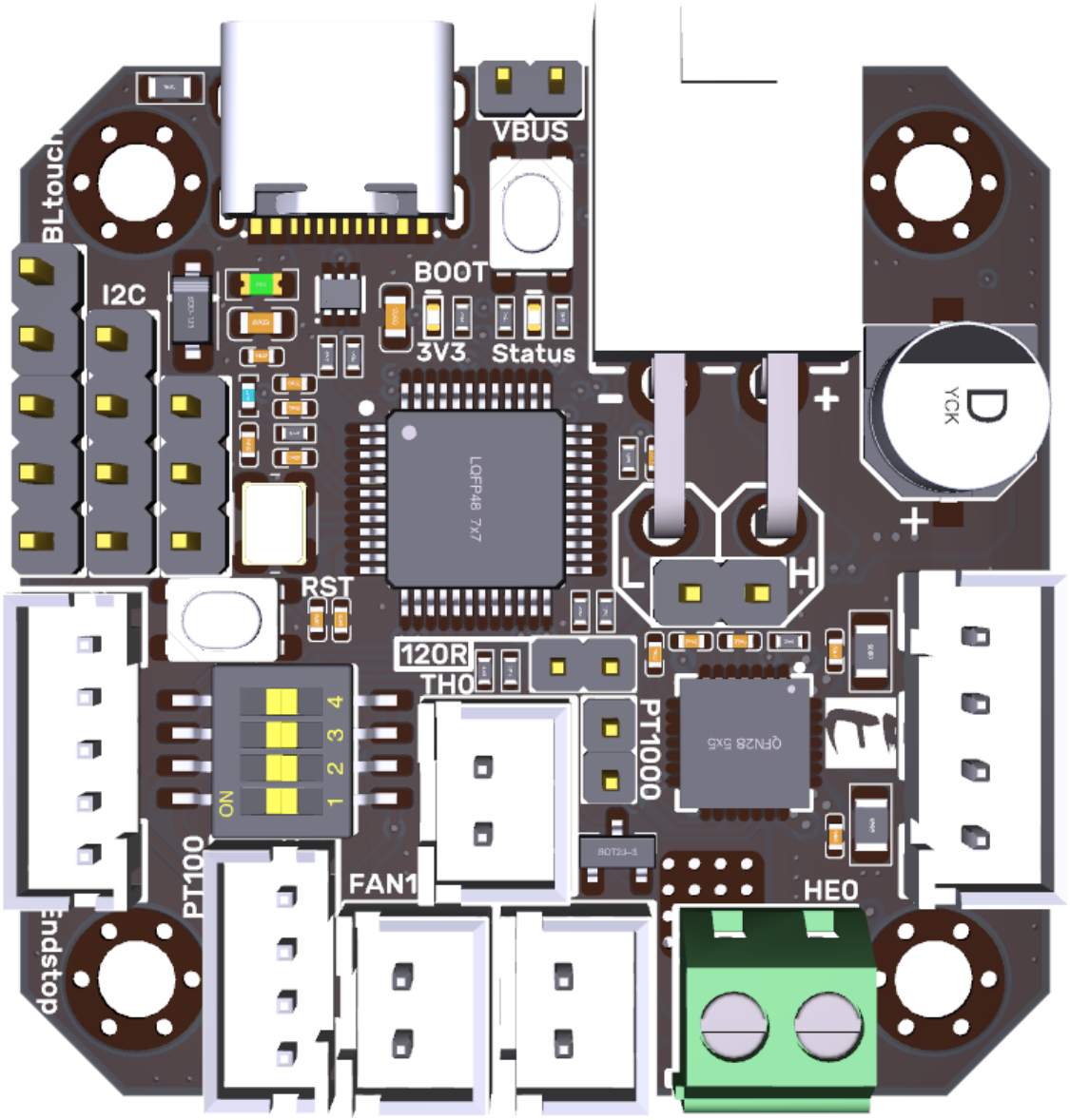
BIGTREETECH
EBB42 CAN V1.1
User Manual



Contents

Revision History

Revision	Description	Date
01.00	First Draft	2022/05/16
01.01	Add precautions for hotend when DFU updates firmware	2022/05/25
01.02	More detailed pin Instruction	2022/06/16

1 Brief Introduction

BIGTREETECH EBB42 CAN V1.1 is a nozzle adapter board made by the 3D printing team of Shenzhen Big Tree Technology Co., Ltd. for a 42 stepper motor extruder. It can communicate via a USB or CAN, which greatly simplifies wiring.

