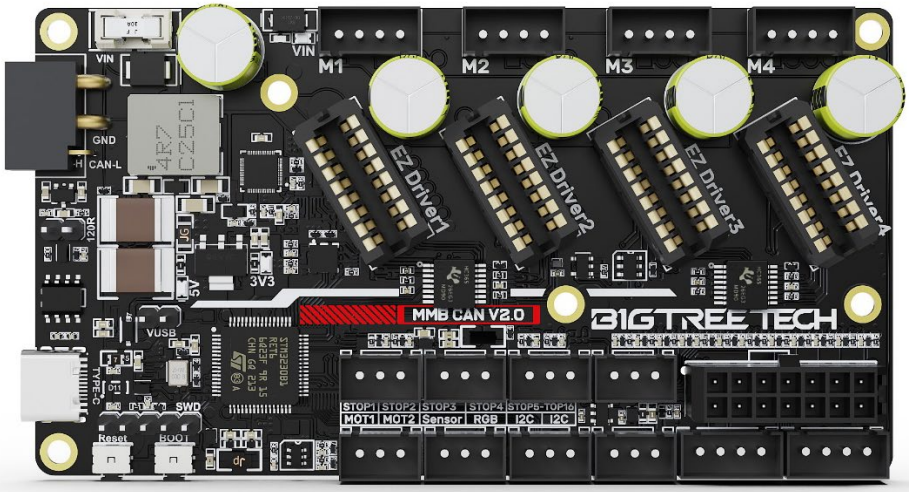
BIGTREETECH

# MMB CAN V2.0

## User Manual



# Revision Log

| Version | Date | Revisions |
| --- | --- | --- |
| v1.00 | June 27th, 2024 | Initial Version |

# Table of Contents

# 1. Product Profile

The BIGTREETECH MMB CAN V2.0 is a buddy board designed specifically for the ERCF V2, co-developed by BIGTREETECH and the ERCF Team. It is optimized to enhance the stability and responsiveness of multi-color operation in the ERCF V2, ensuring excellent compatibility and performance for this kit.

## 1.1. Features Highlights

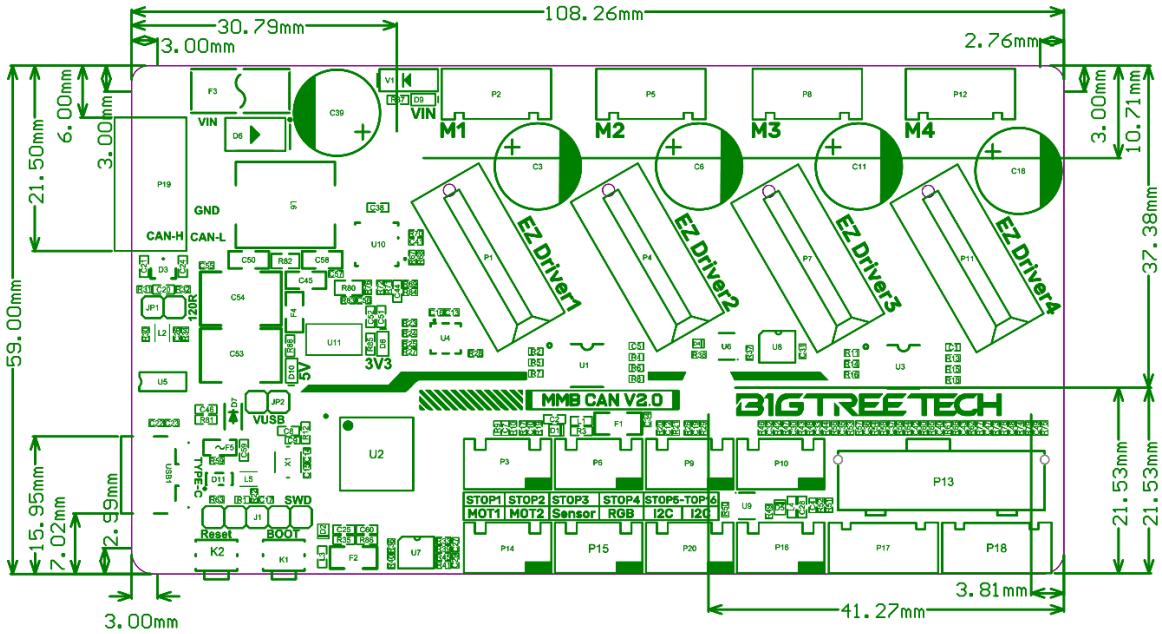
- The board contains BOOT and RESET buttons to enter DFU mode when updating firmware using USB.
- An I2C interface is provided for filament runout detection, blockage detection, or DIY capabilities.
- The power input interface is equipped with reverse polarity protection to prevent damage to the board if the power supply is connected incorrectly during DIY projects.
- The board supports both CAN and USB communication protocols. The CAN terminal resistor (120Ω) can be selected via a jumper, and an additional CAN expansion interface is provided for future upgrades.
- The USB port features an ESD protection chip to safeguard the MCU against potential static discharge damage.
- The board utilizes an XT30 interface for both CAN communication and power supply, streamlining the wiring process.
- The stepper motor driver interface supports high-voltage operation, enabling enhanced DIY customization options.

