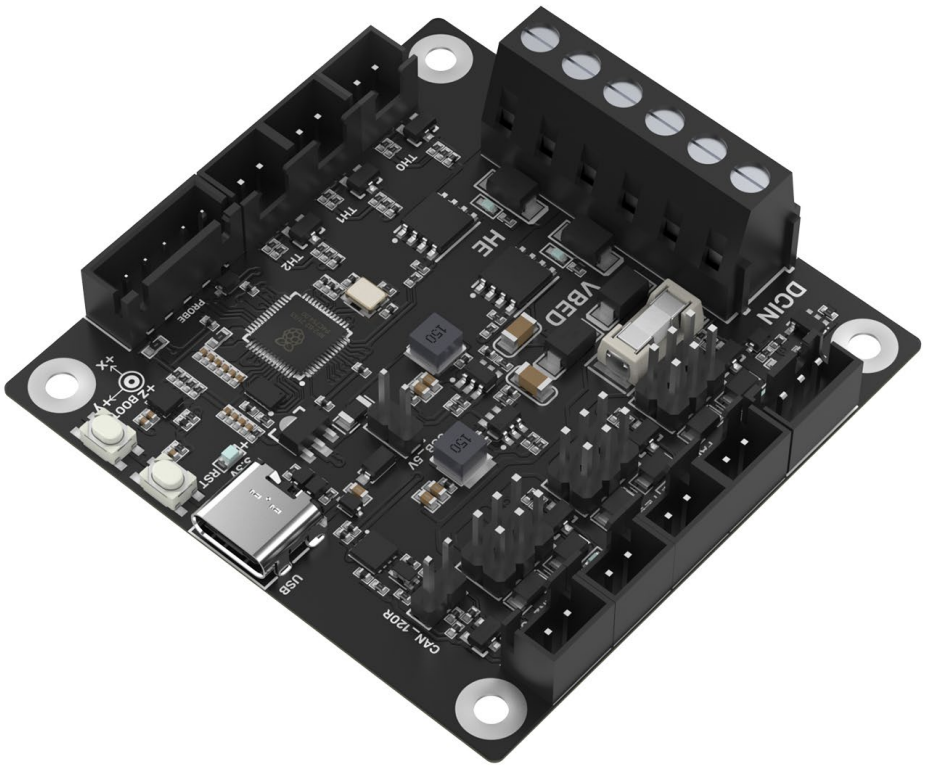
BIGTREETECH

MMB Cubic V1.0

User Manual



Revision Log

Version	Date	Revisions
v1.00	July 18th, 2024	Initial Version
v1.01	September 9th, 2024	Corrected the fan voltage selection diagram.

TABLE OF CONTENTS

Revision Log 2
1. Product Profile 4
1.1. Feature Highlights 4
1.2. Specifications 4
1.3. Dimension 5
2. Interface 6
2.1. Interface Diagram 6
2.2. Pin Description 7
3. Interface Introduction 8
3.1. USB Power Supply 8
3.2. PWM Fan Voltage Selection 8
3.3. Temperature Measurement Interfaces 9
3.4. RGB Interface 9
3.5. CAN Interface 10
3.6. PROBE Interface 10
3.7. Heater Cartridge Interface 10
3.8. Heated Bed Interface 11
4. Klipper Firmware 12
4.1. Flash Katapult(Canboot) 12
4.2. Compiling Klipper Firmware 13
4.3. Firmware Update via KATAPULT 14
4.4. Firmware Update via DFU 15
4.5. CAN Bus Configuration 17
4.6. Configuring Klipper 17
5. Precautions 19

1. Product Profile

The BIGTREETECH MMB Cubic is the ideal solution for fitting additional interfaces onto your Klipper control board, allowing you to connect more fans, hotends, sensors, etc., effortlessly.

1.1. Feature Highlights

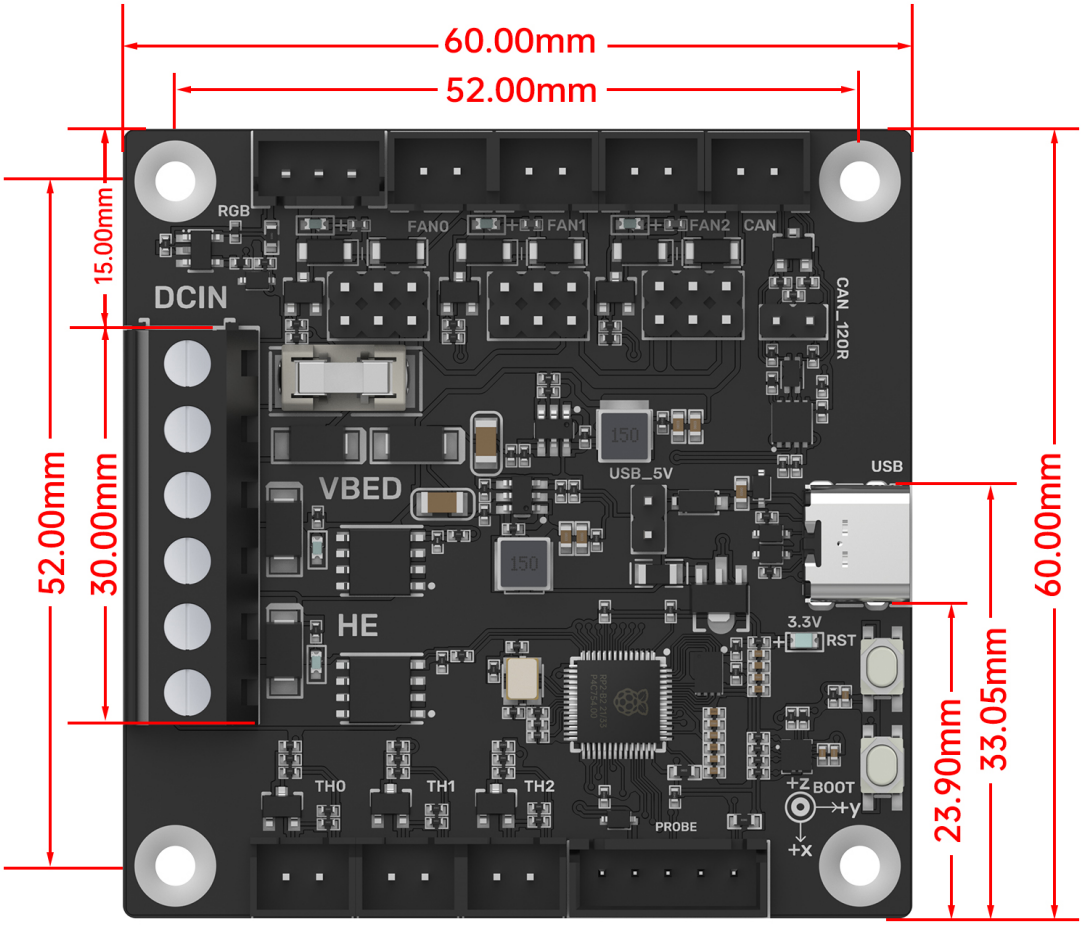
- Employs a 32-bit ARM Cortex-M0+ RP2040 MCU, operating at 133 MHz.
- Features the LN5016-1.5A power chip, supporting a 24V (MAX: 36V) power input, capable of delivering a maximum output current of 1.5A.
- A reserved BOOT button within the board allows users to update the bootloader using the DFU mode.
- A specially designed circuit on the board protects the signal coming back from the thermistor, preventing MCU damage from shorted heated beds and heater cartridge connections.
- The PWM fan includes a range of selectable voltage options (DCIN, 12V, 5V) which eliminates the need for external voltage modules and reduces the risk of damage to the board.
- Firmware updates can be performed via USB in DFU mode or through the make flash command in Klipper via DFU mode.
- High-performance MOSFETs assist in reducing heat generation by controlling the flow of electrical current.
- Includes replaceable fuses.