1.1 Main Features

1. With a BOOT and RESET buttons reserved, users can update the firmware via DFU mode by USB.
2. Adding a protection circuit on the thermistor avoids burning the main control chip caused by the electric leakage of the heater cartridge.
3. The thermistor can select the pull-up resistor value via a jumper, in this way, it can support PT1000 (2.2K pull-up resistor), which is convenient for DIY.
4. The USB power is selected through a jumper cap, which effectively isolates the motherboard DC-DC and USB 5V.
5. Reserve the I2C interface, this interface can also be used for filament runout/blocking detection, or DIY for other functions.
6. A follow current diodes are added to the inductive load interface (fan) to ensure that when the fan MOS is turned off, the fan winding current has a follow current loop, which effectively prevents the high voltage generated by the winding drain at the MOS tube when it is turned off. Considering the size of the board and the switch characteristics of the extruder fan, a Schottky diode in a SOD-323 package is used.
7. The DCDC step-down circuit is reversely connected to a diode to prevent the subsequent circuit from being damaged due to the reverse connection of the power line.
8. Onboard MAX31865 (optional function, the version without 31865 does not have this function, but there are reserved pads), supports 2-wire/4-wire PT100/PT1000 selection.
9. Support communication via CAN or USB. The terminal resistor 120R of CAN can be selected through a jumper cap, and it reserves a CAN expansion interface.
10. Equipping the ESD protection chip on the USB port prevents the main control from being broken down by the static electricity of the USB port.
11. Limit switch hardware debouncing circuit.

12. The adaptor board is equipped with terminals, female reeds, double-way studs and screws, which are required for DIY, greatly meeting the DIY needs of customers.
13. Support CAN bus connection, which has long data transmission, strong anti-noise ability, strong real-time performance and high reliability.

