BIGTREETECH

Eddy series

(Eddy, Eddy Coil, Eddy Duo)

User Manual

Revision Log

Version	Date	Revision
v1.00	April 19th, 2024	Initial Version
v1.01	April 25th, 2024	1. Mark the BOOT button's location on the product image. 2. Added instructions for restarting Klipper.
v1.02	April 26th, 2024	Include methods for updating firmware via computer.
v1.03	April 30th, 2024	Revised the configuration and calibration instructions.
v1.04	May 15th, 2024	Indicated the position of the center point.
v1.05	June 6th, 2024	Added the z_virtual_endstop instruction to Section 5.1: Important Notes.
v1.06	August 19th, 2024	Merged Klipper branch and adjusted Z-offset configuration.
v1.07	September 24th, 2024	Updated to include CAN communication instruction.

Table of Contents

1. Product Information

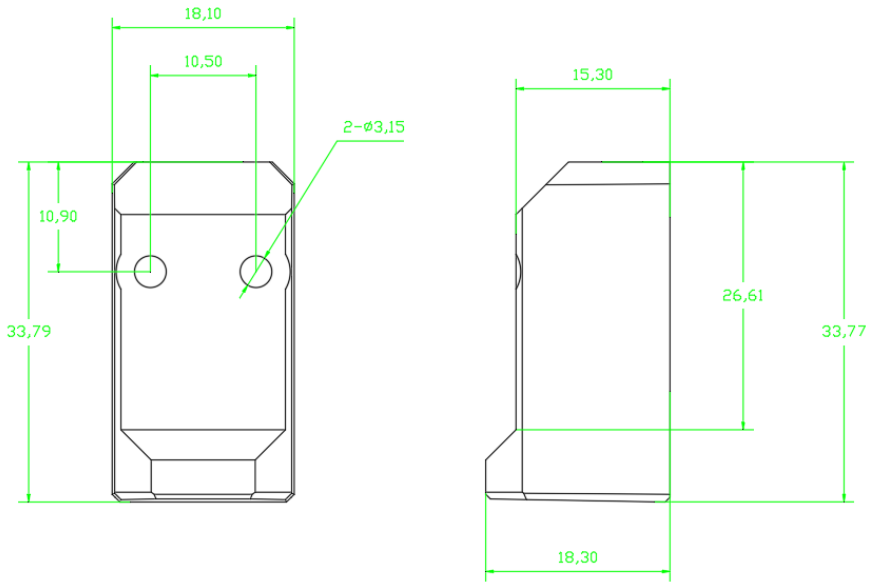
Name	Eddy, Eddy Coil, Eddy Duo
Weight	6g
Voltage	5V
Static Current	30mA
Operating Current	30mA
Cable Length	2.5 m (Eddy, Eddy Duo), 15 cm (Eddy Coil)
Connection	Eddy, Eddy Duo: 4-pin, 1.5mm pitch Eddy Coil: 4-2.54mm DuPont female header, one end with ZH1 5mm 4P connector
Operating Temperature	≤60℃ Ambient
Standard Error	0.5μm
Compatible Models	All FDM printers that use the Klipper firmware.

2. Feature Highlights

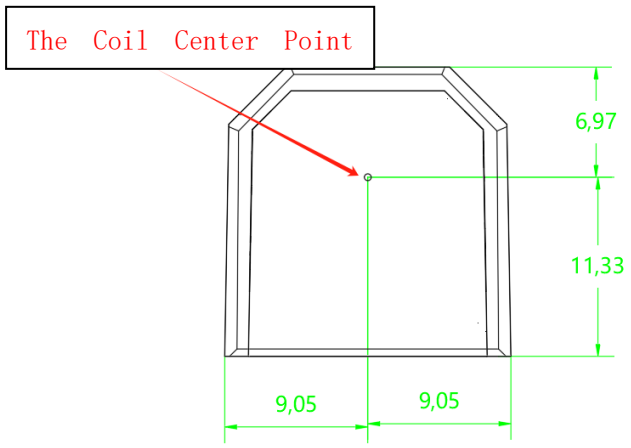
- Compact size and lightweight;
- Equipped with thermal compensation;
- Highly efficient leveling;
- Broad application, strong compatibility;
- High precision thermal stability;
- Non-contact operation.

3. Product Dimensions and Interfaces

3.1. Dimension Diagram



The coil's center point, used for calculating XY offsets, is as follows:



3.2. Location of the BOOT Button

The BOOT button is used when programming Klipper onto the Eddy for the first time; afterward, Klipper can automatically reprogram the RP2040.

Note: Only Eddy, Eddy Duo has the BOOT function; the button on Eddy Coil is non-functional and will not operate.

4. Installation Guide

4.1. Installation Height

Install Eddy 2 to 3 mm above the nozzle to ensure optimal performance. If you encounter any errors during the calibration process, these may be related to Eddy's installation height. For solutions, please refer to the troubleshooting section of this manual.

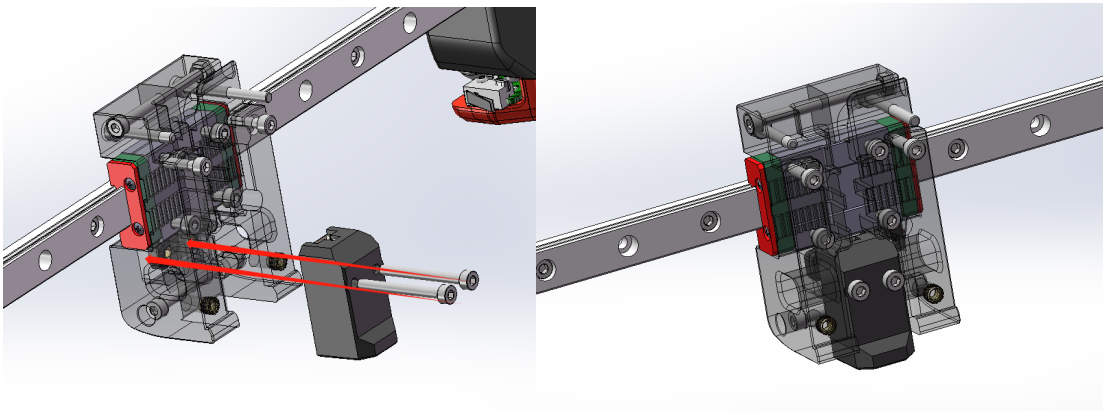
Important:

Some people confuse the current calibration height of 20mm with the mounting height of 2-3mm. The 20mm height is only used when calibrating the coil current later in this guide.

4.2. Example using Voron 2.4

Installation replaces the original PL-08N position.

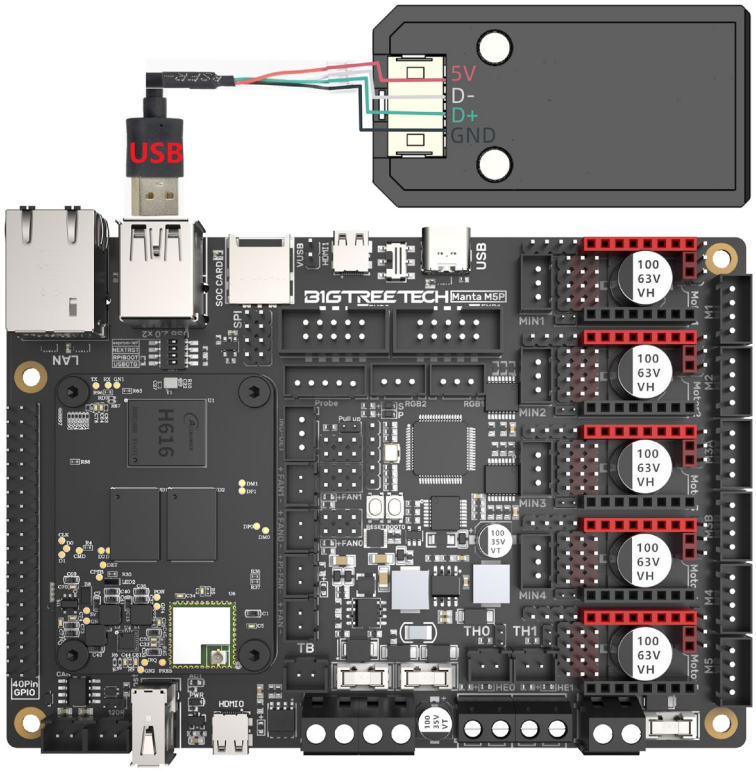
Use two M3x25 screws (included in the package) to secure the Eddy to the X Carriage, as shown in the diagram.



