

BIGTREE TECH

Eddy 系列

(Eddy, Eddy Coil, Eddy Duo)

用户手册



目录

| | |
|--|----|
| 目录..... | 2 |
| 修订历史..... | 4 |
| 一、产品信息 | 5 |
| 二、产品优点 | 6 |
| 三、产品尺寸及接口 | 7 |
| 3.1 尺寸图 | 7 |
| 3.2 B00T 按键位置 | 8 |
| 四、安装指南 | 9 |
| 4.1 安装高度..... | 9 |
| 4.2 以 Voron2.4 为例..... | 9 |
| 4.3 在其他机器上的安装 | 10 |
| 4.4 Eddy + Manta M5P..... | 10 |
| 4.5 Eddy + Manta M8P V2.0..... | 11 |
| 4.6 Eddy Coil + EBB36 V1.2..... | 11 |
| 4.7 Eddy Coil + EBB42 V1.2..... | 12 |
| 4.8 Eddy Duo + MANTA M5P (USB) | 12 |
| 4.9 Eddy Duo + MANTA M5P (CAN) | 13 |
| 4.10 Eddy Duo + MANTA M8P V2.0 (USB) | 14 |
| 4.11 Eddy Duo + MANTA M8P V2.0 (CAN) | 14 |
| 4.12 Eddy Duo + EBB36..... | 15 |
| 4.13 Eddy Duo + EBB42..... | 16 |
| 4.14 Eddy Duo + EBB SB..... | 16 |
| 4.15 Octopus V1.1/Pro V1.0/Pro V1.0.1 + Eddy Duo (USB) | 17 |
| 4.16 Octopus V1.1/Pro V1.0/Pro V1.0.1 + Eddy Duo (CAN) | 17 |
| 4.17 Octopus Pro V1.1 + Eddy Duo (USB) | 18 |
| 4.18 Octopus Pro V1.1 + Eddy Duo (CAN) | 18 |
| 五、固件..... | 19 |
| 5.1 编译固件..... | 19 |

| | |
|--|----|
| 5.2 更新固件..... | 21 |
| 5.2.1. 通过电脑更新固件..... | 21 |
| 5.2.2 通过 DFU 更新固件..... | 22 |
| 5.3 实现 USB 通信到 CAN 通信的转换..... | 23 |
| 5.4 烧录 KATAPULT | 24 |
| 5.5 更新固件..... | 24 |
| 5.5.1 通过电脑更新固件..... | 24 |
| 5.5.2 使用 DFU 进行更新 Katapult 到 Eddy 中..... | 26 |
| 5.6 通过 KATAPULT 进行固件更新 | 26 |
| 六、Klipper 和 Eddy 配置 | 28 |
| 6.1 打印机配置文件模板 | 28 |
| 6.2 Z 轴限位开关配置..... | 28 |
| 七、校准..... | 30 |
| 7.1 驱动电流校准 | 30 |
| 7.2 将 Eddy 读数映射到喷嘴高度 | 30 |
| 7.3 床面网格校准 | 31 |
| 7.4 温度补偿校准 (coil 版本无温度补偿, 忽略此步骤即可) | 31 |
| 八、附加信息 | 33 |
| 8.1 Z 轴偏移..... | 33 |
| 8.2. 打印床网格校准参数 | 33 |
| 8.3. 网格扫描高度 | 33 |
| 8.4. 快速 (连续) 扫描 | 34 |
| 九、FAQ 常见问题解答..... | 35 |

修订历史

| 版本 | 日期 | 修改说明 |
|-------|------------|--|
| v1.00 | 2024/4/19 | 初稿 |
| v1.01 | 2024/4/25 | 1. 新增 BOOT 按键标识 2. 增加重启 Klipper 的说明 |
| v1.02 | 2024/4/26 | 增加了电脑更新固件的方法 |
| v1.03 | 2024/4/30 | 重新整理配置、校准部分的说明 |
| v1.04 | 2024/5/15 | 添加中心点位置尺寸 |
| v1.05 | 2024/6/06 | 在 5.1 重要提示添加了 z_virtual_enstop 说明 |
| v1.06 | 2024/8/19 | 合并了 Klipper 分支并调整了 Z-offset 配置 |
| v1.07 | 2024/9/24 | 增加有关 CAN 通信的说明 |
| v1.08 | 2024/12/10 | 增加与 Octopus 系列的接线图 |
| v1.09 | 2025/05/07 | menuconfig 配置修改为 Flash chip (GENERIC_03H with CLKDIV 4) |

一、产品信息

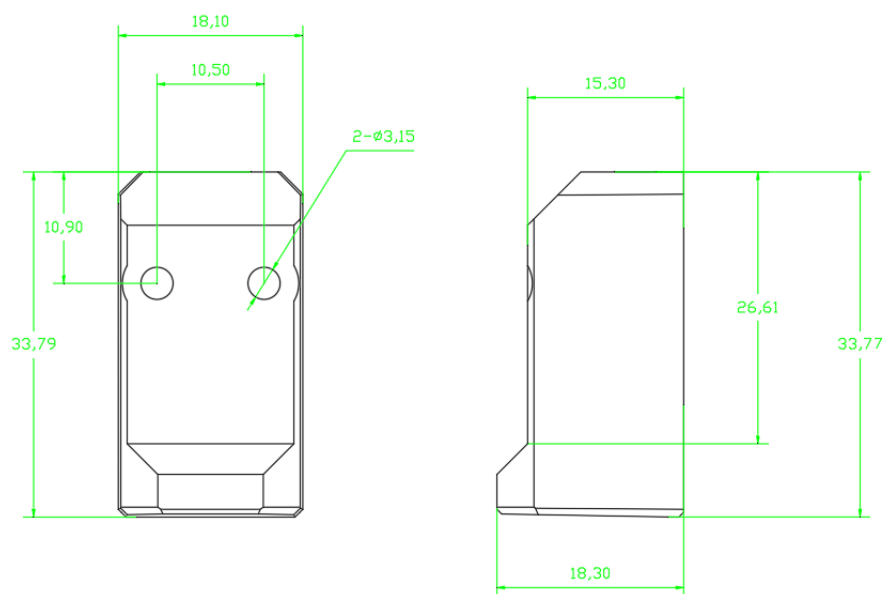
| | |
|------|---|
| 产品名称 | Eddy, Eddy Coil, Eddy Duo |
| 产品重量 | 6g |
| 电压 | 5V |
| 静态电流 | 30mA |
| 工作电流 | 30mA |
| 电缆长度 | 2.5 m (Eddy, Eddy Duo), 15cm (Eddy Coil) |
| 接线 | Eddy, Eddy Duo: 4pin, 间距 1.5mm Eddy Coil: 4-2.54mm 杜邦母头, 一头 ZH1 5mm 4P 连接器 |
| 使用温度 | $\leq 60^{\circ}\text{C}$ 环境温度 |
| 标准误差 | 0.5 μm |
| 适配机型 | 所有使用 Klipper 的 FDM 打印机 |

二、产品优点

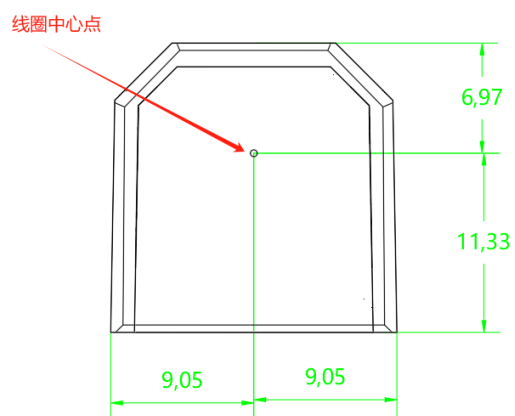
- 整体体积小，重量轻；
- 具备温度补偿功能；
- 高效率调平；
- 应用广泛，兼容性强；
- 精度高，稳定性强；
- 非接触式。

三、产品尺寸及接口

3.1 尺寸图



用于计算 XY 偏移的线圈中心点如下：



3.2 BOOT 按键位置



BOOT 按钮在首次对 Eddy 进行 Klipper 编程时使用；之后，Klipper 可以自动重新编程 RP2040。

注意： 仅仅 Eddy，Eddy Duo 拥有 BOOT 功能，Eddy Coil 上的按键无功能。

四、安装指南

4.1 安装高度

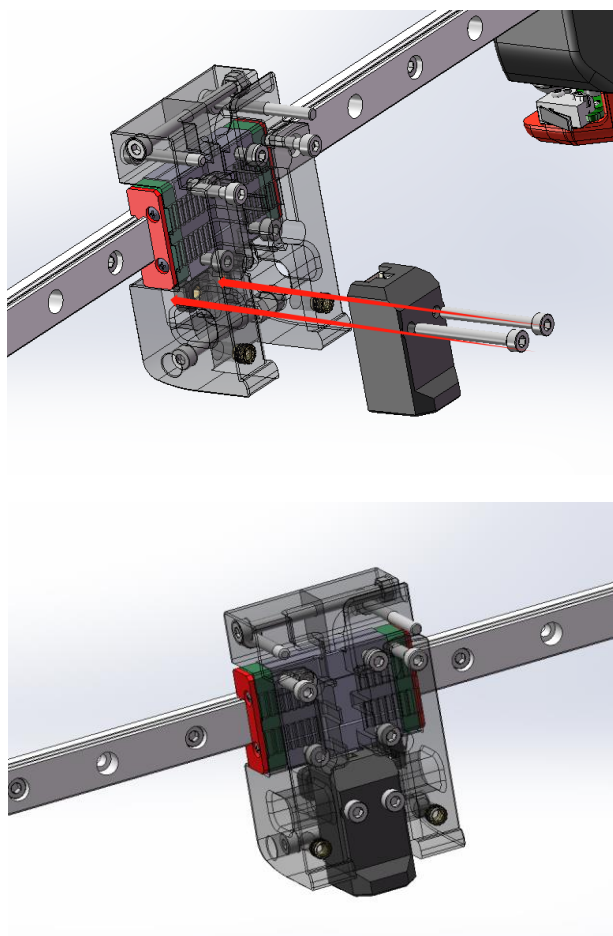
确保 Eddy 的安装位置高于喷嘴 2 至 3 毫米，以确保最佳性能。如果在校准过程中遇到错误，这些可能与 Eddy 的安装高度有关。有关解决方案，请参阅本手册的故障排除部分。

