BIGTREETECH

Eddy V1.0

User Manual

Revision Log

Version	Date	Revisions
v1.00	April 19th, 2024	Initial Version

TABLE OF CONTENTS

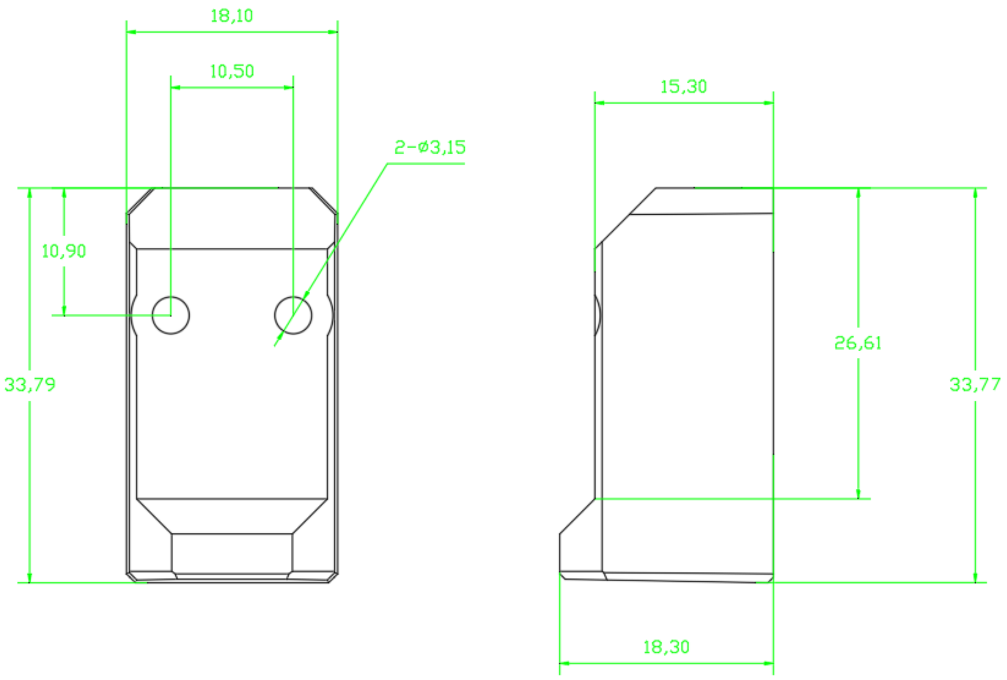
1. Product Information

Name	Eddy
Weight	6g
Voltage	5V
Static Current	30mA
Operating Current	30mA
Cable Length	2.5 m (USB Version), 15 cm (Coil Version)
Connection	USB: 4-pin, 1.5mm pitch Coil: 4-2.54mm DuPont female header, one end with ZH1 5mm 4P connector
Operating Temperature	≤60℃ Ambient
Standard Error	5μm
Compatible Models	All FDM printers using the Klipper firmware.

2. Feature Highlights

- Compact size and lightweight;
- Equipped with temperature compensation;
- Highly efficient leveling;
- Broad application, strong compatibility;
- High precision, strong stability;
- Non-contact operation.

3. Product Dimensions



Dimensional Diagram

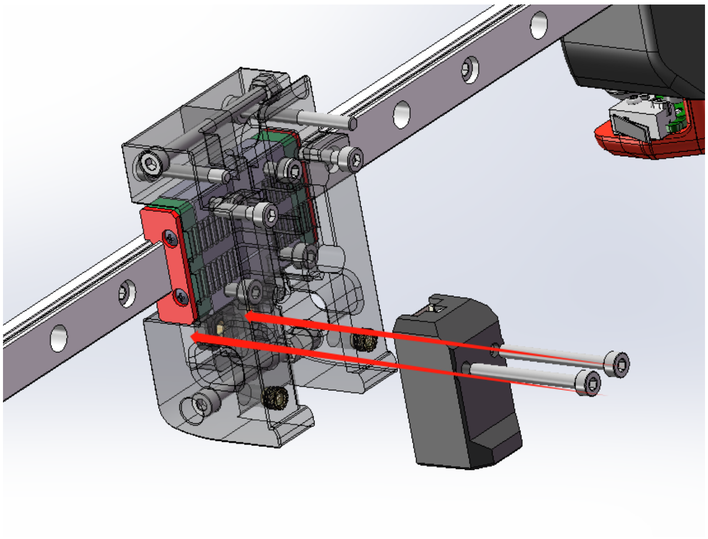
Note: When installing Eddy, ensure the bottom is at least 1-2 mm above the nozzle.