## 1.2. Specifications

| | |
|---|---|
| Dimensions | 108.26mm x 59mm |
| Mounting Dimensions | For detailed information, please refer to BIGTREETECH MMB CAN V2.0-SIZE.pdf |
| MCU | ARM Cortex-M0+ STM32G0B1RET6 64MHz |
| Input Voltage | DC12V-DC60V |
| Input Current | 9A |
| Logic Voltage | DC 3.3V |

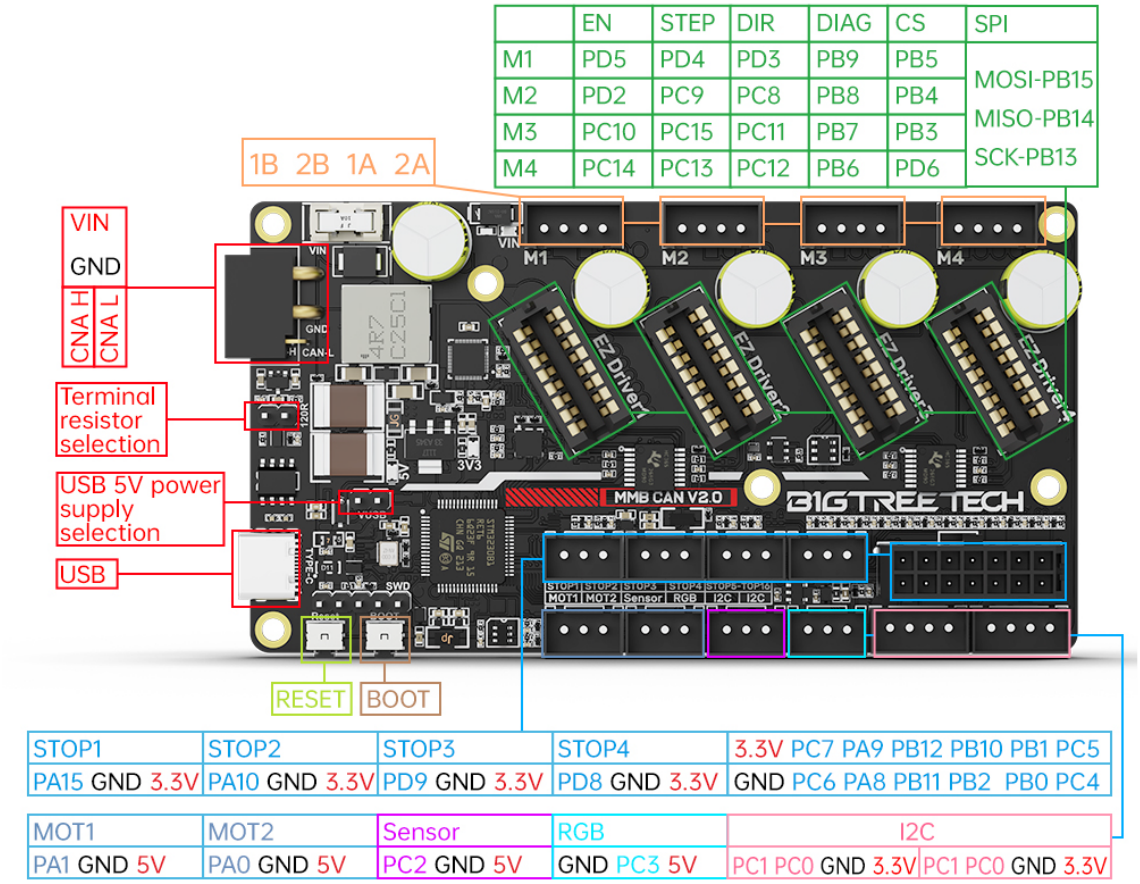
| | |
|---|---|
| Servo Interface (MOT) Max Output | 5V 2A, Peak 2.5A |
| Expansion Interfaces | STP1-STP11, I2C, RGB, Sensor (Infrared Sensor Interface), USB Interface, CAN Interface |
| Supported Motor Drivers | EZ Drivers |
| Driver Operating Modes | STEP/DIR, UART, SPI |
| Stepper Motor Interfaces | M1, M2, M3, M4 |
| USB Communication Interface | USB Type-C |
| DCDC 5V Output Max Current | 7A |

## 1.3. Dimensions



# 2. Peripheral Interfaces

## 2.1. Pin Description




| | EN | STEP | DIR | DIAG | CS | SPI |
|---|---|---|---|---|---|---|
| M1 | PD5 | PD4 | PD3 | PB9 | PB5 | MOSI-PB15 |
| M2 | PD2 | PC9 | PC8 | PB8 | PB4 | MISO-PB14 |
| M3 | PC10 | PC15 | PC11 | PB7 | PB3 | SCK-PB13 |
| M4 | PC14 | PC13 | PC12 | PB6 | PD6 | |