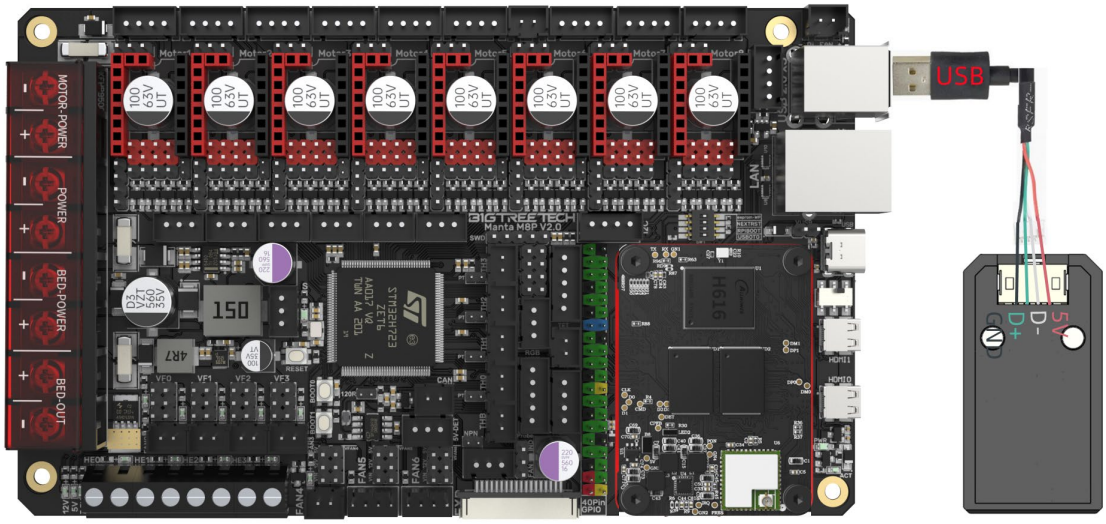
4.3. Installation on Other Machines

Brackets for various common machines are available in our GitHub repository and other popular model-sharing platforms. When installing Eddy, make sure the side with the PCB (the rear side) is positioned as far from the hotend as possible. This placement helps reduce heat transfer from the hotend to Eddy.

