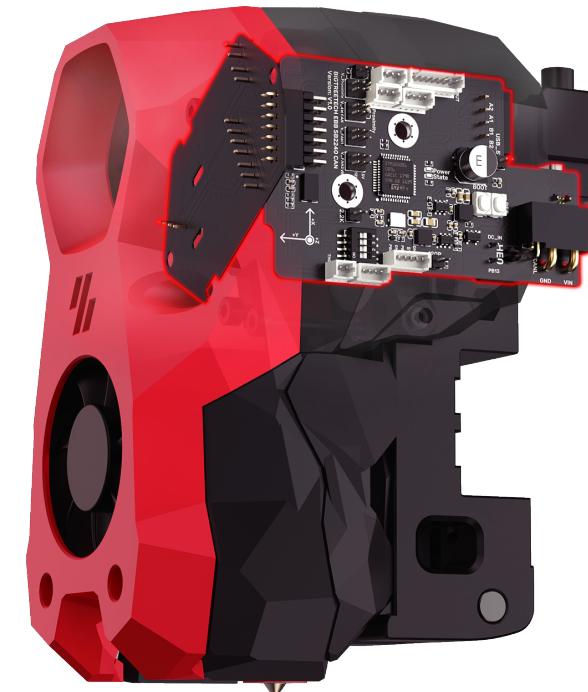


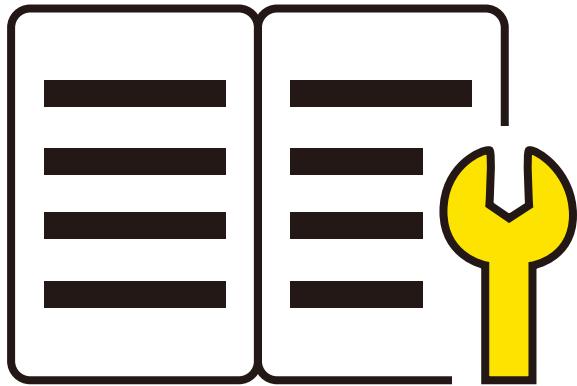
# BIGTREETECH



## EBB SB2240\2209 CAN V1.0 BUILD GUIDE

---

VERSION 2023-05-11



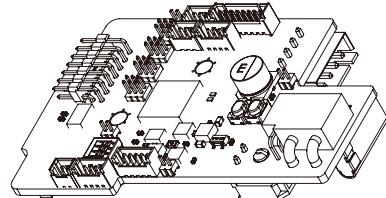
Thanks to **CHAOTICLAB** for providing guidance on Voron's official style build guide, and designing the printed part model for EBB SB2240\2209 CAN.

- Highlighted in blue are included in this EBB SB2240\2209 CAN v1.0.
- Highlighted in red are other accessories of the Voron StealthBurner, which are not included in this EBB SB2240\2209 CAN v1.0 and will need to be purchased by yourself.

## Table of Contents

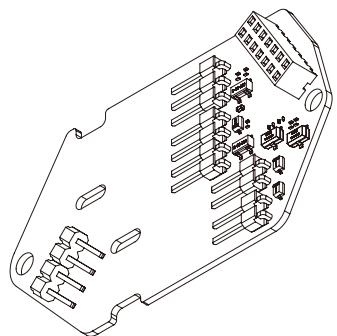
### Packing List

StealthBurner	1
EBB SB0000 CAN v1.0 Overview	1
Turbine Fan	3
Axial Fan	4
PWM Fan	5
LED	6
ClockWork2	7
EBB SB2240\2209 CAN v1.0 Overview	7
Install EBB SB2240\2209 CAN v1.0	8
CAN BUS 120R Termination Resistor	10
Extruder Motor	11
Thermistor	12
PT100/1000	13
Heater Cartridge	14
X-Carriage	15
Proximity Switch	15
X Axis Limit Switch	16
Voron Tap	17
BLTouch	18
StealthBurner Final Assembly	19
CW2 Cable Bridge	20
Fix CAN / USB-C Cable	22
StealthBurner to Mainboard Wiring	24
Klipper	25
Flashing CanBoot	25
Compile Firmware	30
Firmware Update (via Canboot)	31
Firmware Update (via DFU)	33
CAN bus Configure	37
Klipper Configure	38



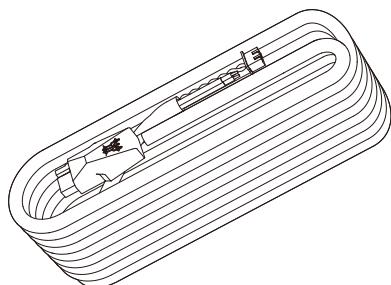
BIGTREETECH EBB SB2240/2209 CAN V1.0

1pc



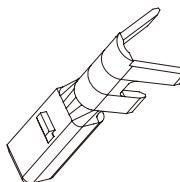
BIGTREETECH EBB SB0000 CAN V1.0

1pc



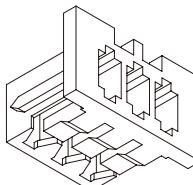
CAN bus Cable

1pc



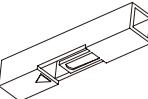
Crimp Pin

1.25 mm pitch	40pcs
3.0 mm pitch	5pcs
2.54 mm pitch	20pcs



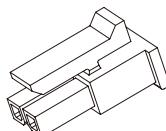
1.25 mm Pitch Pin Connector Housing

2Way	1pc
3Way	2pcs
4Way	2pcs
5Way	1pc
8Way	1pc



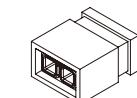
2.54mm Pitch Pin Connector Housing

1Way	15pcs
------	-------



3.0 mm Pitch Pin Connector Housing

2Way	1pc
------	-----



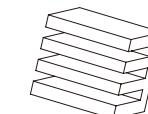
Jumper Cap

15pcs
-------



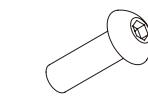
Insulated Wire Ferrule

2pcs
------



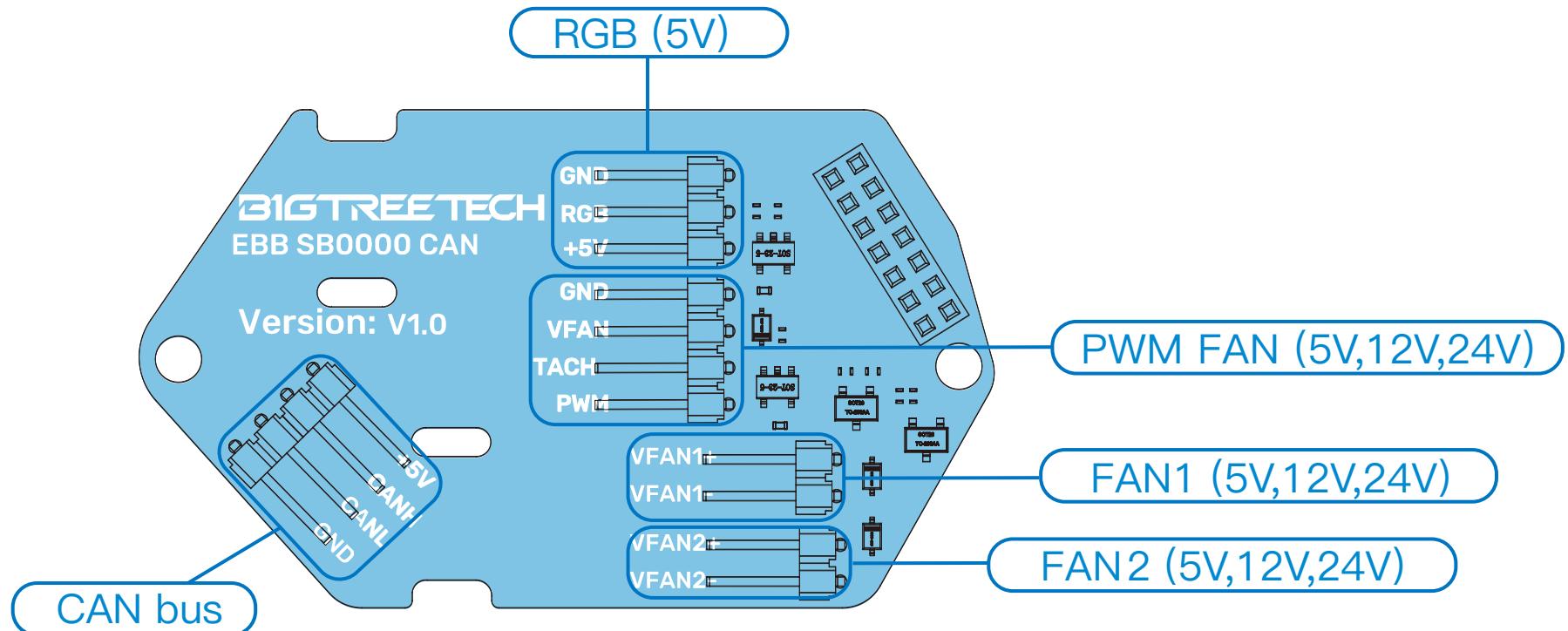
Heat Sink

1pc
-----



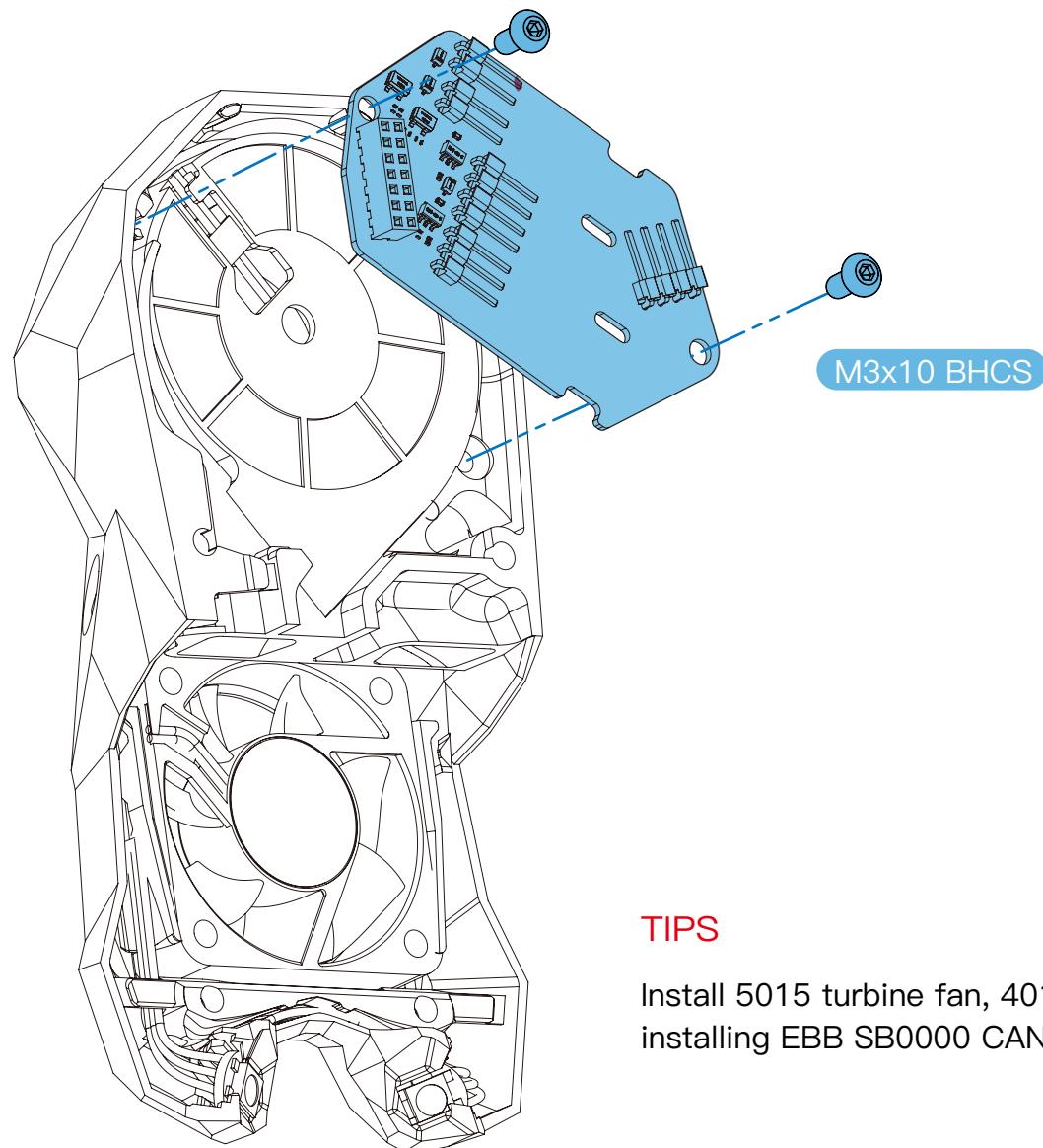
M3 x 10 Button Head Cap Screw (BHCS)

4pcs
------



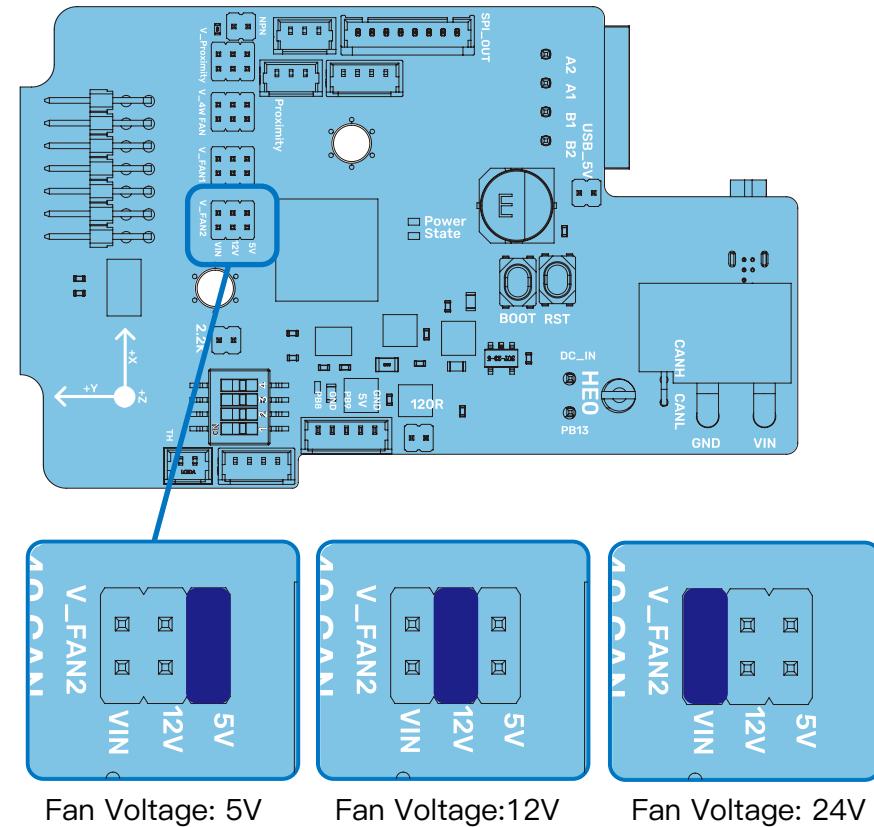
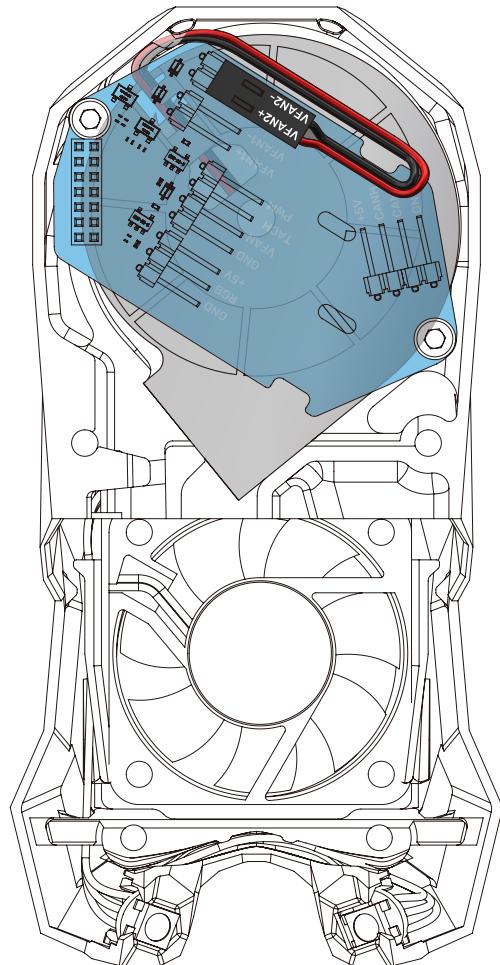
### WIRE TERMINAL

EBB SB0000 CAN v1.0 use 2.54 pitch DuPont connector. Therefore, all fans and LED lights on StealthBurner Body should be 2.54 pitch DuPont connector.