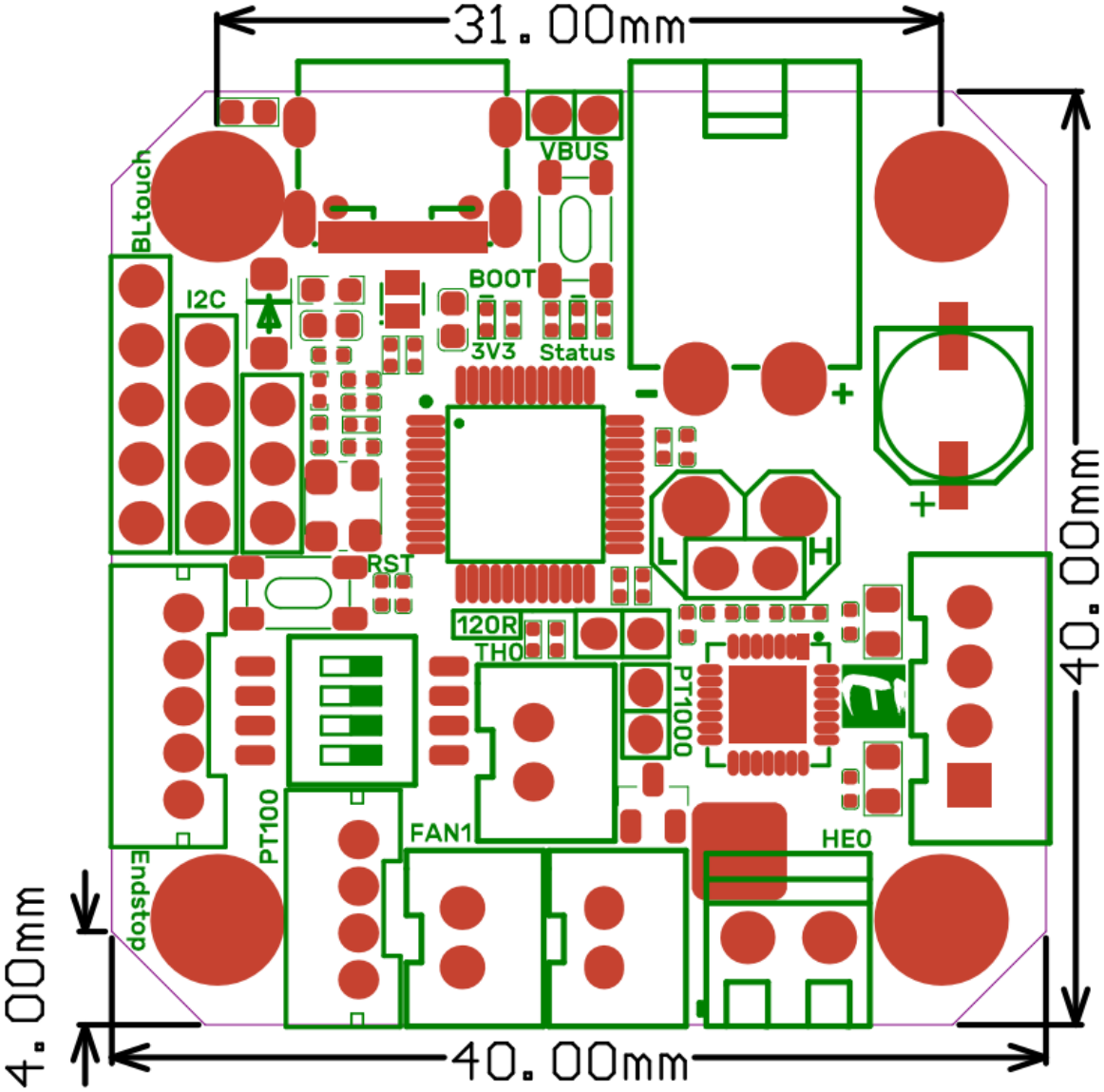
1.2 Product Parameters

1. Product Size: 40mm x 40mm, for further details please read: **BIGTREETECH EBB42 CAN V1.1-SIZE.pdf**
2. Installation Dimensions: Hole Spacing 31mm x 31mm, M3 Screw Holes x 4
3. Microprocessor: ARM Cortex-M0+ STM32G0B1CBT6 64MHz
4. Input Voltage: DC12V-DC24V 9A
5. Logic Voltage: DC 3.3V
6. Heating Interface: Heater Cartridge (E0), maximum output current: 5A
7. Onboard Sensor: ADXL345
8. Fan Interfaces: Two CNC Fans (FAN0, FAN1)
9. Maximum Output Current of Fan Interface: 1A, Peak Value: 1.5A
10. Expansion Interfaces: EndStop, I2C, Probe, RGB, PT100/PT1000, USB Interface, CAN Interface
11. Motor Drive: Onboard TMC2209, Hardware Address: 00, Rsense: 0.11R
12. Driver Working Mode: UART
13. Stepper Motor Interface: E
14. Temperature Sensor Interface Optional: One for 100K NTC or PT1000(TH0), one for PT100/PT1000
15. USB Communication Interface: USB-Type-C
16. DC-DC 5V Maximum Output Current: 1.5A

1.3 Firmware Support

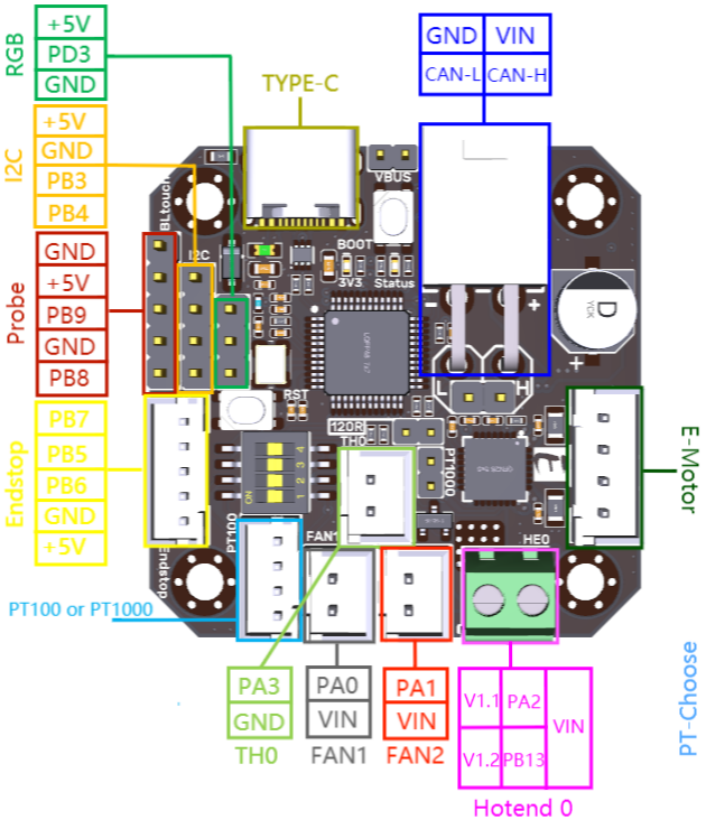
This product currently only supports Klipper.