重要提示：需要注意的是，用户可能会将电流校准高度 20 毫米与 Eddy 的安装高度 2 至 3 毫米混淆。20 毫米的高度仅在本手册后续部分进行线圈电流校准时使用。

4.2 以 Voron2.4 为例

安装位置，完全替代原有 PL-08N 安装位

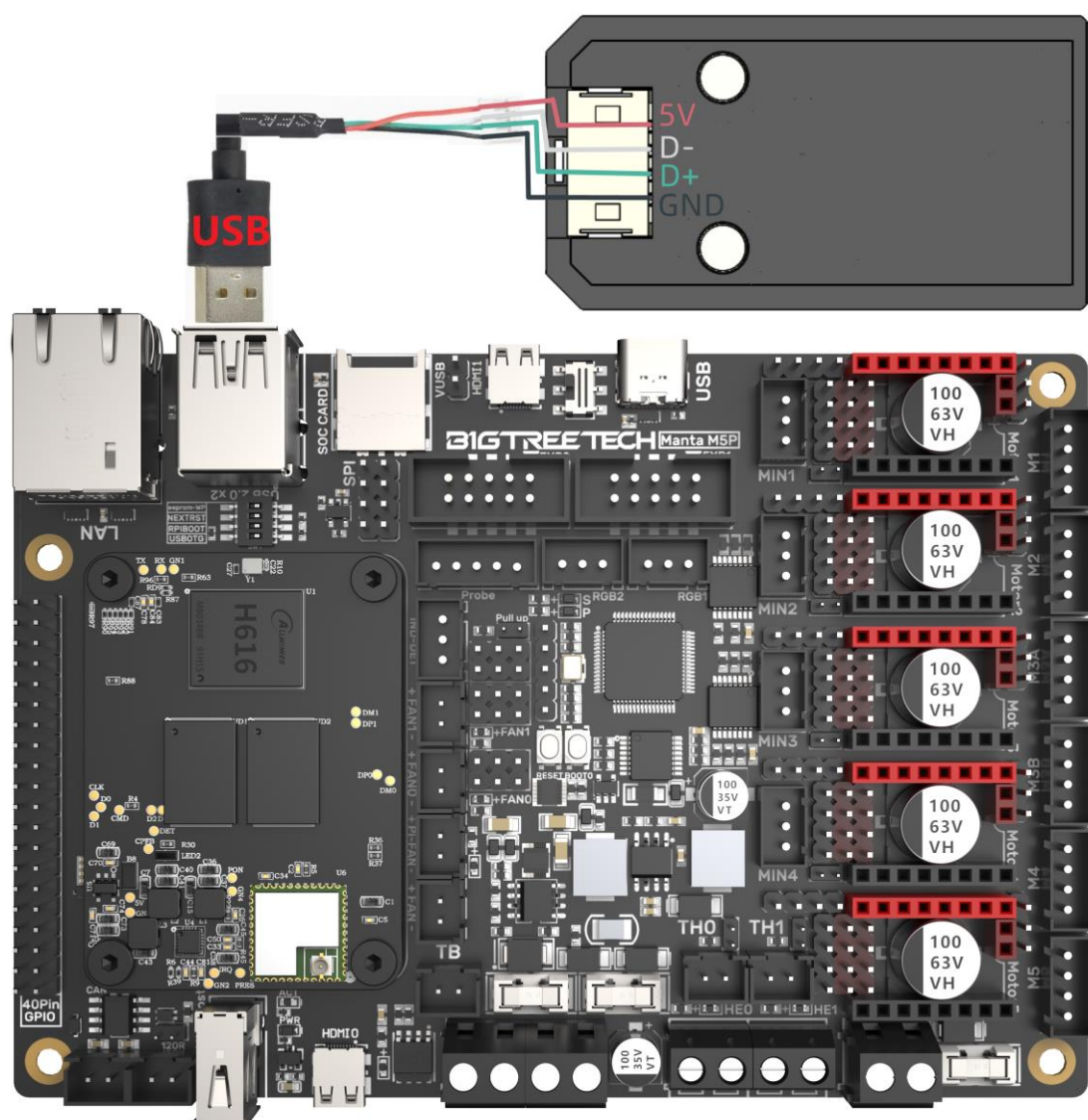
使用两颗 M3*25 螺丝（包装内附）将模块固定再 X Carriage 打印件上，如视图



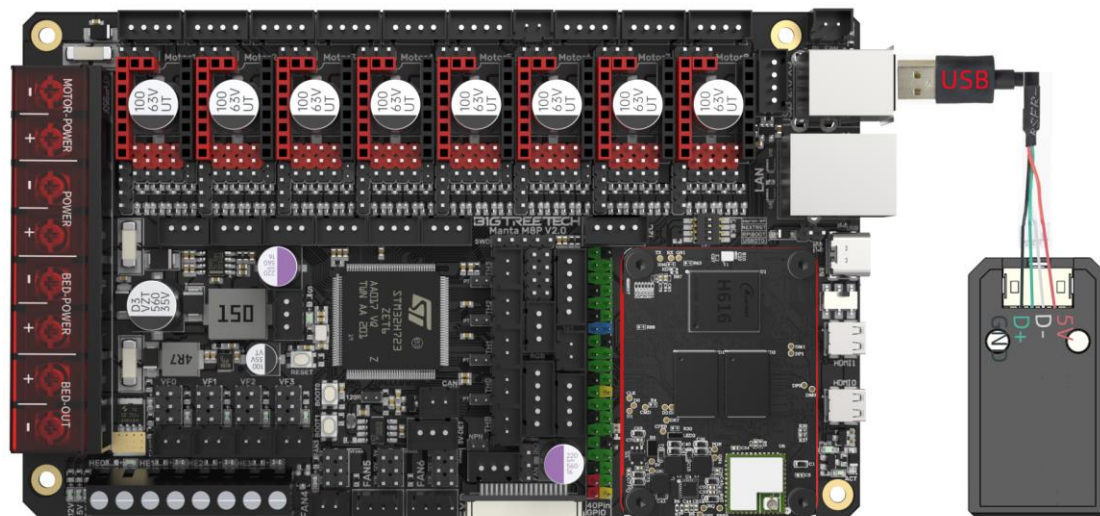
4.3 在其他机器上的安装

用户可在我们的 GitHub 仓库以及其他常见模型分享平台中找到适用于多种常见机型的支架。在安装 Eddy 的过程中，请确保 PCB 侧（背面）与热端保持尽可能大的距离。此类布局有助于最小化从热端到 Eddy 的热传导。