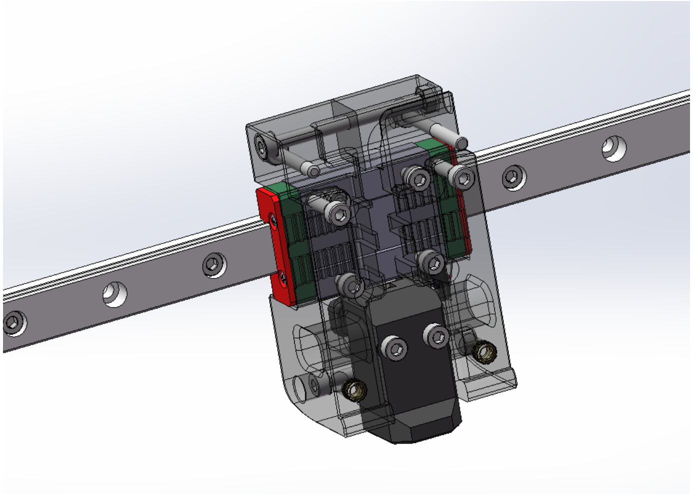
4. Installation Guide

4.1. Example using Voron 2.4

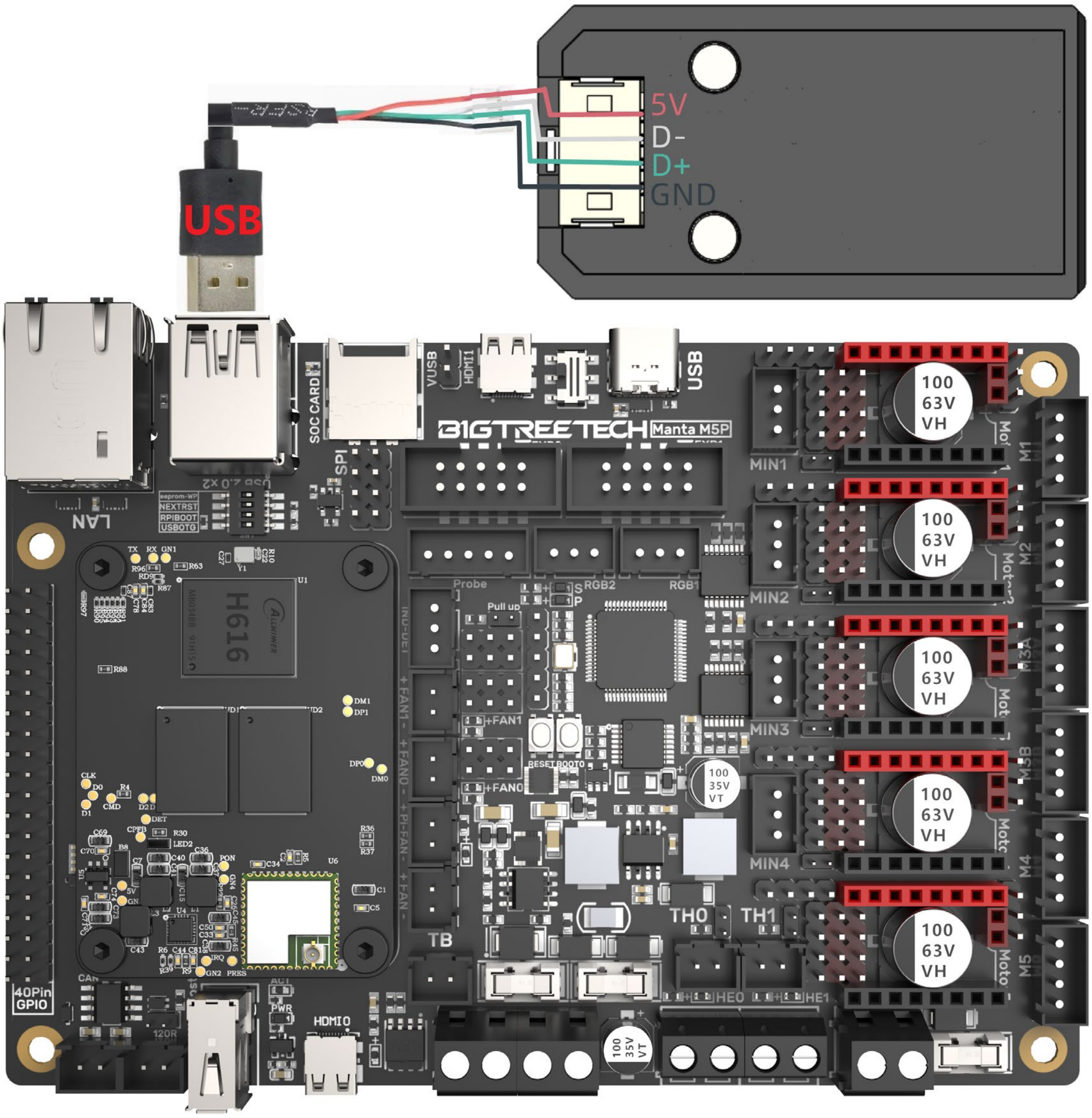
Installation replaces the original PL-08N position.