1.4 Product Size



2 Peripheral Interface

2.1 Pin Instruction



ADXL345-SPI 2			
CS	SCLK	MISO	MOSI
PB12	PB10	PB2	PB11

MAX31865-SPI 1			
CS	SCLK	MISO	MOSI
PA4	PA5	PA6	PA7

TMC2209 Driver			
EN	STEP	DIR	UART
PD2	PD0	PD1	PA15

CAN		USB	
TX	RX	DP	DM
PB1	PB0	PA12	PA11

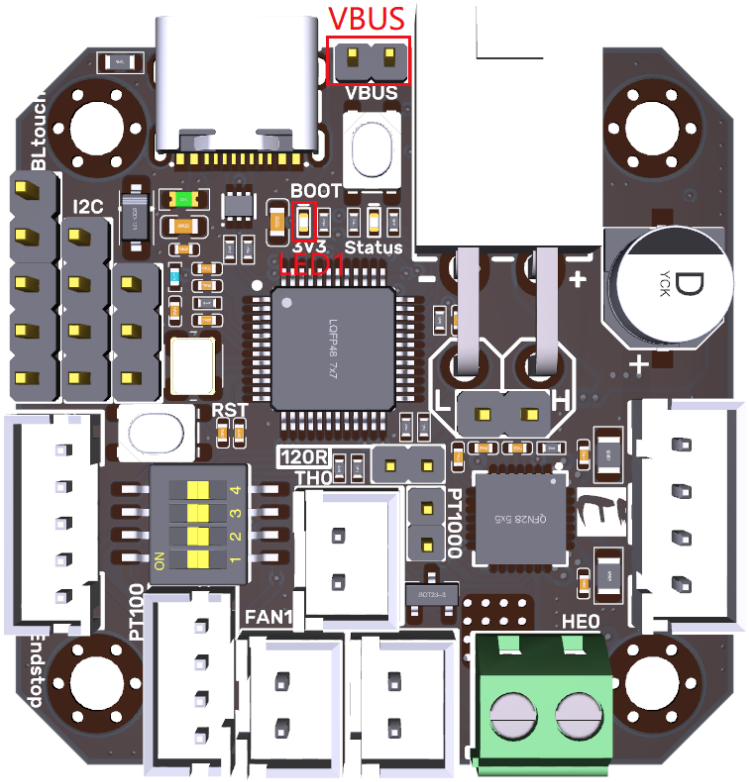
1	2	3	4	SensorModel
ON	ON	ON	OFF	2Lines PT100
ON	ON	OFF	ON	2Lines PT1000
OFF	OFF	ON	OFF	4Lines PT100
OFF	OFF	OFF	ON	4Lines PT1000

PT-Choose

3 Introduction of Interface

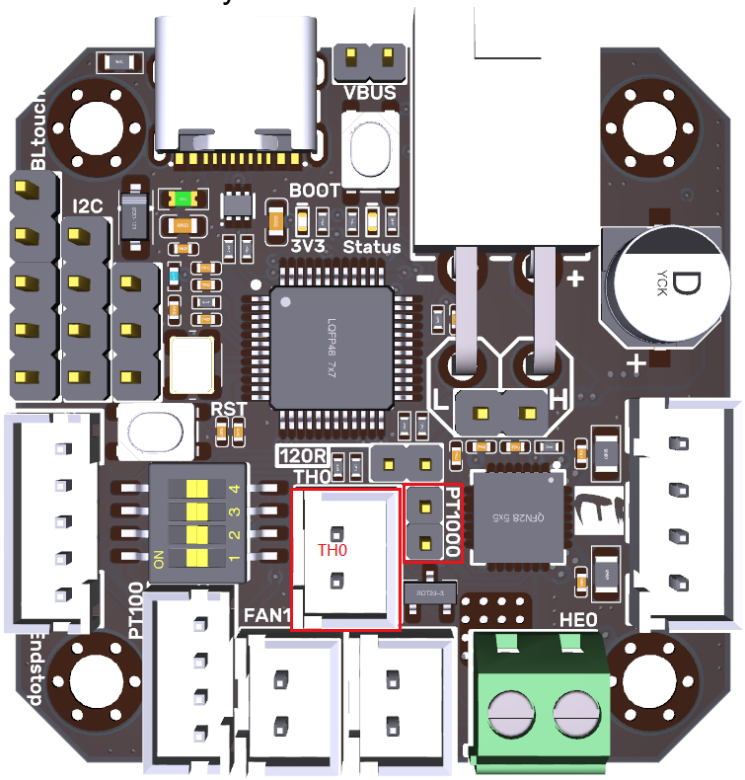
3.1 Power up via USB

After the motherboard is powered on, the yellow-green LED1 lights will light up, indicating a normal power supply. The VUSB in the middle of the board is the power selection part. Only when using USB to supply power to the motherboard or need to supply power through USB, do you need to use the jumper cap to connect VUSB.