1.2. Specifications

Dimensions	60mm x 60mm
MCU	ARM Cortex-M0+ RP2040 133 MHz
Mainboard Input Voltage	DC24V-36V
Heated Bed Input Voltage	VBED DCIN
Logic Voltage	DC3.3V
Heating Interface	Heated Bed (VBED), Heater Cartridge (HE)
Max Heated Bed Output Current	10A
Heater Cartridge Port Power	120W（24V 5A）
Fan Interfaces	3x 2-pin PWM Fans (FAN0, FAN1, FAN2)
Max Fan Output Current	Total: 1A, Peak 1.5A
Total Current for Heater Cartridge + Fans	Maximum 10A
Mainboard Max 5V Output Current	Peak 1.5A
Mainboard Max 12V Output Current	Peak 1.5A

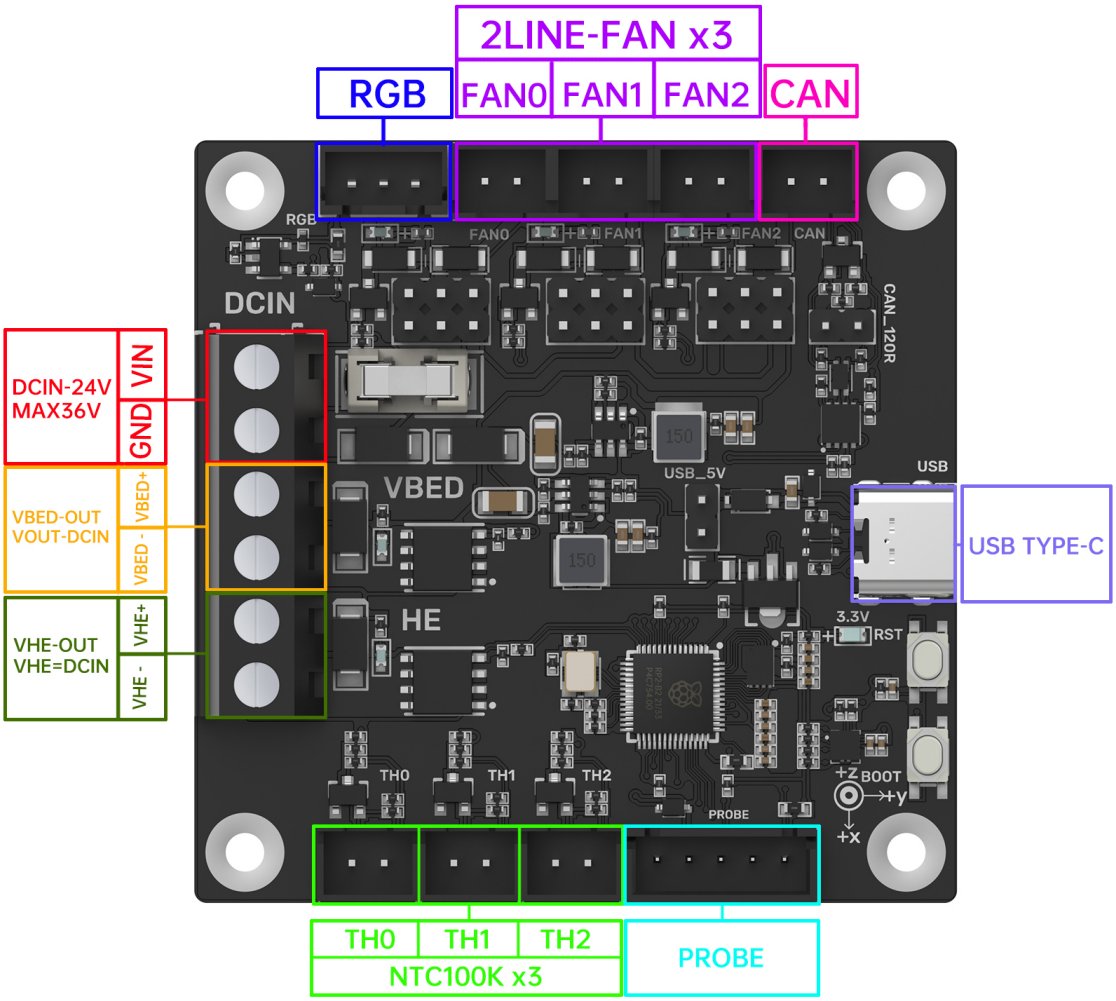
Expansion Interfaces	Probe (Servos, Probe), RGB, CAN, TH x3 (NTC100K), 5V-TYPE-C power interface, etc.