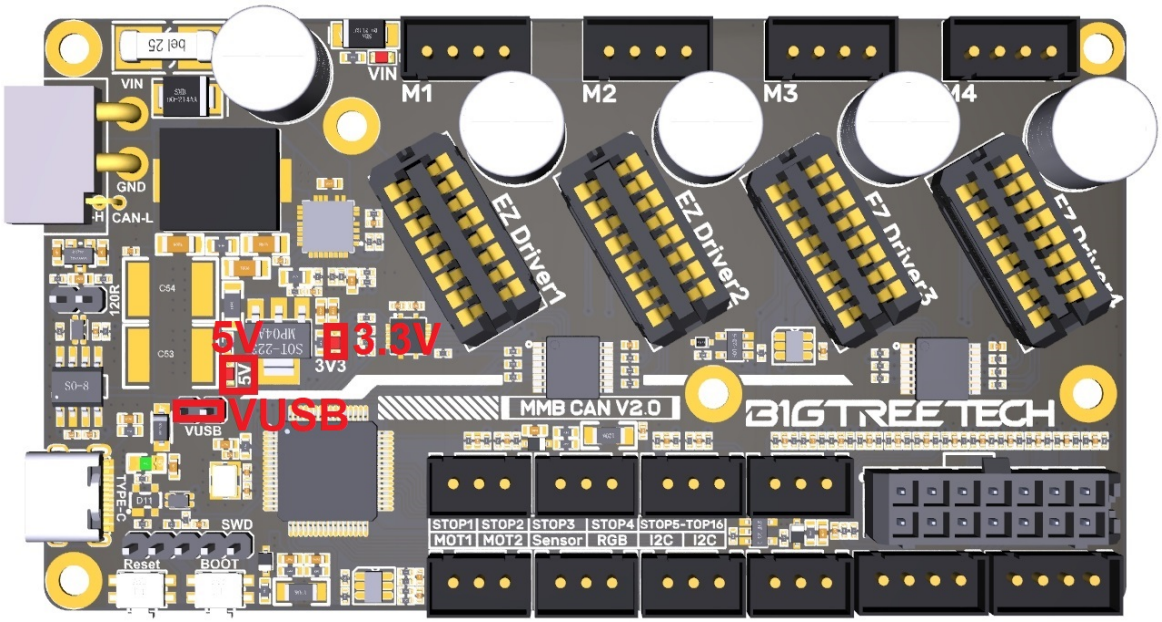
| STOP1 | STOP2 | STOP3 | STOP4 | 3.3V PC7 PA9 PB12 PB10 PB1 PC5 |
|---|---|---|---|---|
| PA15 GND 3.3V | PA10 GND 3.3V | PD9 GND 3.3V | PD8 GND 3.3V | GND PC6 PA8 PB11 PB2 PB0 PC4 |

| MOT1 | MOT2 | Sensor | RGB | I2C | |
|---|---|---|---|---|---|
| PA1 GND 5V | PA0 GND 5V | PC2 GND 5V | GND PC3 5V | PC1 PC0 GND 3.3V | PC1 PC0 GND 3.3V |