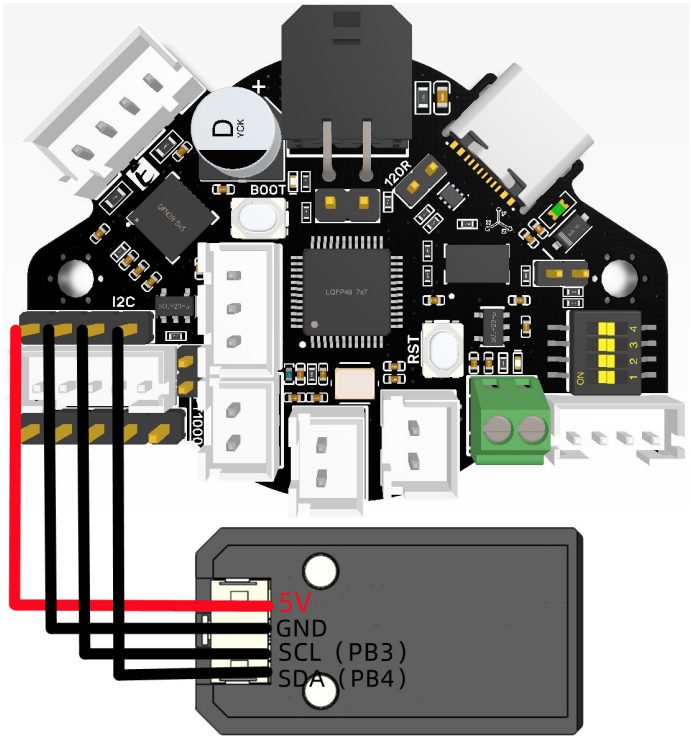
4.4. Eddy + Manta M5P



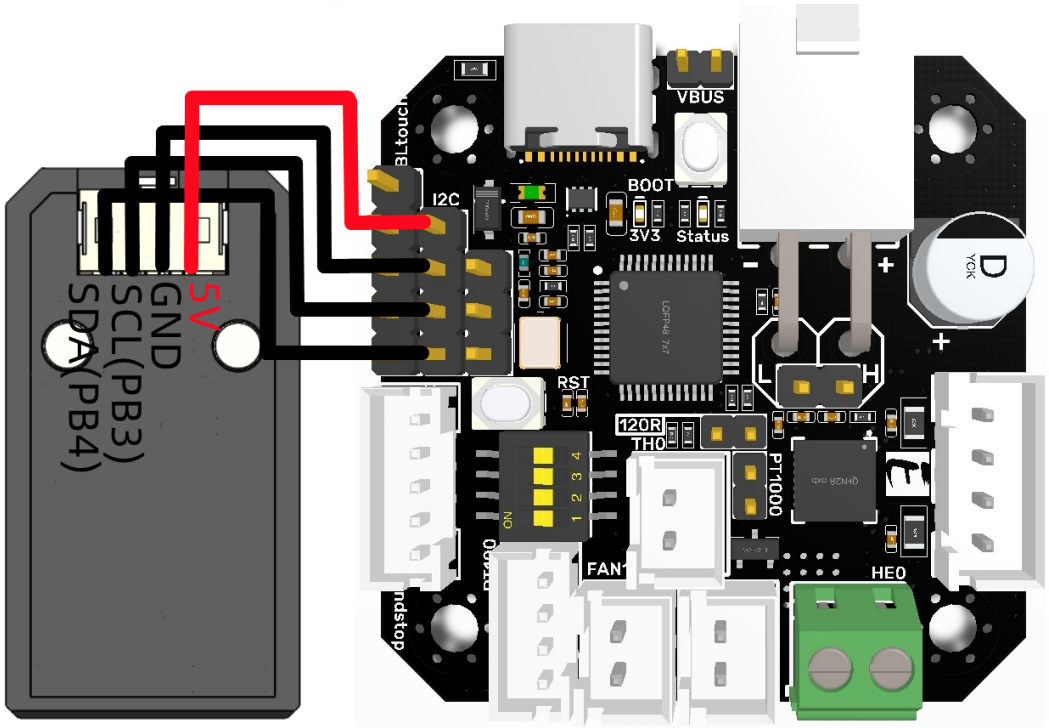
4.5. Eddy + Manta M8P V2.0



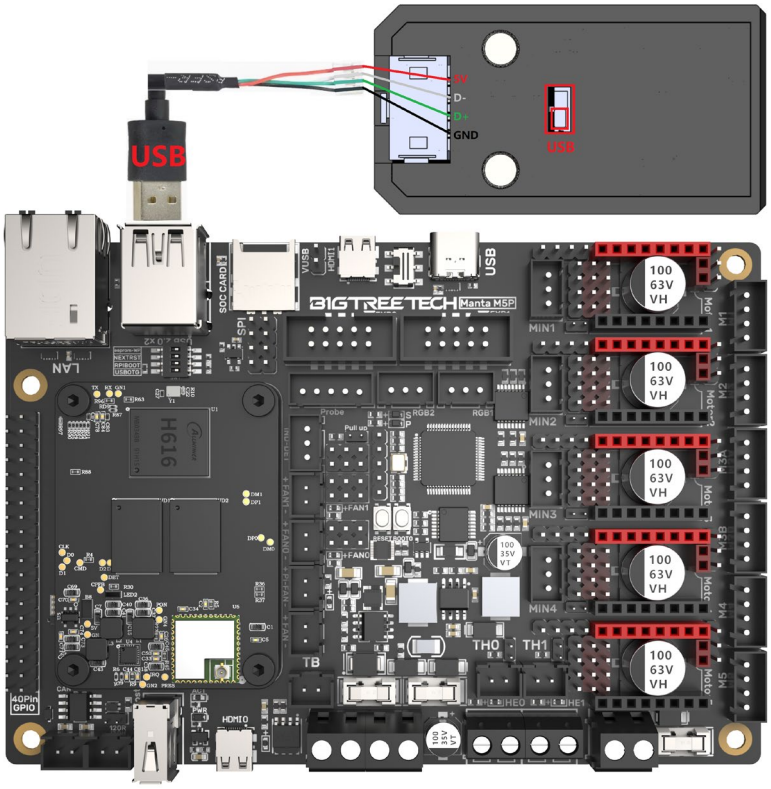
4.6. Eddy Coil + EBB36 V1.2



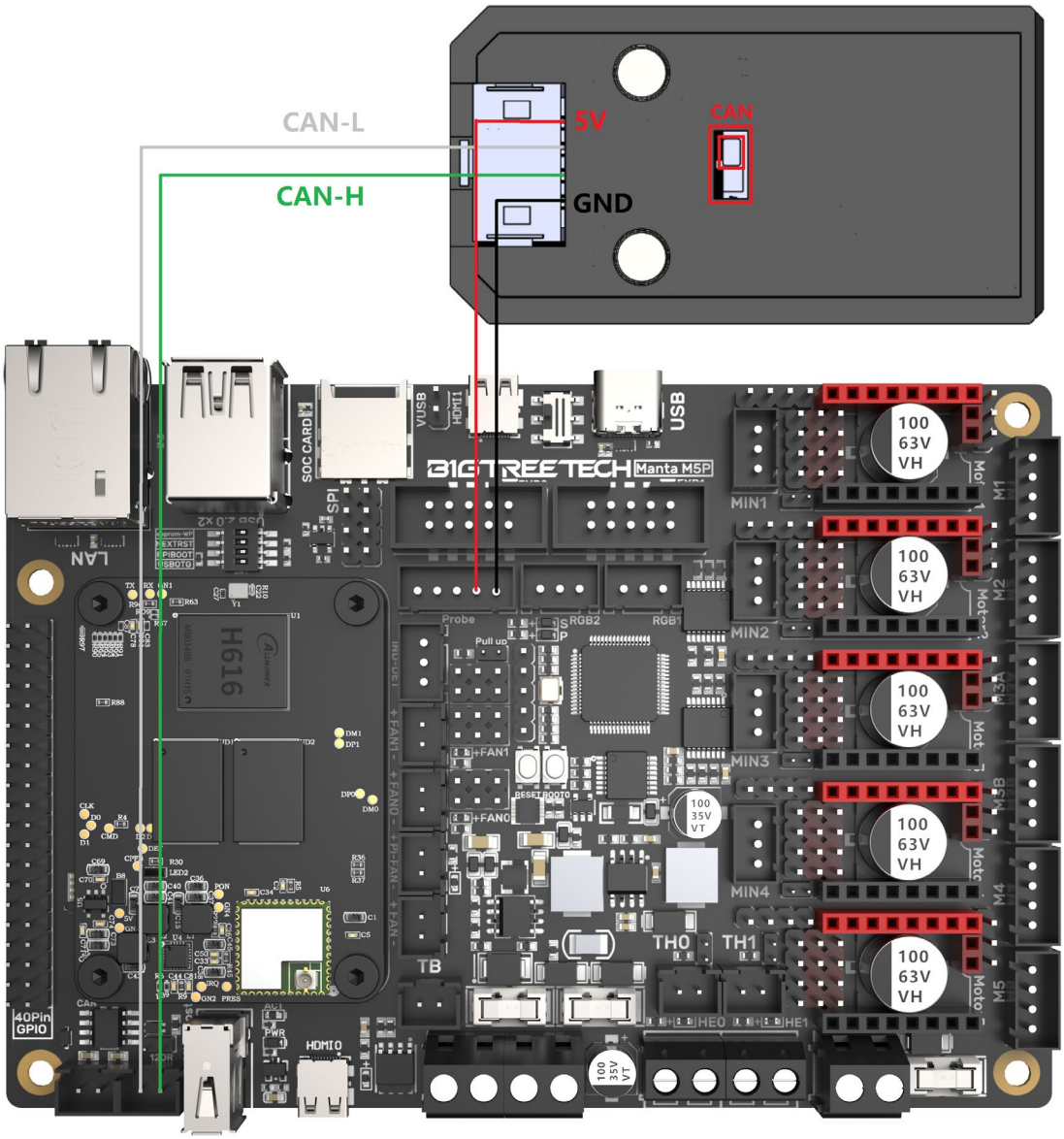
4.7. Eddy Coil + EBB42 V1.2



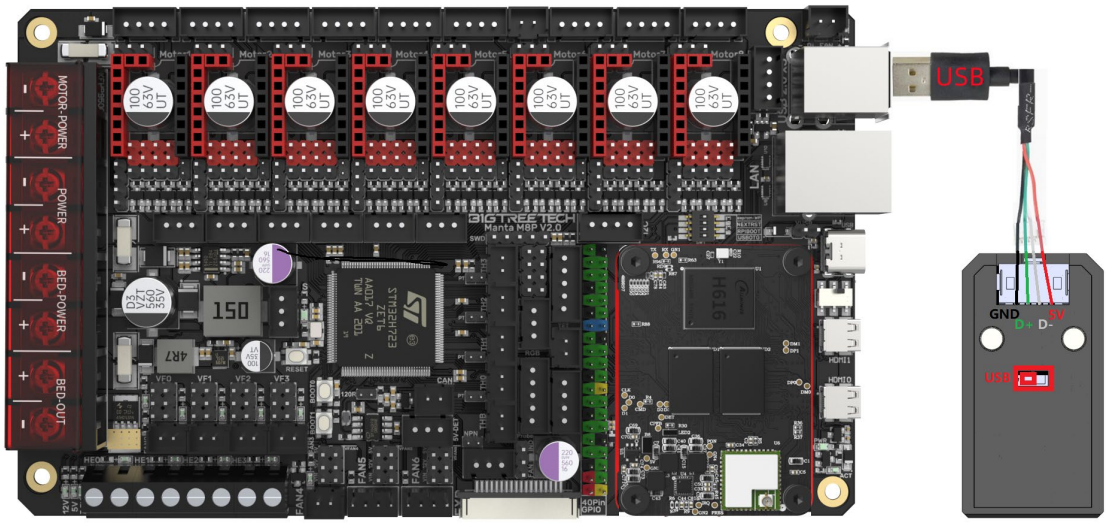
4.8. Eddy Duo + MANTA M5P（USB）



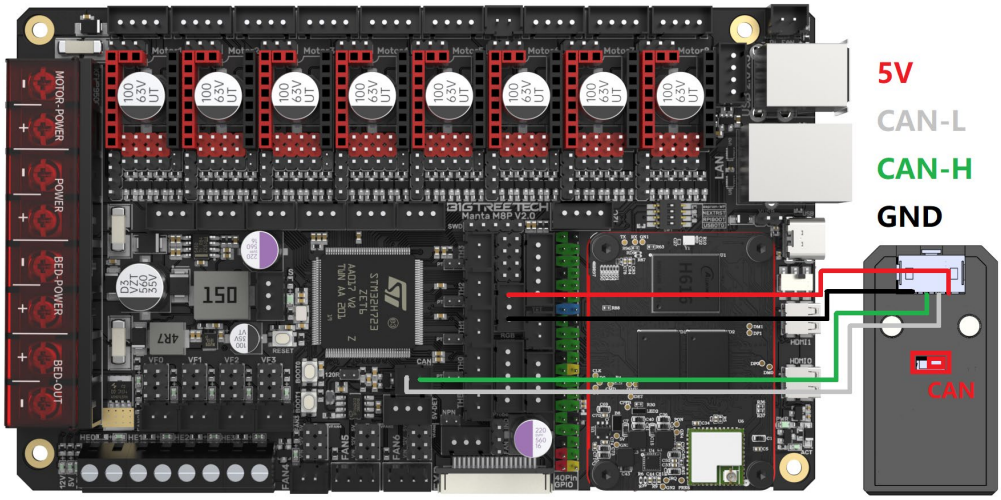
4.9. Eddy Duo + MANTA M5P（CAN）



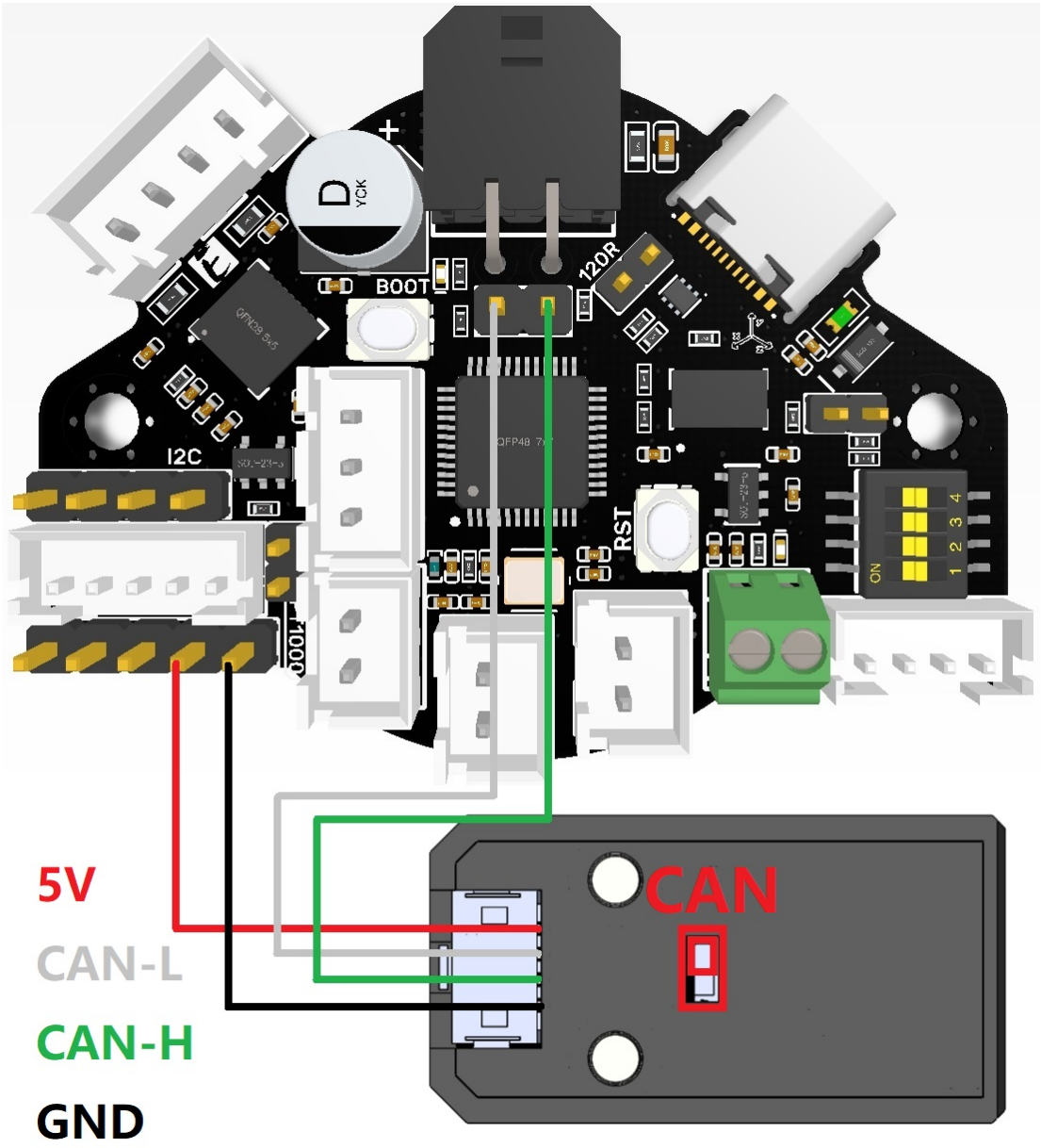
4.10. Eddy Duo + MANTA M8P V2.0（USB）



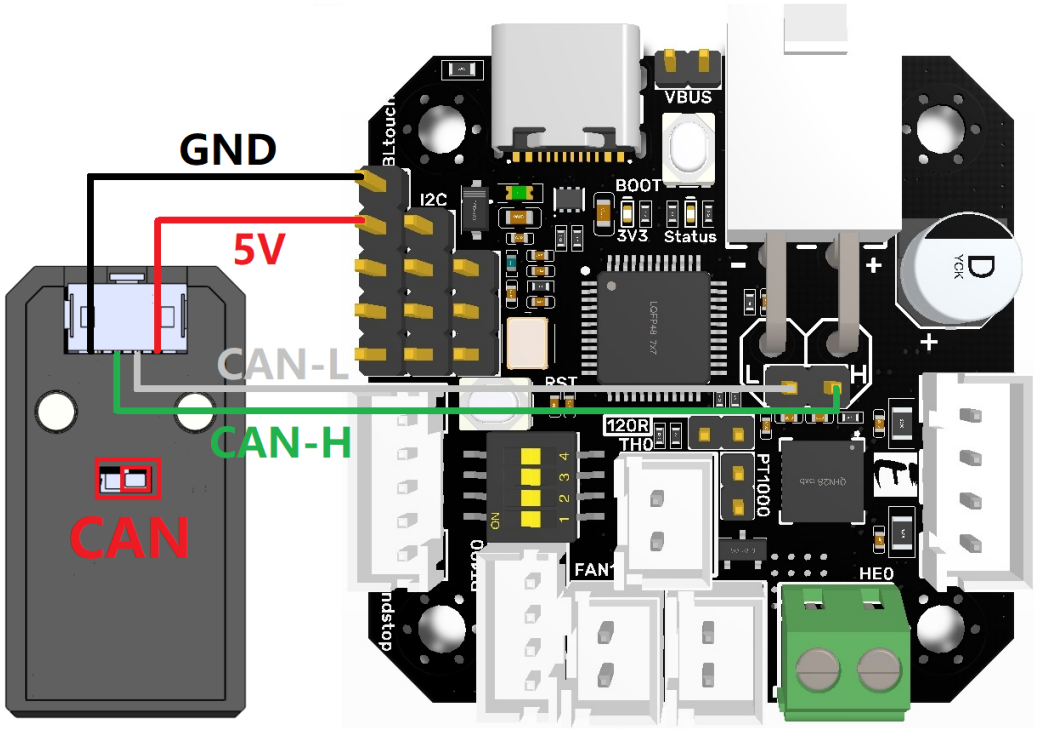
4.11. Eddy Duo + MANTA M8P V2.0（CAN）



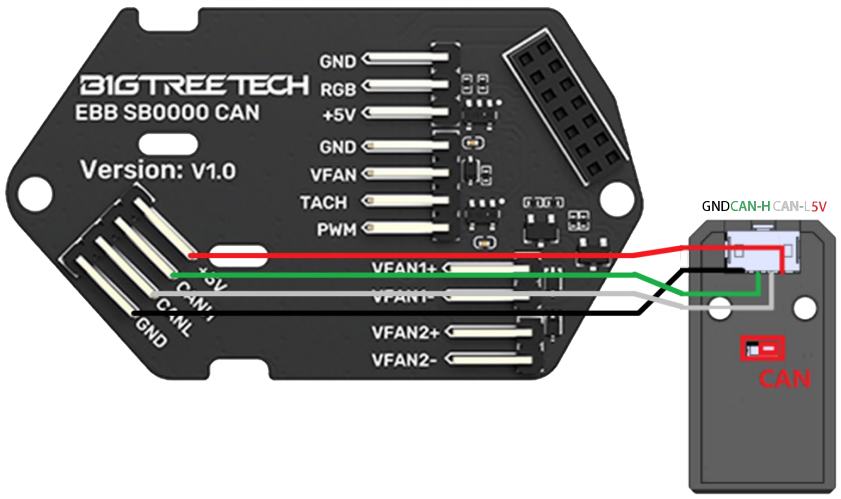
4.12. Eddy Duo + EBB36



4.13. Eddy Duo + EBB42



4.14. Eddy Duo + EBB SB



5. Firmware

Important:

1. Eddy and Eddy Coil are ONLY compatible with Klipper installations that use a virtual environment based on Python 3. Even if Python 3 is installed on your system, this is not an indication of your klippy virtual environment being created using Python 3. If you receive an error that says something like: Internal error during connect: split() takes no keyword arguments then you have a klippy host that is based on a Python 2 virtual environment and you need to get it upgraded. The easiest way to do this is to use KIAUH which will allow you to reinstall the klippy host without overwriting your configs when selecting the Python 3 option.
2. Eddy Duo supports two communication methods: USB and CAN. When switching from USB communication to CAN communication, it is recommended to update the firmware required for CAN communication while still in USB communication mode. This simplifies the process, as updating via DFU is easier than using KATAPULT.