1.3. Dimension

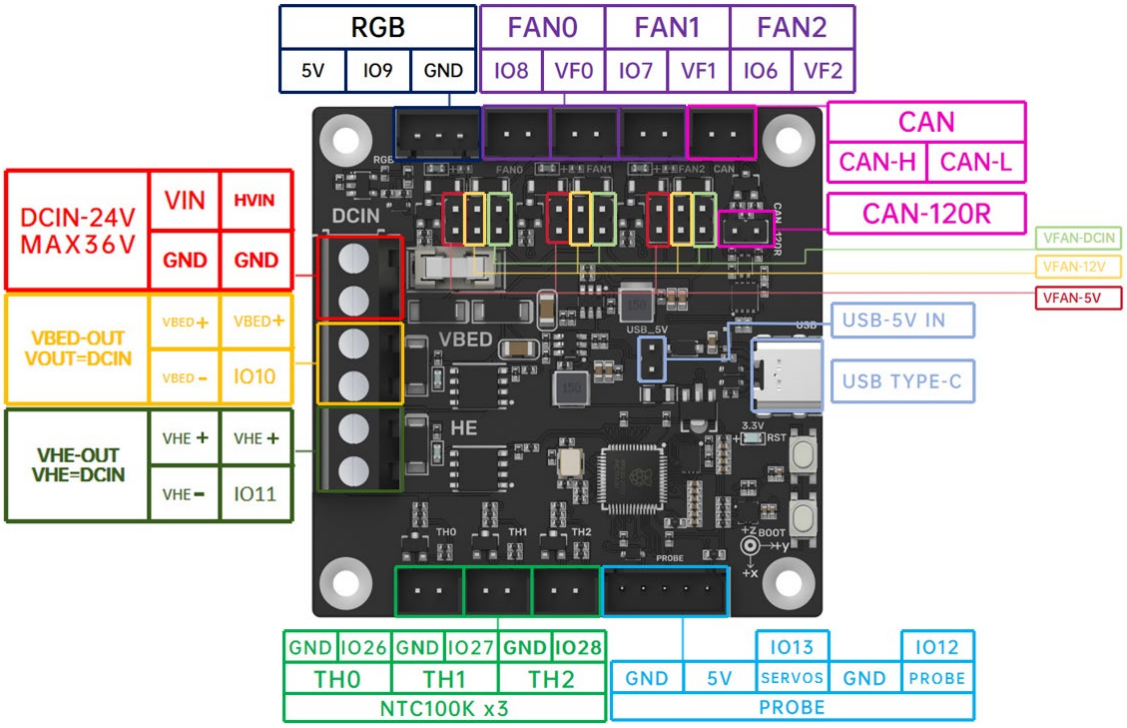


2. Interface

2.1. Interface Diagram



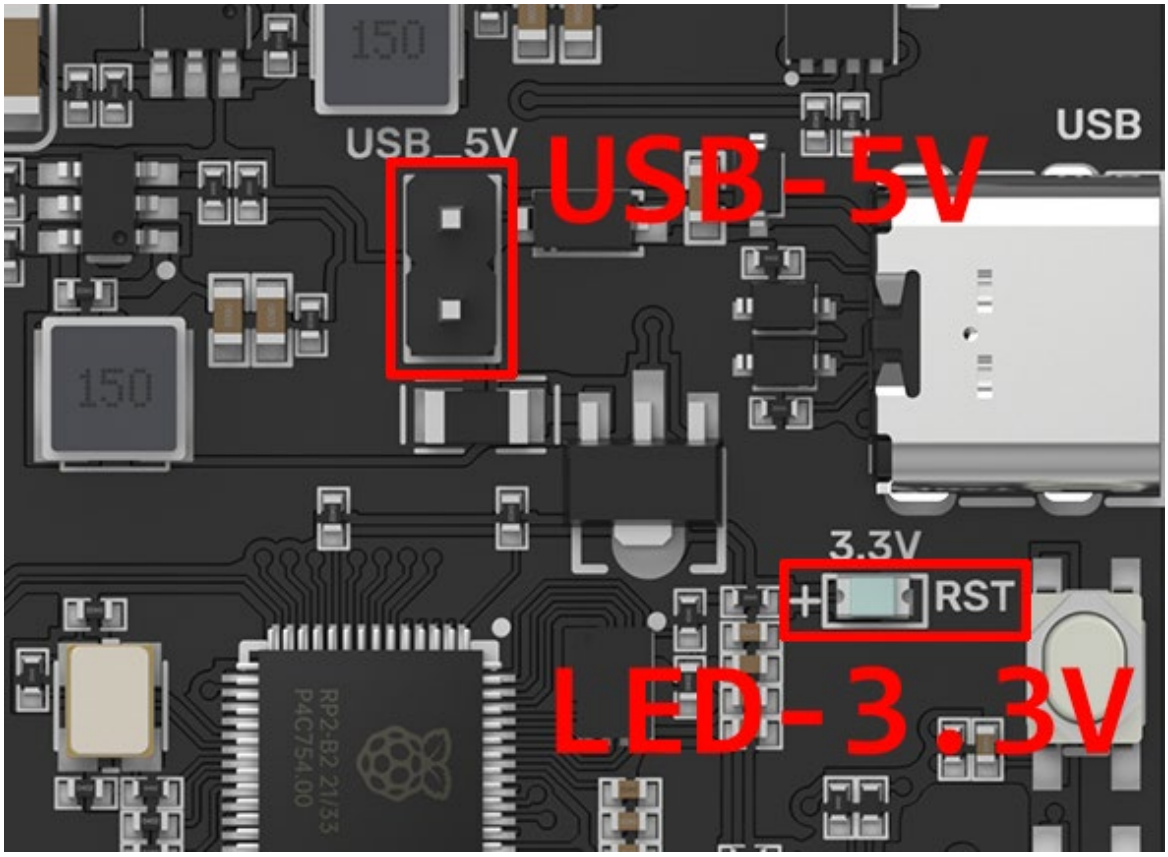
2.2. Pin Description



3. Interface Introduction

3.1. USB Power Supply

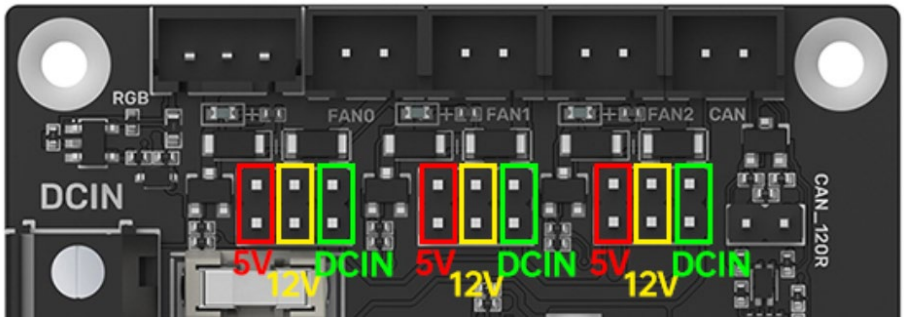
To use USB power supply, short-circuit the USB-5V pin header. The 3.3V indicator will light up under normal power conditions.



3.2. PWM Fan Voltage Selection