Use two M3*25 screws (included in the package) to secure the Eddy to the X Carriage as shown in the diagram.

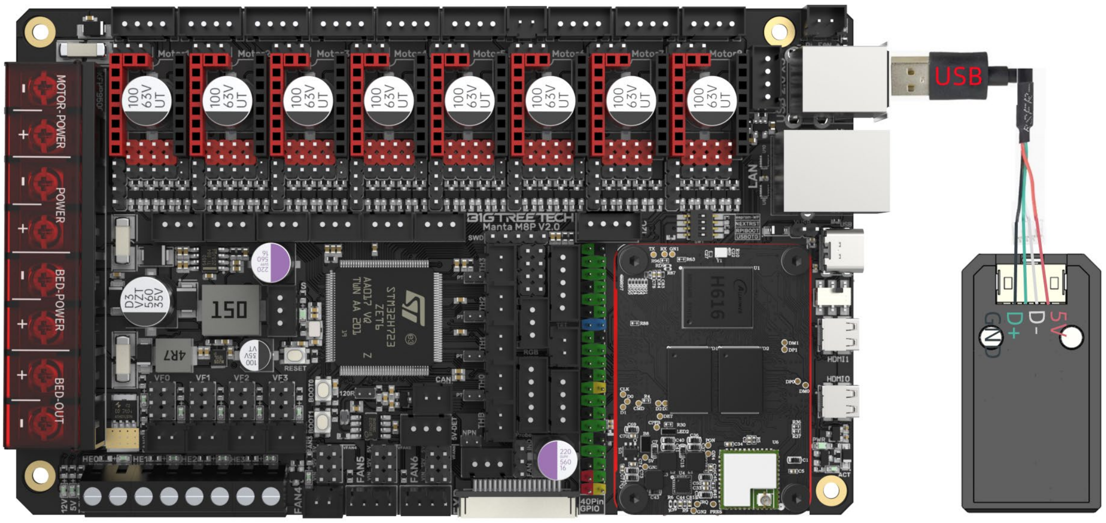




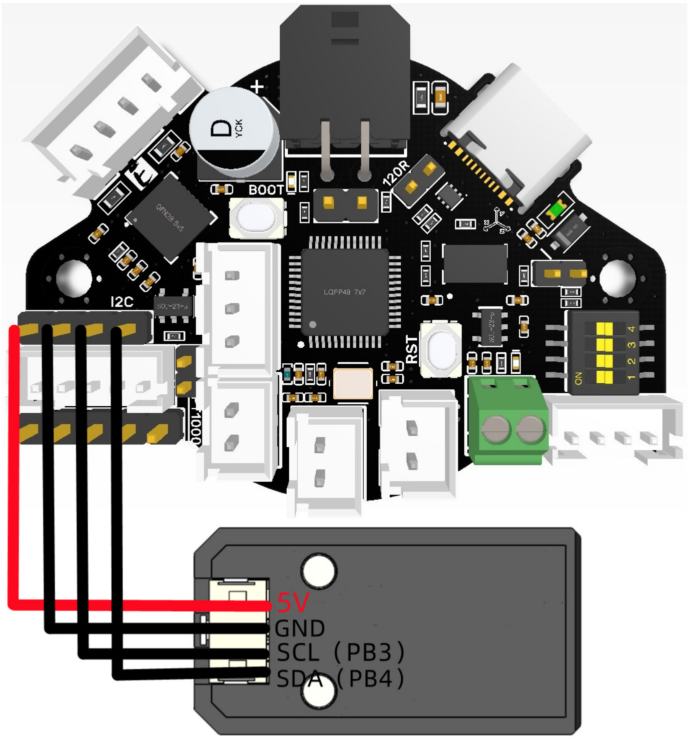
4.2. Eddy + Manta M5P



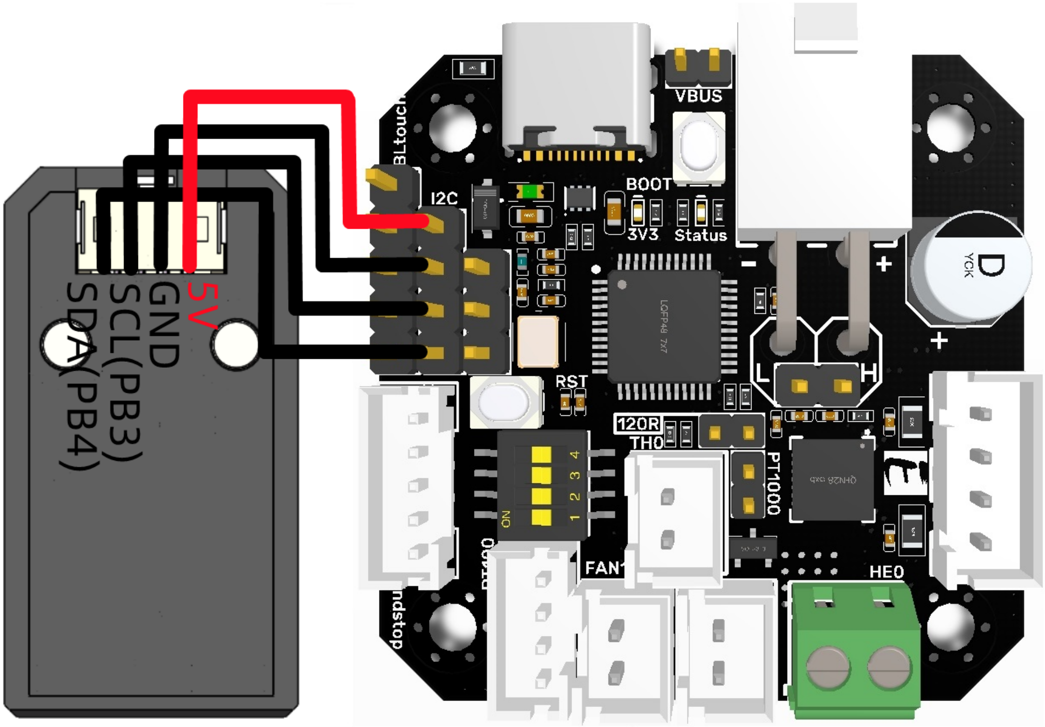
4.3. Eddy + Manta M8P V2.0



4.4. Eddy Coil + EBB36 V1.2



4.5. Eddy Coil + EBB42 V1.2