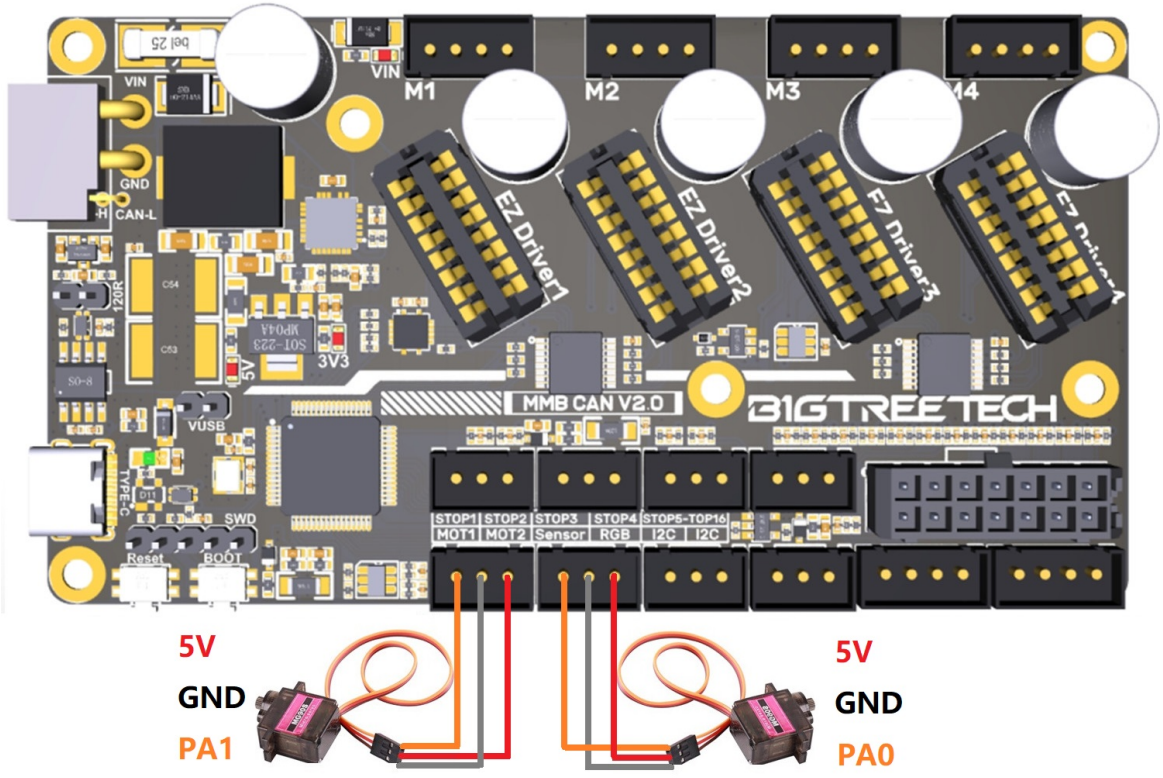
# 3. Interface Introduction

## 3.1. USB Power Supply

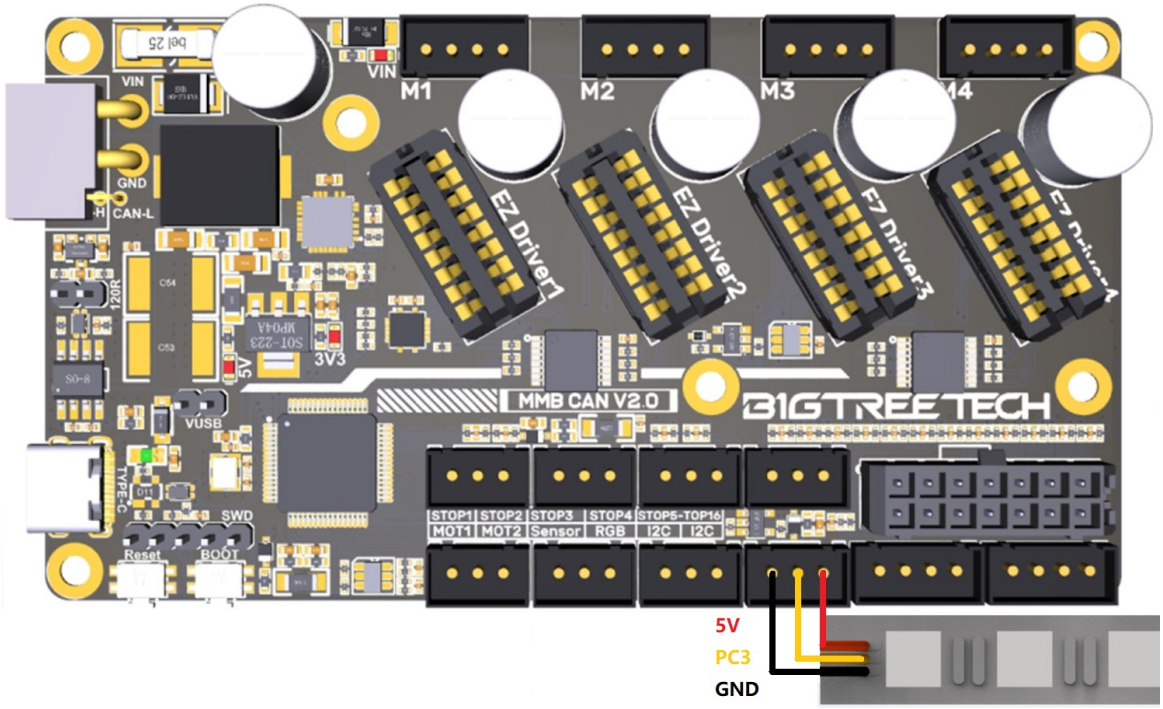
When the board is powered on, the power indicator light will turn on to confirm that the power supply is functioning correctly. The VUSB terminal on the board is the power selection terminal, which should only be shorted with a jumper when powering through USB.