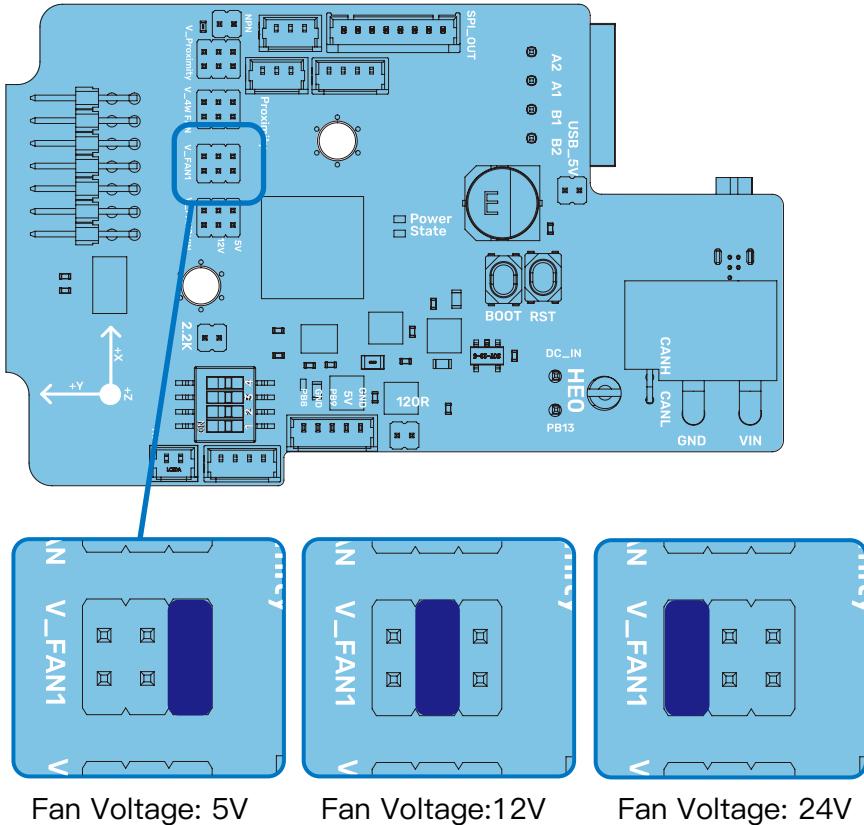
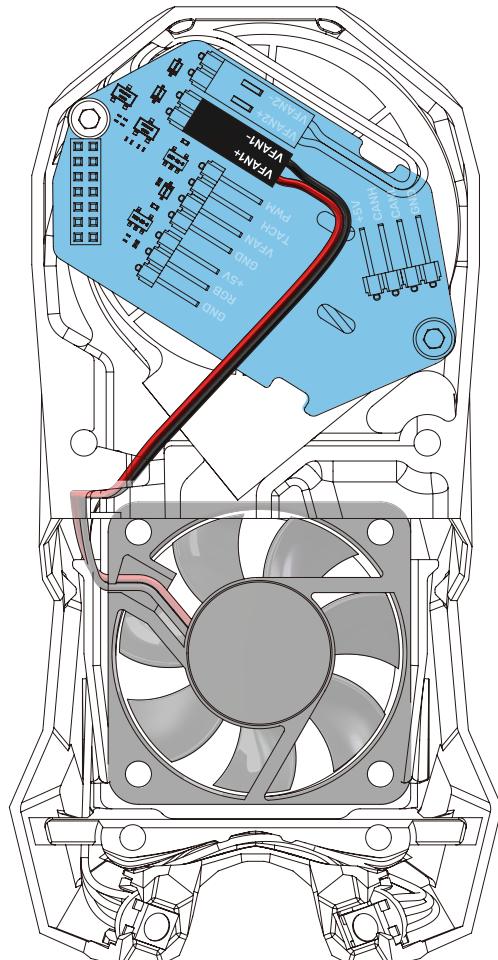
TIPS

Install 5015 turbine fan, 4010 axial fan, and LED lamp before installing EBB SB0000 CAN v1.0.



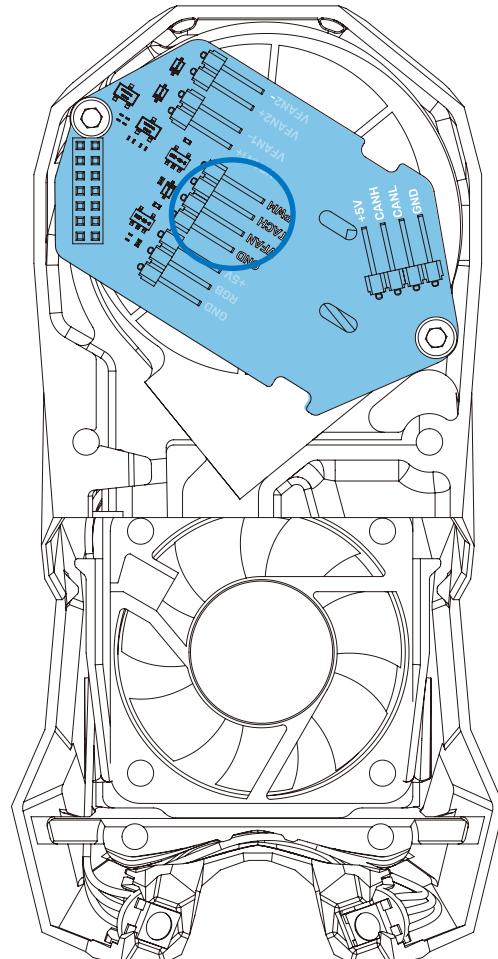
## FAN VOLTAGE

All fans on the EBB SB0000 CAN v1.0 support multi voltage selection, as shown in the figure, and jumper caps are inserted or removed according to the actual voltage used.



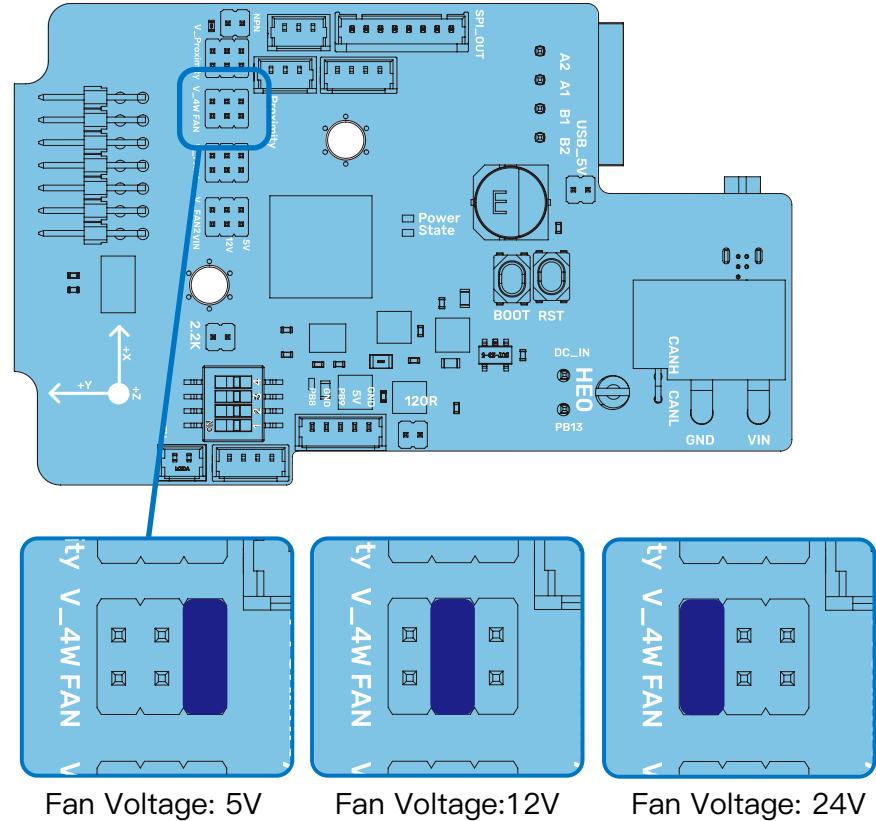
## FAN VOLTAGE

All fans on the EBB SB0000 CAN v1.0 support multi voltage selection, as shown in the figure, and jumper caps are inserted or removed according to the actual voltage used.



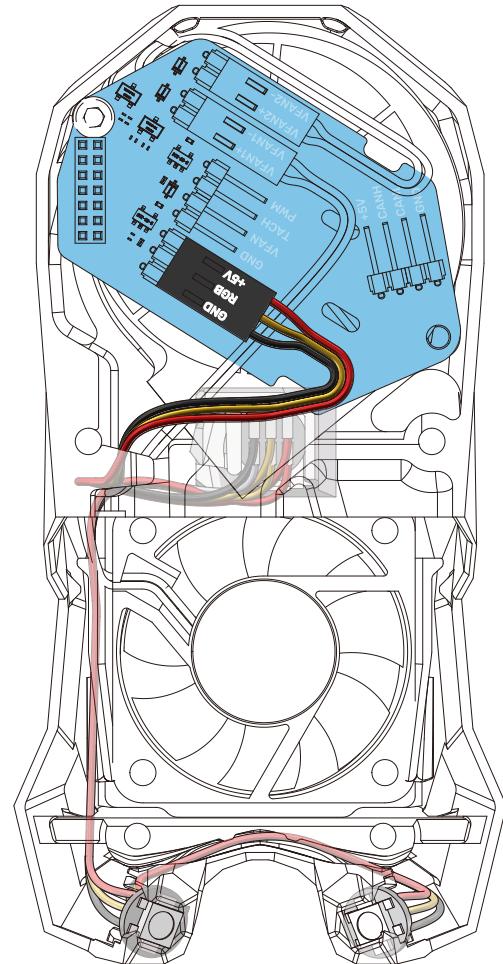
### PWM FAN

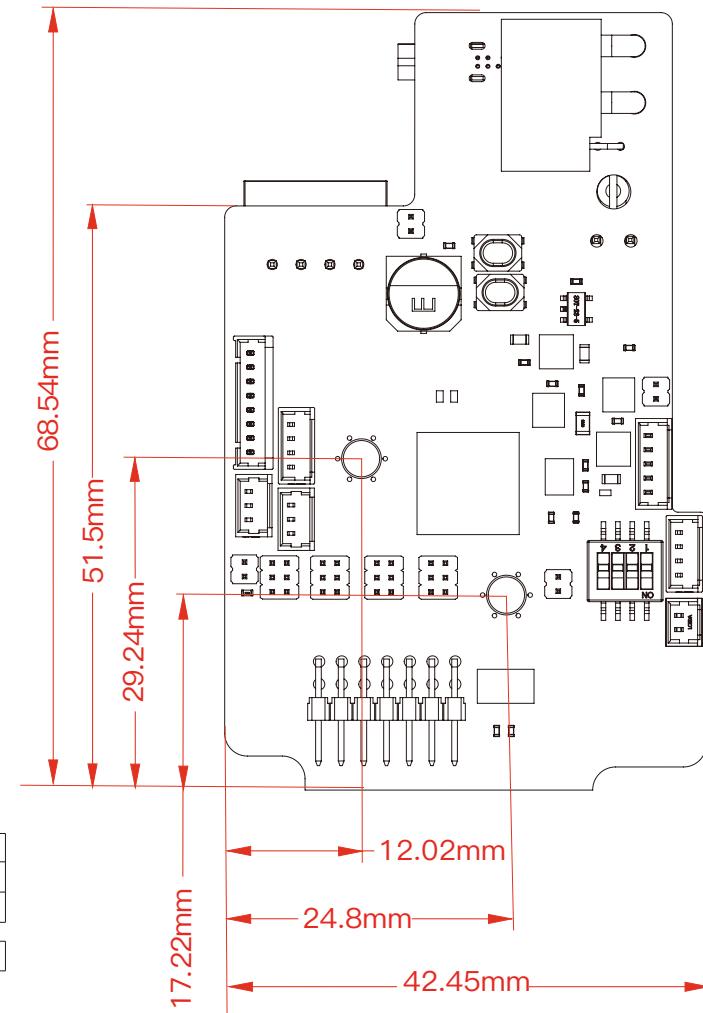
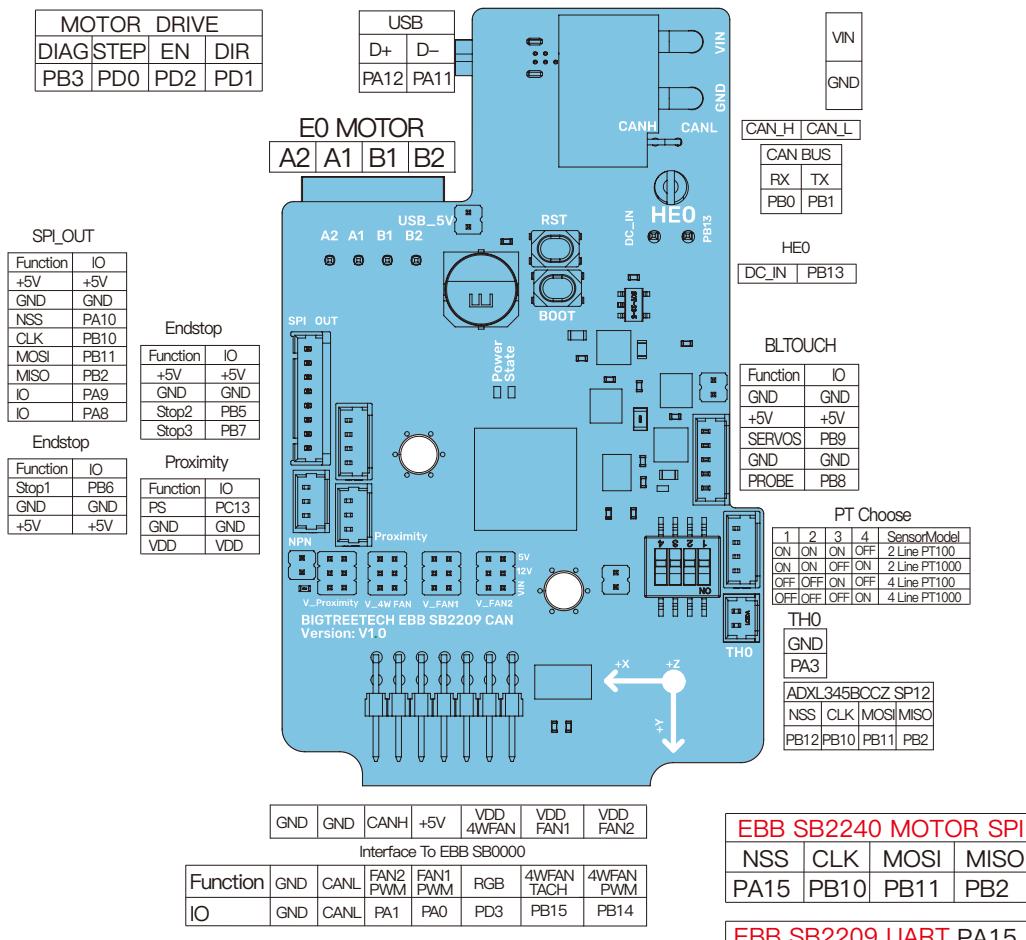
You can also use PWM fan on EBB SB0000 CAN v1.0. As shown in the figure, it contains a 4 pins PWM fan header (2.54 pitch and 4 pins DuPont connector).