5.1. Compiling Firmware

The firmware compilation instructions below only apply to the Eddy, Eddy Duo. If you are using an Eddy Coil, you will have it connected to the I2C port on a toolboard. You will need to compile firmware for that toolboard using the master Klipper branch and then install it onto that same toolboard. When configuring the Eddy within Klipper, you will just need to specify that it does communicate using the I2C port on that toolboard which will depend on the pins for that board.

If you are coming from the old BIGTREETECH branch of Klipper then we recommend using KIAUH to move back to the mainline branch. We also recommend updating the firmware on all of your Klipper devices so that it too is running on a binary compiled from mainline.

1. Ensure that you are using mainline klipper by typing the following commands via SSH:

cd ~/klipper

git checkout master

2. Next, type:

make menuconfig

3. Use these settings to compile the firmware.

(1) Eddy V1.0 (USB)

```
(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
```

[*] Enable extra low-level configuration options Micro-controller

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

(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) --->

If not using KATAPULT

Bootloader offset (No bootloader) --->

Flash chip (W25Q080 with CLKDIV 2) --->

If using KATAPULT (first carry out the procedure in section 5.4. Flashing KATAPULT)

Bootloader offset (16KiB bootloader) --->

If using USB communication

Communication interface (USB) --->

If using CAN bus communication

Communication interface (CAN bus) --->

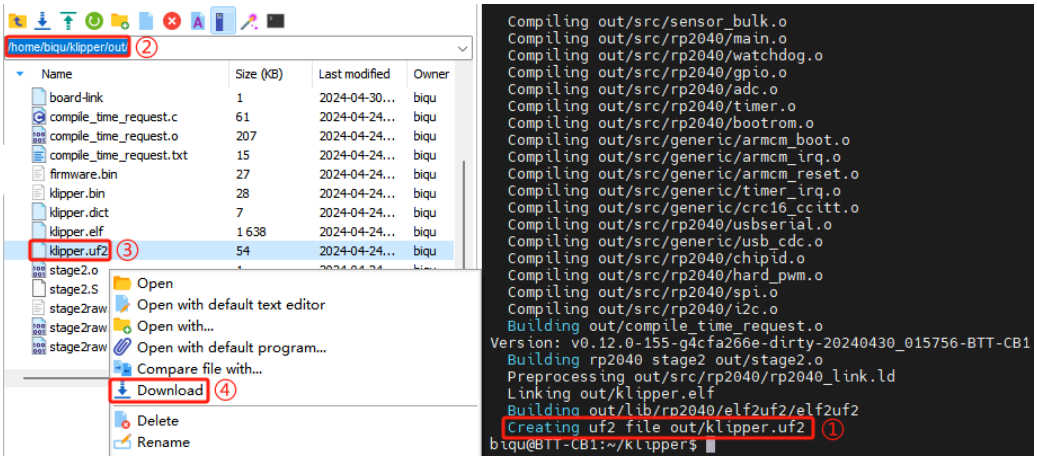
(4) CAN RX gpio number

(5) CAN TX gpio number

(1000000) CAN bus speed

4. Once set, hit 'Q' and when asked, select yes to save.

5. 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.



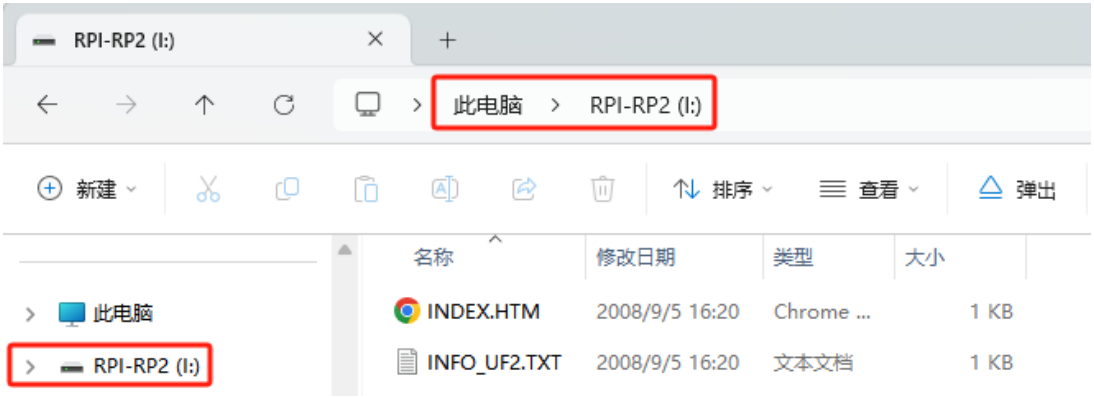
5.2. Update Firmware

5.2.1. Update Firmware via Computer

1. Press and hold the Boot button, then connect Eddy to your computer's

USB port using a USB cable (Do not disassemble your Eddy. The button is fully accessible without disassembly. The image is shown in an exploded view merely to make the button easier to see and should not be used as an example.)

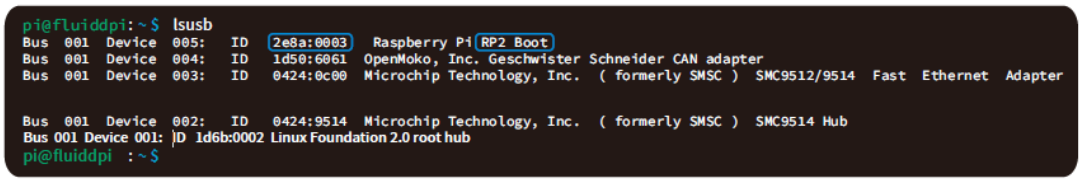
2. Once recognized as a storage device, copy the downloaded klipper.uf2 into it. Eddy will automatically update its firmware and restart. The update will be complete after the restart.



5.2.2. Update Firmware via DFU

1. Press and hold the Boot button, then connect Eddy to the USB port of your Raspberry Pi/BIGTREETECH Pi using a USB cable. (Do not disassemble your Eddy. The button is fully accessible without disassembly. The image is shown in an exploded view merely to make the button easier to see and should not be used as an example.)

2. Type lsusb into the command line. You should see Eddy.



3. Type cd ~/klipper into command line.
4. Type make flash FLASH_DEVICE=2e8a:0003 Remember to change 2e8a:0003 to your device ID you found in step 2.
5. Type ls /dev/serial/by-id/* into the command line. The found device should be what you enter into your klipper config under [mcu eddy] for the Serial variable.

5.3. Switching from USB Communication to CAN Communication

1. While in USB communication mode, compile the firmware for CAN communication.

Since you already flashed with USB firmware previously, you do not need

press the Boot button to flash.

2. While connected via USB, simply run the following command

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

to flash the firmware (Note: Replace /dev/serial/by-id/xxx with the actual USB communication ID found). To locate this ID, use the following command: ls /dev/serial/by-id/*

Then, switch the Eddy Duo DIP switch to the CAN communication mode.

3. CAN Bus Configuration

(1) In the SSH terminal, enter

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

and add the following content:

```
allow-hotplug can0
iface can0 can static
    bitrate 1000000
    up ifconfig $IFACE txqueuelen 1024
```

Set the CAN bus speed to 1M (speed must match the speed set in the firmware (1000000) CAN bus speed), save the changes (Ctrl + S) and exit (Ctrl + X), then enter

sudo reboot

to restart Raspberry Pi.

(2) Each device on the CAN Bus will generate a canbus_uuid based on the MCU's UID. To find each microcontroller device ID, ensure the hardware is powered on and properly wired, then run:

~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0

(3) If an uninitialized CAN device is detected, the above command will report the device's canbus_uuid:

Found canbus_uuid=0e0d81e4210c

(4) If Klipper is already running and connected to this device, the canbus_uuid will not be reported.

At this point, CAN communication should be functioning normally.