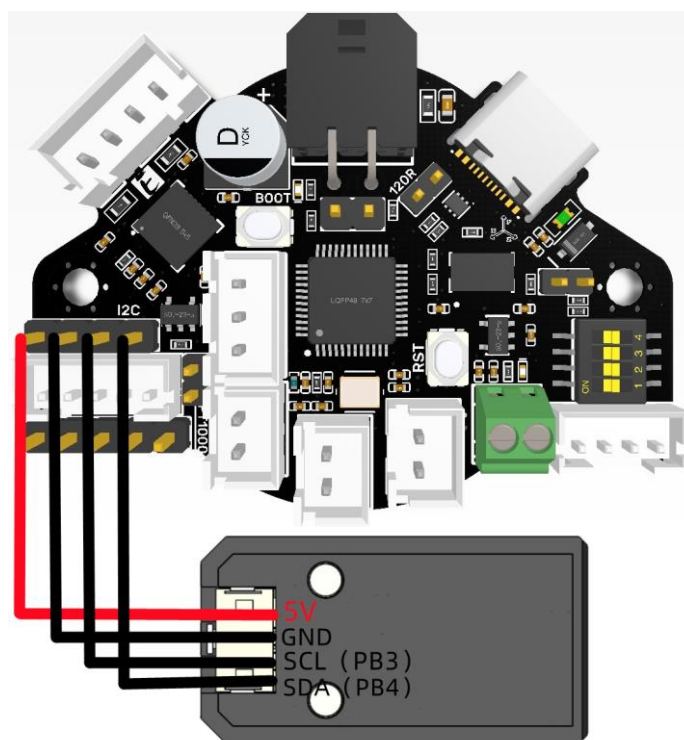
4.4 Eddy + Manta M5P



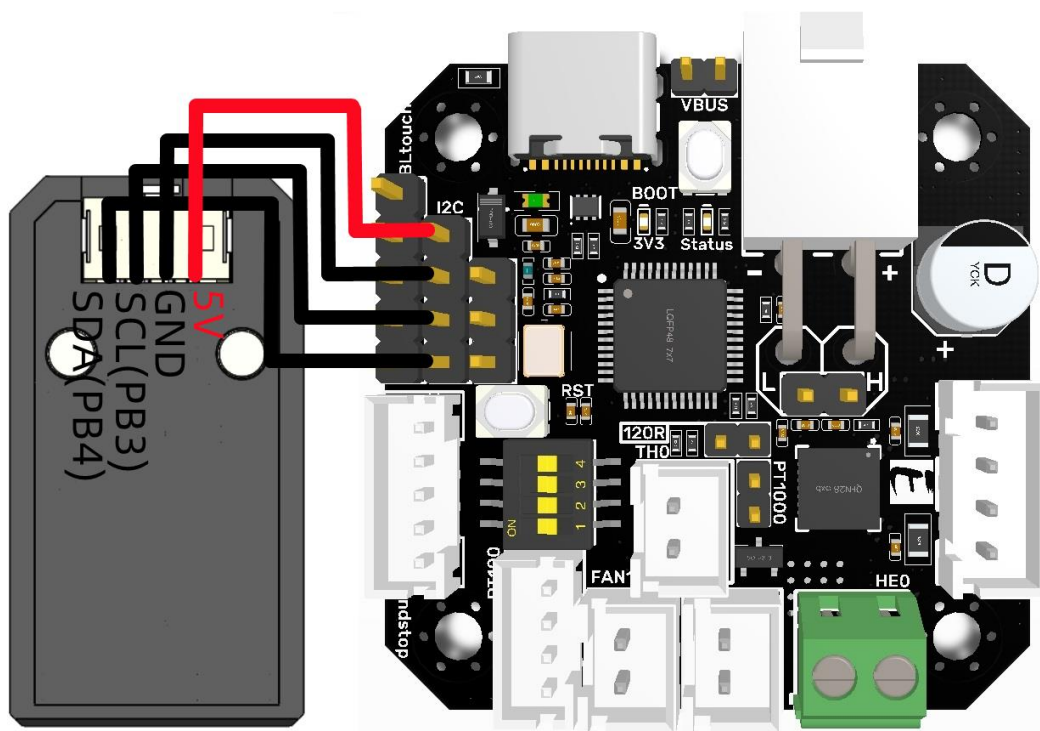
4.5 Eddy + Manta M8P V2.0



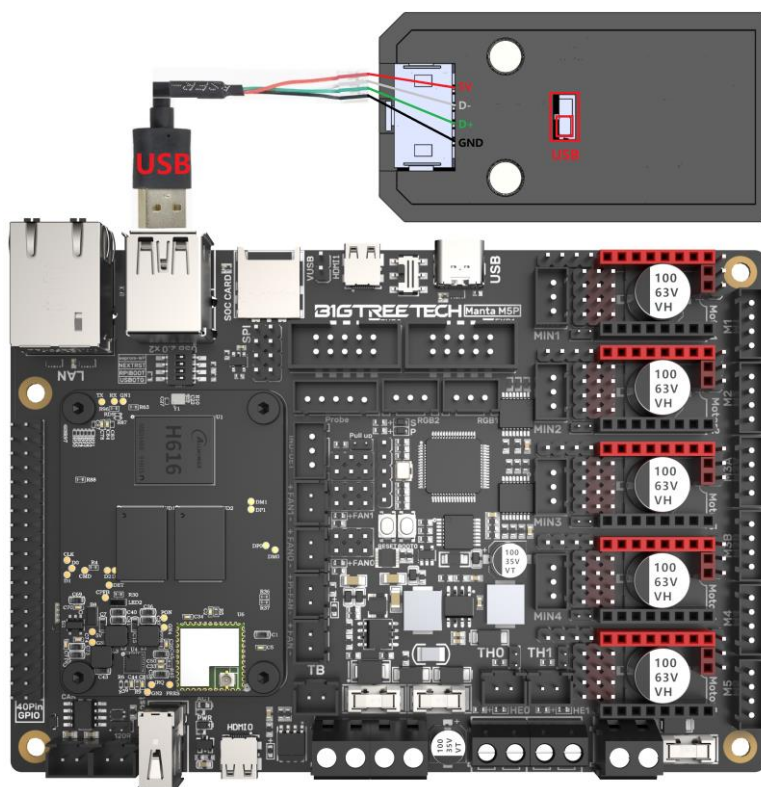
4.6 Eddy Coil + EBB36 V1.2



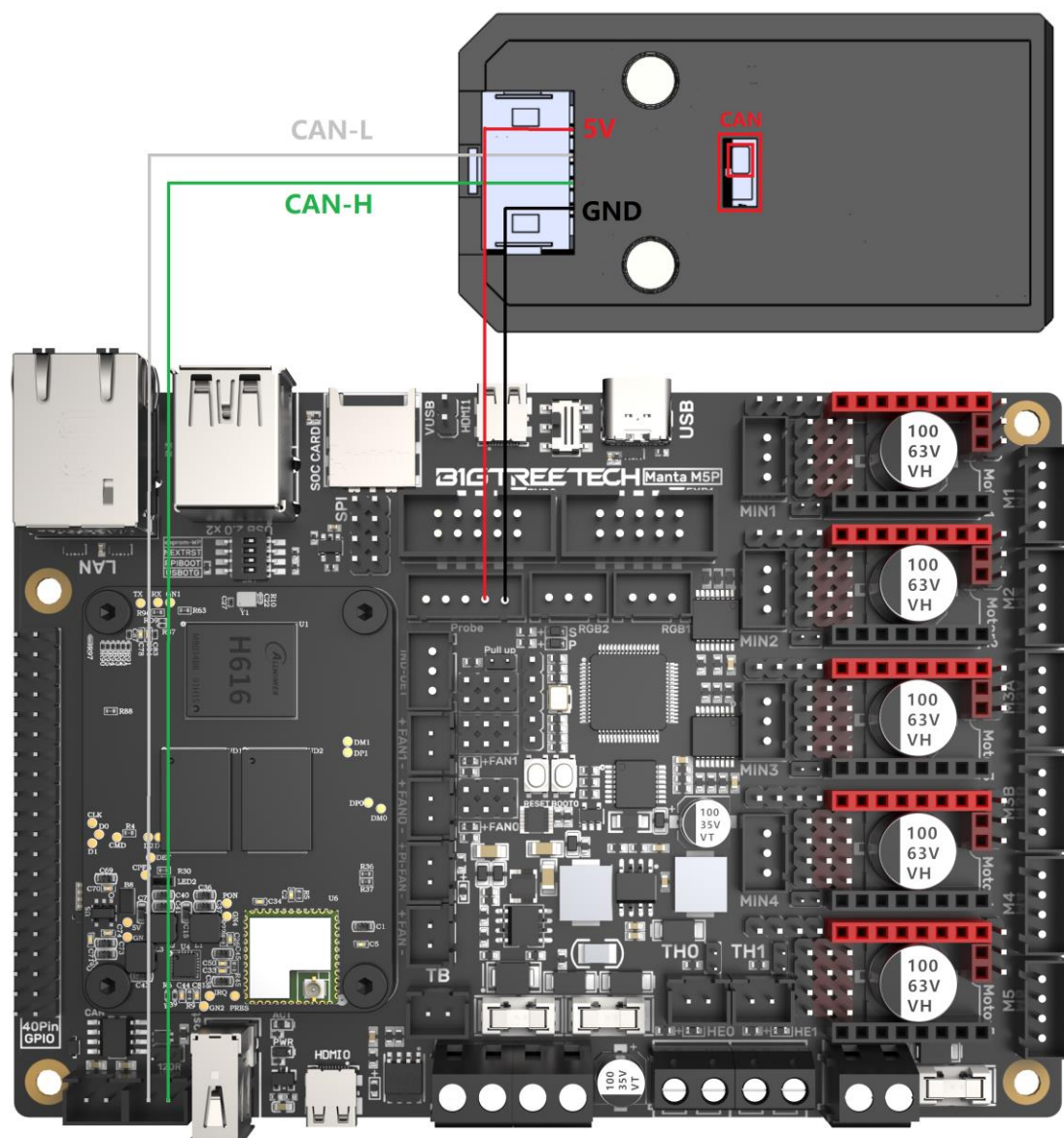
4.7 Eddy Coil + EBB42 V1.2