5. Firmware

5.1. Important Notes

When Eddy performs temperature compensation, the heated bed temperature can be high. Please be cautious to avoid burns.

5.2. Compiling Firmware

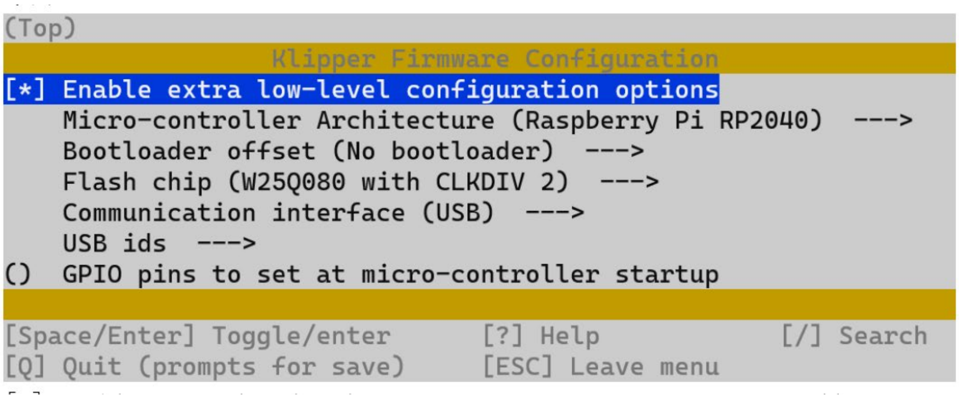
The USB version requires firmware update, while the coil version needs firmware update based on the module or motherboard it is attached to.