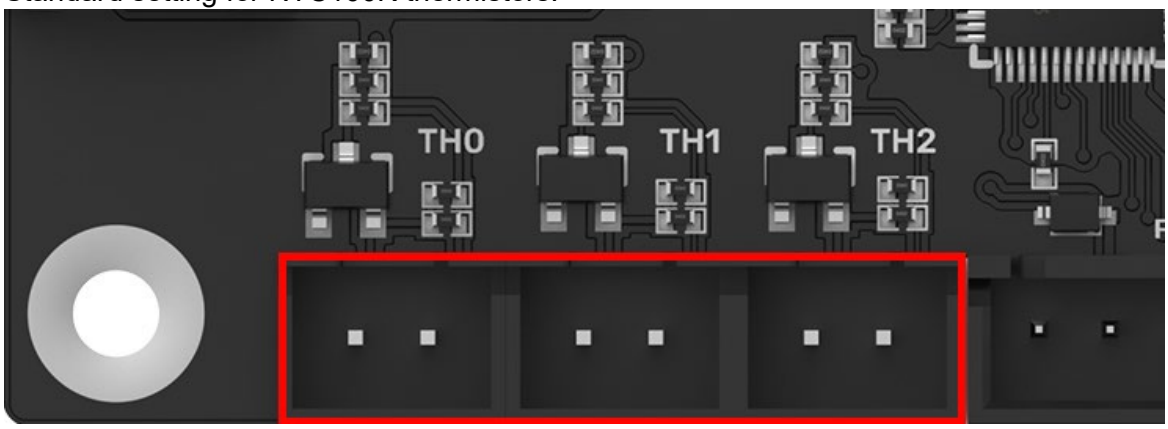
Set the output voltage to 5V, 12V, or 24V using a jumper.

Note: Please ensure that the supported voltage of the fan is confirmed before a selection is made in order to avoid damaging the board. Our company will not be held liable for failure to do so.

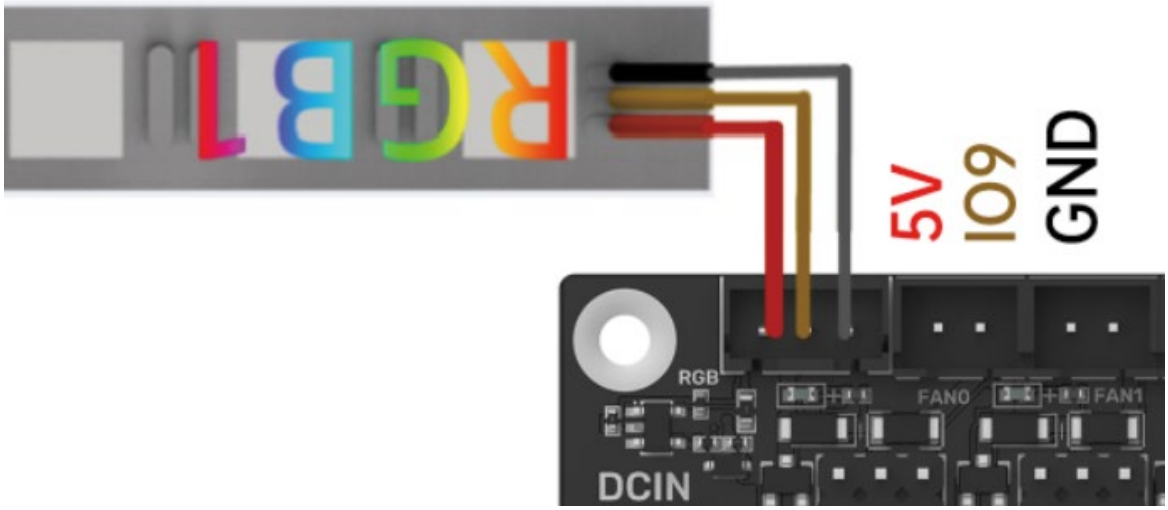


3.3. Temperature Measurement Interfaces

Standard setting for NTC100K thermistors.