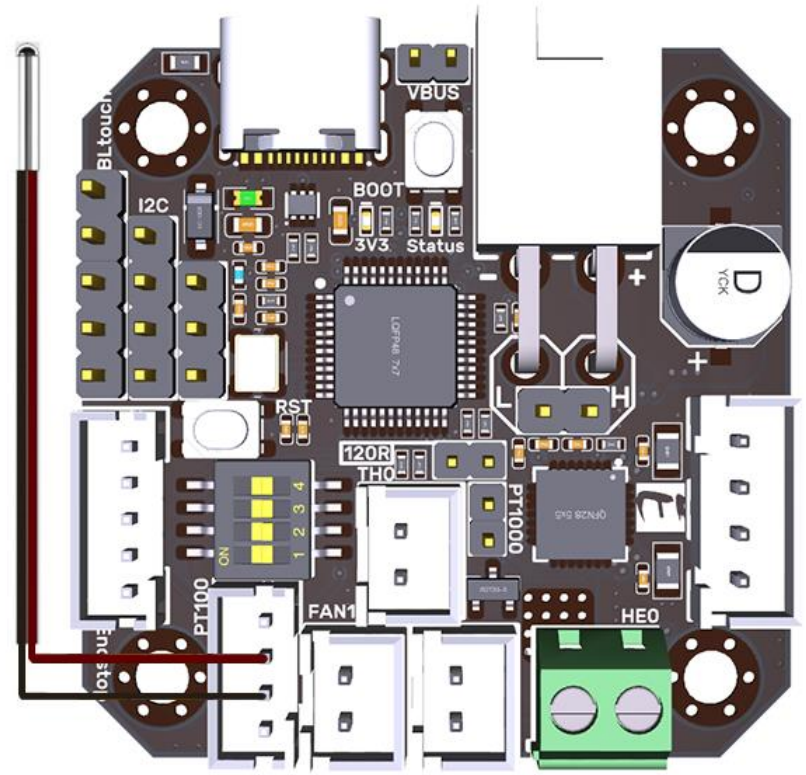
3.2 100K NTC or PT1000 Settings

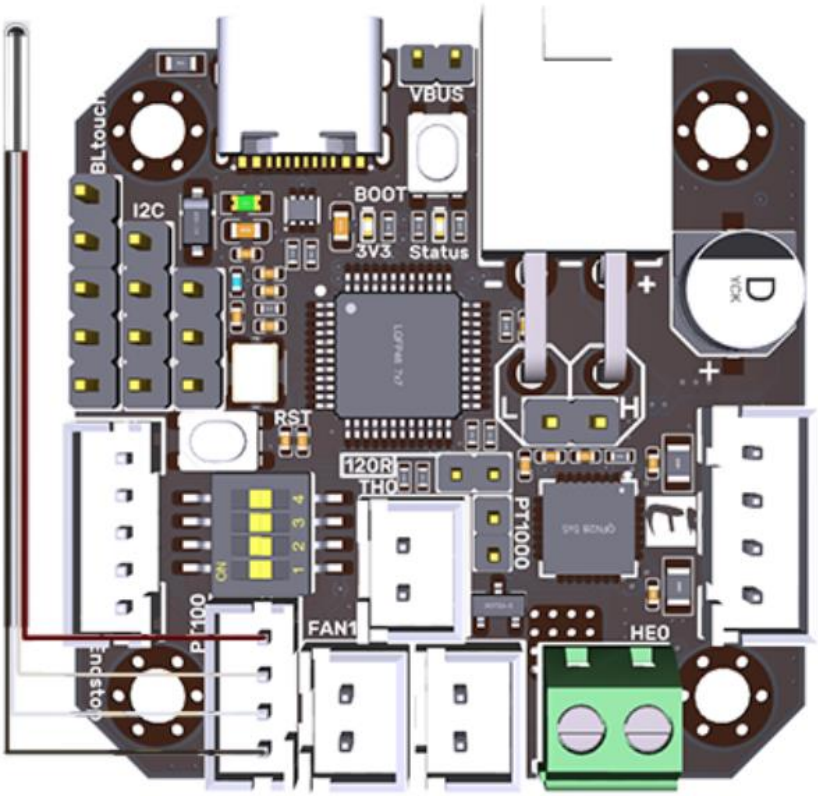
1. Version without 31865: No need to plug a jumper cap when using a 100K NTC thermistor, and TH0's pull-up resistor value is 4.7K. When using PT1000, you need to use the jumper cap to short the two pins, as shown below picture. At this time, TH0's pull-up resistor value is 2.2K (Notice: the temperature accuracy that's read out this way will be less accurate than that of MAX31865).



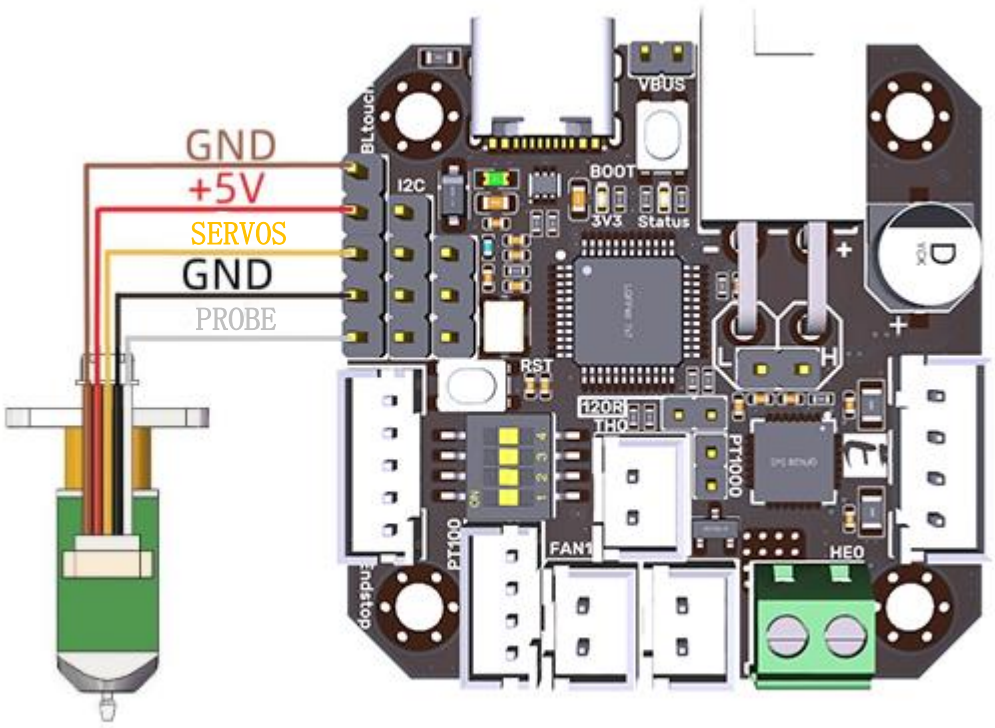
2. Version with 31865: Select PT100/PT1000 by DIP switch, two-line or four-wire:

1	2	3	4	Sensor Model
ON	ON	ON	OFF	Two lines PT100
ON	ON	OFF	ON	Two lines PT1000
OFF	OFF	ON	OFF	Four-wire PT100
OFF	OFF	OFF	ON	Four-wire PT1000

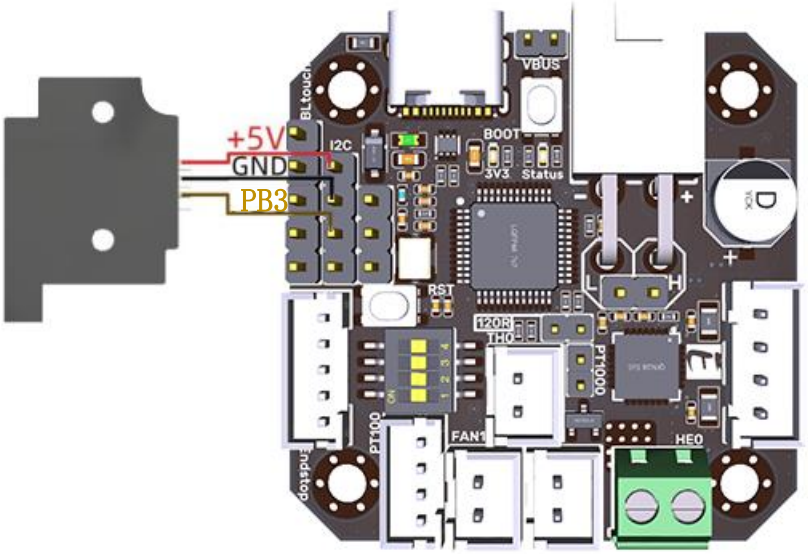

VBUS
BLtouch
I2C
BOOT
3V3
Status
RST
120R
TH0
PT1000
PT100
FAN1
HE0
ON
1 2 3 4


VBUS
BLtouch
I2C
BOOT
3V3
Status
RST
120R
TH0
PT1000
PT100
FAN1
HE0
ON
1 2 3 4

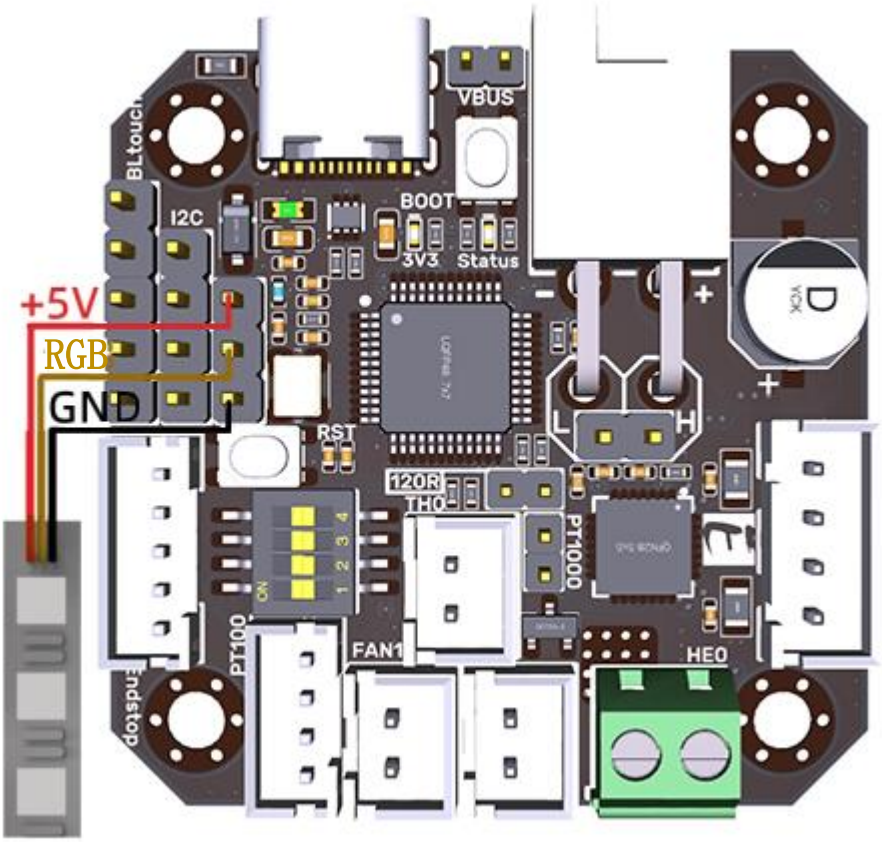
3.3 BL-Touch Wiring



3.4 Filament Runout Detection Wiring



3.5 RGB Wiring

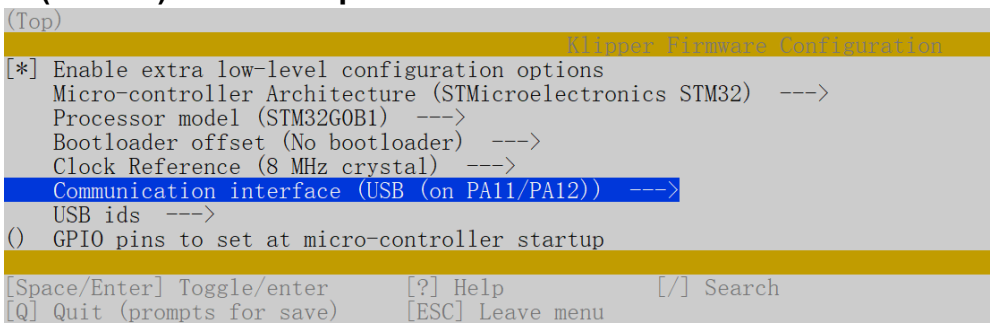


4 Klipper

4.1 Firmware Compilation

1. After connecting to the Raspberry Pi via SSH, type the below sentences at the command line:
   `cd ~/klipper/`
   `make menuconfig`
   Compile the firmware with the following configuration (if the following options are not available, please update the Klipper source to the latest version).