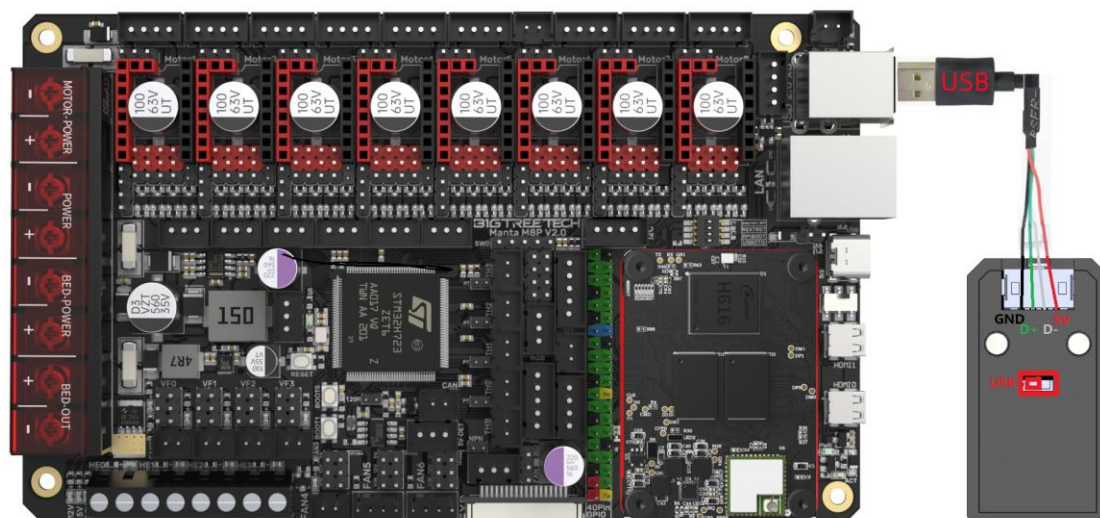
4.8 Eddy Duo + MANTA M5P (USB)



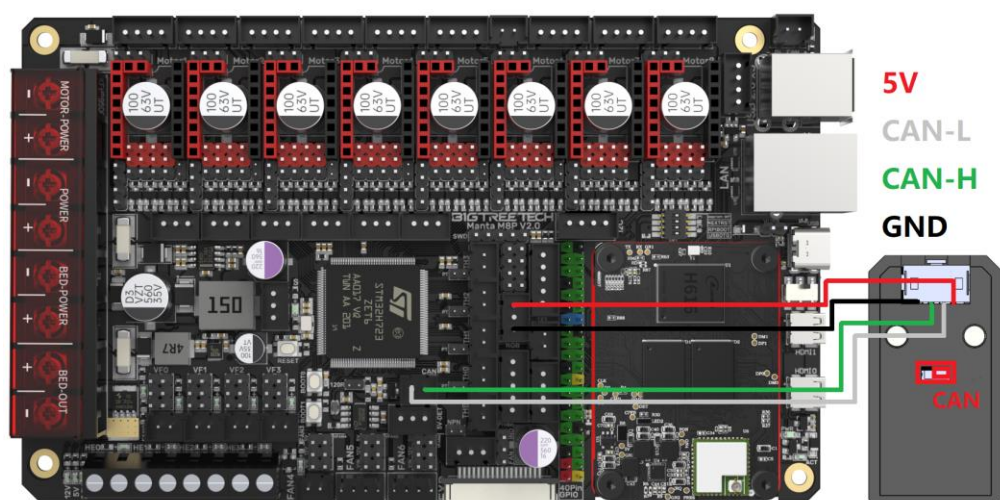
4.9 Eddy Duo + MANTA M5P (CAN)



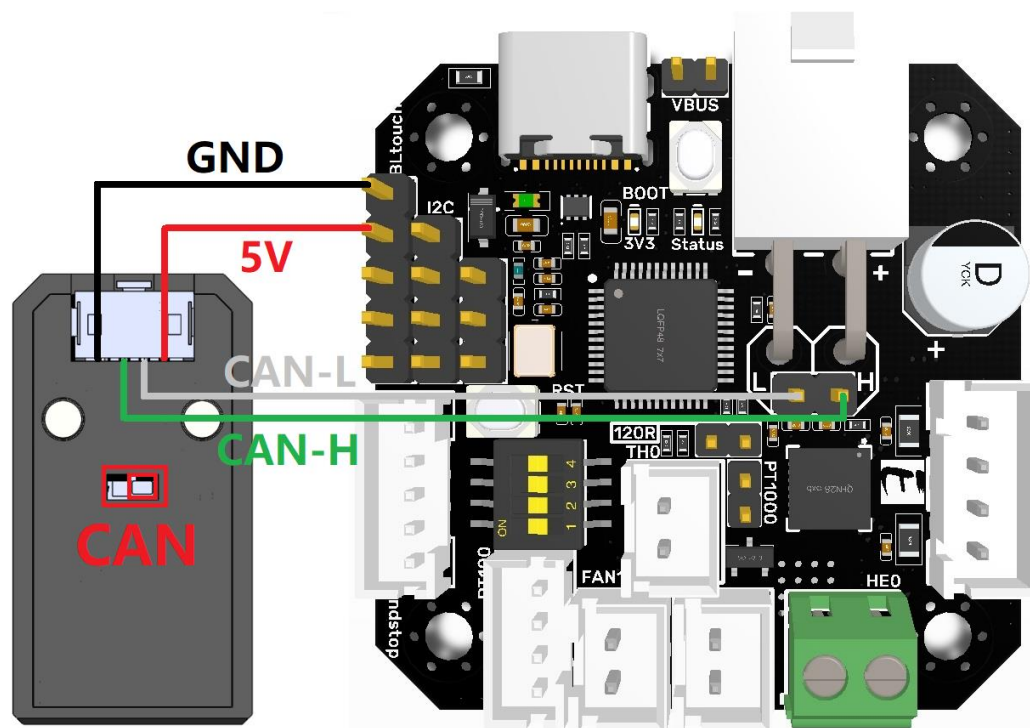
4.10 Eddy Duo + MANTA M8P V2.0 (USB)



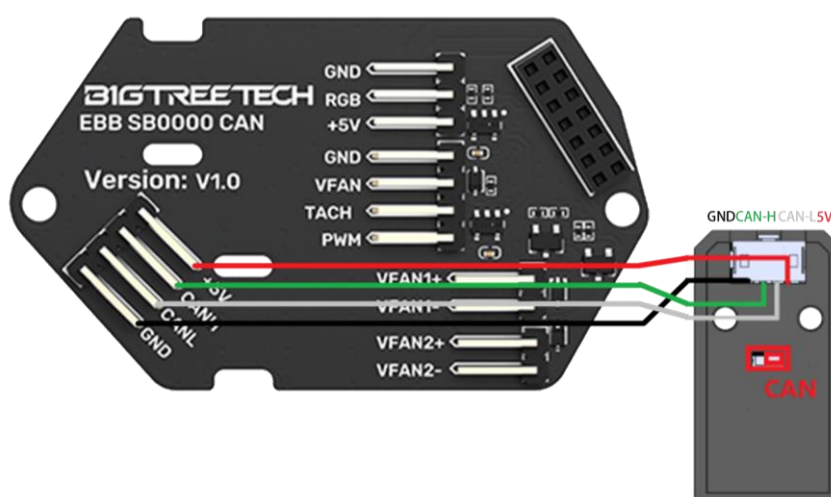
4.11 Eddy Duo + MANTA M8P V2.0 (CAN)