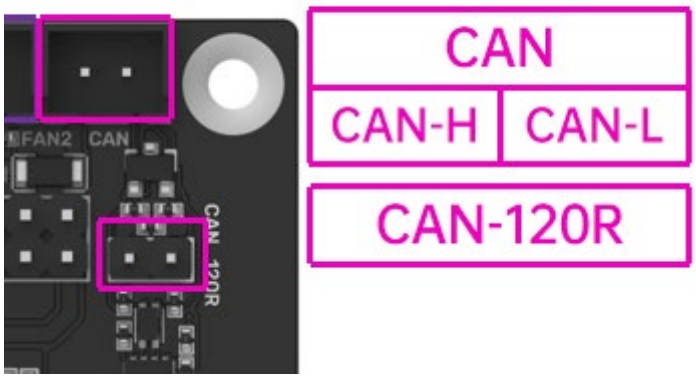


3.4. RGB Interface

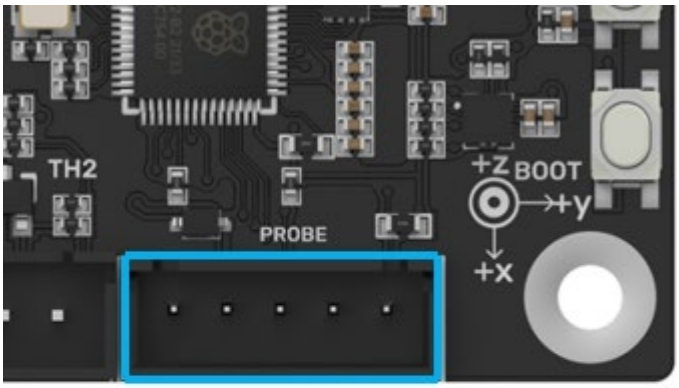


3.5. CAN Interface

If the MMB Cubic is used as a terminating device (one of the two devices at either end of the bus), ensure that the 120R position has a jumper inserted.

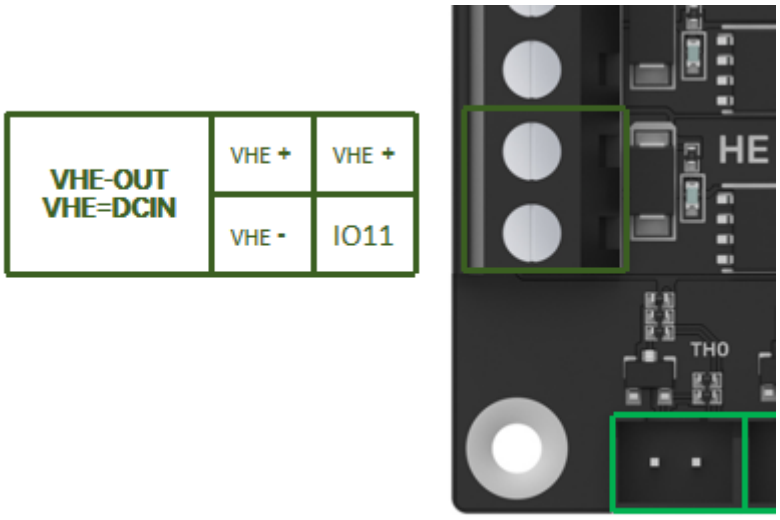


3.6. PROBE Interface



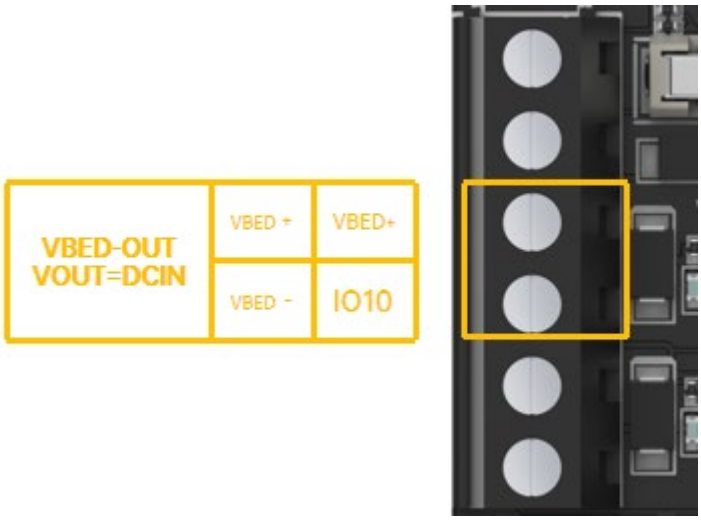
3.7. Heater Cartridge Interface

Note：Ensure the heater cartridge supports the input voltage, which matches the board's input.



3.8. Heated Bed Interface

Note: Ensure the heated bed supports the input voltage, which matches the board's input.