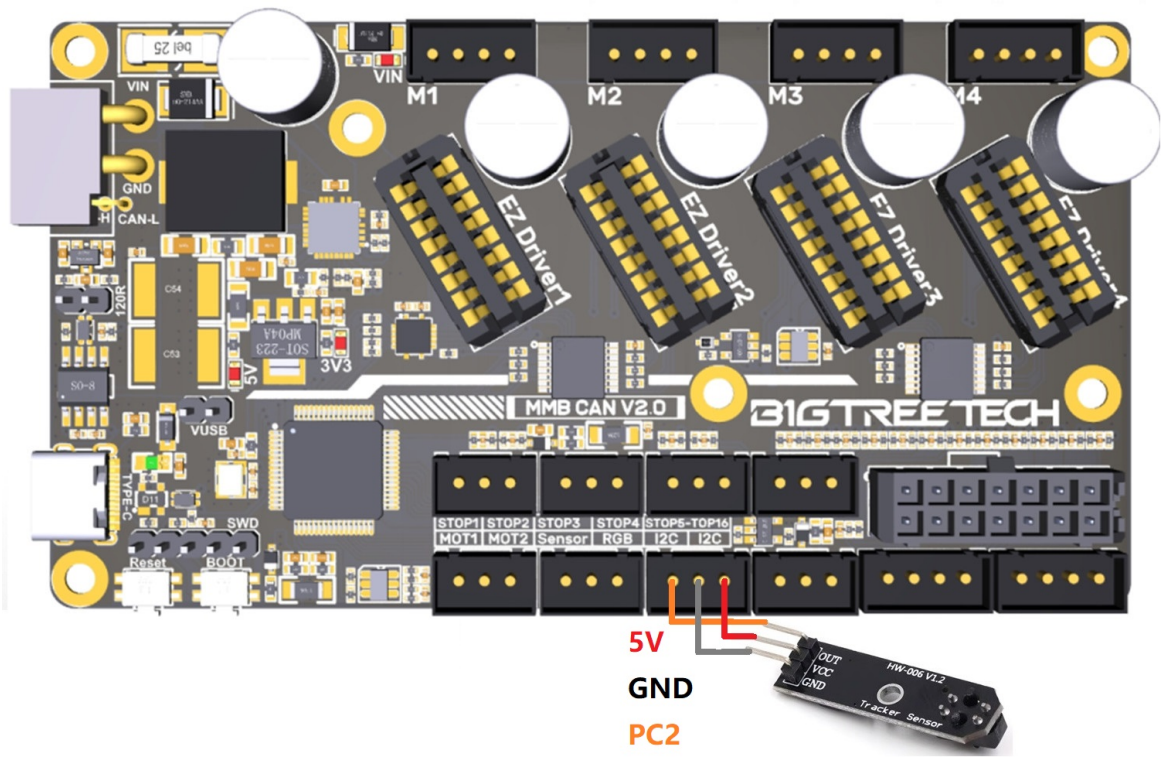

## 3.2. MOT Interfaces



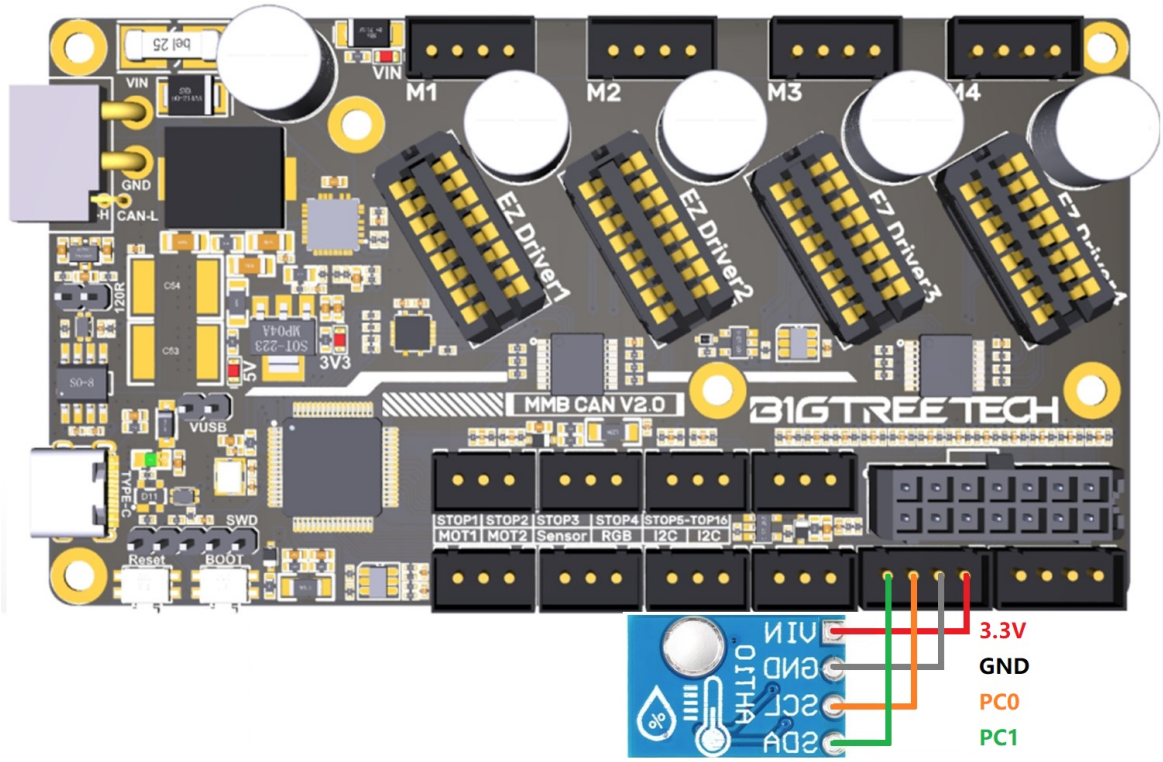
## 3.3. RGB-WS2812 Wiring



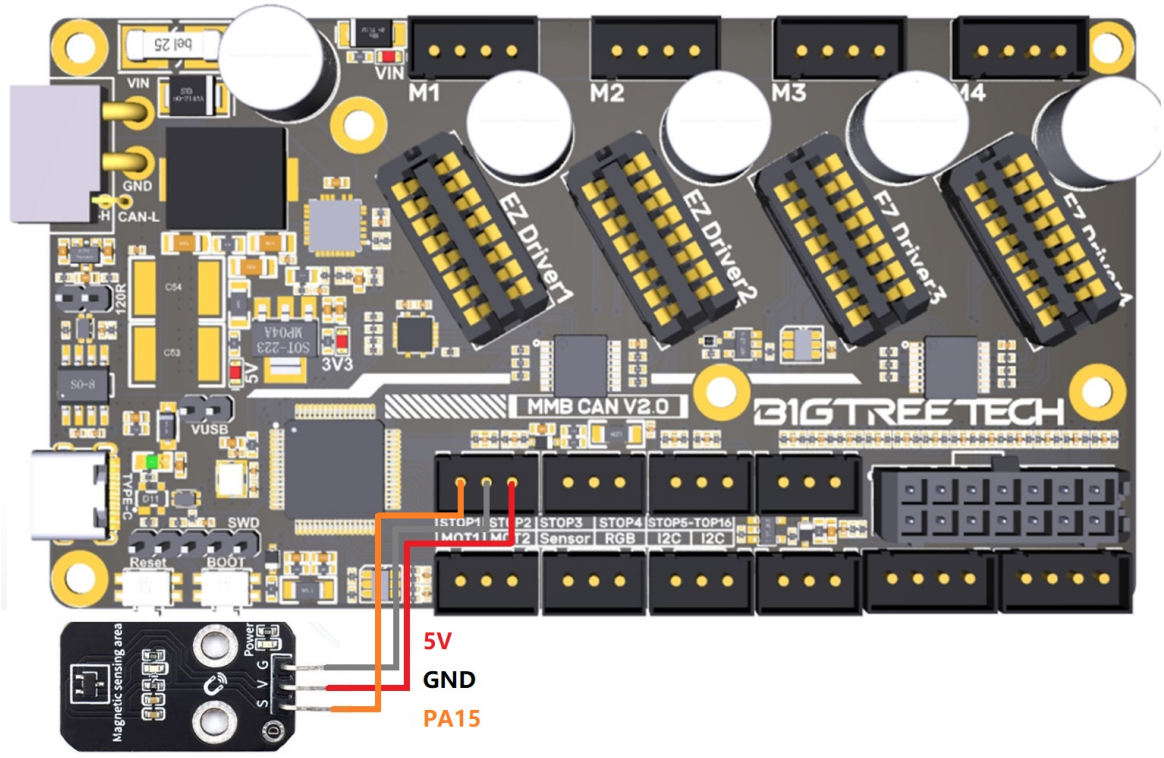

## 3.4. Sensor (e.g., CRT5000 Infrared Sensor) Wiring



## 3.5. I2C (e.g., AHT10 Temperature and Humidity Sensor) Wiring




## 3.6. Endstop (e.g., Hall Sensor) Wiring



# 4. Klipper

## 4.1. Flashing Katapult (formerly CanBoot)

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

To flash Katapult on Raspberry Pi or CB1, 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.

```
(Top)
                        Katapult Configuration v0.0.1-64-g3e23332
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32G0B1)  --->
    Build Katapult deployment application (Do not build)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (CAN bus (on PD0/PD1))  --->
    Application start offset (8KiB offset)  --->
(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
```

3. Enter make to compile the firmware. When “make” is completed, the required “katapult.bin” firmware will be generated in the home/biqu/katapult/out folder.

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. Use the following command to identify the DFU device ID
   lsusb

```
pi@fluiddpi:~ $ lsusb