### FAN VOLTAGE

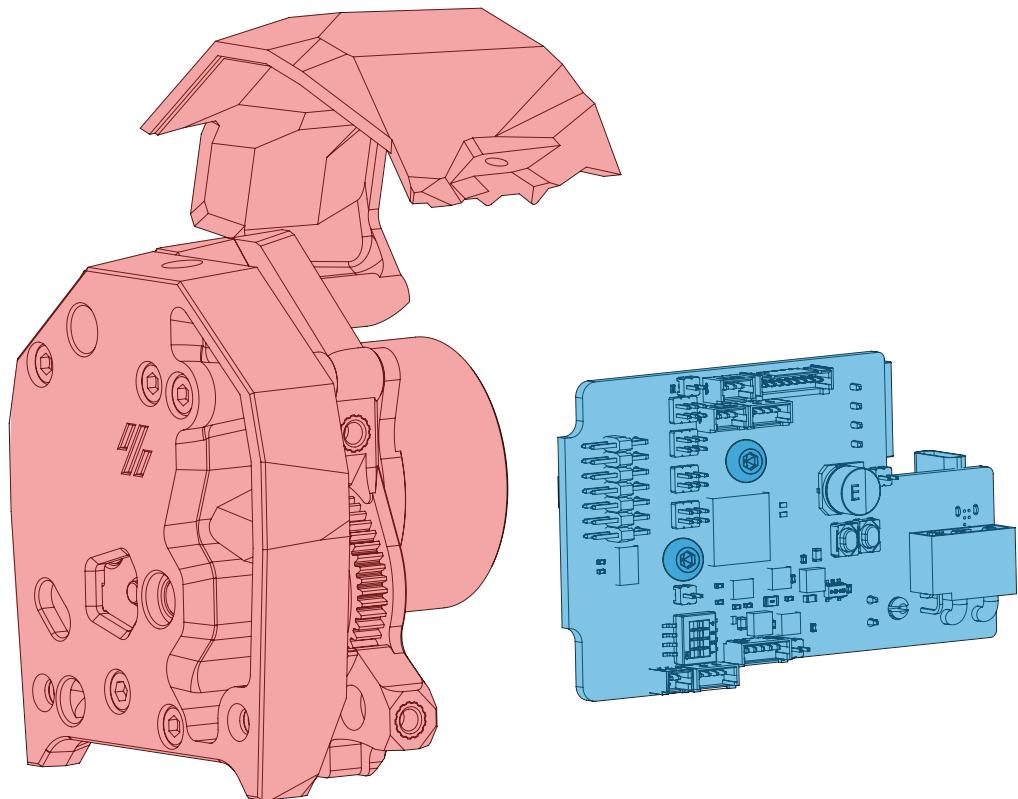
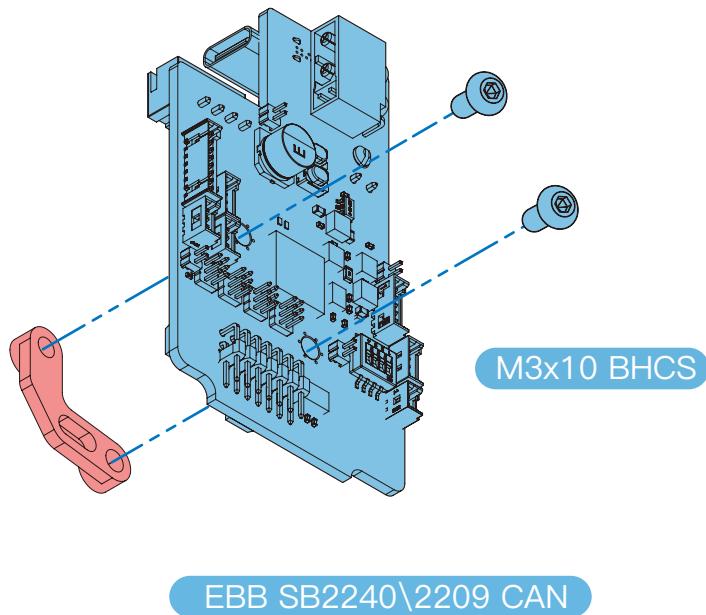
All fans on the EBB SB0000 CAN v1.0 support multi voltage selection, as shown in the figure, and jumper caps are inserted or removed according to the actual voltage used.

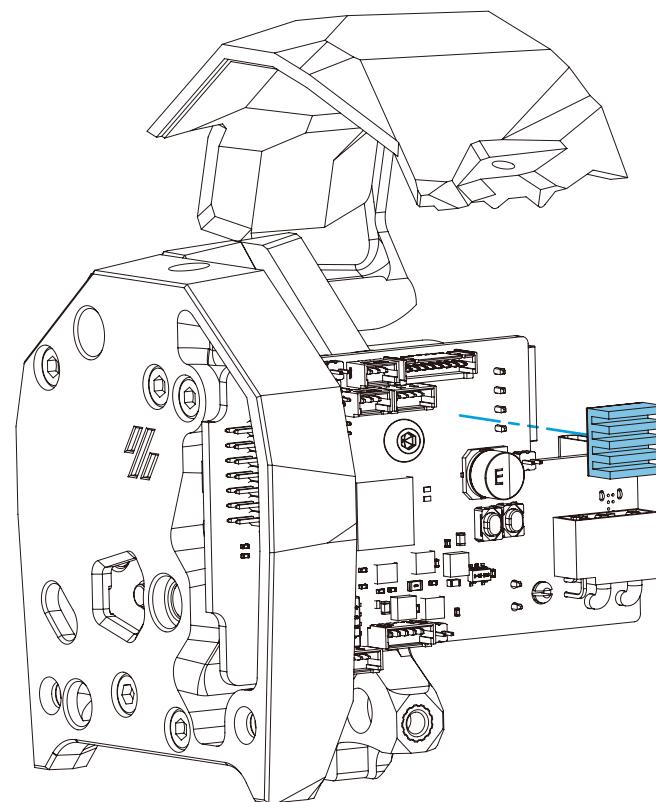
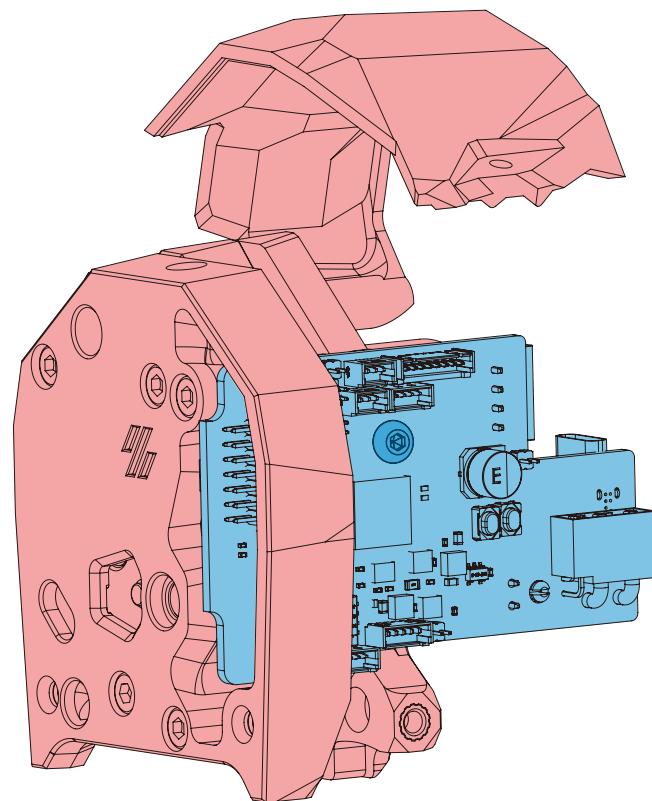




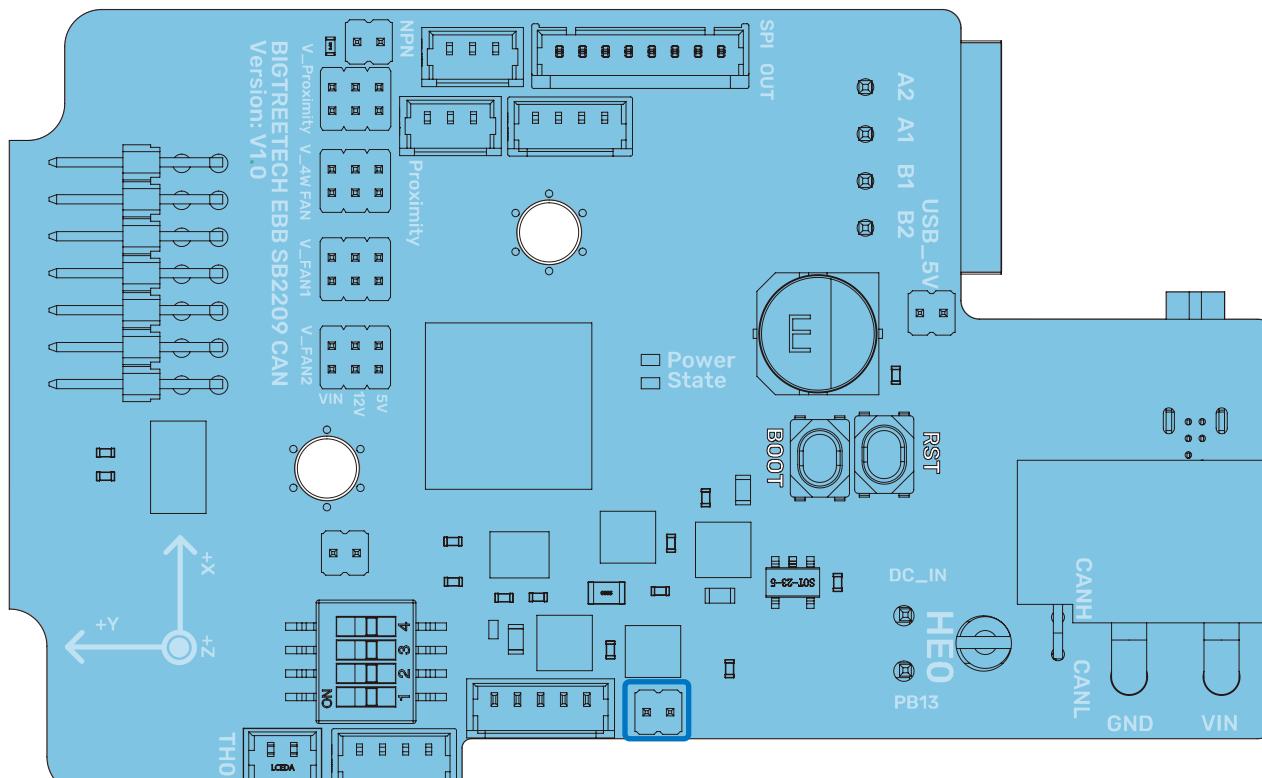
## MOTOR DRIVE

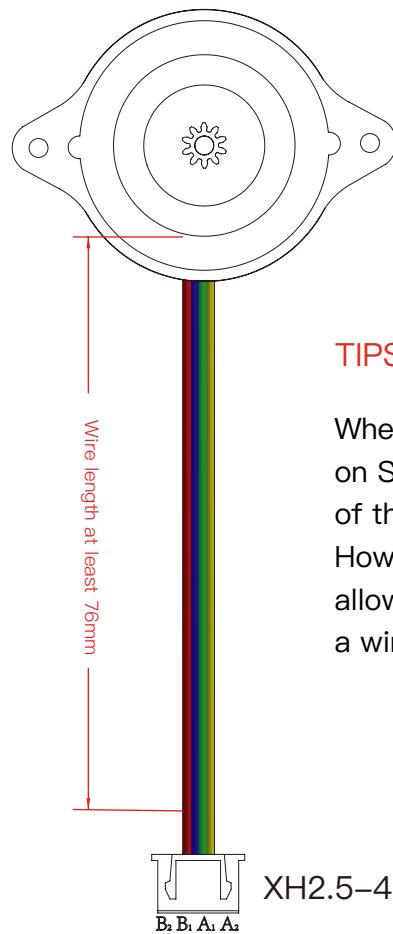
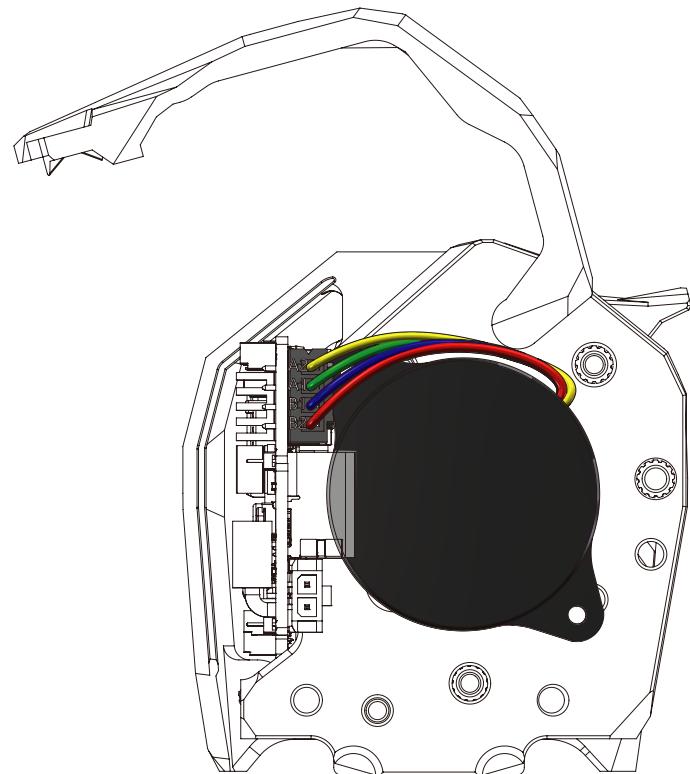
The EBB SB2240 CAN uses SPI mode, whereas the EBB SB2209 CAN uses UART.





When the EBB SB2240/2209 CAN device uses CAN bus communication, if it is the final device in the CAN bus chain, you must plug a jumper at the 120R position.

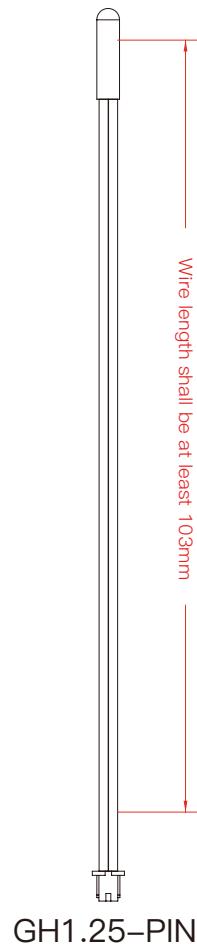
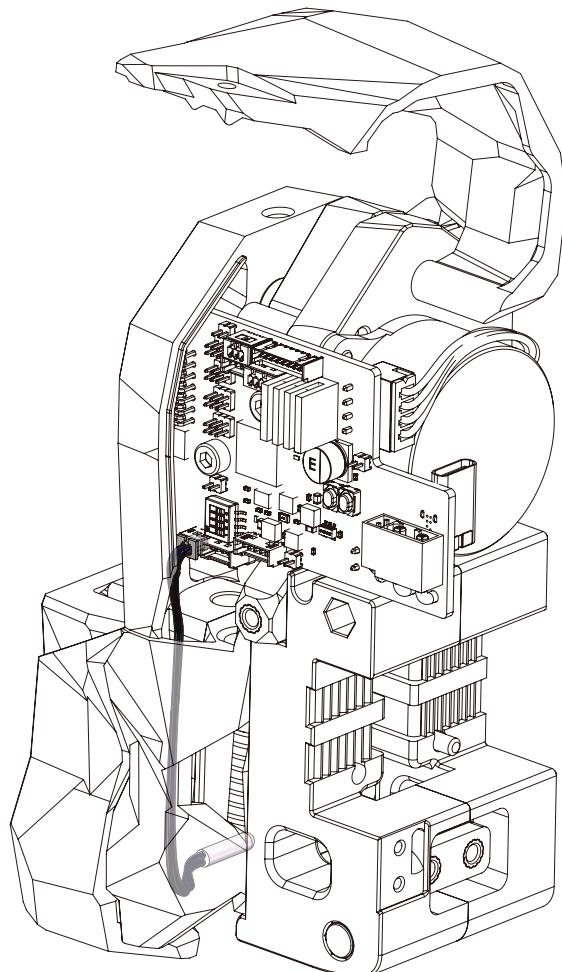