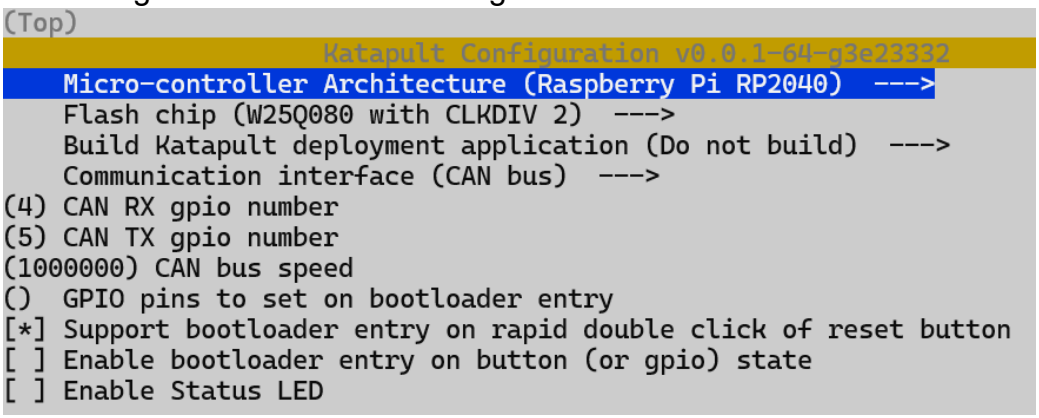
4. Klipper Firmware

4.1. Flash Katapult(Canboot)

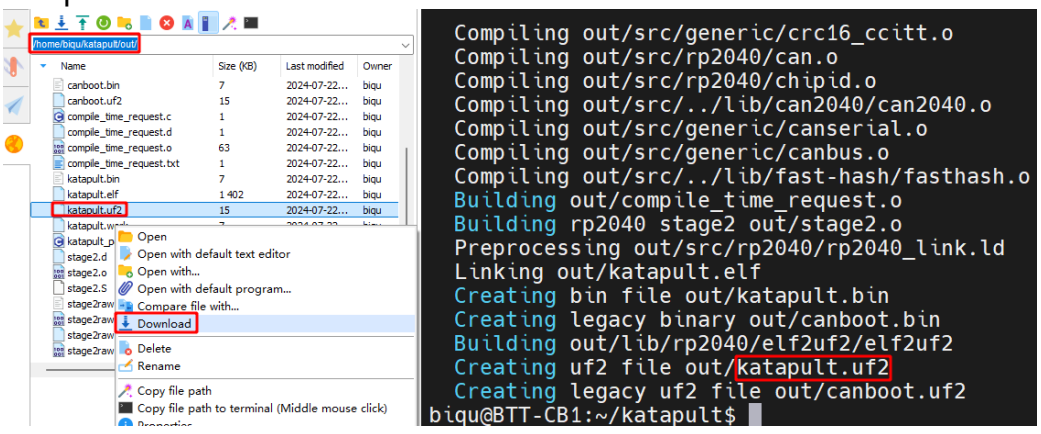
Note: Katapult is for direct firmware updates via CAN bus. Skip this step if using DFU.

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 and can be directly downloaded to the computer on the left side of the SSH software.



4) Hold down the Boot button and connect to Raspberry Pi/CB1 with a Type-C cable. This allows the chip to enter DFU mode.

5) In the SSH terminal command line, enter **lsusb** to query the DFU device ID.

```
biqu@BTT-CB1:~/klipper$ lsusb
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 003: ID 2e8a:0003 Raspberry Pi RP2 Boot
Bus 002 Device 002: ID 1a40:0101 Terminus Technology Inc. Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
biqu@BTT-CB1:~/klipper$
```

6) Enter the following command to flash Katapult

make flash FLASH_DEVICE=2e8a:0003
Replace **2e8a:0003** with the actual device ID obtained in the previous step.

7) After flashing, unplug the Type-C data cable.

4.2. Compiling Klipper Firmware

1. After SSH connects to CB1/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 (CAN bus)  --->
(4) CAN RX gpio number
(5) CAN TX gpio number
(1000000) CAN bus speed
()  GPIO pins to set at micro-controller startup
```

[*] Enable extra low-level configuration options
Micro-controller Architecture (Raspberry Pi RP2040) --->
If not using Katapult:
Bootloader offset (No bootloader) --->
If using Katapult:
Bootloader offset (16KiB bootloader) --->

Flash chip (W25Q080 with CLKDIV 2) --->
If using CAN Bus communication:
Communication Interface (CAN bus) --->
(4) CAN RX gpio number
(5) CAN TX gpio number
(1000000) CAN bus speed
If using USB communication:
Communication Interface (USBSERIAL) --->