1. After SSH connects to Raspberry Pi, enter the following in the command line:

cd ~/klipper/

make menuconfig

Compile the firmware using the configuration below (if these options are not available, update the Klipper firmware source code to the latest version).

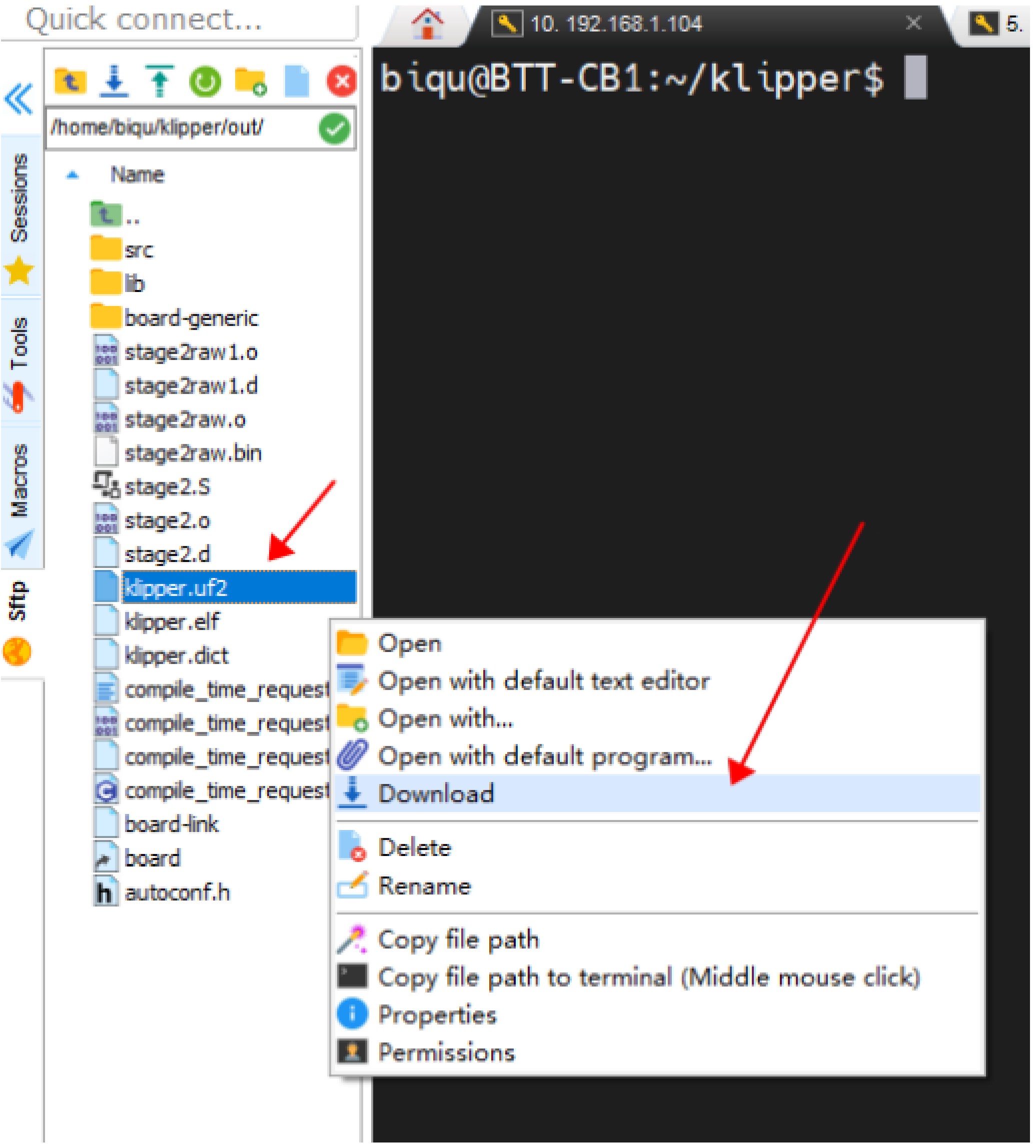


```
(Top)
                    Klipper Firmware Configuration
[*] Enable extra low-level configuration options
    Micro-controller Architecture (Raspberry Pi RP2040)  --->
    Bootloader offset (No bootloader)  --->
    Flash chip (W25Q080 with CLKDIV 2)  --->
    Communication interface (USB)  --->
    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. After configuring, enter ‘q’ to exit the configuration interface. When asked to save configuration, select ‘Yes’.

3. Enter make to compile the firmware. When make is completed, the required klipper.uf2 firmware will be generated in the home/pi/klipper/out folder and can be directly downloaded to the computer on the left side of the SSH software.

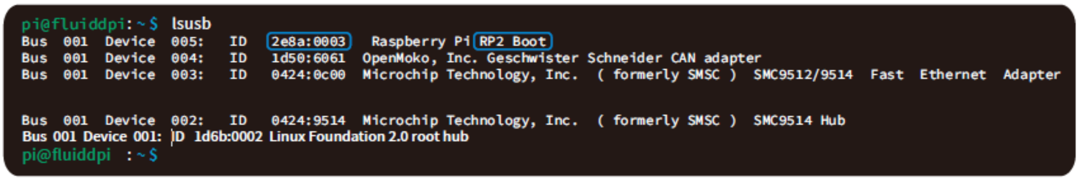