   [*] Enable extra low-level configuration options
   Micro-controller Architecture (STMicroelectronics STM32) --->
   Processor model (STM32G0B1) --->
   Bootloader offset (No bootloader) --->
   Clock Reference (8 MHz crystal) --->
   If using USB communication over Type-C

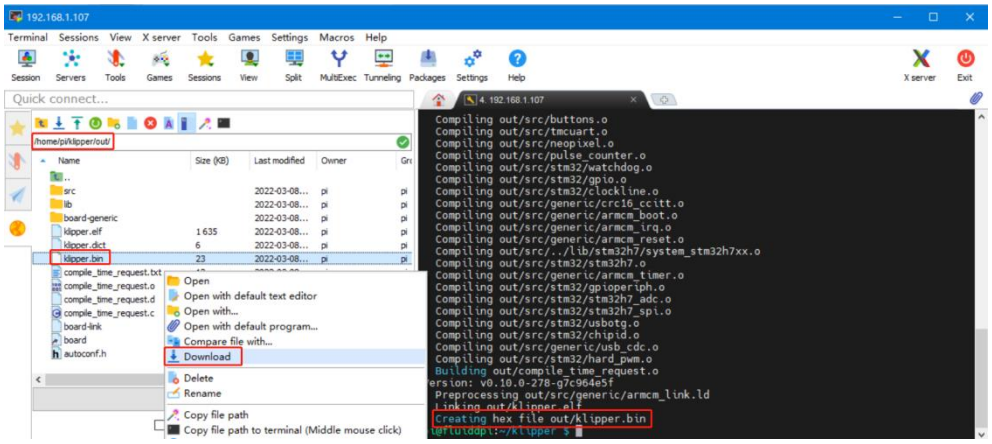
   Communication interface (USB (on PA11/PA12)) --->
   If using CANBus communication
   Communication interface (CAN bus (on PB0/PB1)) --->
   (250000) CAN bus speed



   Note: Only after https://github.com/Klipper3d/klipper/pull/5488 is merged into the main branch of Klipper, will the official firmware support the CAN bus function of STM32G0B1. If you use CANBus communication, you can use the firmware_canbus.bin firmware compiled by us on our GitHub, or use our source code to compile it yourself:
   https://github.com/bigtreetech/klipper/tree/stm32g0b1-canbus