5.4. Flashing KATAPULT (formerly CanBoot)

Note: Katapult is for direct firmware updates via CAN bus

Refer to the following instructions to download the Katapult project: https://github.com/Arksine/katapult

1. Enter cd ~

to go to the home directory, enter

git clone https://github.com/Arksine/katapult

to download the Katapult project, then enter

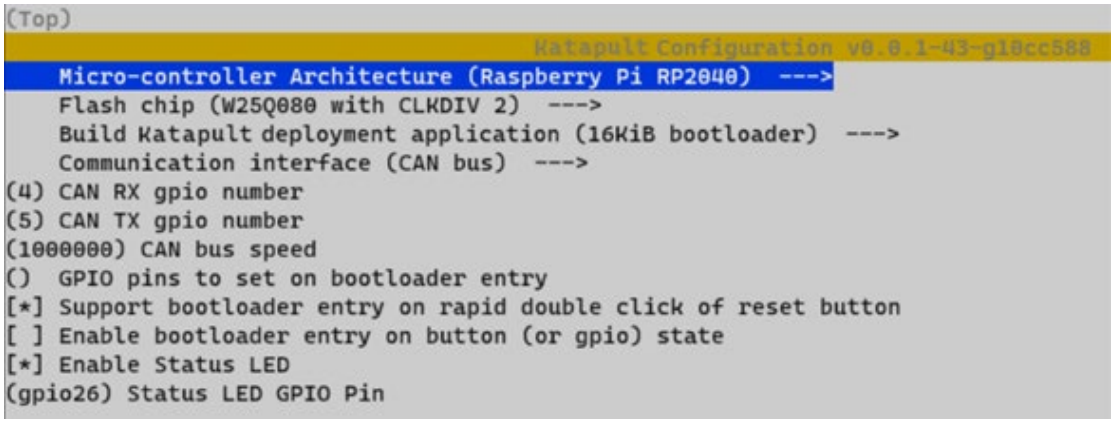
cd katapult

to navigate to the Katapult directory.

2. Enter

make menuconfig

and configure as shown in the image below.

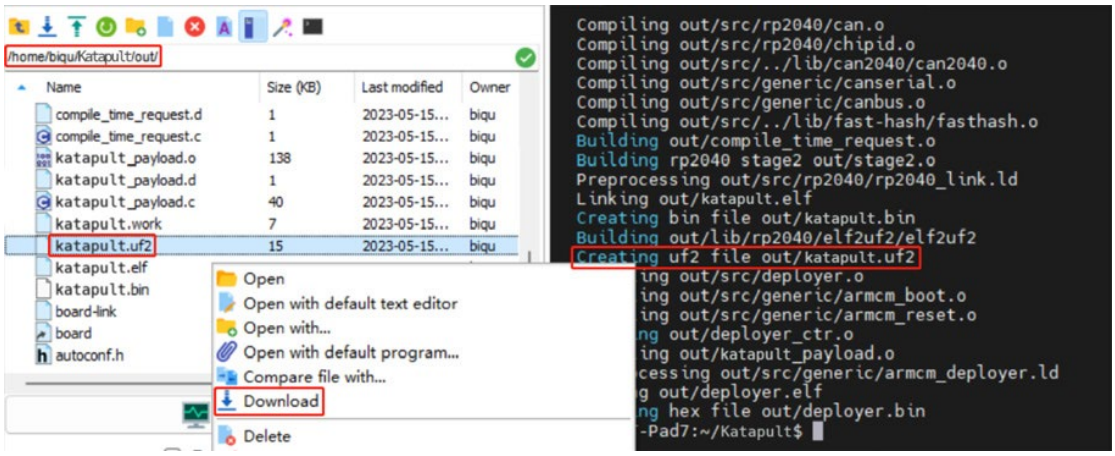


3. Enter ‘make’ to compile the firmware. When ‘make’ is completed, the required katapult.uf2 firmware will be generated in the home/biqu/Katapult/out folder.

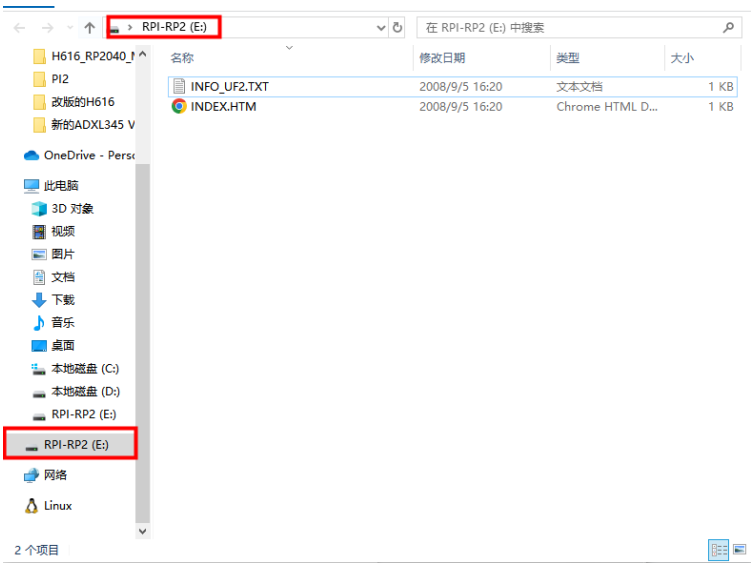
5.5. Firmware Update

5.5.1. Updating Firmware via a Computer

（1）In the SSH software on the left side, you may directly download it to the computer



（2）Connect the Eddy module to your computer via USB while pressing the Boot button. The device will show up as a storage drive.



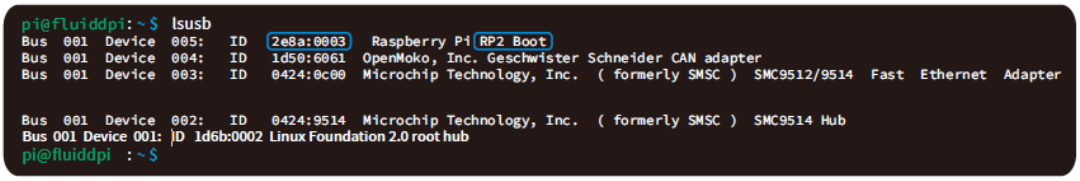
（3）Copy the klipper.uf2 file downloaded in the previous steps to this storage device to complete the firmware update.

5.5.2. Using DFU to Update Katapult on Eddy

（1）Press the Boot button and power on the device to enter DFU mode

（2）Use the following command to identify the DFU device

lsusb



（3）Flash Katapult by running: make flash FLASH_DEVICE=2e8a:0003

(Replace `2e8a:0003` with your device ID.)

Once the flashing is complete, follow the instructions in section 5.1 to compile the firmware. After compiling, you can update the firmware via KATAPULT.

5.6. Updating Firmware via Katapult

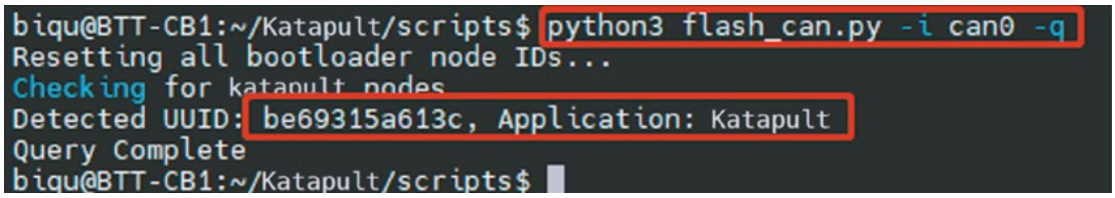
Note: The klipper.bin file mentioned below needs to be generated by ‘make’ in advance.

1. Ensure the CAN bus cable is connected, and the 120R termination resistor jumper is in place.

2. Input cd ~/katapult/scripts

Then input python3 flash_can.py -i can0 -q

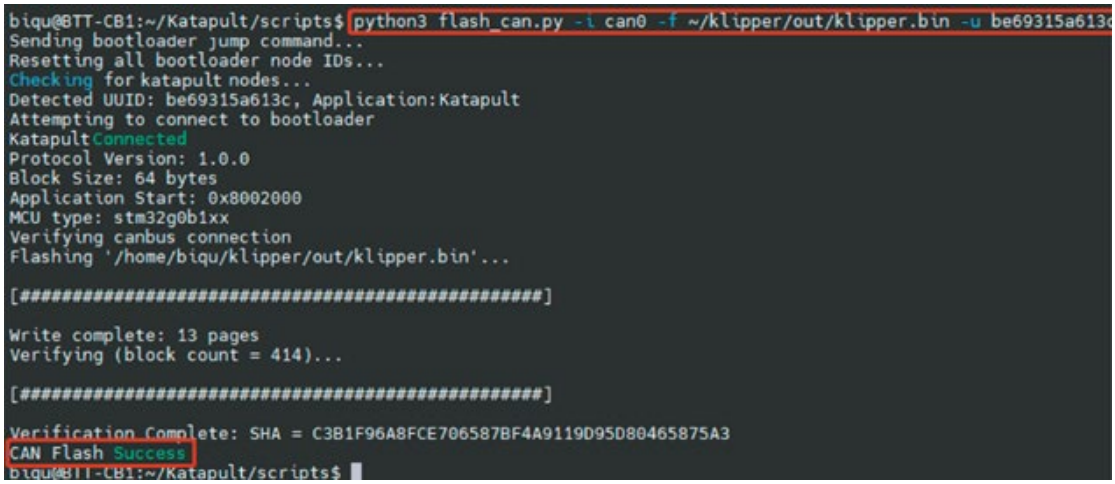
to query the CANbus ID (ensure the CAN cable is properly connected and powered beforehand). As shown in the image, the UUID of the device has been found.