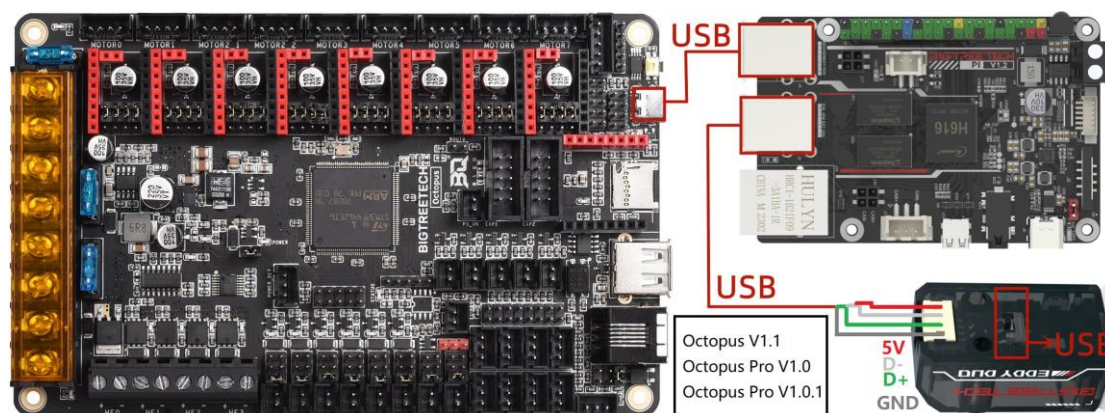
4.13 Eddy Duo + EBB42



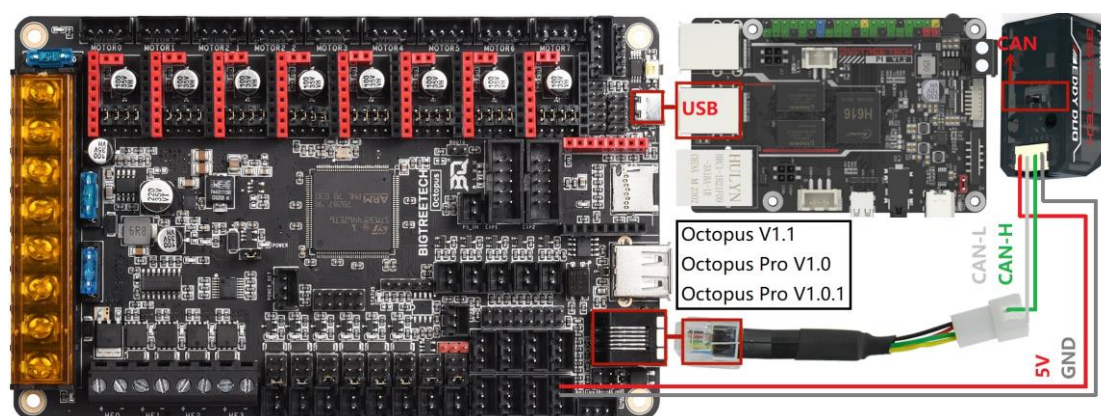
4.14 Eddy Duo + EBB SB



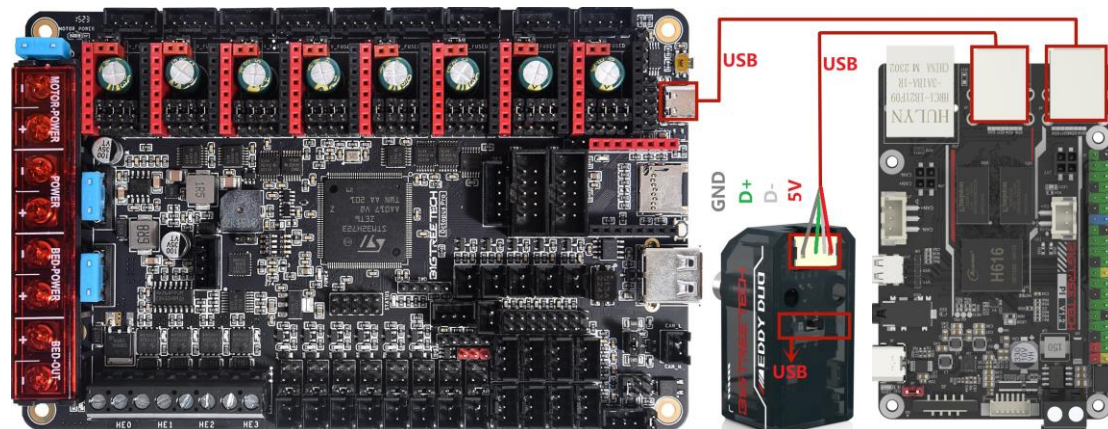
4.15 Octopus V1.1/Pro V1.0/Pro V1.0.1 + Eddy Duo (USB)



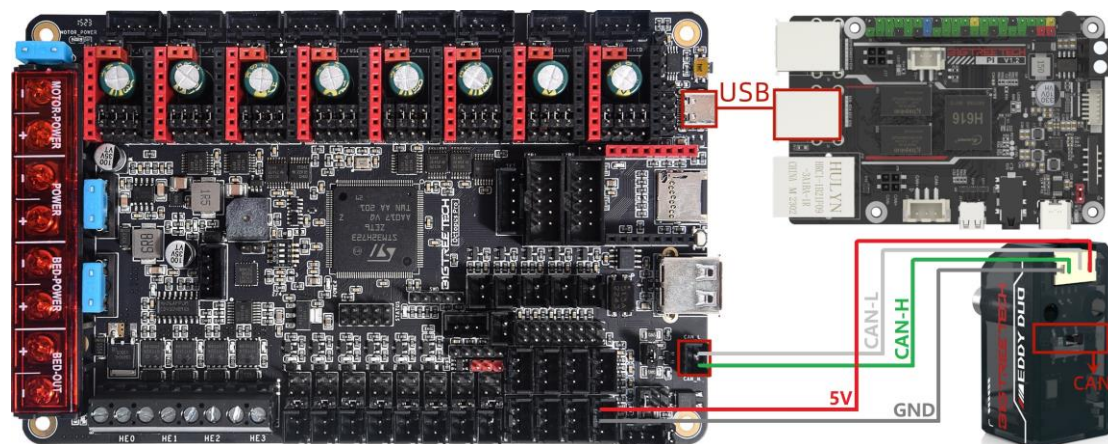
4.16 Octopus V1.1/Pro V1.0/Pro V1.0.1 + Eddy Duo (CAN)



4.17 Octopus Pro V1.1 + Eddy Duo (USB)



4.18 Octopus Pro V1.1 + Eddy Duo (CAN)



五、固件

重要提示:

1. Eddy 和 Eddy Coil 只支持基于 Python 3 的 Klipper 虚拟环境。即使系统已安装 Python 3, 也不代表 klippy 虚拟环境是用 Python 3 建立的。若出现 'Internal error during connect: split() takes no keyword arguments' 等错误信息, 说明您的 klippy 主机可能还在使用基于 Python 2 的虚拟环境, 需要进行升级。推荐使用 KIAUH 工具, 这是在不覆盖现有配置的情况下, 更新 klippy 主机至 Python 3 环境的最简便方法。
2. Eddy Duo 有两种通信方式。如果要从 USB 通信方式转换到 CAN 通信方式时, 建议在 USB 通信模式下更新 CAN 通信所需要的固件会让过程更简单。(使用 DFU 更新会比 KATAPULT 更新简单)

5.1 编译固件

此编译教程仅适用于 Eddy, Eddy Duo。如果您使用的是 Eddy Coil, 则需将其连接到工具板上的 I²C 端口。首先在 Klipper 的 master 分支上为该工具板编译固件, 编译完成后, 将新固件上传并更新到工具板上。在 Klipper 中配置 Eddy 时, 只需设置其通过该工具板的 I²C 端口进行通信, 具体通信配置取决于工具板的引脚设置。

如果您是从 Klipper 的旧 BIGTREETECH 分支转来, 建议使用 KIAUH 迁移到主线分支, 并推荐更新所有 Klipper 设备的固件, 确保它们也运行在从主线编译的二进制文件上。

编译步骤如下:

1. 确保您使用的是主线 Klipper, 在 ssh 终端中运行以下命令

```
cd ~/klipper/  
git checkout master
```

2. 接下来, 输入: `make menuconfig`
3. 使用下面的配置编译固件

(1) Eddy V1.0 (USB)

```
(Top)
Klipper Firmware Configuration
[*] Enable extra low-level configuration options
    Micro-controller Architecture (Raspberry Pi RP2040/RP235x) ---->
    Processor model (rp2040) ---->
    Bootloader offset (No bootloader) ---->
    Flash chip (GENERIC_03H with CLKDIV 4) ---->
    Communication Interface (USB SERIAL) ---->
    USB ids ---->
[*] Optimize stepper code for 'step on both edges'
() GPIO pins to set at micro-controller startup
```

```
[*] Enable extra low-level configuration options
    Micro-controller Architecture (Raspberry Pi RP2040/RP235x) ---->
    Processor model (rp2040) ---->
    Bootloader offset (No bootloader) ---->
    Flash chip (GENERIC_03H with CLKDIV 4) ---->
    Communication Interface (USB SERIAL) ---->
```

(2) Eddy Duo

```
(Top)
Klipper Firmware Configuration
[*] Enable extra low-level configuration options
    Micro-controller Architecture (Raspberry Pi RP2040) ---->
    Bootloader offset (16KiB bootloader) ---->
    Communication interface (CAN bus) ---->
(4) CAN RX gpio number (NEW)
(5) CAN TX gpio number (NEW)
(1000000) CAN bus speed
() GPIO pins to set at micro-controller startup (NEW)
```

```
[*] Enable extra low-level configuration options
    Micro-controller Architecture (Raspberry Pi RP2040/RP235x) ---->
    Processor model (rp2040) ---->
```

如果不使用 KATAPULT

```
    Bootloader offset (No bootloader) ---->
    Flash chip (GENERIC_03H with CLKDIV 4) ---->
```

如果使用 KATAPULT (使用此方式需要先进行 5.4 的操作)

```
    Bootloader offset (16KiB bootloader) ---->
```

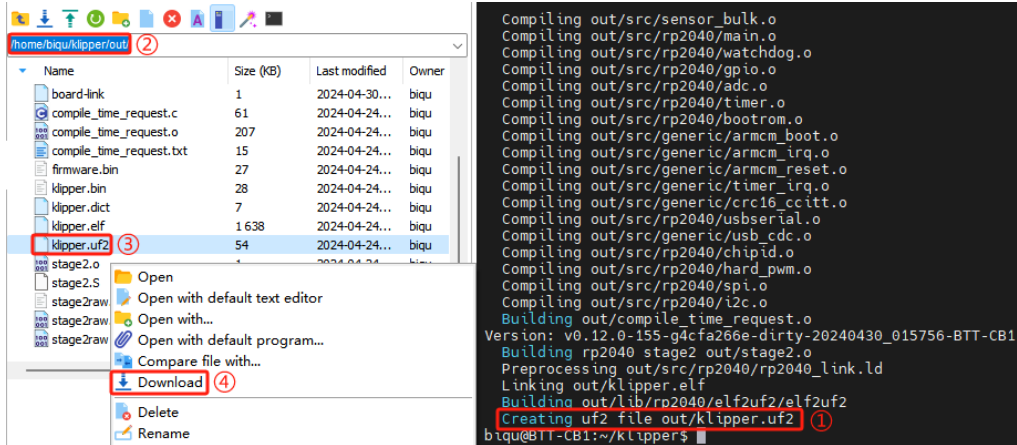
如果使用 USB 通信

```
    Communication Interface (USB SERIAL) ---->
```