USB ids --->

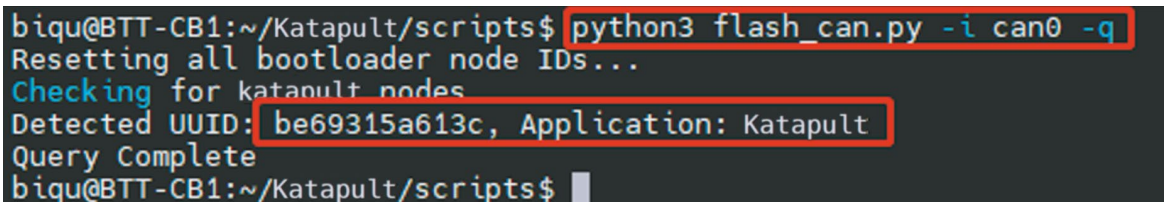
() GPIO pins to set at micro-controller startup

2. After configuring, enter ‘q’ to exit the configuration interface. When asked to save configuration, select ‘Yes’.

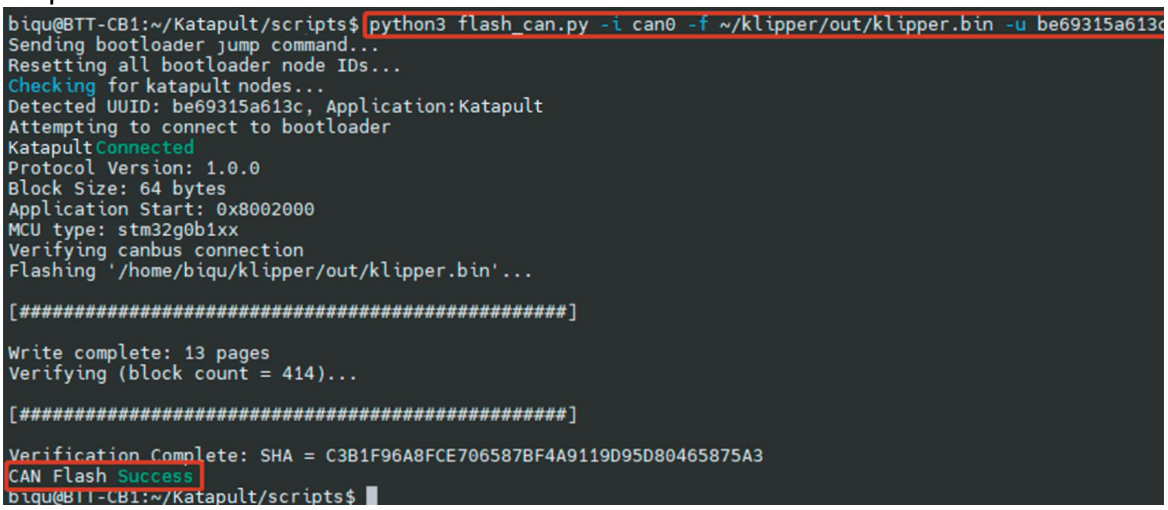
3. Enter **make** to compile the firmware. When **make** is completed, the required **klipper.bin** firmware will be generated in the **home/pi/klipper/out** folder.

4.3. Firmware Update via KATAPULT

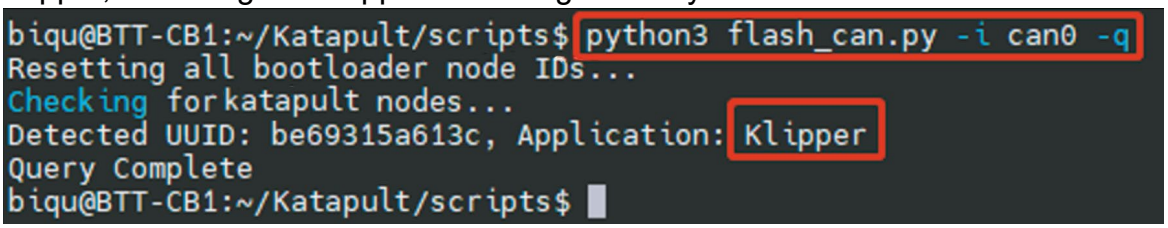
1. To use the CAN bus, ensure that the CAN bus cables are properly connected and that the jumper is inserted at the position of the 120R termination resistor. Enter
cd ~/Katapult/scripts
and then enter
python3 flash_can.py -i can0 -q
to query the CAN bus ID (connect the CAN cable and power-on in advance). As shown in the image below, the UUID of the device is found.



2. Enter
python3 flash_can.py -i can0 -f ~/klipper/out/klipper.bin -u be69315a613c
replacing the UUID parameter after "-u" with the actual UUID on your board. **Note:** by this point, you should have already compiled klipper.bin using "make". Additionally, when selecting the bootloader offset in the Klipper menuconfig, use the 16KiB option since Katapult's Application start offset is 16KiB. The image below shows a successful flashing sequence.



3. Re-enter
python3 flash_can.py -i can0 -q
to query. At this stage, the ‘Application’ has changed from Katapult to Klipper, indicating that Klipper is running normally.



4.4. Firmware Update via DFU

Raspberry Pi or CB1 firmware update through DFU:

1. Hold down the Boot button and connect to Raspberry Pi/CB1 with a Type-C cable. This allows the chip to enter DFU mode.

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