Bus 001 Device 005: ID 0483:df11 STHicroolactronics STM Device in DFU Mode
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:~ $
```

6. Enter the following command to flash Katapult:
   make flash FLASH_DEVICE=0483:df11

Replace “**0483:df11**” 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 (should these options not be available, update the Klipper firmware source code to the latest version).

```
(Top)
                                    Klipper Firmware Configuration
[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32G0B1)  --->
    Bootloader offset (8KiB bootloader)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (CAN bus (on PD0/PD1))  --->
(1000000) CAN bus speed
()  GPIO pins to set at micro-controller startup
```

   [*] Enable extra low-level configuration options

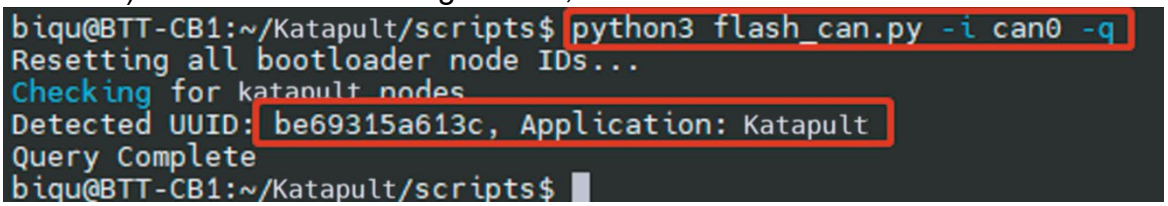
   Micro-controller Architecture (STMicroelectronics STM32) --->
   Processor model (STM32G0B1) --->
   If not using Katapult:
   Bootloader offset (No bootloader) --->
   If using Katapult:
   Bootloader offset (8KiB bootloader) --->
   If using USB communication on Type-C:
   Communication interface (USB (on PA11/PA12)) --->
   If using CAN Bus communication:
   Communication interface (CAN bus (on PD0/PD1)) --->
   (1000000) CAN bus speed