```
biqu@BTT-CB1:~/klipper$
```

5.3. Firmware Update via DFU

1. Hold down the Boot button, then connect the power supply to enter DFU mode.

2. In the SSH terminal command line, enter lsusb to query the DFU device ID.



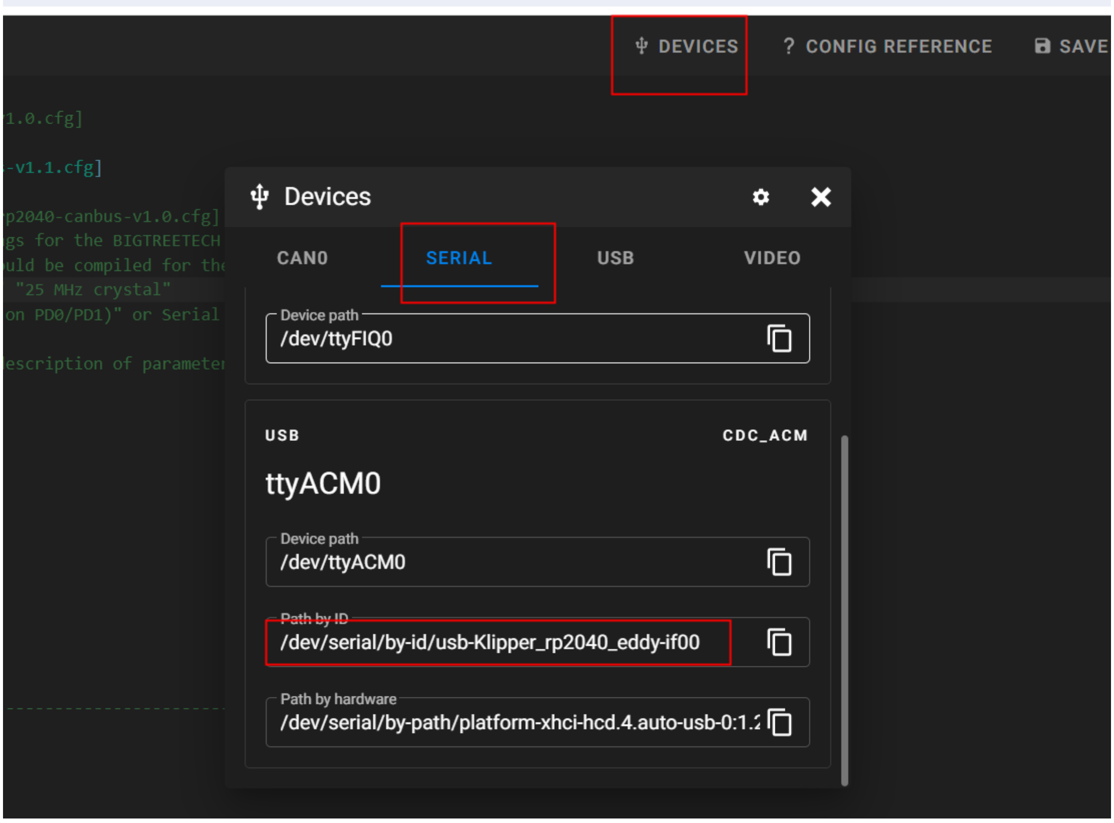
```
pi@fluiddpi:~$ lsusb
Bus  001  Device  005:   ID   2e8a:0003   Raspberry Pi RP2 Boot
Bus  001  Device  004:   ID   1d50:6061  OpenMoko, Inc. Geschwister Schneider CAN adapter
Bus  001  Device  003:   ID   0424:0c00  Microchip Technology, Inc.  ( formerly SMSC )  SMC9512/9514  Fast  Ethernet  Adapter