如果使用 CAN BUS 通信

```
    Communication interface (CAN bus) ---->
(4) CAN RX gpio number
(5) CAN TX gpio number
(1000000) CAN bus speed
```


- 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” ；
- 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。



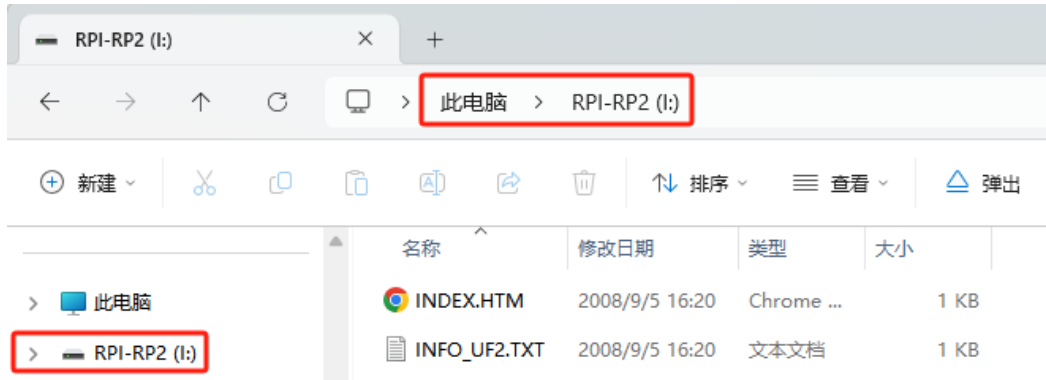
5.2 更新固件

5.2.1. 通过电脑更新固件

- 按住 Boot 按钮，用 USB 线将 Eddy 接到电脑的 USB 端口。（无需拆开 Eddy，Boot 按钮位于如图所示的位置，可直接操作。）



- 电脑会识别出一个存储设备，将上面步骤下载下来的 klipper.uf2 文件拷贝到这个存储设备，设备会自动更新固件并重启，重启后即完成了更新固件。



5.2.2 通过 DFU 更新固件

1. 按住 Boot 按钮，用 USB 线将 Eddy 接到树莓派/BIGTREETECH Pi 的 USB 端口。
(无需拆开 Eddy，Boot 按钮位于如图所示的位置，可直接操作。)



2. 在 SSH 终端命令行中运行 `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. 运行
`cd ~/klipper`
`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)
4. 输入 `ls /dev/serial/by-id/*` 到命令行。找到的设备将被输入到您的

klipper 配置中，作为[mcu eddy]下的 Serial 变量。

5.3 实现 USB 通信到 CAN 通信的转换

1. 首先在 USB 通信模式下将 CAN 通信的固件编译好

由于之前已经烧录过 USB 固件，这次烧录时无需按下 Boot 按钮。

2. 如果使用 USB 通信，可以直接输入

```
make flash FLASH_DEVICE=/dev/serial/by-id/usb-KLIPPER_RP2040....
```

烧录固件（注意：将/dev/serial/by-id/xxx 更换为查询到的 USB 通信 ID，查询命令为：ls /dev/serial/by-id/*）。

然后把 Eddy Duo 拨码开关拨到 CAN 的通信方式

3. CAN BUS 配置

（1）在 SSH 终端中输入命令

```
sudo nano /etc/network/interfaces.d/can0
```

并执行

```
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。

到这里 can 通信基本就可以正常使用了。

5.4 烧录 KATAPULT

之前被称为「CanBoot」的引导加载程序软件已更名为「Katapult」。

注意：Katapult 旨在通过 CAN bus 接口直接更新 MCU 固件。

参考此处说明下载 Katapult 工程 <https://github.com/Arksine/katapult>

1. 输入 `cd ~`

跳转到主目录，输入 `git clone https://github.com/Arksine/katapult`

下载 Katapult 工程，

输入 `cd katapult`

跳转到 Katapult 目录中。

2. 输入

`make menuconfig` 并按照下图配置

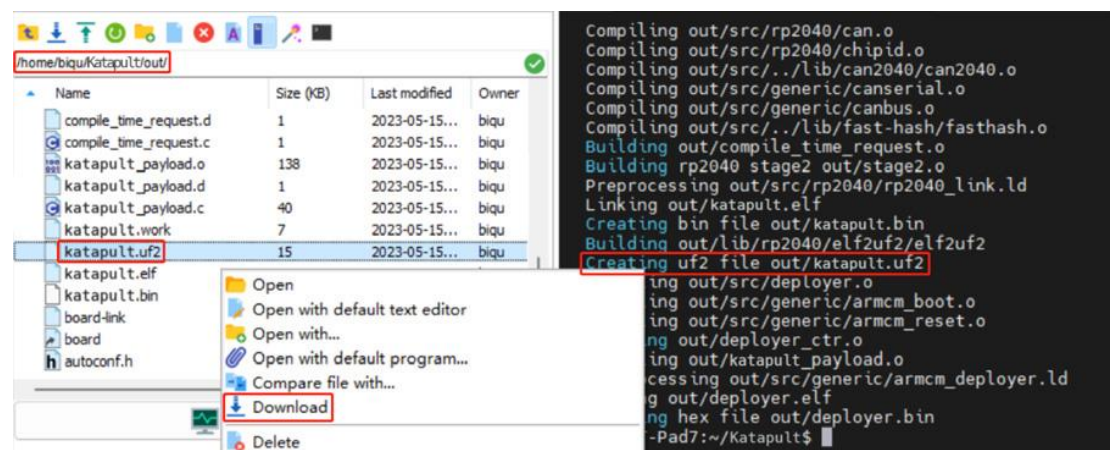
```
(Top)
Katapult Configuration v0.0.1-96-gbceceea
Micro-controller Architecture (Raspberry Pi RP2040/RP235x) --->
Processor model (rp2040) --->
Flash chip (GENERIC_03H with CLKDIV 4) --->
Build Katapult deployment application (16KiB bootloader) --->
Communication Interface (CAN bus) --->
(4) CAN RX gpio number (NEW)
(5) CAN TX gpio number (NEW)
(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
(gpio26) Status LED GPIO Pin
```

3. 输入 `make` 编译固件，当 make 执行完成后会在 `home/biqu/Katapult/out` 文件夹中生成我们所需要的 ‘katapult.uf2’ 固件，

5.5 更新固件

5.5.1 通过电脑更新固件

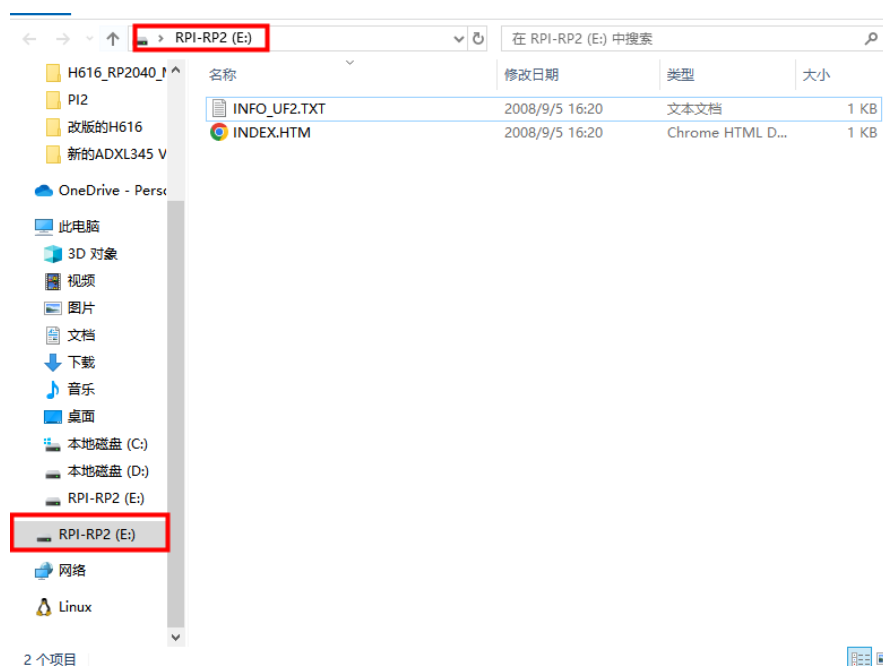
(1) 在 SSH 软件左侧可以直接下载到电脑中；



(2) Eddy 模块连接上我们配送的 USB 接线，然后按住 Boot 按钮，将 USB 接线连接到电脑的

USB 端口，电脑会识别出一个存储设备，如图：





(3) 将上面步骤下载下来的 klipper.uf2 文件拷贝到这个存储设备，即完成了更新固件。

5.5.2 使用 DFU 进行更新 Katapult 到 Eddy 中

- (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 `make flash FLASH_DEVICE=2e8a:0003` 开始烧录 Katapult

(注意：将 2e8a:0003 更换为上一步中查询到的实际的设备的 ID)

烧录完成，按照上述 5.1 进行编译固件后

我们就可以通过 KATAPULT 进行固件更新

5.6 通过 KATAPULT 进行固件更新

注意：下面提到的 klipper.bin 需要提前 make 生成出来，

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 ~/katapult/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。