### TIPS

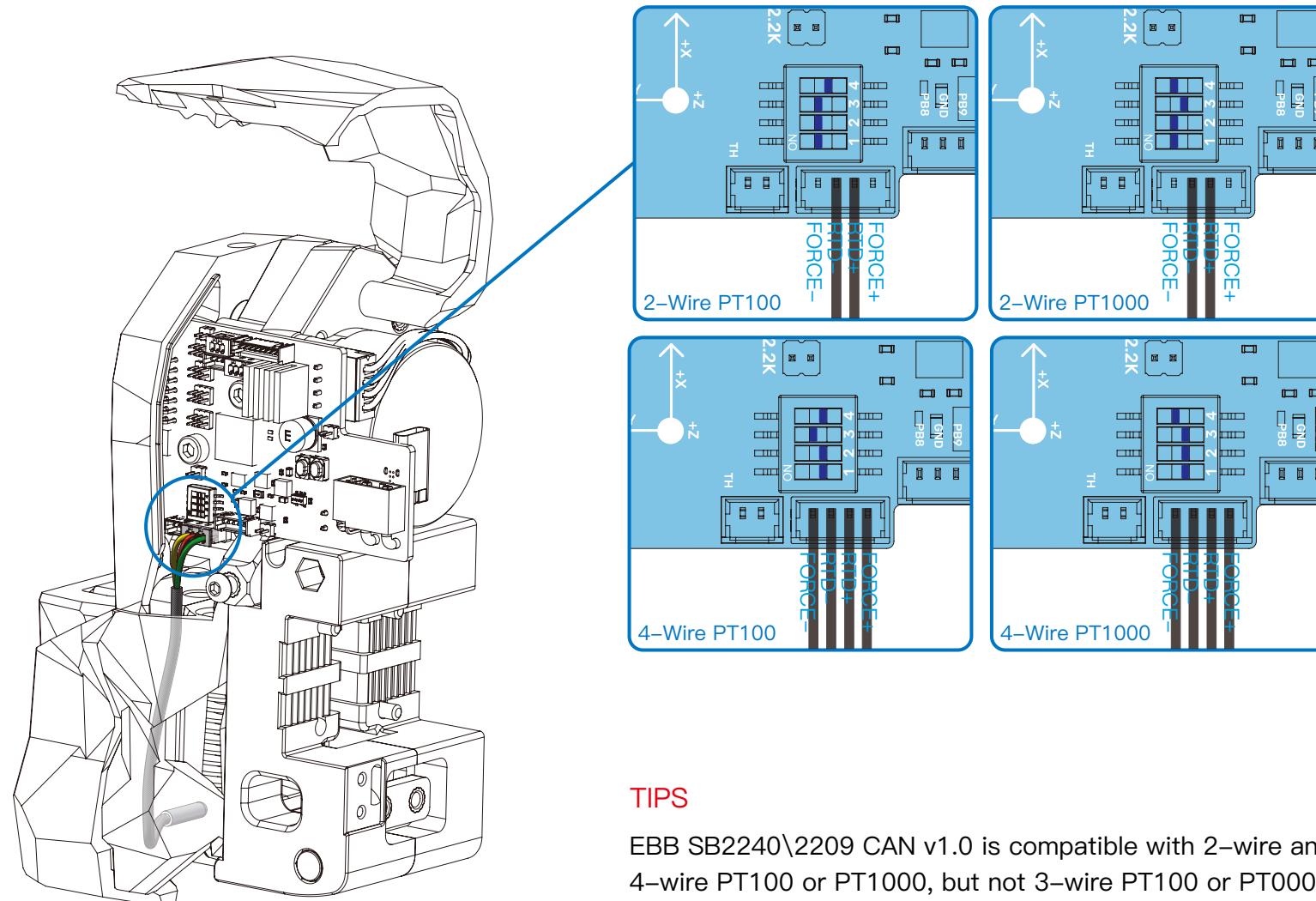
When EBB SB2240\2209 CAN v1.0 is used on StealthBurner, the minimum wire length of the motor is 76mm.

However, it is recommended to leave some allowance for wiring and assembly, such as a wire length of 100mm or 110mm.



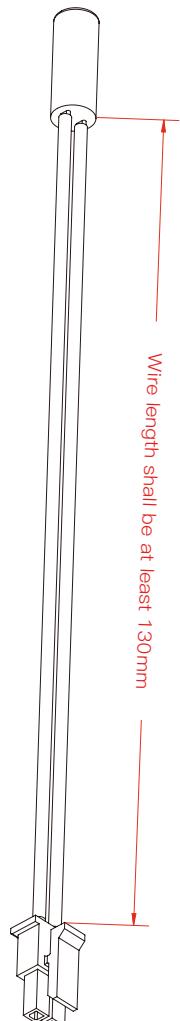
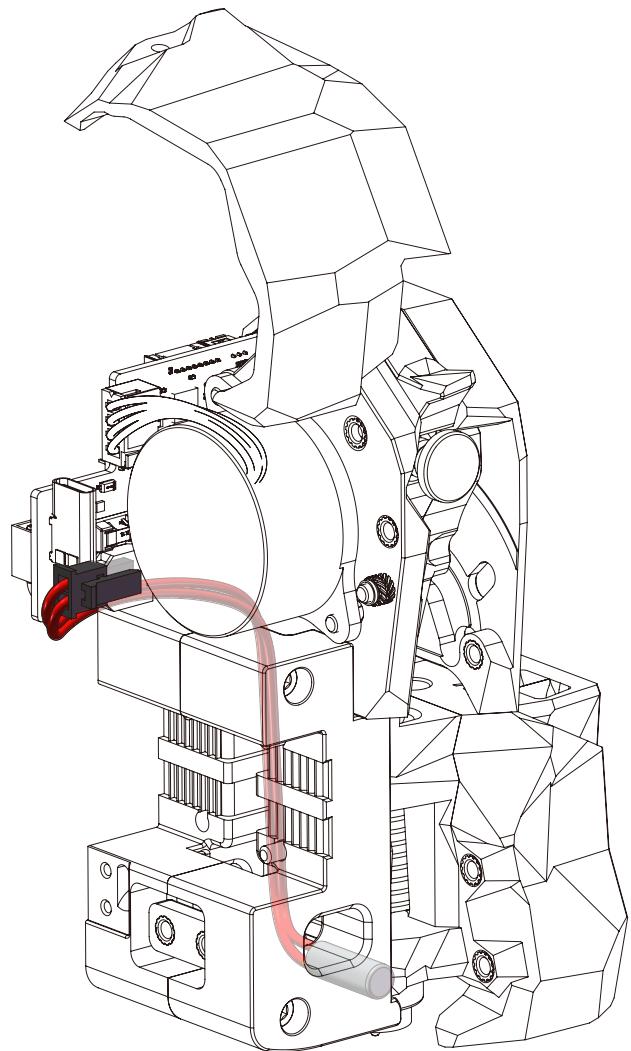
### TIPS

When EBB SB2240\2209 CAN v1.0 is used on StealthBurner, the minimum wire length of the thermistor is 103mm. However, it is recommended to leave some allowance for wiring and assembly, such as a wire length of 120mm or 130mm.



### TIPS

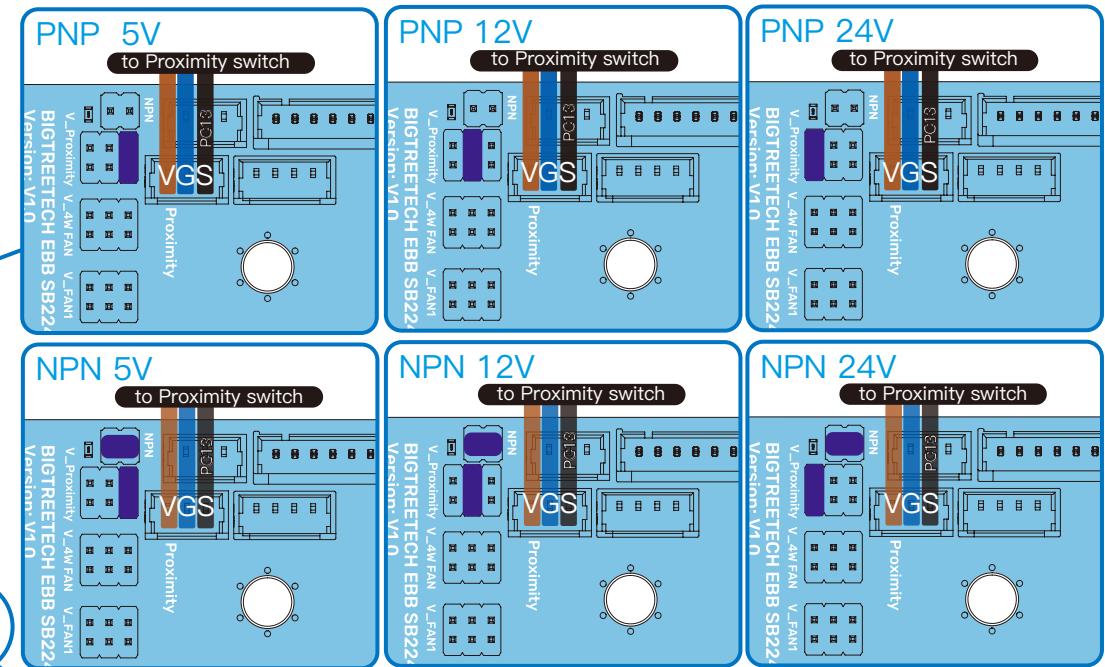
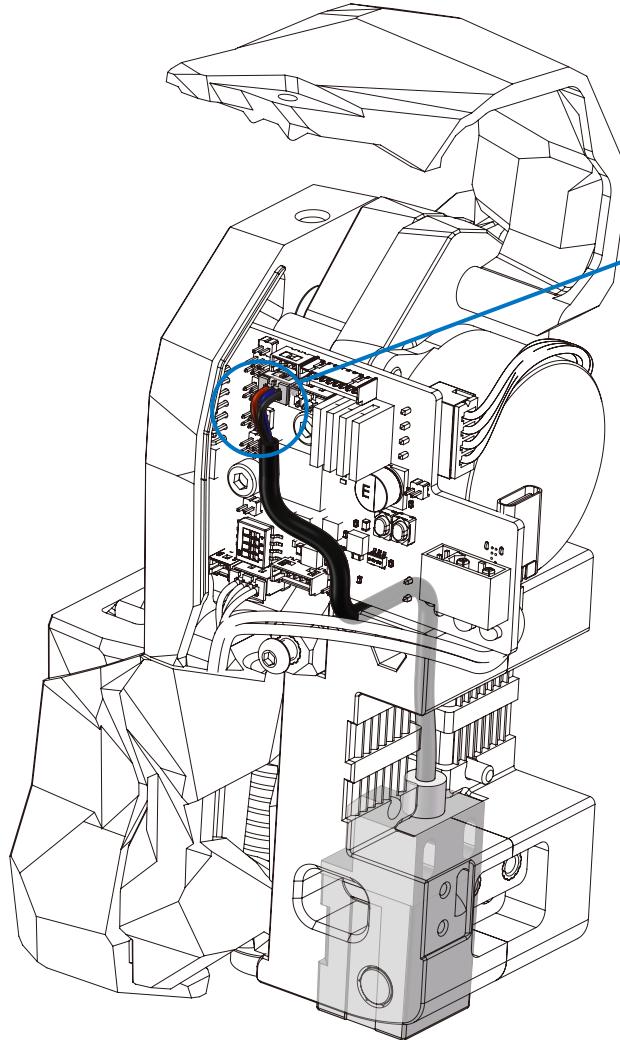
EBB SB2240\2209 CAN v1.0 is compatible with 2-wire and 4-wire PT100 or PT1000, but not 3-wire PT100 or PT000. As shown in the figure, wire and adjust the dial switch according to the actual model. And the terminals are GH1.25.



MX3.0-2PIN

**TIPS**

When EBB SB2240\2209 CAN v1.0 is used on StealthBurner, the minimum wire length of the heater cartridge is 130mm. However, it is recommended to leave some allowance for wiring and assembly, such as a wire length of 150mm or 160mm.

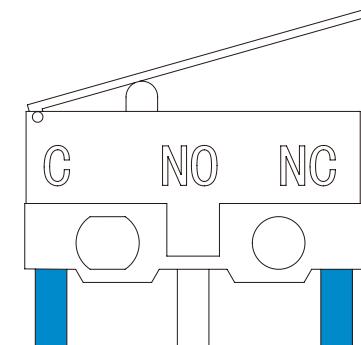
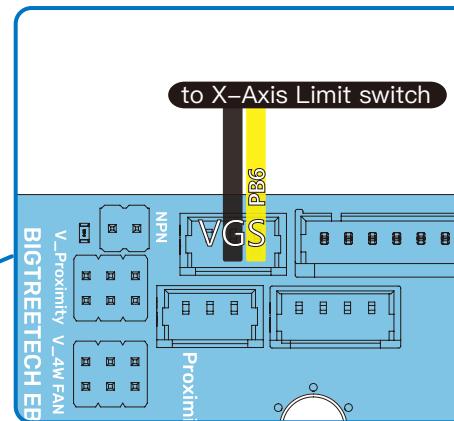
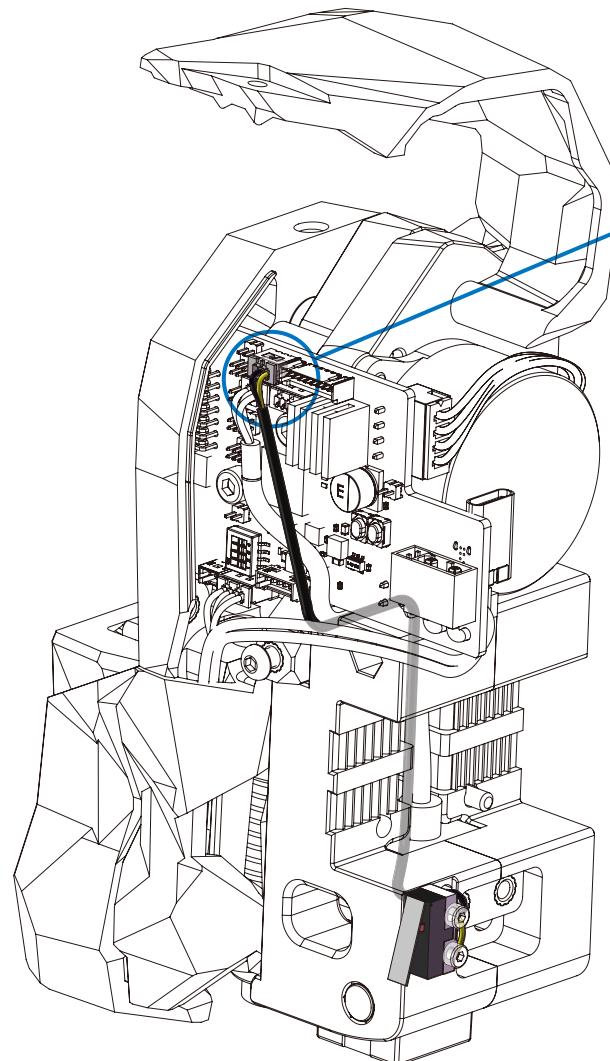


### PROXIMITY SWITCH

The proximity switch is divided into PNP and NPN and has various voltages. As shown in the figure, the jumper cap is inserted or removed at the corresponding position according to the actual model.