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

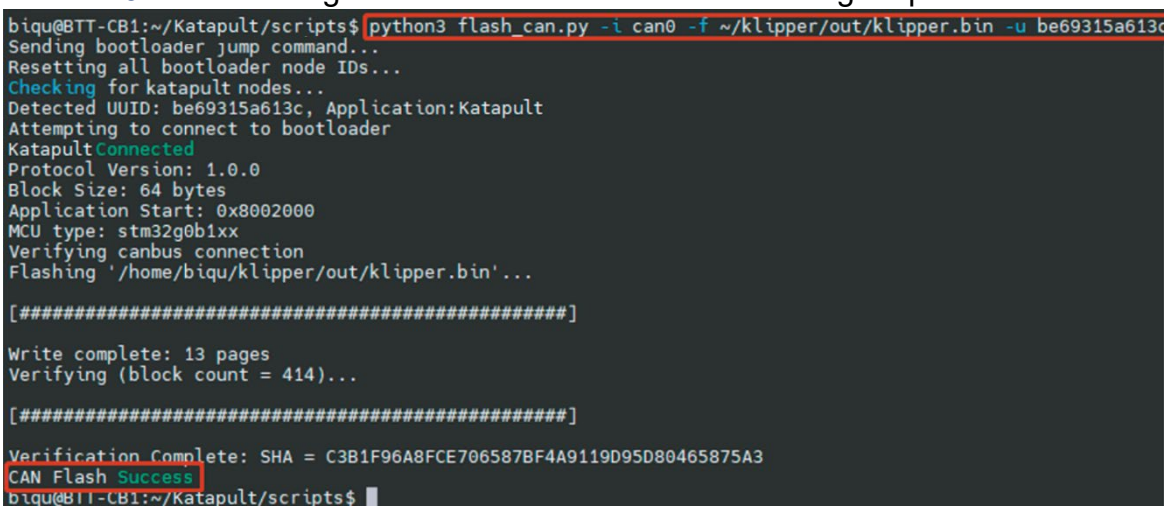
3. Enter make to compile the firmware. When make is completed, the required klipper.bin firmware will be generated in the home/pi/klipper/out folder.

## 4.3. Firmware Update via KATAPULT

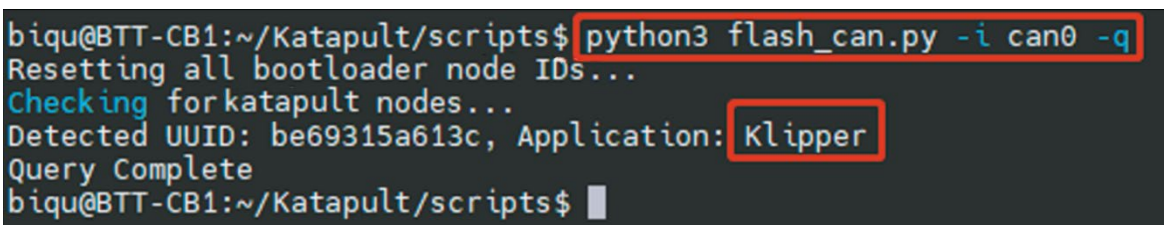
1. To use the CAN bus, ensure that the CAN bus cables are properly connected and that the jumper is inserted at the position of the 120R termination resistor. Enter
   python3 ~/katapult/scripts/flash_can.py -i can0 -q
   to query the CAN bus ID (connect the CAN cable and power-on in advance). As shown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




2. Enter
   python3 ~/katapult/scripts/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: At this point, you should have already compiled klipper.bin using "make". Additionally, when selecting the bootloader offset in the Klipper menuconfig, use the 8KiB option since Katapult's Application start offset is 8KiB. The image below shows a successful flashing sequence.