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@fluiddpi  :~$
```

3. Enter cd klipper to navigate to the klipper directory, then enter
make flash FLASH_DEVICE=2e8a:0003
to start flashing the firmware (note: replace 2e8a:0003 with the actual device ID obtained in the previous step).
4. After flashing, enter ls /dev/serial/by-id/*， in the command line to query the USB communication ID. You can then update this ID in the Klipper configuration file under the serial section.

After completing the steps above, the USB communication should be functioning normally.

You can also search for the USB communication ID directly in Mainsail.



Device path
/dev/ttyFIQ0

USB CDC_ACM

ttyACM0

Device path
/dev/ttyACM0

Path by ID
/dev/serial/by-id/usb-Klipper_rp2040_eddy-if00

Path by hardware
/dev/serial/by-path/platform-xhci-hcd.4.auto-usb-0:1.2

5.4 Klipper

5.4.1 (USB) MAIN Configuration

1. Execute the following commands in the SSH terminal:

```
cd ~/klipper/

git remote add eddy https://github.com/bigtreetech/klipper

git fetch eddy

git checkout eddy/eddy
```

2. Configure eddy in printer.cfg:

```
[mcu eddy]
```

serial: /dev/serial/by-id/ (the actual ID found using the above method or in Mainsail)

```
[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 # Simply avoid setting it to 0.
i2c_mcu: eddy
i2c_bus: i2c0f
x_offset: 0 # Set according to the actual offset relative to the nozzle
y_offset: 20 # Set according to the actual offset relative to the nozzle
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) MAIN Configuration

1. Execute the following commands in the SSH terminal

```
cd ~/klipper/
git remote add eddy https://github.com/bigtreetech/klipper
git fetch eddy
git checkout eddy/eddy
```

2. Configure eddy in printer.cfg:

```
[mcu eddy]
serial: /dev/serial/by-id/usb-Klipper_stm32g0b1xx_3D0047001150425539393020-if00

[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 # Simply avoid setting it to 0.
i2c_mcu: eddy
i2c_bus: i2c3_PB3_PB4 # Write according to the actual I2C port pins of the motherboard or expansion module in use.
x_offset: 40 # Set according to the actual offset relative to the nozzle
y_offset: -30 # Set according to the actual offset relative to the nozzle
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

# The speed (in mm/s) of non-scanning moves during the calibration

horizontal_move_z: 5

# The height (in mm) that the head should be commanded to move to just prior to start the scanning operation.

mesh_min: 10, 10

# Defines the minimum X, Y coordinate of the mesh for rectangular beds. This coordinate is relative to the EDDY's location. This will be the first scanning point, nearest to the origin. This parameter must be provided for rectangular beds.

mesh_max: 220, 220

# Defines the maximum X, Y coordinate of the mesh for rectangular beds. Adheres to the same principle as mesh_min, however this will be the furthest point scanning from the bed's origin. This parameter must be provided for rectangular beds.

probe_count: 5, 5

# For rectangular beds, this is a comma separate pair of integer values X, Y defining the number of points to probe along each axis. A single value is also

valid, in which case that value will be applied to both axes.

Refer to https://www.klipper3d.org/Config_Reference.html#bed_mesh

For the [bed_mesh], set horizontal_move_z to 2 to keep Eddy as close as possible to the bed during scanning.

5.5 Temperature Compensation

1. After completing the above configuration, the first step is to calibrate the drive current of Eddy:

Eddy Drive Current Calibration

Place Eddy approximately 20mm above the bed.

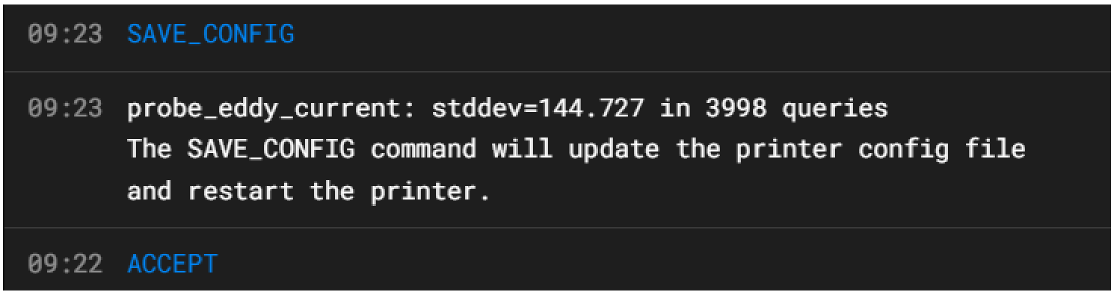
Execute LDC_CALIBRATE_DRIVE_CURRENT CHIP=btt_eddy to automatically acquire the drive current, then use SAVE_CONFIG to save the parameters.

2. Eddy Frequency and Z-Axis Height Relationship Calibration

Home X and Y axes with G28 X Y.

Move the nozzle to the center of the bed with G0 X150 Y150 F6000 (ensure there is no height map activated during this step).

Start the manual Z-offset calibration with PROBE_EDDY_CURRENT_CALIBRATE CHIP=btt_eddy. Once calibration is completed, use SAVE_CONFIG to save the settings as shown.