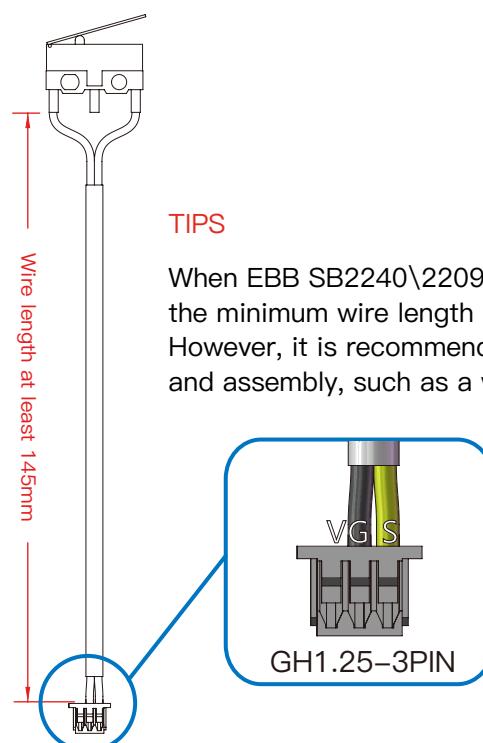
### TIPS

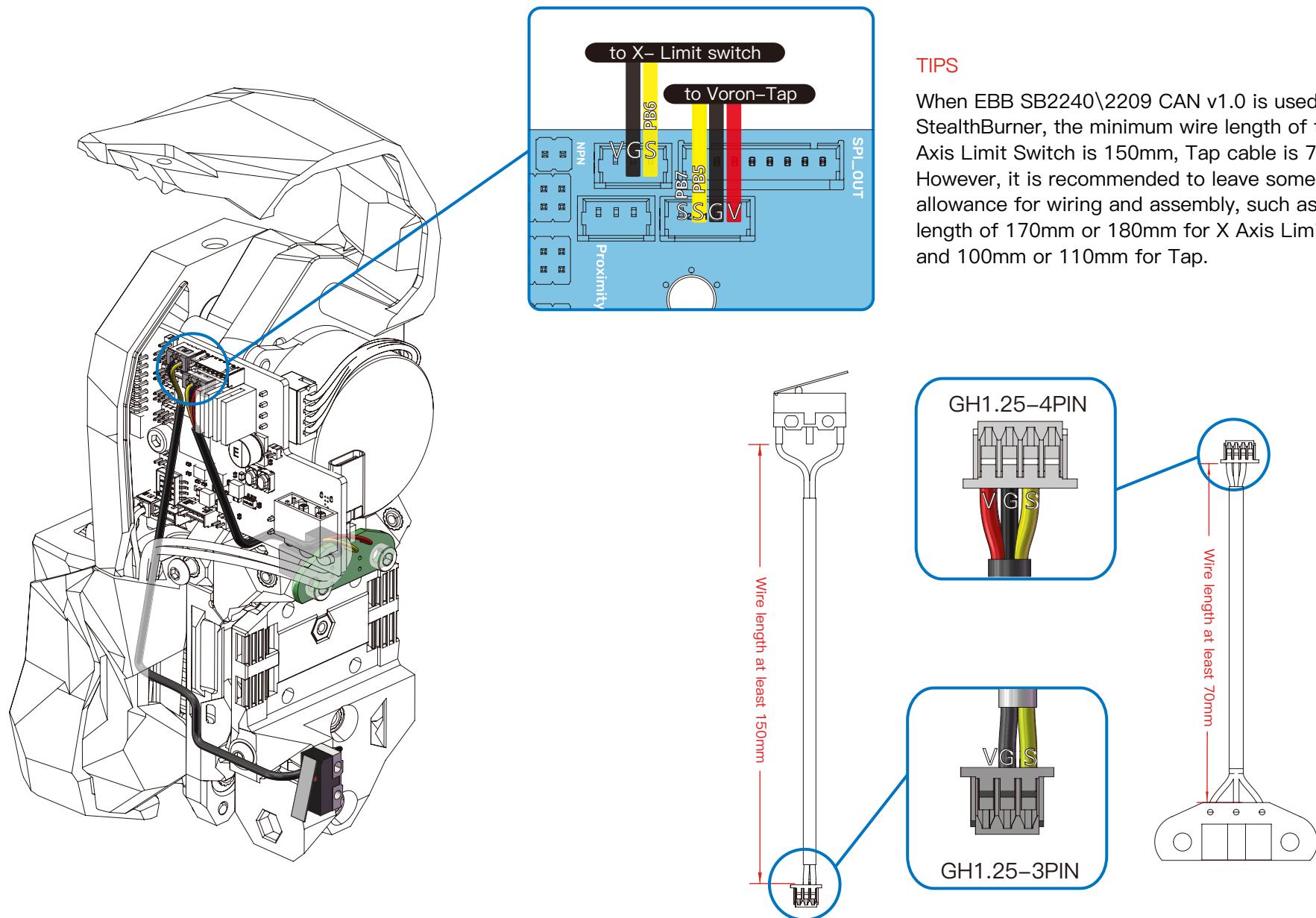
When EBB SB2240\2209 CAN v1.0 is used on StealthBurner, the minimum wire length of the PL-08N is 100mm. However, it is recommended to leave some allowance for wiring and assembly, such as a wire length of 130mm or 140mm.



### END-STOP SWITCHES FOR X

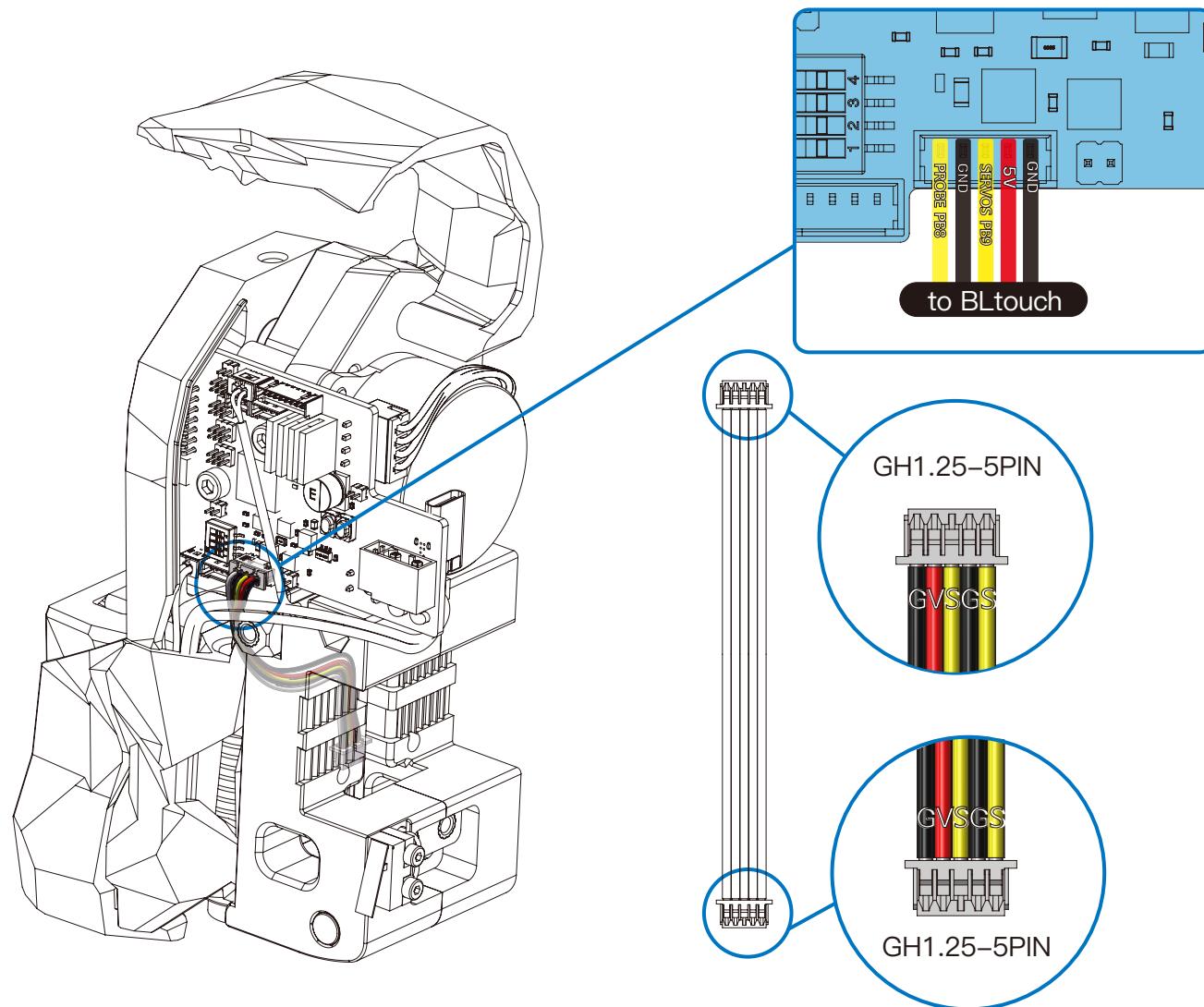
End-stops are wired in a "Normally Closed" configuration. On microswitches those are the 2 outer terminals indicated by C and NC.

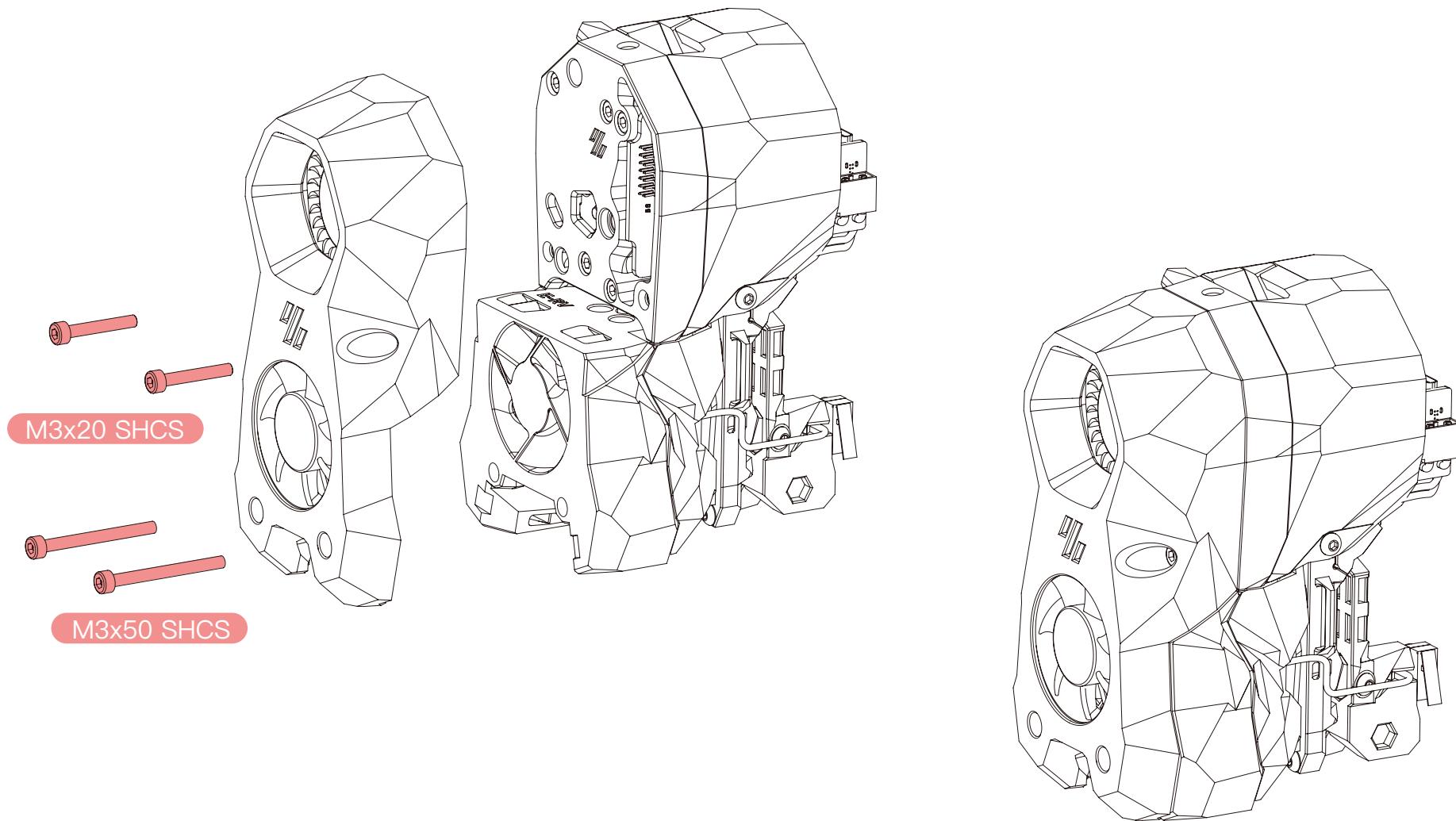


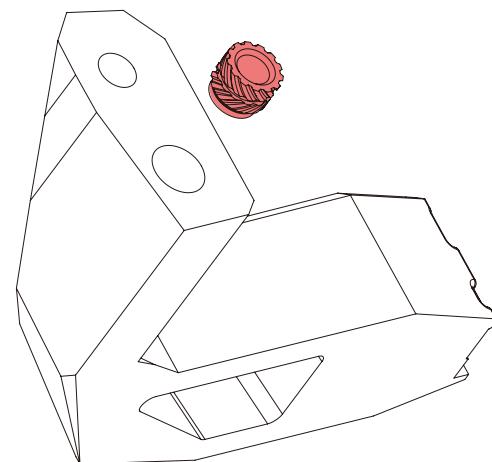
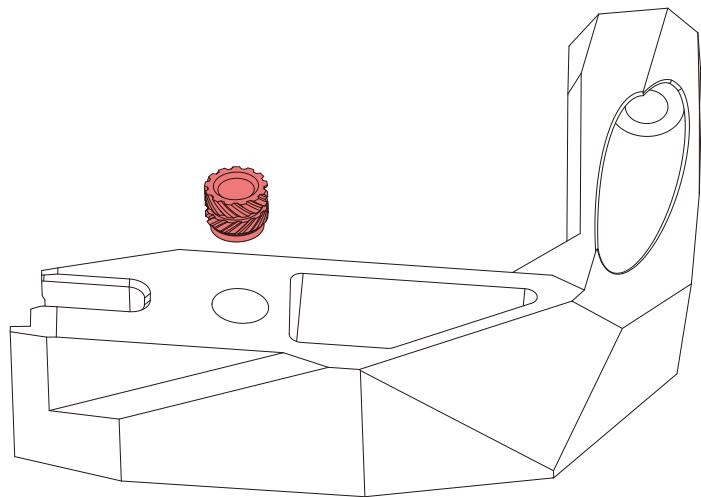


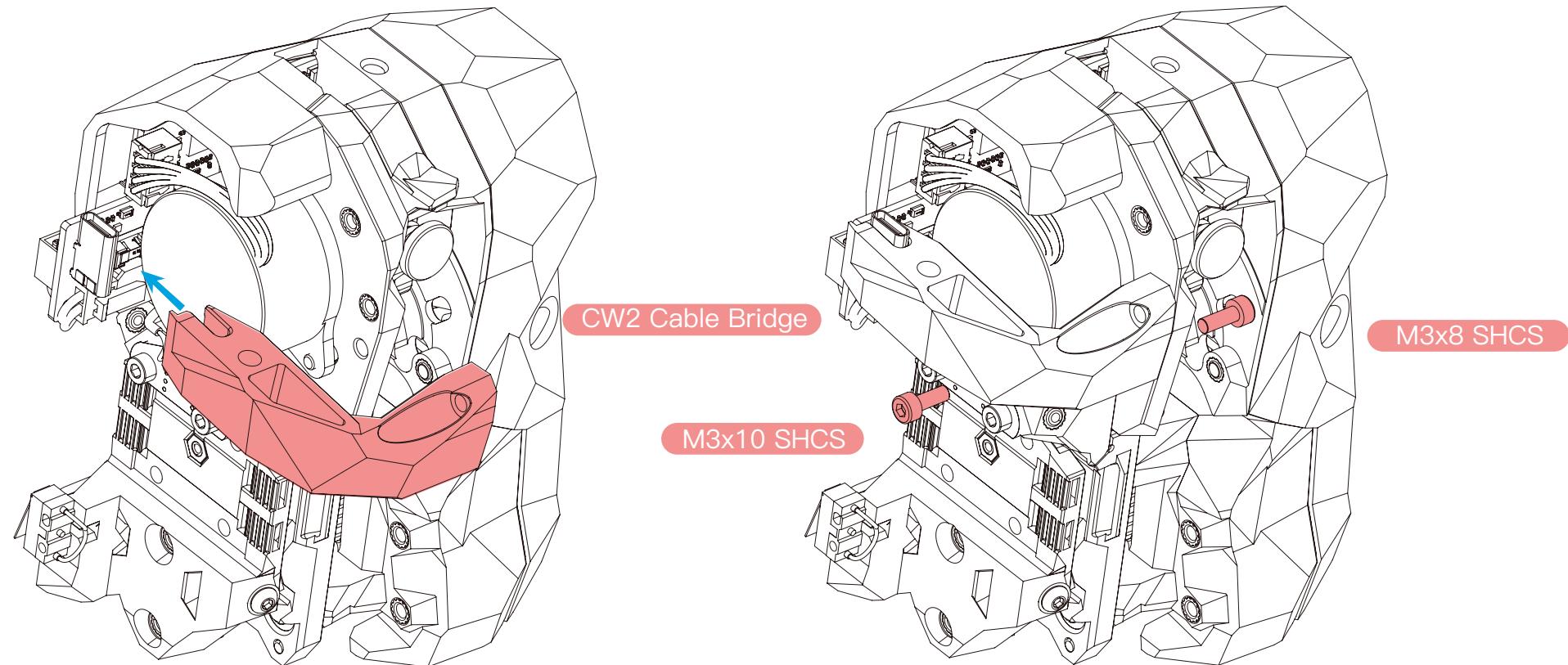
## TIPS

When EBB SB2240\2209 CAN v1.0 is used on StealthBurner, the minimum wire length of the X Axis Limit Switch is 150mm, Tap cable is 70mm. However, it is recommended to leave some allowance for wiring and assembly, such as a wire length of 170mm or 180mm for X Axis Limit Switch and 100mm or 110mm for Tap.