```
biqu@BTT-CB1:~/klipper$ lsusb
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 003: ID 2e8a:0003 Raspberry Pi RP2 Boot
Bus 002 Device 002: ID 1a40:0101 Terminus Technology Inc. Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
biqu@BTT-CB1:~/klipper$
```

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/
to query the device Serial ID (this ID is only available for USB communication, this step can be ignored when using CAN Bus communication).

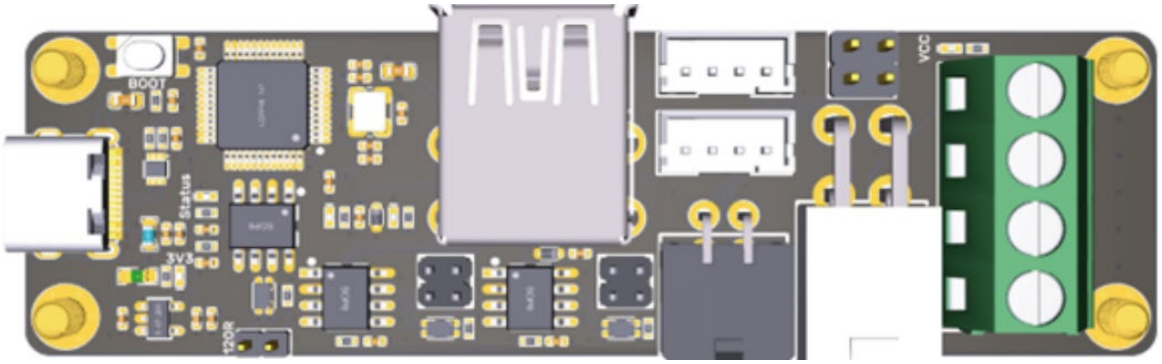
5. If using USB communication, there is no need to manually press the Boot button to enter DFU mode for subsequent updates after the first flashing is completed. Directly enter

   make flash FLASH_DEVICE=/dev/serial/by-id/usb-Klipper_rp2040_mmb_cubic-if00
   to flash the firmware (note: replace **/dev/serial/by-id/xxx** with the actual ID obtained in the previous step).

6. If using CAN bus communication, unplug the Type-C data cable after flashing.

4.5. CAN Bus Configuration

For use with BIGTREETECH U2C module:



(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

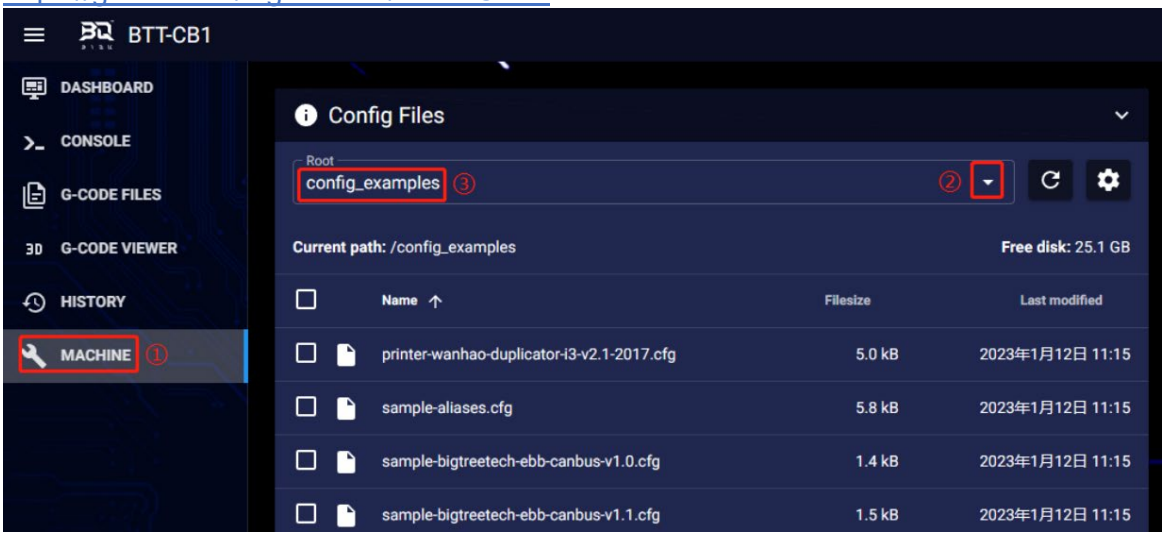
(3) If an uninitialized CAN device is detected, the above command will report the device's canbus_uuid:
Found canbus_uuid=0e0d81e4210c

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

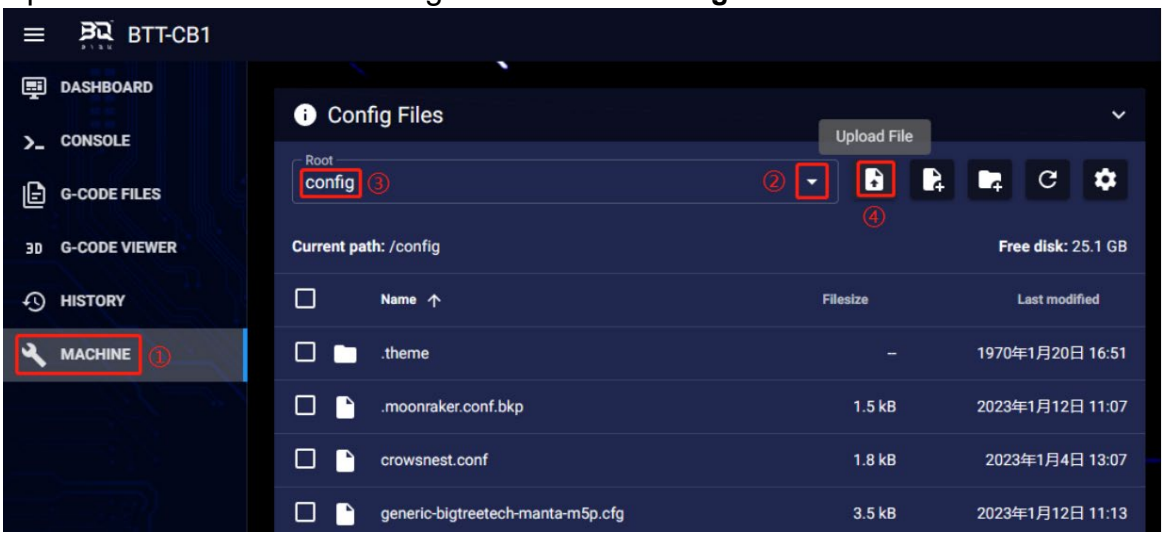
4.6. Configuring Klipper

1. Access the mainsail web UI by entering the IP address of the Raspberry Pi into the browser. Using the path shown in the image below, download the reference configuration named **sample-bigtreetech-mmb-cubic.cfg**. If this file is not found, update the Klipper firmware source code to the latest

version or use the link to download it from GitHub:
https://github.com/bigtreetech/MMB-Cubic



2. Upload the motherboard configuration file to **Configuration Files**.



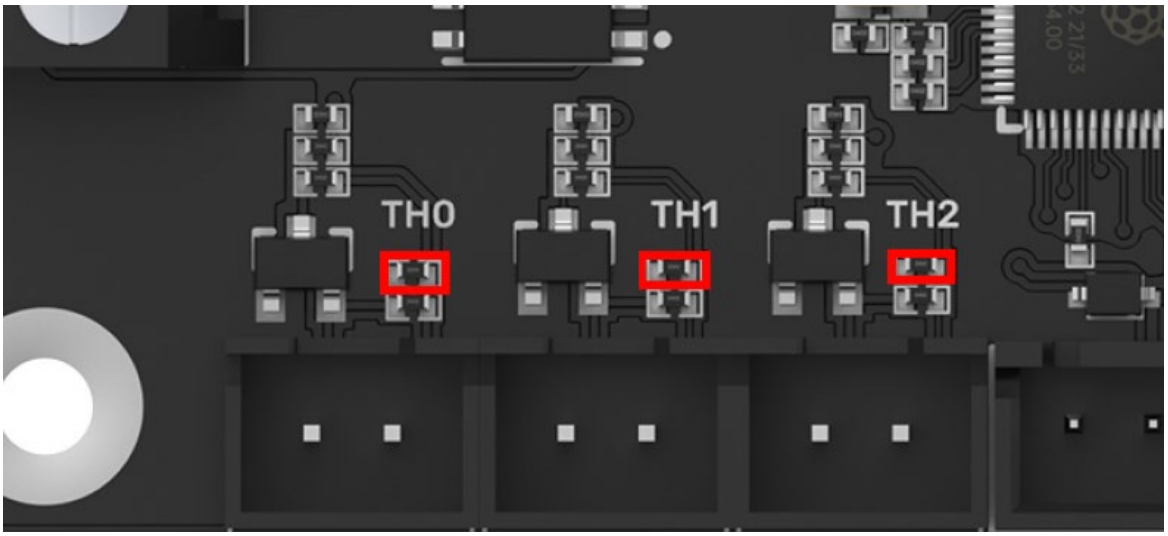
3. Add the MMB Cubic configuration to the **printer.cfg** file:
[include sample-bigtreetech-mmb-cubic.cfg]

4. Change the USB serial or CAN bus ID within the configuration file to match the actual ID of the board (USB serial or canbus).

5. Configure the specific functions of the module according to the instructions in the following link:
https://www.klipper3d.org/Overview.html

5. Precautions

- When using 24V, each of the three fan interfaces can support up to 1A; at 12V or 5V, the combined maximum is 1.5A.
- The temperature interfaces default to NTC100K thermistors; use a 4.12K resistor for PT1000 sensors.



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.