3. Input

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

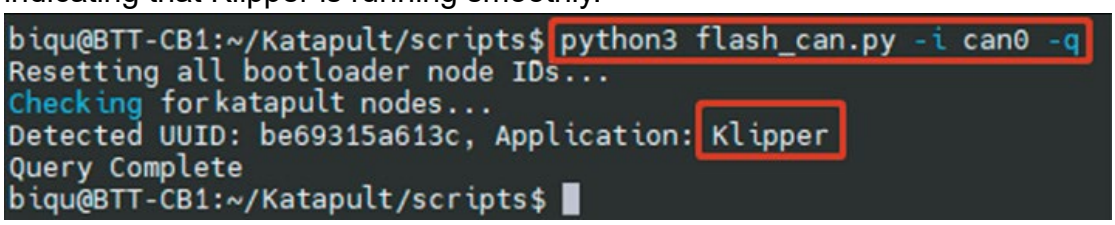
Replace with the actual UUID. Note: The Katapult application start offset is set to 16KiB; therefore, the Klipper bootloader offset in menuconfig should also be set to 16KiB, as shown in the image. The flashing process is successful.



4. Input again

python3 flash_can.py -i can0 -q

to query. At this point, the application has changed from Katapult to Klipper, indicating that Klipper is running smoothly.



6. Klipper & Eddy Configuration

6.1. Printer.cfg Templates

Now that you have the right firmware loaded onto your Eddy, it is time to complete the Klipper configuration. BIGTREETECH provides three different sample configuration files to get you going. You will need to determine which file suits your needs the best. Choose the file that works best for you based on the criteria below. Read the comments in the selected configuration file carefully as they will help you to understand how to modify certain parameters to your installation. Each configuration file is linked below, but you can also find them by visiting https://github.com/bigtreetech/Eddy

- You wish to use the Eddy as a probe but will use another device as the z-endstop - Use this config with no homing
- You wish to use the Eddy as a probe AND as the z-endstop - Use this config which includes homing
- You wish to use the Eddy as a probe AND as the z-endstop and would like to use the beta z-offset functionality - Use this config which includes homing and z-offset

Whichever config you select, copy the entire contents into your printer.cfg file. If you experience that the gcode macro conflicts, then please see the section 'Error: gcode command < ANY GCODE COMMAND > already registered' in the FAQ.

6.2. Z-Endstop Configuration

You can use the Eddy as the Z endstop or you can use another device as an endstop. If you decide to use another device as an endstop, set up your homing and endstop according to that device.

If you want to enable Z-Homing/Endstop for the eddy do the following things:

a. Under your [stepper_z] in printer.cfg change endstop_pin: PA5 to endstop_pin:

probe:z_virtual_endstop and comment out or remove position_endstop: 0. Note that your current endstop may not be PA5 so just look for the line that matches your current endstop and change it.

b. Ensure that you have selected the correct sample configuration file and that the entire contents of that file have been copied into your printer.cfg file. Take note that if you are using a KNOMI, then there may be some macros which conflict with the macros in the KNOMI.cfg file. To resolve these conflicts, comment out the macros in the KNOMI.cfg file and uncomment the lines from the Eddy macros that deal with KNOMI functionality.

c. Edit the parts of the config file that are unique to your setup. These may be things mentioned in the list below. Follow the comments in the config file to help you to edit the values so that they work best with your setup.

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

Important:

The sample configuration requires you to adjust the x_offset and y_offset to match your probe position relative to your nozzle. The settings for the standard Voron X carriage mount are included in all sample configuration files.

7. Calibration

7.1. Drive Current Calibration

With the firmware and configuration done, you are now ready to begin the Eddy's drive current calibration.

1. Place Eddy approx. 20mm above the bed. If you plan to use the Eddy as an endstop then you will not yet be able to home with it and you will need to manually move the gantry or bed such that the Eddy is 20mm above the bed.
2. From Mainsail or Fluidd run the command

   LDC_CALIBRATE_DRIVE_CURRENT CHIP=btt_eddy
3. Type SAVE_CONFIG to save the drive current to your config.

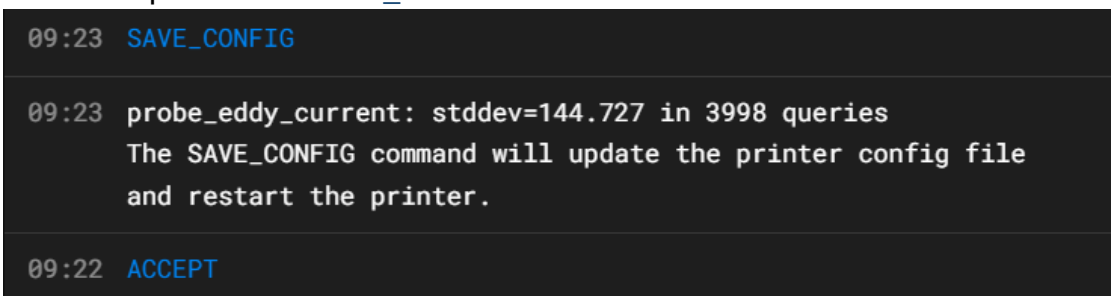
7.2. Mapping Eddy Readings to Nozzle Heights

Now that the drive current has been calibrated, the Eddy will be able to obtain readings from the print bed. Klipper needs to know how those readings correspond to the height of the nozzle. The following calibration procedure positions the nozzle on the bed so that the Z height is = 0. It then takes readings from the Eddy as it gradually increases the nozzle height so that it can map those readings to known heights.

- If you have used one of the recommended configuration templates, you can simply follow the steps below to run the mapping procedure.
  1. Send the command PROBE_EDDY_CURRENT_CALIBRATE_AUTO CHIP=btt_eddy
  2. Follow the prompts on the klipper UI to lower the nozzle until it sandwiches a piece of paper between itself and the bed, but be careful not to dig into the bed. The paper should still be able to move with some force applied.
  3. Click accept and watch as the Eddy performs the mapping. Be sure to send SAVE_CONFIG when it is done.
  4. You can now move on to **Bed Mesh Calibration.**

- If you have not used a configuration template, follow the steps below to run the mapping procedure.

  1. Home X and Y axes with command G28 X Y
  2. Make sure you don't have a bed heightmap loaded.
     Send BED_MESH_CLEAR from the console to clear the heightmap.
  3. Move Nozzle to Centre of the bed with G0 X150 Y150 F6000. The given command assumes a 300x300 printer but you will need to adjust it for your bed size.
  4. Start the Manual Z Offset Calibration (Paper test) by typing PROBE_EDDY_CURRENT_CALIBRATE CHIP=btt_eddy. You will see an adjustment box that allows you to lower the nozzle. Lower the nozzle until it sandwiches a piece of paper between it and the bed but be careful not to dig into the bed. The paper should still be able to move with some force applied.
  5. Once completed use SAVE_CONFIG.



7.3. Bed Mesh Calibration

1. Home All Axes
2. If your printer supports z_tilt or quad_gantry_level (QGL), run: Z_TILT_ADJUST or QUAD_GANTRY_LEVEL to prevent nozzle collisions during the mesh scan
3. Use command BED_MESH_CALIBRATE METHOD=scan SCAN_MODE=rapid
4. Once completed use SAVE_CONFIG

7.4. Temperature Compensation Calibration (Eddy and Eddy Duo ONLY)

Important:

The following steps are for Eddy and Eddy Duo Only. Eddy Coil does not have temperature compensation so these steps should be disregarded.

When Eddy performs temperature compensation, exercise caution, as the heated bed can reach extremely high temperatures.