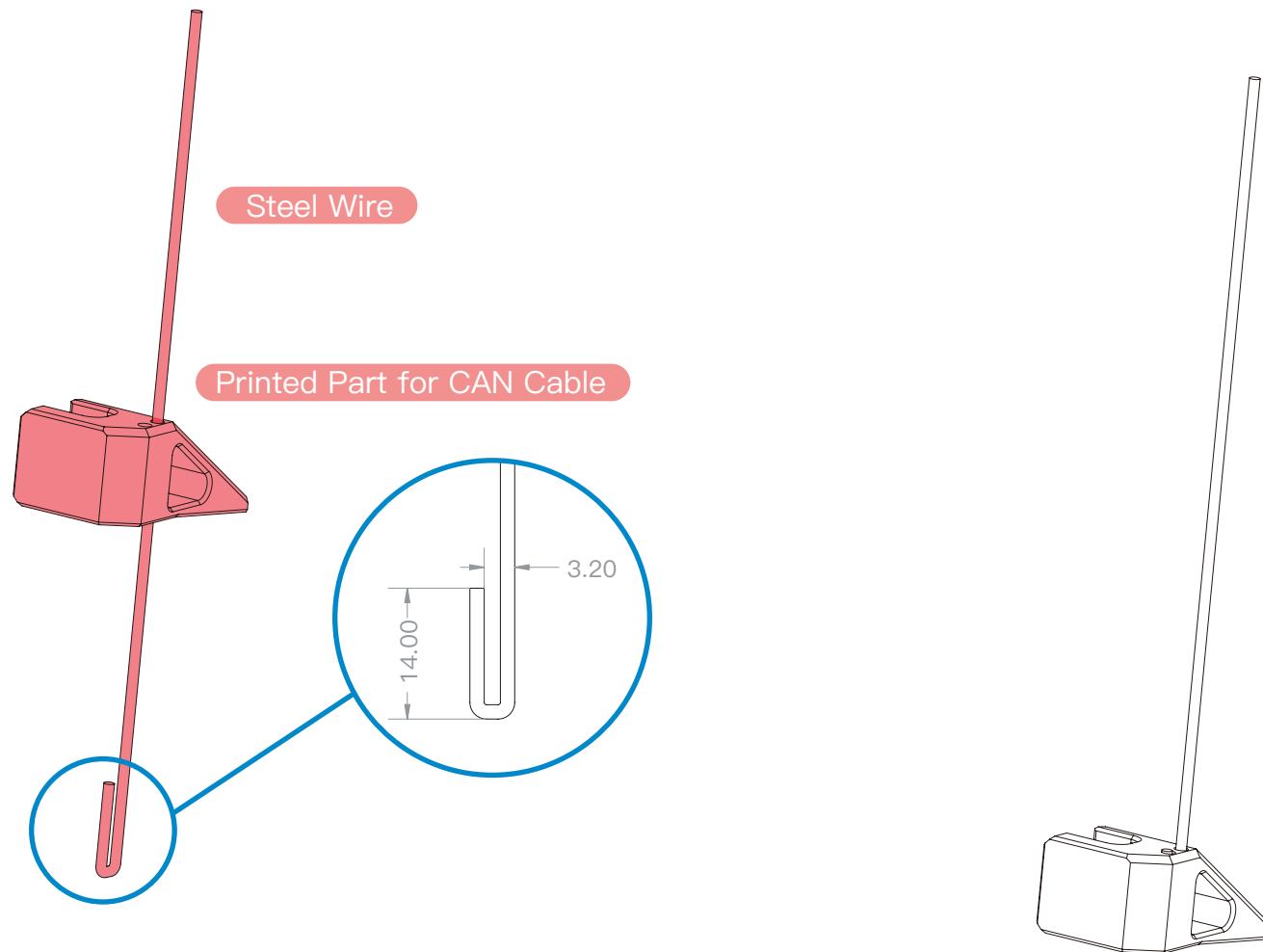






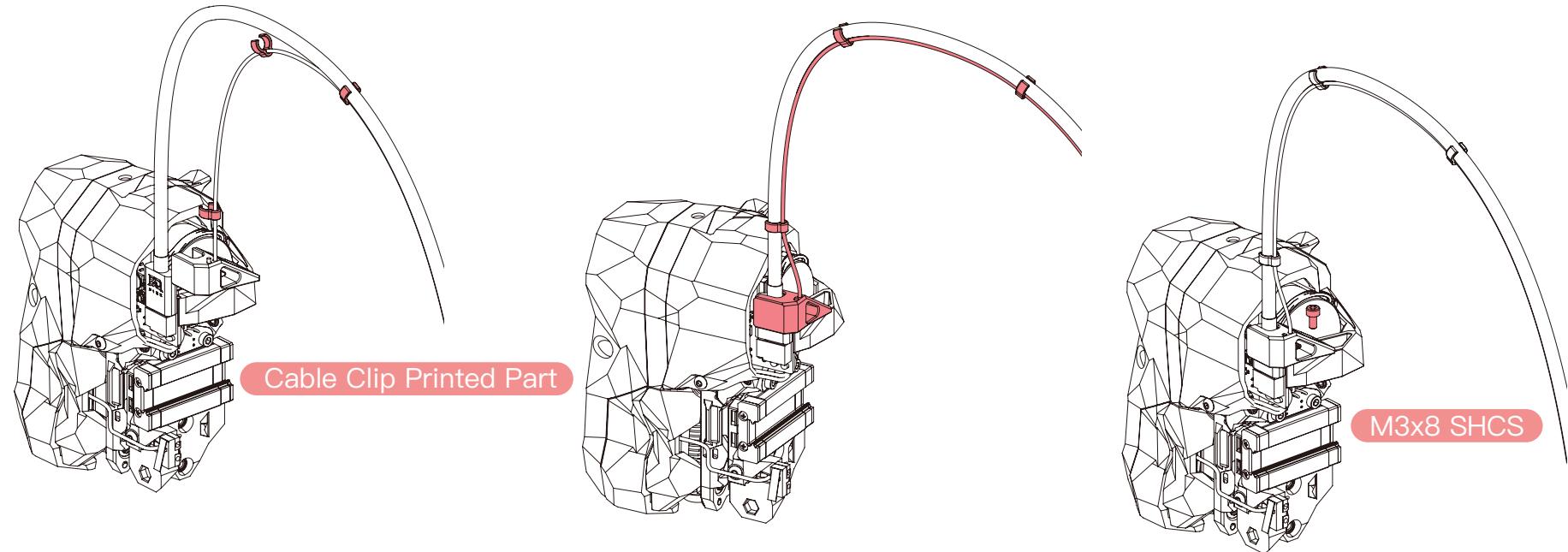
**TIPS**

The CW2 cable bridge is different from the official version, it is a custom printed part that can be downloaded directly from our GitHub repository, in EBB\EBB SB2240\_2209 CAN\Custom Printed Parts.



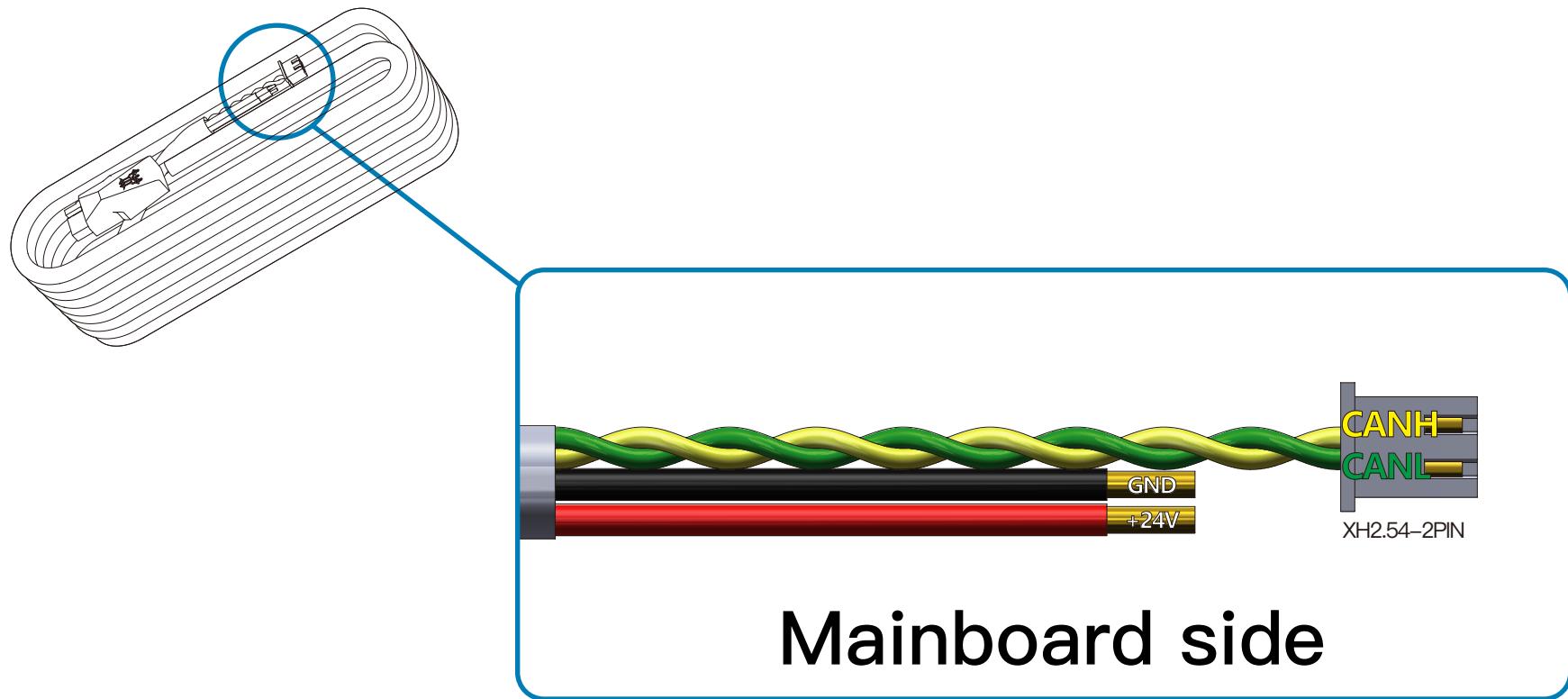
### TIPS

The printed part for CAN cable is a custom printed part that can be downloaded directly from our GitHub repository, in EBB\EBB SB2240\_2209 CAN\Custom Printed Parts, as well as the printed part for USB-C cable.



### CABLE CLIP

The cable clip printed part can be downloaded from our GitHub repository, in EBB\EBB SB2240\_2209 CAN\Custom Printed Parts, or you can use zip ties instead.



Note: Please note that CanBoot is designed for the purpose of directly updating the MCU firmware via the CAN bus interface. If you prefer the DFU update method, you may skip this step.

### “Flashing CanBoot on a CB1/Raspberry Pi”

Refer to the instructions here to download the CanBoot project

<https://github.com/Arksine/CanBoot>

1.Run:

```
cd ~
```

to enter the home directory, then run:

```
git clone https://github.com/Arksine/CanBoot
```

to download CanBoot project.

run:

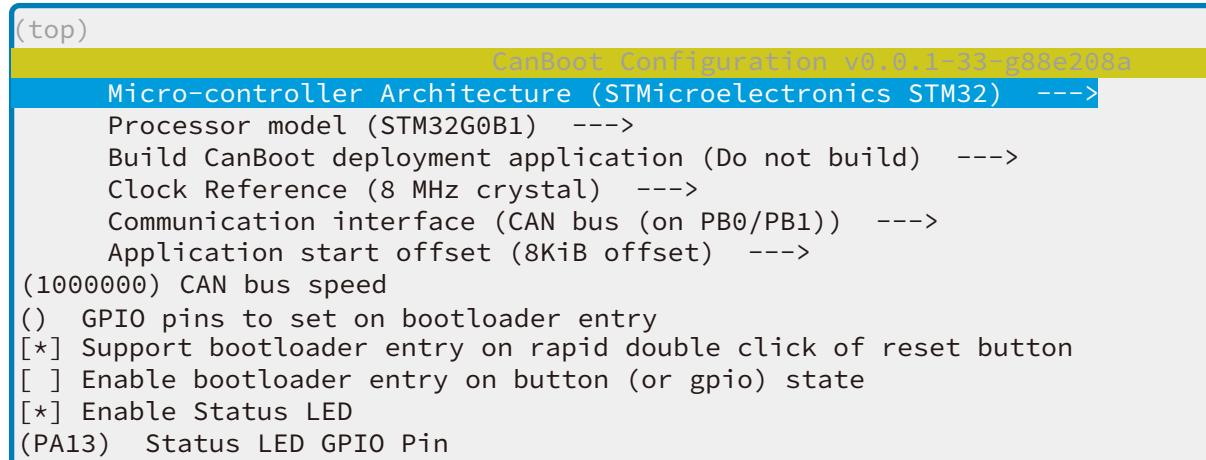
```
cd CanBoot
```

to enter the CanBoot directory.

2.Run:

```
make menuconfig
```

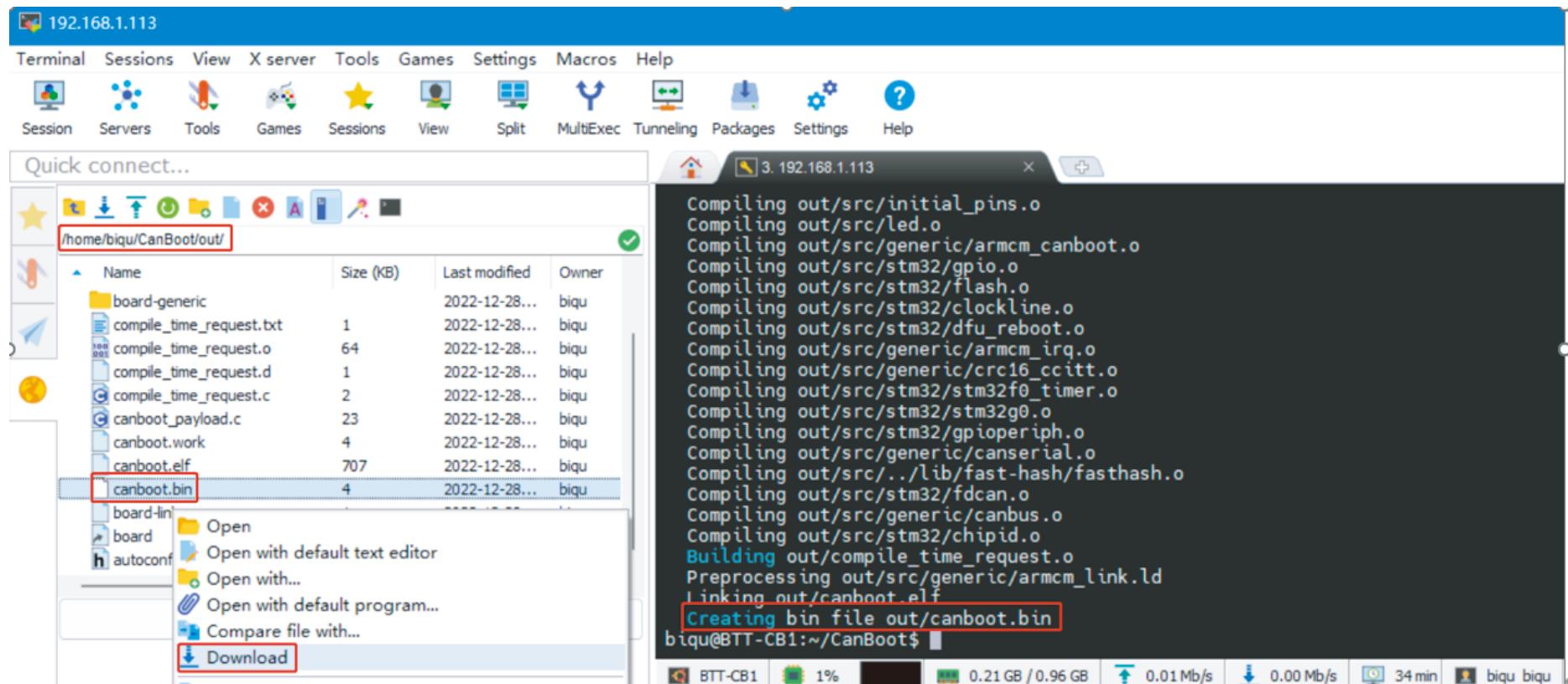
and configure according to the following figure