3. Re-enter
   python3 ~/katapult/scripts/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:

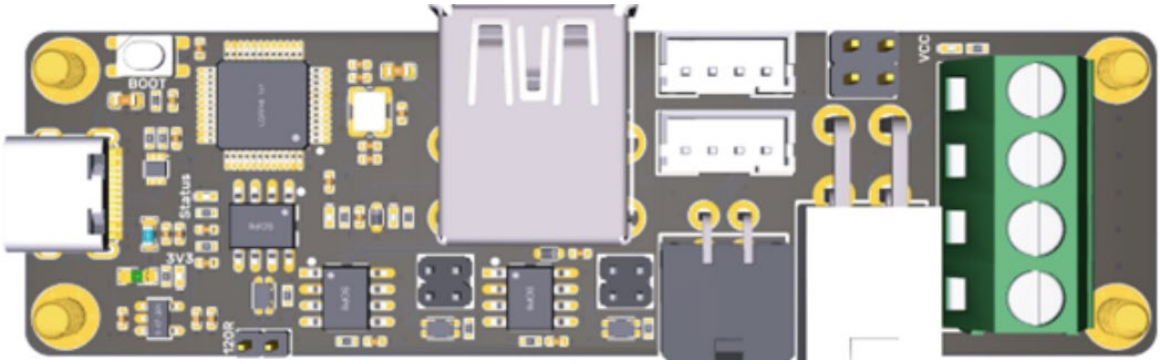
1. Hold down the Boot button and connect to Raspberry Pi/CB1 with a Type-C cable. This allows the chip to enter DFU mode.
2. In the SSH terminal command line, enter lsusb to query the DFU device ID.

```
pi@fluiddpi:~ $ lsusb
Bus 001 Device 005:  ID 0483:df11 STHicroolactronics STM Device in DFU Mode
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=0483:df11
   to start flashing the firmware (note: replace 0483:df11 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_stm32g0b1xx_4550357128922FC8-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:

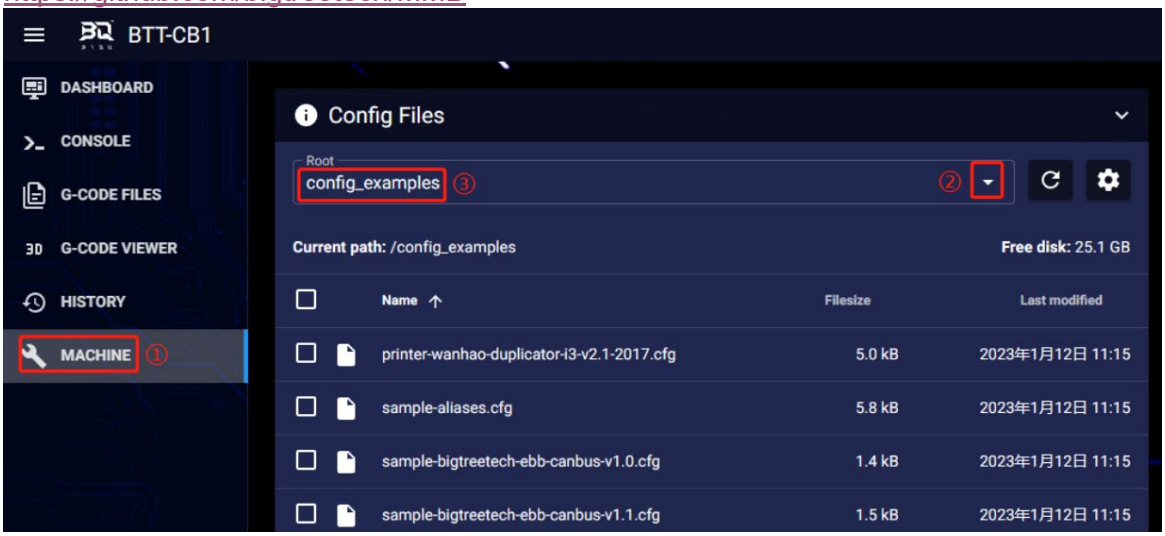
   ~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0

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

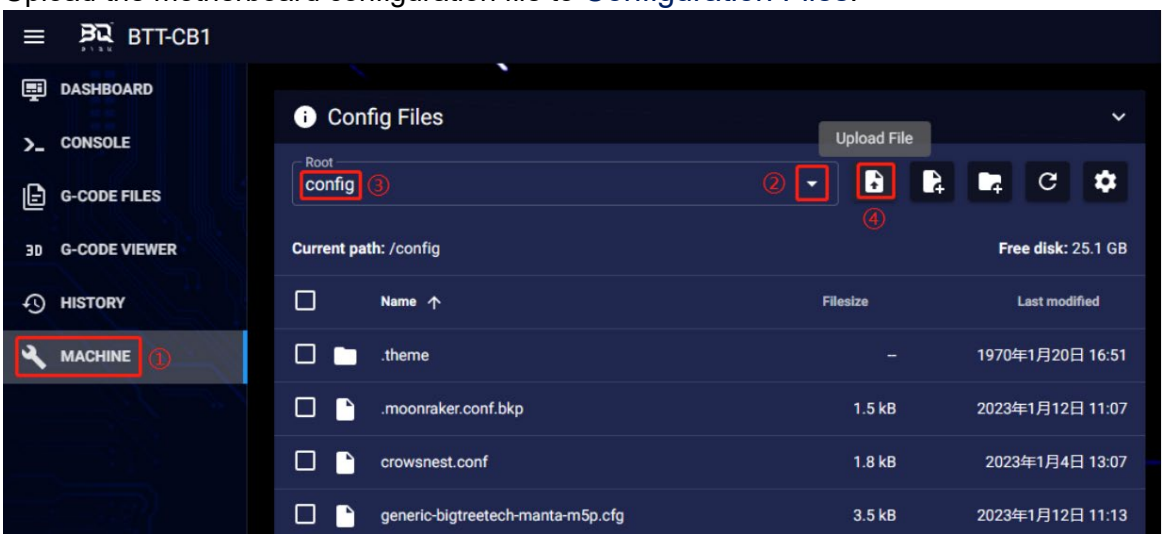
4. If Klipper is already running and connected to this device, the canbus_uuid will not be reported.

## 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-canbus-v2.0.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



2. Upload the motherboard configuration file to Configuration Files.




3. Add the MMB CAN V2 configuration to the printer.cfg file:

   [include sample-bigtreetech-mmb-canbus-v2.0.cfg]

4. Change the USB serial or CAN UUID within the configuration file to match the actual ID of the motherboard (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

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.