```
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
```

For Voron printers, perform a Quick Gantry Leveling (QGL) to ensure the gantry is level and prevent the nozzle from rubbing against the heated bed during the subsequent mesh scan. Once this is done, you can home all axes with home all, and then conduct a quick bed mesh scan using the following command: BED_MESH_CALIBRATE METHOD=scan SCAN_MODE=rapid

After completing the scan, save the settings using SAVE_CONFIG.

3. Temperature Compensation (Eddy coil without temperature compensation)

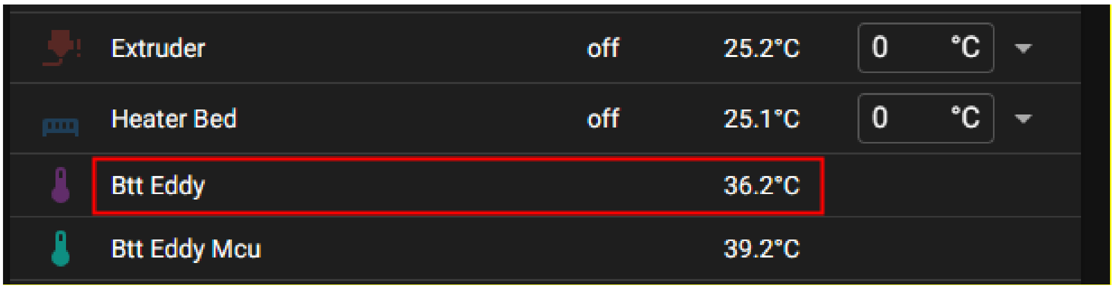
(1) Home All Axes: Use home all to home all axes.

(2) Set Idle Timeout: Execute SET_IDLE_TIMEOUT TIMEOUT=36000. This increases the idle timeout to prevent the machine from timing out during the heating process.

(3) Record the ambient temperature of the BTT eddy sensor.

Set a maximum temperature for the heated bed and set a typical temperature for the tool head.

Wait for the BTT eddy temperature to stabilize at this set maximum temperature and record it.



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	

The recorded maximum BTT eddy temperature serves as the target temperature for subsequent operations. Ensure that the target temperature does not exceed this maximum, ideally being one or two degrees lower to maintain safe and effective operation.

(4) After returning to room temperature, execute PROBE_DRIFT_CALIBRATE PROBE=btt_eddy TARGET=50 STEP=5. Here, "TARGET=50" refers to reaching a target temperature of 50°C, and "STEP=5" indicates that each node's temperature increment is 5°C.

Be cautious: the maximum temperature of the heated bed can be very high. Take care to avoid burns.

For example, if the current temperature is 30°C and the target temperature is 50°C, the total temperature range would be 50-30=20°C, with each node representing a 5°C increment. Therefore, there will be a sampling of 20 / 5 = 4 nodes.

Executing the above command will immediately prompt for manual Z-offset calibration. Then manually heat the bed and nozzle, waiting for the temperature to rise. Wait for the temperature of the BTT Eddy to rise. The BTT Eddy will prompt for another manual Z-offset calibration at the next node, which is 35°C. Then it will prompt for another manual Z-offset calibration at the next node, which is 40°C, and so on.

Note: In the example above, the temperature at the second node is 35°C. However, manual Z-offset calibration takes time, and the temperature may still be rising. If the calibration is completed when the actual temperature is 36°C, then the temperature for the next node will be increased by one degree, which is 41°C.

Should you require further resources for this product, you can find them at [GitHub](https://github.com/bigtreetech/). If you cannot find what you need, you may contact our after-sales support (service005@biqu3d.com).

If you encounter any other problems during use or have suggestions or feedback, please contact us. Thank you for choosing BIGTREETECH products.