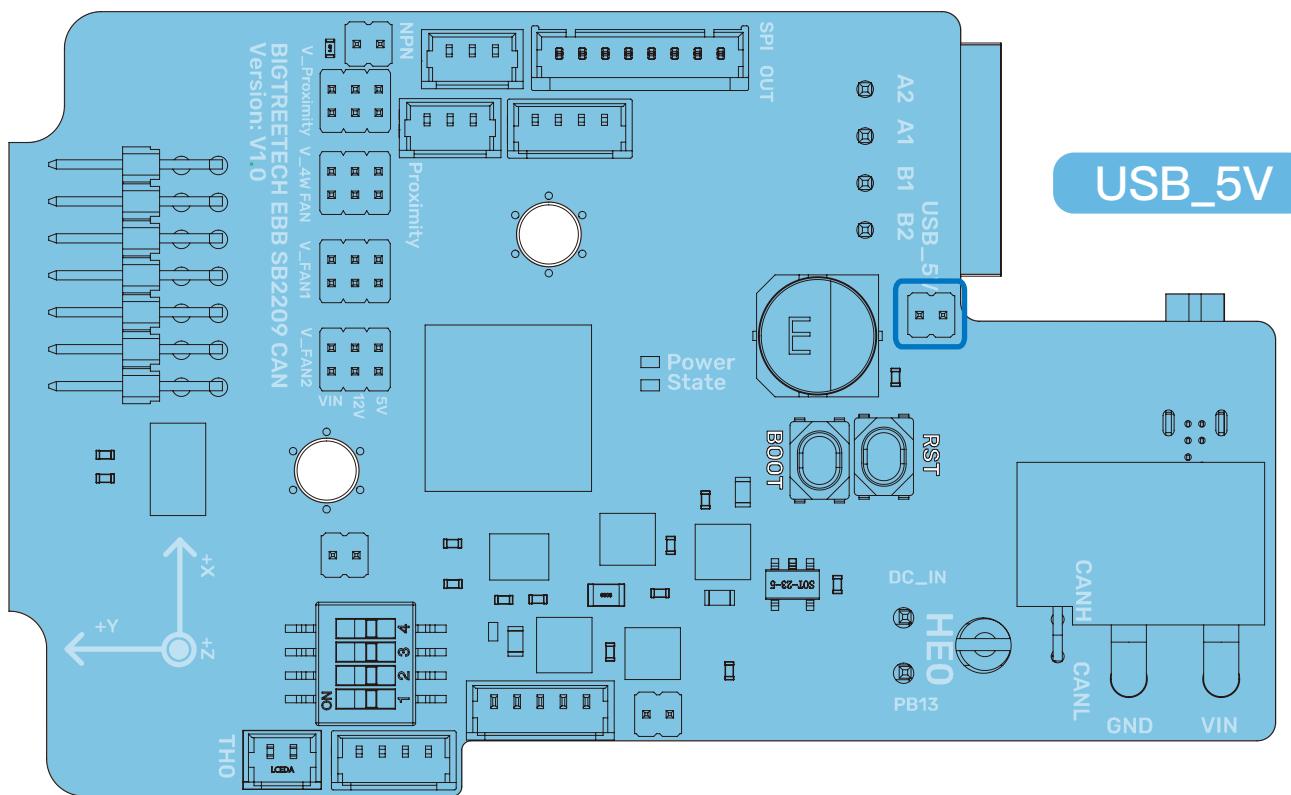
## 3.Run

`make`

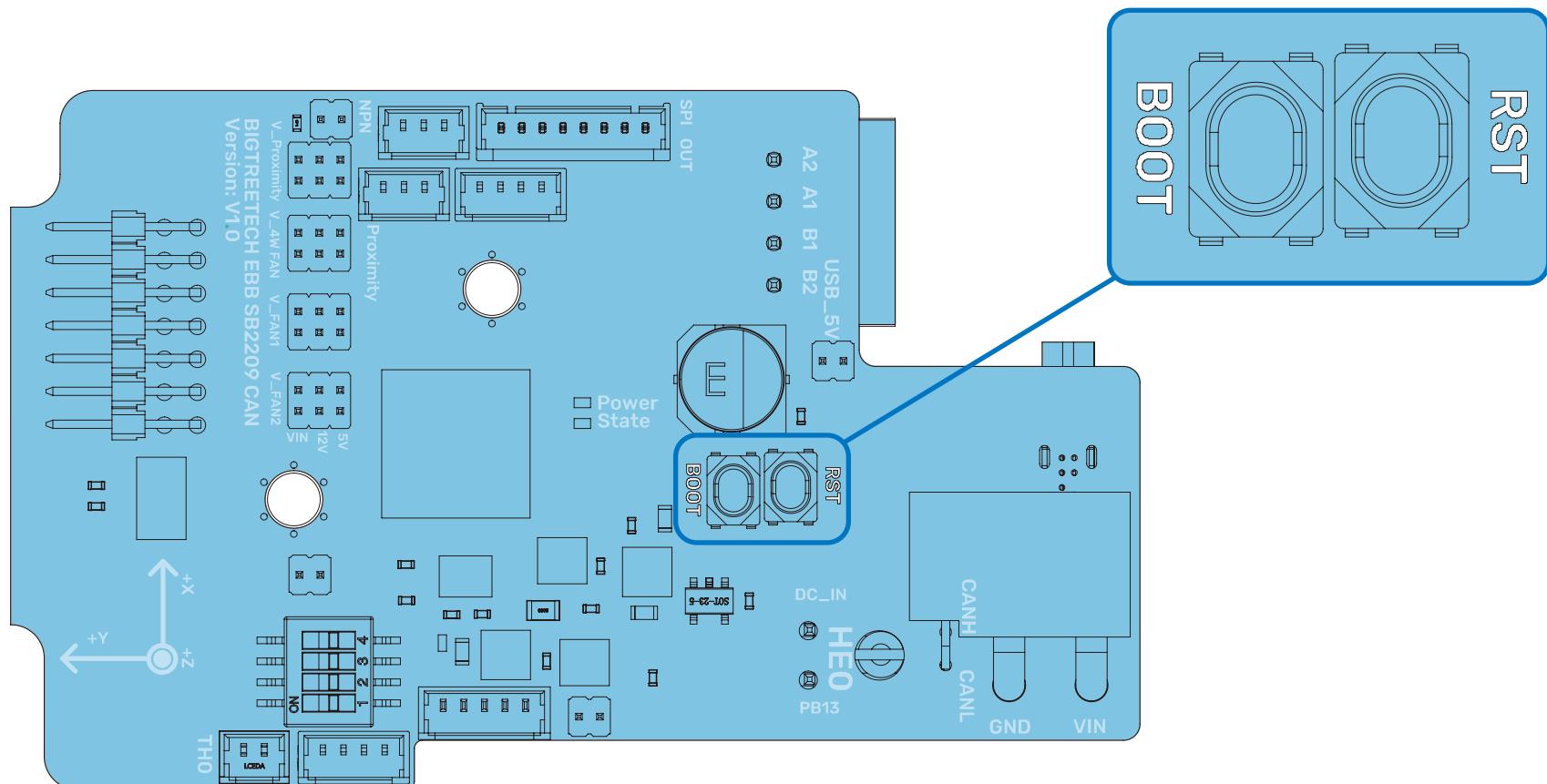
to compile firmware, 'canboot.bin' file will be generated in `home/biqu/CanBoot/out` folder when `make` is finished, download it onto your computer using the SSH application.



4.1 Please use a Type-C cable to connect the EBB SB2240/2209 CAN to the Raspberry Pi/CB1, and ensure that the USB\_5V jumper is connected, in order to supply power to the EBB SB2240/2209 CAN via Type-C.



4.2. Press and hold the Boot button, and then click the Reset button to enter the DFU mode.



#### 4.3. Enter in the SSH terminal command line

```
lsusb
```

Query DFU device ID

```
pi@fluiddp1:~ $ lsusb
Bus 001 Device 005: ID 0483:df11 STMicroelectronics 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@fluiddp1:~ $
```

#### 4.4. Please enter the following command to flash CanBoot:

```
sudo dfu-util -a 0 -d 0483:df11 -s 0x08000000:mass-erase:force -D ~/CanBoot/out/canboot.bin
```

where "0483:df11" should be replaced with the actual device ID obtained in step 4.3.

#### 4.5. After flashing is complete, remove the USB\_5V jumper and the Type-C cable.

1.) After SSH is successfully connected to Raspberry Pi, run

```
cd ~/klipper/
make menuconfig
```

Compile the firmware with the following configuration (if the options below are not available, please update your Klipper source code to the newest version).

[\*] Enable extra low-level configuration options

Micro-controller Architecture (STMicroelectronics STM32) --->

Processor model (STM32G0B1) --->

If you do not use CanBoot

Bootloader offset (No bootloader) --->

If CanBoot is used

Bootloader offset (8KiB bootloader) --->

Clock Reference (8 MHz crystal) --->

If USB communication on Type-C is used

Communication interface (USB (on PA11/PA12)) --->

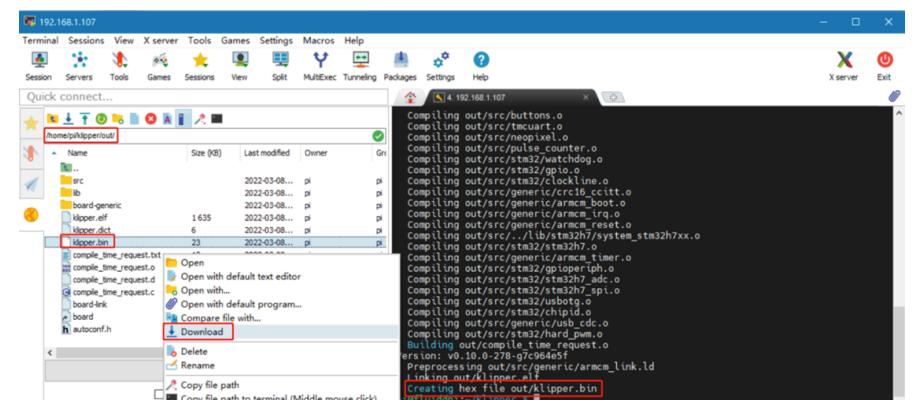
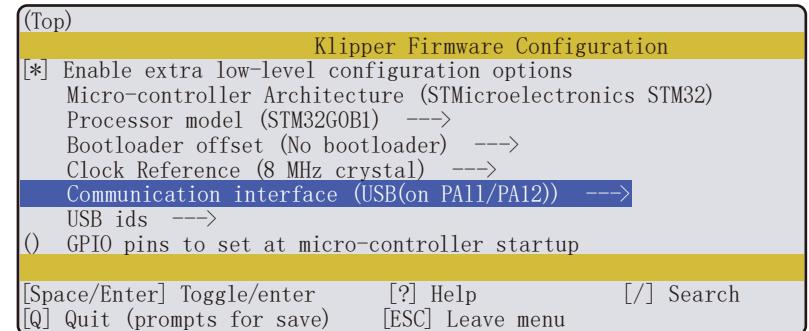
If CAN-Bus communication is used

Communication interface (CAN bus (on PB0/PB1)) --->

(1000000) CAN bus speed

2.) Press 'q' to exit, and "Yes" when asked to save the configuration.

3.) Run `make` to compile firmware, "klipper. bin" file will be generated in `home/pi/klipper/out` folder when `make` is finished, download it onto your computer using the SSH application.



1.To use the CAN bus, you need to connect the CAN bus cable and insert a jumper at the position of the 120R termination resistor.

2. Run:

```
cd ~/CanBoot/scripts
```

then run:

```
python3 flash_can.py -i can0 -q
```

to query the canbus ID (connect the CAN cable and power it on in advance), as shown in the figure below, the UUID of the device has been found

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

## 3. Run:

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

The be69315a613c is replaced with the actual UUID. Note: klipper.bin needs to be made in advance, and the application start offset of CanBoot is 8KiB offset, so Klipper's menuconfig Bootloader offset should also be 8KiB bootloader, as shown in the following figure.

```
biqu@BTT-CB1:~/CanBoot/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 canboot nodes...
Detected UUID: be69315a613c, Application: CanBoot
Attempting to connect to bootloader
CanBoot 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 = C3B1F96A8FCE706587BF4A9119D95D80465875A3
CAN Flash Success
biqu@BTT-CB1:~/CanBoot/scripts$
```

## 4. Run:

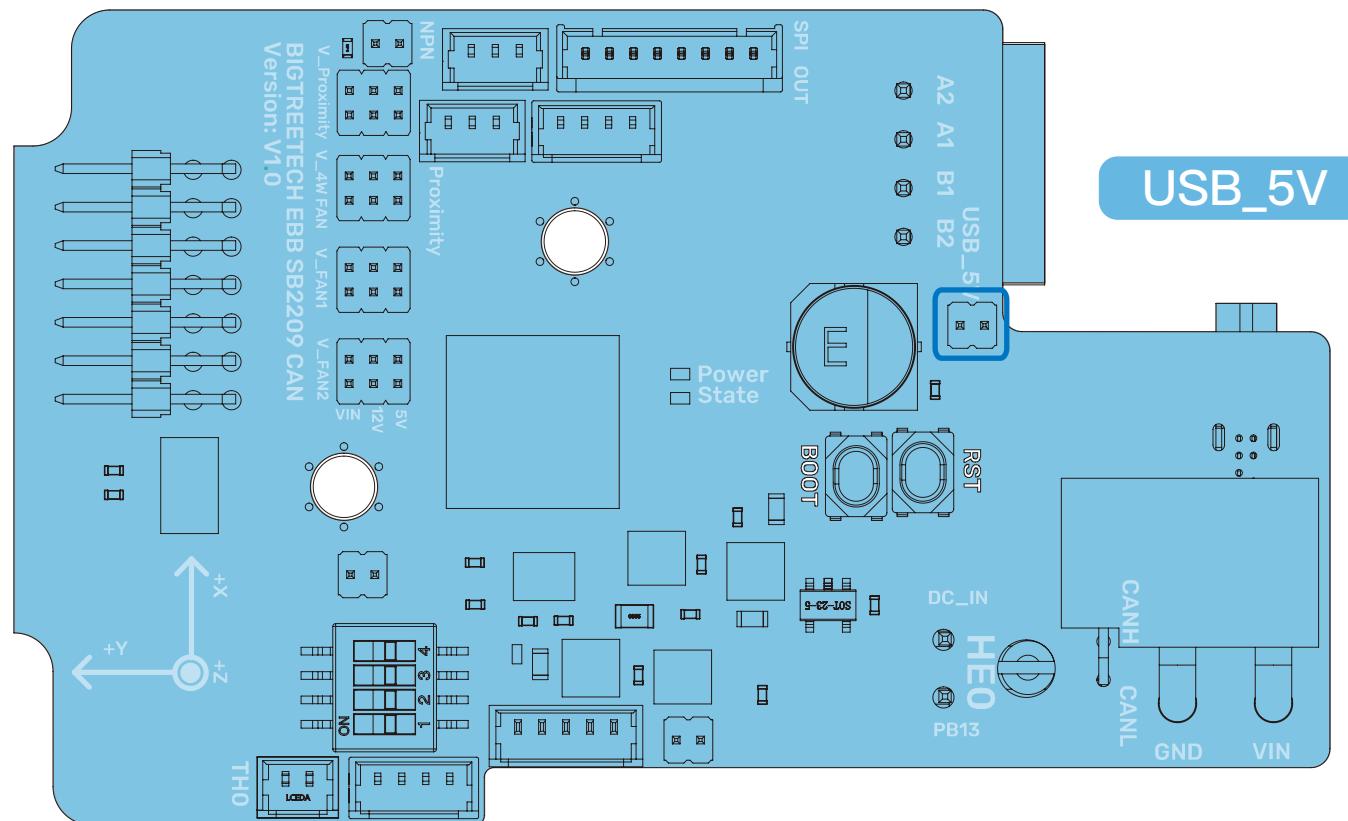
```
python3 flash_can.py -i can0 -q
```

to query. At this time, the Application changes from CanBoot to Klipper, indicating that Klipper has been running normally