```
biqu@BTT-CB1:~/Katapult/scripts$ python3 flash_can.py -i can0 -q
Resetting all bootloader node IDs...
Checking for katapult nodes...
Detected UUID: be69315a613c, Application: Katapult
Query Complete
biqu@BTT-CB1:~/Katapult/scripts$
```

3. 输入

`python3 flash_can.py -i can0 -f ~/klipper/out/klipper.bin -u be69315a613c`

替换为实际的 UUID，注意：Katapult 的 Application start offset 为 16KiB offset，

所以 Klipper 的 menuconfig 中 Bootloader offset 也要为 16KiB bootloader，如下图已经烧录成功。

```
biqu@BTT-CB1:~/Katapult/scripts$ python3 flash_can.py -i can0 -f ~/klipper/out/klipper.bin -u be69315a613c
Sending bootloader jump command...
Resetting all bootloader node IDs...
Checking for katapult nodes...
Detected UUID: be69315a613c, Application: Katapult
Attempting to connect to bootloader...
Katapult connected
Protocol Version: 1.0.0
Block Size: 64 bytes
Application Start: 0x8002000
MCU type: stm32g0b1xx
Verifying canbus connection
Flashing '/home/biqu/klipper/out/klipper.bin'...

[#####]
Write complete: 13 pages
Verifying (block count = 414)...

[#####]
Verification Complete: SHA = C3B1F96A8FCE706587BF4A9119D95080465875A3
CAN Flash Success
biqu@BTT-CB1:~/Katapult/scripts$
```

4. 再次输入

`python3 flash_can.py -i can0 -q`

查询，此时 Application 由之前的 Katapult 变为 Klipper，代表 Klipper 已经正常运行。

```
biqu@BTT-CB1:~/Katapult/scripts$ python3 flash_can.py -i can0 -q
Resetting all bootloader node IDs...
Checking for katapult nodes...
Detected UUID: be69315a613c, Application: Klipper
Query Complete
biqu@BTT-CB1:~/Katapult/scripts$
```

六、Klipper 和 Eddy 配置

6.1 打印机配置文件模板

现在您已经成功安装了正确的固件到您的 Eddy 设备上，接下来是完成 Klipper 配置的步骤。BIGTREETECH 提供了三种不同的示例配置文件，帮助您开始设置。您需要根据下面的标准选择最适合您需求的文件。仔细阅读选定配置文件中的注释，它们将帮助您了解如何修改安装中的特定参数。每个配置文件的链接如下所示，您也可以通过访问 <https://github.com/bigtreotech/Eddy> 找到它们：

- 如果您希望使用 Eddy 作为自动调平传感器，但使用另一设备作为 Z 轴限位开关，[使用该无归零功能的配置](#)。
- 如果您希望 Eddy 同时作为自动调平传感器和 Z 轴限位开关，[使用该包含归零功能的配置](#)。
- 如果您希望 Eddy 同时作为自动调平传感器和 Z 轴限位开关，并希望使用 beta Z 轴偏移功能，[使用该包含归零和 Z 轴偏移的配置](#)。

选择任何配置后，将其全部内容复制到您的 `printer.cfg` 文件中。如果出现 gcode 宏冲突，请查看 FAQ 中的“Error: gcode command < ANY GCODE COMMAND > already registered”部分。

6.2 Z 轴限位开关配置

您可以将 Eddy 配置为 Z 轴限位开关，或者选择使用其他设备作为限位开关。如果您选择使用其他设备作为 Z 轴限位开关，请根据该设备调整您的归零(homing)和限位开关设置。

如果您希望为 Eddy 启用 Z 轴归零/限位开关功能，请按照以下步骤操作：

- a. 在 `printer.cfg` 文件中的 `[stepper_z]` 部分，将 `endstop_pin: PA5` 修改为 `endstop_pin: probe:z_virtual_endstop`，并注释或删除 `position_endstop: 0`。**请注意**，您当前使用的限位开关可能不是 PA5，因此请查找与您的限位开关相匹配的行并进行更改。
- b. 确保您选择了正确的示例配置文件，并将该文件的全部内容复制到您的 `printer.cfg` 文件中。如果您使用着 KNOMI，可能会有一些宏与 `KNOMI.cfg` 文件中的宏存在冲突。解决这些冲突的方法是注释掉 `KNOMI.cfg` 文件中的宏，并取消

注释 Eddy 宏中处理 KNOMI 功能的行。

c. 根据您的设备特定需求编辑配置文件的相关部分。可能需要调整的设置包括

- MCU serial
- X offset 和 Y offset
- Mesh_min 和 mesh_max
- Home_xy_position

重要提示： 示例配置要求您调整 x_offset 和 y_offset 以匹配您的 Eddy 位置相对于您的喷嘴的位置。所有示例配置文件中均包含标准 Voron X 滑车的设置。

七、校准

7.1 驱动电流校准

完成固件和配置后，您现在可以开始对 Eddy 的驱动电流进行校准。

1. 将 Eddy 置于热床上方约 20mm 处。如果您计划将 Eddy 用作限位开关，则尚不能使用其进行归零操作，您需要手动移动龙门或床，使 Eddy 位于床面上方 20mm。
2. 在 Mainsail 或 Fluidd 中运行以下命令：`LDC_CALIBRATE_DRIVE_CURRENT CHIP=btt_eddy`
3. 输入 `SAVE_CONFIG` 以将驱动电流设置保存到配置中。

7.2 将 Eddy 读数映射到喷嘴高度

驱动电流校准完成后，Eddy 将能够从打印床获取读数。Klipper 需要知道这些读数与喷嘴的高度如何对应。以下校准程序将喷嘴定位在床面上，使 Z 高度等于 0。然后，它从 Eddy 获取读数，随着喷嘴高度逐渐增加，将这些读数映射到已知的高度。

- 如果您使用了推荐的配置模板，可以简单按照以下步骤执行映射程序。
 1. 发送命令 `PROBE_EDDY_CURRENT_CALIBRATE_AUTO CHIP=btt_eddy`
 2. 按照 klipper UI 的提示，逐步降低喷嘴，直至其接触到放在打印床上的纸张。确保纸张在适当的压力下可以平滑移动，同时感受到轻微的摩擦力。注意，在此过程中要避免喷嘴对打印床造成过大的压力或损坏。
 3. 点击接受，并观察 Eddy 执行映射过程。完成后确保发送 `SAVE_CONFIG`。
 4. 跳到床面网格校准。
- 如果您没有使用配置模板，按以下步骤执行映射程序：
 1. 使用命令 `G28 X Y` 归零 X 轴和 Y 轴。
 2. 确保没有加载床面高度图。从控制台发送 `BED_MESH_CLEAR` 以清除高度图。
 3. 使用命令 `G0 X150 Y150 F6000` 将喷嘴移动到平台中心。此命令基于

300x300mm 的打印机设置，但您可能需要根据您的打印床尺寸进行相应调整。

4. 开始手动 Z 轴偏移校准 ([Paper Test](#))：

输入 `PROBE_EDDY_CURRENT_CALIBRATE CHIP=btt_eddy`。您将看到一个调整框，允许您降低喷嘴。降低喷嘴直至其接触到放在打印床上的纸张。确保纸张在适当的压力下可以平滑移动，同时感受到轻微的摩擦力。

5. 完成后使用 `SAVE_CONFIG` 保存配置。