1. Home All Axes and move Z 5 mm above the bed by typing G0 Z5 or using the movement UI.
2. Set idle timeout by typing SET_IDLE_TIMEOUT TIMEOUT=36000
3. Run TEMPERATURE_PROBE_CALIBRATE PROBE=btt_eddy TARGET=56 STEP=4

   Tips:

   In the command above, the target is set to 56. That is a good value for many machines; however, if your chamber reaches higher than that value, you may set the target higher. Note that it will take slightly longer to complete the calibration the higher you go as the probe needs to soak for longer.
4. This will cause the UI to display the z axis adjustment box. Use the paper method mentioned here to pinch a sheet of paper between the nozzle and the bed and then accept the value.
5. After accepting the value, turn on your heat bed to the maximum value and your nozzle to 220℃.
6. If you are in a room with an air-conditioner or an open window, it would be good to turn it off and/or close the window. We want the temperature of the Eddy to rise and breezes will stop that. Uninterrupted room temperature works best.
7. As the Eddy temp rises, you will automatically be asked to perform the paper pinch method at each 4℃ interval. Be careful not to burn yourself on the bed as the bed can get quite hot.
8. Repeat the paper test method until the calibration completes. If you find that the temperature of the Eddy is no longer increasing, you can end

the calibration early using the relevant command below.

The following additional gcode commands are available during drift calibration:

TEMPERATURE_PROBE_NEXT - may be used to force a new sample before the step delta has been reached.

TEMPERATURE_PROBE_COMPLETE - may be used to complete calibration before the TARGET has been reached.

ABORT - may be used to end calibration and discard results.

Tip: The Eddy thermal calibration process not only accounts for Eddy probe drift but also for thermal expansion of the mechanical components within your machine. This expansion can be very significant and it can result in poor first layers when using other probes. It is important to keep in mind that if you perform the thermal calibration with the nozzle and the heated bed turned on then there will be thermal expansion from both the hotend and the heated bed. Therefore, if you later try to perform a paper test and only have either the nozzle or the heated bed turned on you may find that there is about a 0.05 gap (not enough to cause a first layer issue but enough to feel less of a pinch on the paper). If this all sounds a bit confusing then don't worry. All you need to know is that you should perform the calibration with the bed and the nozzle both hot and then subsequently print with the bed and the nozzle both hot, then you will get fantastic first layers.

You’re all set and your Eddy will give you a beautiful first layer across a wide temperature range!

8. Extra Info

8.1. Z-Offset

This section only applies to those who are using the Eddy for homing.

The Eddy should not need the use of a z-offset since it is calibrated to understand where z=0 is. Nevertheless, if you would like to use a z-offset then you should use the sample config file that includes z-offset functionality.

To determine the correct Z-offset, follow the steps below.

1. Home your printer.

2. Place a piece of paper beneath the nozzle.

3. Use mainsail or fluidd to set the z height to z=0. DO NOT babystep to get the nozzle to z=0! Set it as the z-axis height.

4. After setting the z-axis height to z=0 check if the pinch on your paper is just right. If not, then use babystepping to go up or down.

5. After babystepping to the correct height, save the adjustment using the button on the mainsail or fluidd UI.

8.2. Bed Mesh Calibrate Parameters

The Eddy allows you to perform a very rapid bed mesh scan before each print to ensure that you get the best first layer possible. To do this, we recommend replacing the standard BED_MESH_CALIBRATE macro with our modified version from the sample configuration file and then including a BED_MESH_CALIBRATE call in your print start macro.

8.3. Bed Mesh Scan Height

The scan height is set by the horizontal_move_z option in [bed_mesh]. In addition, it can be supplied with the BED_MESH_CALIBRATE gcode command via the HORIZONTAL_MOVE_Z parameter.

The scan height must be sufficiently low to avoid scanning errors. Typically, a height of 2mm (ie: HORIZONTAL_MOVE_Z=2) should work well, presuming that the probe is mounted correctly.

It should be noted that if the probe is more than 4mm above the surface then the results will be invalid. Thus, scanning is not possible on beds with severe surface deviation or beds with extreme tilt that hasn't been corrected.

8.4. Rapid (Continuous) Scanning

When performing a rapid bed mesh scan there is little time to accumulate many samples per point so that they can be averaged and have noise removed. Therefore, a rapid scan may not be as accurate as a standard bed mesh scan but in most cases, it will still produce a fine first layer.

Rapid scans can be improved by allowing the travel planner to slightly overshoot the scanned bed mesh and smooth the moves. You can configure this overshoot in the bed_mesh configuration section using the scan_overshoot: parameter. Note that you will need to ensure that the axis can travel to the mesh boundary in addition to this overshoot value on your printer so be careful not to specify a value that is too high. Normally, 8mm is plenty.

9. FAQ - Frequently Asked Questions

1. Sometimes I get "Error during homing probe: Eddy current sensor error"

- This generally indicates that the oscillator within the Eddy sensor is not at a valid value before the probe/homing attempt starts. We recommend trying the following steps:

1) Double check your probe height. It may be that it is too close to the bed or too high. Remember that we recommend that it is at 2mm-3mm above the bed when the nozzle is just touching the bed. Around 2.5mm is optimal in most cases but if you are finding that your probe is having errors at high temperatures then try to drop it just below 2mm. However, if your probe is having errors during QGL attempts then you may need to raise it slightly.
2) After you have adjusted the probe height, remove all of the calibration settings from your config file and recalibrate the eddy.
3) If you still receive this error then increase the reg_drive_current value to 16 from 15 if it is currently set to 15.

2. Sometimes I get a "Probe Triggered Before Movement" Error

- This will happen when you try to execute two successive PROBE commands. Always raise the gantry by a few millimeters between PROBE commands to avoid this.

3. Eddy is performing Z Hops when running Bed Mesh

- Make sure you are using the correct macro call. BED_MESH_CALIBRATE METHOD=rapid_scan

- Remove or alter KAMP - Adaptive Bed Mesh and any custom BED_MESH_CALIBRATE macros. Use klipper adaptive mesh instead or, alternatively, do not include KAMP/Adaptive_Meshing.cfg in your KAMP_Settings.cfg

4. Which Eddy version should I use?

- It depends on your needs. Eddy, Eddy Duo and Eddy Coil are nearly identical; however, Eddy Coil is more for toolhead boards and connects via I2C connectors.
- Eddy Coil does not have temperature compensation and so it may be less reliable for homing if you are using it within a sealed chamber.
- In addition, Eddy Duo supports two communication methods: USB and CAN, while Eddy supports only one method: USB.

5. Error: gcode command < ANY GCODE COMMAND > already registered

- This will happen when you have conflicting gcode macros. Check all of your gcode macros for ones that share the same name and arbitrate the conflicts. Generally, you should select the functionality from the Eddy macros if there is a conflict and you are not sure what to do.

6. My z-offset doesn’t seem to save and resets, is there a work around or fix?

- Coming from a standard probe, this may seem like a bug. However, if you have calibrated the Eddy correctly and are using the special homing macros, then there will be no need for a z-offset. Explaining why is a bit long winded but essentially when it comes to an Eddy, the z-offset parameter does not adjust the height at which the nozzle prints, it just adjusts the height at which homing or probing triggers.
- While we strongly recommend simply performing the Eddy probe calibration in order to get a nozzle height that is just right, you can still simulate a standard z-offset by using the Z-offset beta sample

configuration file. Simply uncomment any macro that is related to the beta z-offset functionality and you will be able to use the standard mainsail buttons to raise/lower the nozzle and save that height as a z-offset.

7. My Eddy Macros Conflict with My KNOMI Macros

- The Eddy and the KNOMI share similar Macros. All of the needed functionality for the KNOMI has been built into the Eddy macros. Please comment out the KNOMI macros which conflict and use the Eddy macros in place of them.
- Note that you may need to uncomment some lines in the Eddy macros that are specifically included for people who run the KNOMI. Check the macros to see which lines have been commented and then uncomment them if they are needed for KNOMI.

8. KAMP and Eddy

- KAMP aka Klipper-Adaptive-Meshing-Purging should be removed from your Klipper prior to using Eddy. Please comment out the include line. ie#[include ./KAMP/adaptive_meshing.cfg] from your KAMP_SETTINGS.cfg
- Instead, KAMP has been integrated into Klipper as of January 2024 and you should use the ADAPTIVE=1 option in your BED_MESH_CALIBRATION calls. You can find more Information on Adaptive Mesh Here

If you need 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.