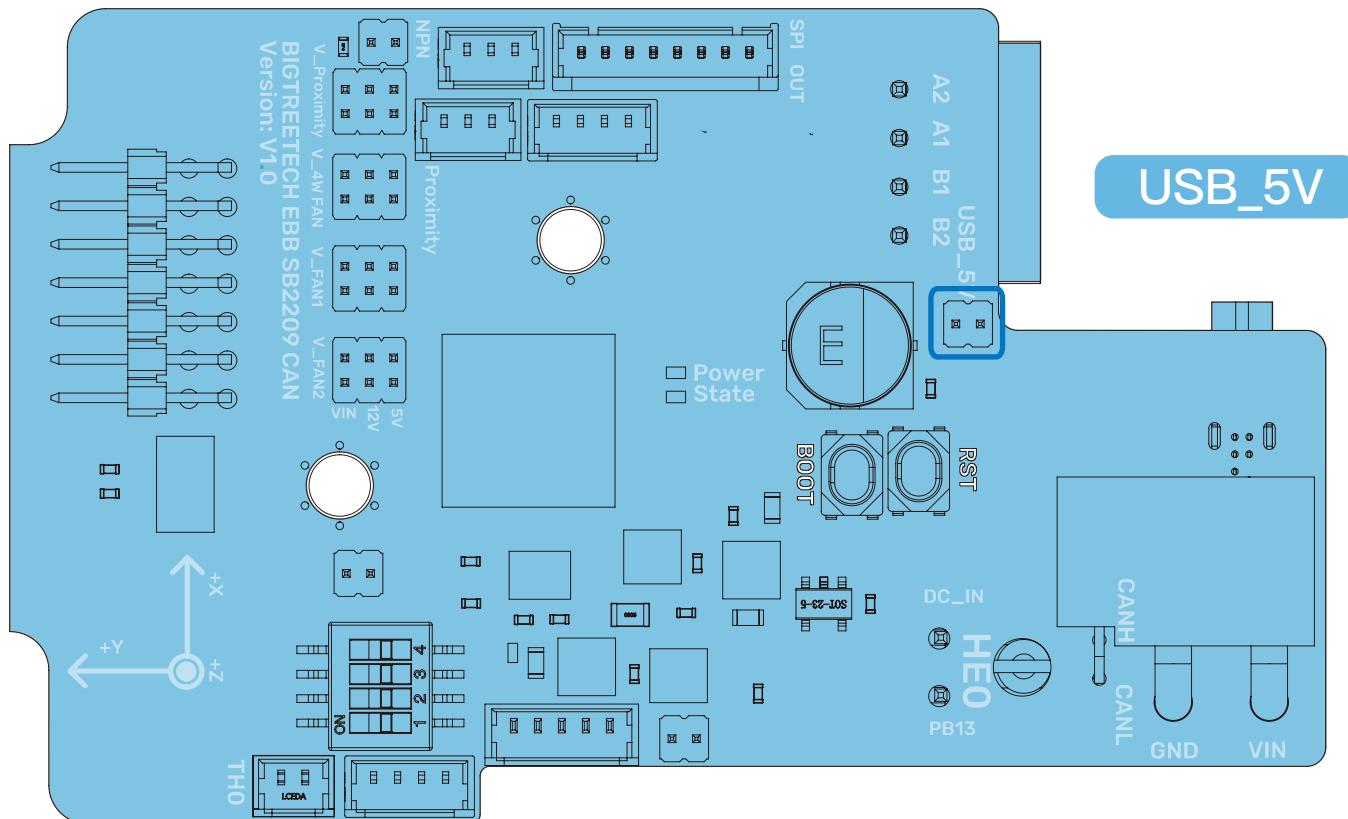
```
biqu@BTT-CB1:~/CanBoot/scripts$ python3 flash_can.py -i can0 -q
Resetting all bootloader node IDs...
Checking for canboot nodes...
Detected UUID: be69315a613c, Application: Klipper
Query Complete
biqu@BTT-CB1:~/CanBoot/scripts$
```

Raspberry Pi or CB1 update via DFU.

1. Please use a Type-C cable to connect the EBB SB2240/2209 CAN to the Raspberry Pi/CB1, and ensure that the USB\_5V jumper is connected, in order to supply power to the EBB SB2240/2209 CAN via Type-C.



2. Press and hold the Boot button, and then click the Reset button to enter the DFU mode.



3.Enter in the SSH terminal command line

lsusb

Query DFU device ID

```
pi@fluidppi:~ $ lsusb
Bus 001 Device 005: ID 0483:df11 STMicroelectronics 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@fluidppi:~ $
```

4.Run:

cd klipper

to enter to the klipper directory, then run the following command to write the firmware:

make flash FLASH\_DEVICE=0483:df11

Note: Replace 0483: df11 with the actual device ID found in the previous step.

5.The prompt **File downloaded successfully** indicates that the writing is completed.

After the writing is completed, there will be an error message: dfu-util: Error during download get\_status, just ignore it.

6. After the firmware is written, run

```
ls /dev/serial/by-id/
```

to query the serial ID of the device (this ID can only be found in the USB communication mode, and this step is ignored in CANBus mode).

7. If USB communication is used, after the first writing, it is not necessary to manually press the Boot and Reset buttons to enter the DFU mode when updating again.

You can directly enter

```
make flash FLASH_DEVICE=/dev/serial/by-id/usb-Klipper_stm32g0b1xx_0F0033000C504B4633373520-if00
```

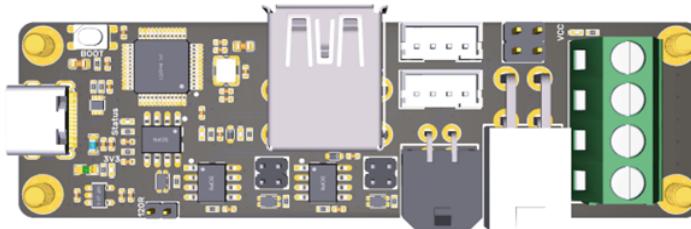
to write the firmware (Note: replace /dev/serial/by-id/xxx with the actual ID found in the previous step).

8. The prompt **File downloaded successfully** indicates that the writing is completed.

After the writing is completed, there will be an error message: dfu-util: Error during download get\_status, just ignore it.

After flashing is complete, remove the USB\_5V jumper and the Type-C cable.

Used with BIGTREETECH U2C module.



### 1. Enter the command

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

in the SSH terminal and execute

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

Set the CAN-BUS speed to 1M (it must be consistent with the speed set in the firmware (1000000) CAN bus speed), save (Ctrl+S) and exit (Ctrl+X) after modification, and enter

```
sudo reboot
```

to restart Raspberry Pi.

2. Each device on CAN bus will generate a canbus\_uuid according to the UID of MCU, to find each microcontroller device ID, make sure the hardware is powered on and wired correctly, and then run:

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

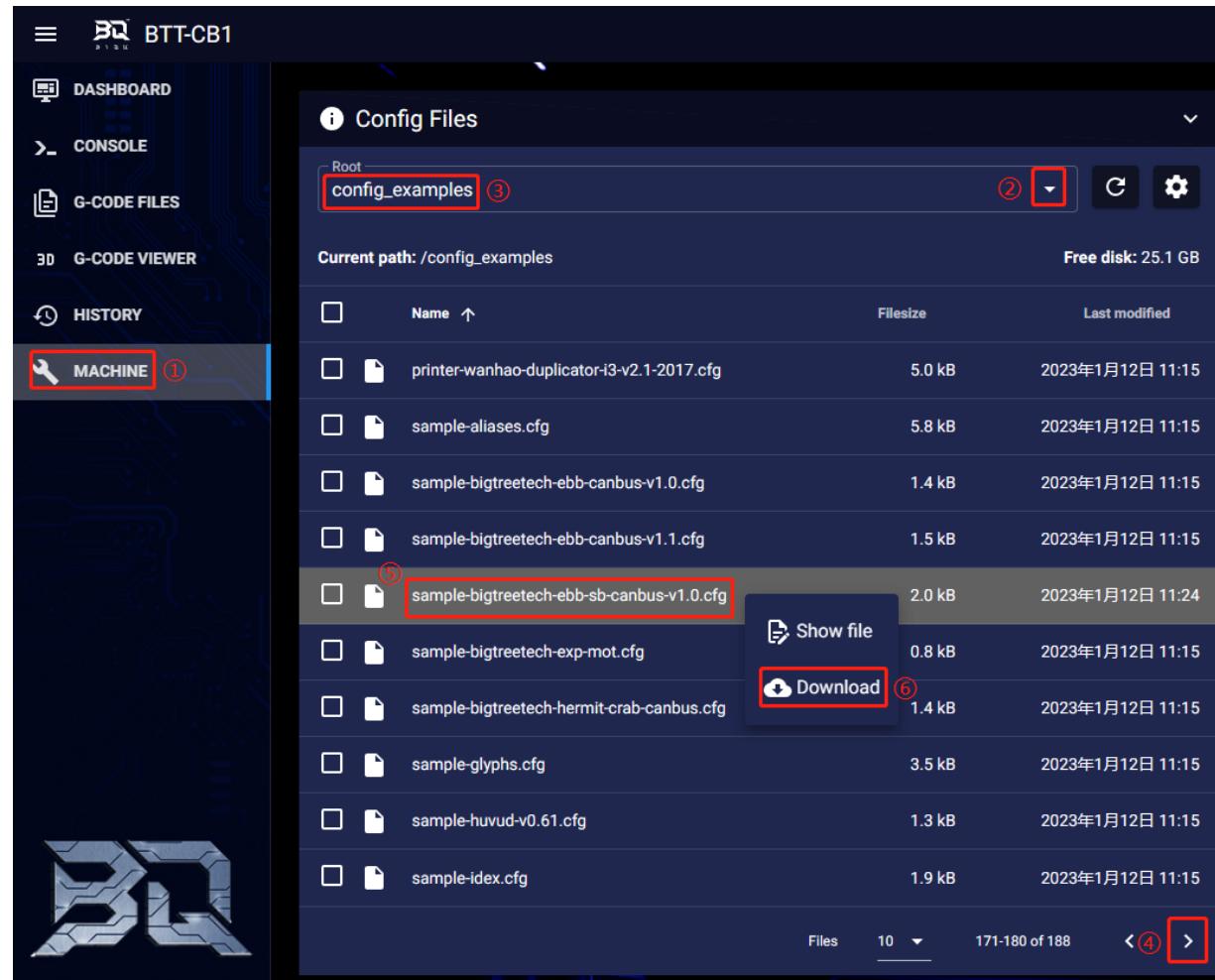
3. If an uninitialized CAN device is detected, the above command will report the device's canbus\_uuid.

```
Found canbus_uuid=0e0d81e4210c
```

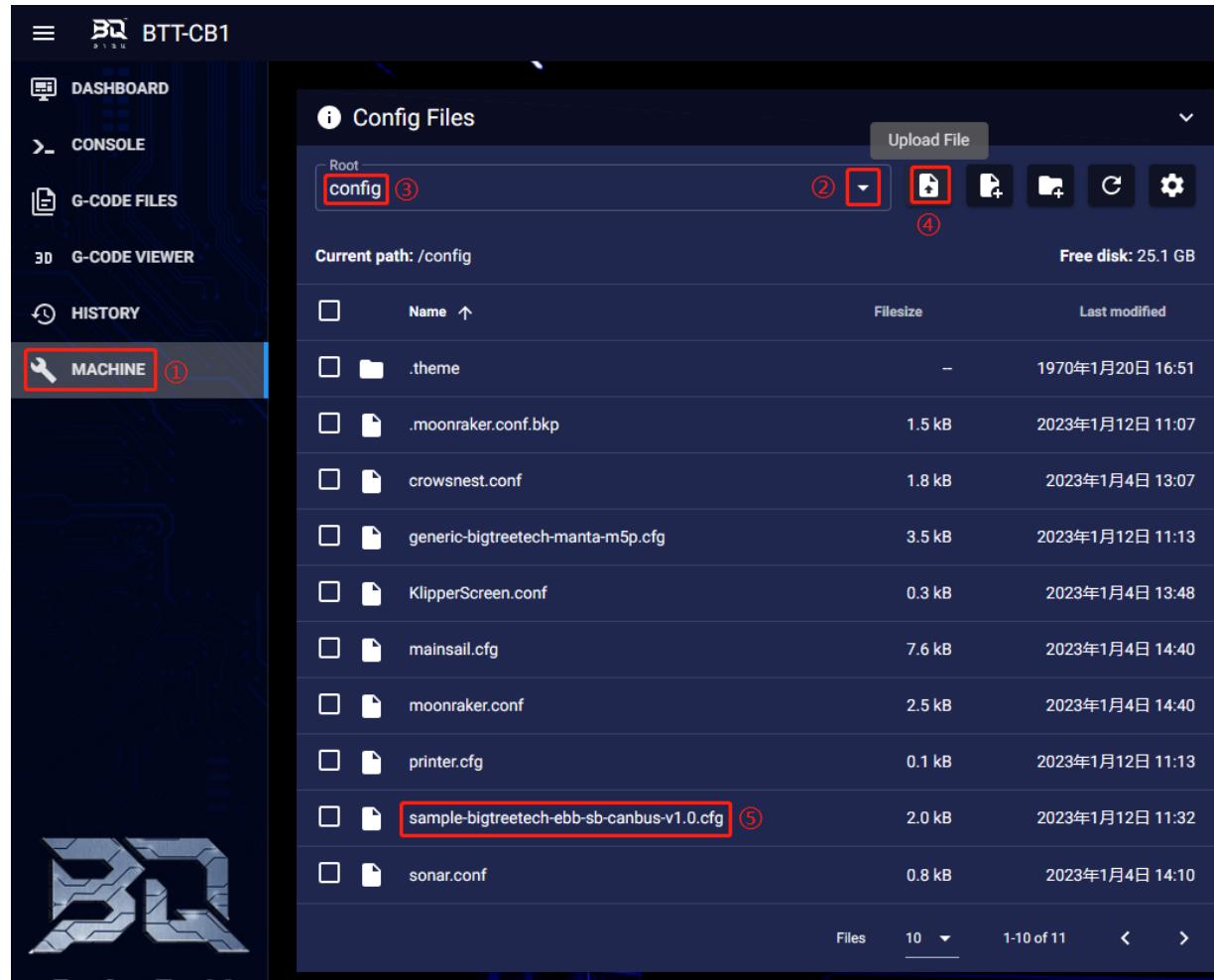
4. If Klipper has been running normally and connected to this device, then canbus\_uuid will not be reported, which is normal.

1. Enter Raspberry Pi IP address into your browser, and find the reference config for the motherboard in the directory shown below, if there is no such config available, update your Klipper source code to the newest version or download it from GitHub:

<https://github.com/bigtreetech/EBB>



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



3. Add the configuration of this motherboard in the "printer.cfg" file:

```
[include sample-bigtreetech-ebb-sb-canbus-v1.0.cfg]
```

```
X     printercfg
```

```
11
```

```
12     [include sample-bigtreetech-ebb-canbus-v1.0.cfg]
```

```
13
```

4. Enter the correct ID (USB serial or canbus).

```
X     sample-bigtreetech-ebb-sb-canbus-v1.0.cfg
```

```
8         [mcu EBBCan]
```

```
9         serial: (/dev/serial/by-id/usb-Klipper_firmware_12345-if00)
```

```
10        #canbus_uuid: 0e0d81e4210c
```

5. Configure the module's specific functions according to

```
https://www.klipper3d.org/Overview.html
```



---

**Website**

[www.bigtree-tech.com](http://www.bigtree-tech.com)

---

**GitHub**

[www.github.com/bigtreeTech](https://www.github.com/bigtreeTech)

---

**Discord**

[www.discord.gg/5jdwbYYZuv](https://www.discord.gg/5jdwbYYZuv)

---