```
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
```

7.3 床面网格校准

1. Home 所有轴。
2. 带有 `z_tilt` 或者 `quad_gantry_level` (QGL) 功能的打印机，先运行一次 `Z_TILT_ADJUST` 或者 `QUAD_GANTRY_LEVEL`，防止网格扫描时喷嘴撞到热床
3. 执行以下命令进行快速的网格扫描 `BED_MESH_CALIBRATE METHOD=scan SCAN_MODE=rapid`
4. 完成后使用 `SAVE_CONFIG` 保存配置。

7.4 温度补偿校准 (coil 版本无温度补偿，忽略此步骤即可)

重要提示： 以下步骤仅适用于 Eddy, Eddy Duo。Eddy Coil 没有温度补偿功能，因此可忽略下面的步骤。当 Eddy 执行温度补偿时，热床极限温度很高，谨防烫伤。

1. Home 所有轴并通过输入 `G0 Z5` 或通过操作界面上的控制按钮，将 Z 轴调整至床面上方 5 毫米。
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

3. 运行 `TEMPERATURE_PROBE_CALIBRATE PROBE=btt_eddy TARGET=56 STEP=4`

提示：在上述命令中，目标温度设置为 56℃，适用于多数设备。如果您的 3D 打印机腔体的实际温度更高，您可以适当提高目标温度。请注意，目标温度越高，校准所需时间将相应延长，因为 Eddy 需要更长时间来达到温度平衡。

4. 执行此操作后，UI 将显示 Z 轴调整框。请使用上述提到的[手动 Z 轴偏移校准（Paper Test）方法](#)，将纸张夹在喷嘴和床面之间，然后确认该值。
5. 接受值后，将热床温度调至最高，喷嘴温度调至 220℃。
6. 如果您在有空调或开窗的房间中，为了确保 Eddy 的温度上升，建议关闭空调或窗户，因为风会影响温度的升高。
7. 随着 Eddy 温度的上升，系统会自动提示您每隔 4℃执行一次手动 Z 偏移校准。**热床温度很高，谨防烫伤。**
8. 重复手动 Z 偏移校准（Paper Test）直到校准完成。如果发现 Eddy 的温度不再上升，可以使用下面的相关命令提前结束校准。

在漂移校准期间可用的额外 gcode 命令包括：

- `TEMPERATURE_PROBE_NEXT` - 可用于在达到步进增量前强制采样新数据。
- `TEMPERATURE_PROBE_COMPLETE` - 可用于在达到目标前完成校准。
- `ABORT` - 可用于终止校准并忽略结果。

提示：Eddy 的热校准过程不仅考虑了 Eddy 的漂移，还考虑了机器内部机械部件的热膨胀。这种膨胀可能非常显著，使用其他传感器时可能导致第一层打印质量不佳。请记住，如果您在喷嘴和热床均开启的情况下进行热校准，则热端和热床会同时发生热膨胀。因此，如果您稍后尝试进行手动 Z 偏移校准（Paper Test），只开启喷嘴或热床中的一个，您可能会发现有大约 0.05mm 的间隙（不足以引起第一层问题，但足以感觉到纸张的夹持力减小）。如果这听起来有些复杂，不用担心。您只需要知道，应在热床和喷嘴均加热的情况下进行校准，然后在热床和喷嘴均加热的情况下进行打印，这样可以获得极佳的首层效果。

完成这些步骤后，Eddy 将能够在广泛的温度范围内提供出色的首层打印效果！

八、附加信息

8.1 Z 轴偏移

此部分仅适用于将 Eddy 用于 Homing 的用户。

由于 Eddy 经过校准，可以准确识别 $z=0$ 的位置，因此通常不需要使用 Z 轴偏移。然而，如果您希望使用 Z 轴偏移，请使用[包含 Z 轴偏移功能的示例配置文件](#)。

要确定正确的 Z 轴偏移，请按照以下步骤操作：

1. Home 所有轴。
2. 在喷嘴下方放置一张纸。
3. 使用 Mainsail 或 Fluidt 设置 Z 轴高度为 $z=0$ 。请勿使用微调 (babystep) 来将喷嘴调整到 $z=0$ ！应将其设置为 Z 轴的实际高度。
4. 设置 Z 轴高度为 $z=0$ 后，检查纸张夹紧的松紧是否合适。如果不合适，则使用微调功能进行调整。
5. 调整到合适的高度后，使用 Mainsail 或 Fluidt UI 上的按钮保存调整结果。

8.2. 打印床网格校准参数

Eddy 允许您在每次打印前进行非常快速的床面网格扫描，以确保获得最佳的首层效果。为此，我们建议将标准的 `BED_MESH_CALIBRATE` 宏替换为示例配置文件中的修改版本，并在打印开始宏中包含 `BED_MESH_CALIBRATE` 调用。

8.3. 网格扫描高度

扫描高度由 `[bed_mesh]` 配置中的 `horizontal_move_z` 选项设置。此外，还可以通过 `BED_MESH_CALIBRATE` G-code 命令中的 `HORIZONTAL_MOVE_Z` 参数指定。

扫描高度必须足够低，以避免扫描误差。通常情况下，设置为 2mm（即：`HORIZONTAL_MOVE_Z=2`）会比较合适，前提是 Eddy 正确安装。

需要注意的是，如果 Eddy 距离床面超过 4mm，则扫描结果将无效。因此，在床

面有严重表面偏差或极端倾斜未被校正的情况下，将无法进行有效的扫描。

8.4. 快速（连续）扫描

在执行快速床面网格扫描时，由于每个点的样本采集时间较短，可能无法积累足够的样本进行平均和去噪。因此，快速扫描的准确性可能不如标准床面网格扫描。但通常情况下，它仍能提供满意的首层效果。

可以通过允许 travel planner 略微超出扫描的床面网格并平滑移动来改善快速扫描。您可以在 bed_mesh 配置部分使用 scan_overshoot: parameter。请注意，您需要确保轴可以移动到网格边界加上此超出值的区域，因此请小心不要指定过高的值。通常情况下，8mm 足够。

九、FAQ 常见问题解答

1. 有时我收到 “Error during homing probe: Eddy current sensor error” 报错

- 这通常表示 Eddy 中的传感器在探测/归零尝试开始前没有达到有效值。我们建议尝试以下步骤：
 - 1) 仔细检查 Eddy 高度。可能 Eddy 离床面太近或太高。我们建议当喷嘴刚触及床面时, Eddy 高度应为 2mm-3mm。大多数情况下, 2.5mm 是最优的, 但如果您发现 Eddy 在高温下出现错误, 可以尝试将其略微降低到 2mm 以下。然而, 如果 Eddy 在 QGL 尝试过程中出现报错, 您可能需要稍微提高 Eddy 高度。
 - 2) 调整 Eddy 高度后, 从配置文件中删除所有校准设置, 并重新校准 Eddy。
 - 3) 如果仍然收到此报错, 如果 `reg_drive_current` 值设置为 15, 请将其增加到 16。

2. 有时我收到 “Probe Triggered Before Movement” 报错

- 这是因为您尝试执行两个连续的 PROBE 命令。请在两个 PROBE 命令之间将龙门架抬起几毫米, 以避免此问题。

3. Eddy 在运行床网格校准时执行 Z Hops

- 确保您使用了正确的宏调用: `BED_MESH_CALIBRATE METHOD=rapid_scan`
- 删除或修改 `KAMP - Adaptive Bed Mesh` 以及任何自定义的 `BED_MESH_CALIBRATE` 宏。请使用 Klipper 自适应网格, 或者不要在 `KAMP_Settings.cfg` 中包含 `KAMP/Adaptive_Meshing.cfg`。

4. Eddy, Eddy Duo 还是 Eddy Coil, 如何选择?

- 这取决于您的需求。Eddy, Eddy Duo 和 Eddy Coil 几乎相同, 但 Eddy Coil 更适用于工具头板, 通过 I²C 连接。

- Eddy Coil 没有温度补偿，因此在密封腔体内使用时可能不如 Eddy，Eddy Duo 稳定。此外，Eddy Duo 具有 USB 和 CAN 两种通讯方式。

5. 错误：gcode command < ANY GCODE COMMAND > already

registered 这通常是因为存在冲突的 G-code 宏。检查所有 G-code 宏，确保没有重复的名称，并解决这些冲突。通常情况下，如果出现冲突且不确定如何处理，请优先选择 Eddy 宏中的功能。

6. 我的 z-offset 似乎无法保存并会重置？

- 对于常见调平传感器，这可能看起来像一个 bug。然而，如果您正确校准了 Eddy 并使用了特殊的归零宏，那么 z-offset 是没有必要的。解释起来有些复杂，但本质上，当使用 Eddy 时，z-offset 参数不会调整喷嘴的打印高度，只会调整归零或探测触发的高度。
- 我们强烈建议通过 Eddy 进行校准以获得准确的喷嘴高度，但您也可以使用 [Z-offset beta 示例配置文件](#) 来模拟标准的 z-offset。只需取消注释与 beta z-offset 功能相关的宏，您就可以使用标准的 Mainsail 按钮来调整喷嘴高度，并将其保存为 z-offset。

7. 我的 Eddy 宏与 KNOMI 宏冲突

- Eddy 和 KNOMI 使用了类似的宏。Eddy 宏中已经包含了 KNOMI 所需的所有功能。如果有冲突，请注释掉有冲突的 KNOMI 宏，并使用 Eddy 宏中的功能。
- 请注意，您可能需要取消注释 Eddy 宏中的某些行，这些行是专门为 KNOMI 用户设置的。检查宏文件，查看哪些行已被注释，并在需要使用 KNOMI 时取消注释这些行。

8. KAMP 与 Eddy

- [KAMP aka Klipper-Adaptive-Meshing-Purging](#) 在使用 Eddy 之前需要从 Klipper 中移除。请在 KAMP_SETTINGS.cfg 文件中注释掉相关的包含行，例如 `#[include ../KAMP/adaptive_meshing.cfg]`。
- 自 2024 年 1 月起，KAMP 已被合并到 Klipper 中，您应在 BED_MESH_CALIBRATION 调用中使用 ADAPTIVE=1 选项。有关[自适应网格的更多信息，请查看相关文档](#)。

如果您还需要此产品的其他资源，可以到 <https://github.com/bigtreotech/> 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（service005@biqu3d.com）。

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！