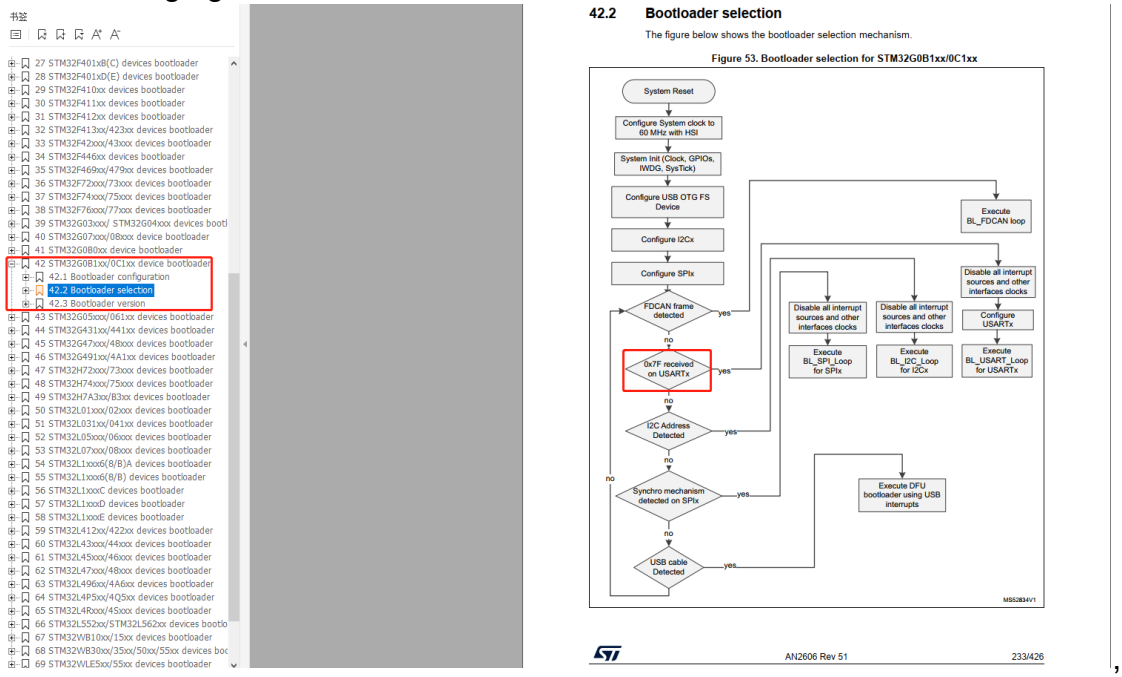
2. When the configuration is completed, type 'q' to exit the configuration interface. Select "Yes" when asked whether to save the configuration.

3. Enter `make` to compile the firmware. The 'klipper.bin' firmware that we need will be generated on the `home/pi/klipper/out` folder of the Raspberry Pi when completed. The firmware is on the left side of SSH software, users can download it directly to the computer.



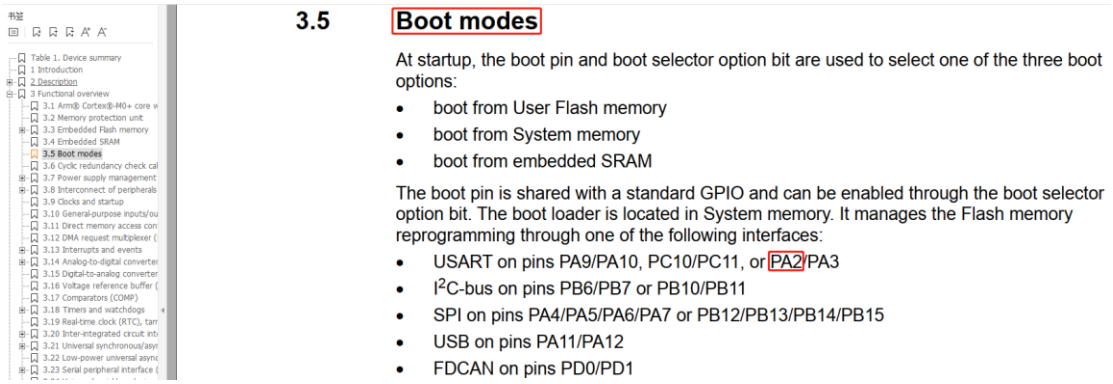
4.2 Firmware Update

Warning：STM32G0B1CB needs to jump to the **System memory** area to run bootloader (written by STMicroelectronics) when using DFU to update firmware through the Type-C port. Referring to the description in manual **AN2606** (https://www.st.com/content/ccc/resource/technical/document/application_note/b9/9b/16/3a/12/1e/40/0c/CD00167594.pdf/files/CD00167594.pdf/jcr:content/translations/en.CD00167594.pdf), The initialization process of this bootloader is shown in the following figure:



The IO of USART will be configured before going to the USB DFU mode.

After going to DFU mode, **PA2** will be configured to output high level by bootloader in **System memory** area refer to the datasheet of STM32G0B1CB (https://www.st.com/resource/en/datasheet/stm32g0b1cb.pdf)



3.5 **Boot modes**

At startup, the boot pin and boot selector option bit are used to select one of the three boot options:

- boot from User Flash memory
- boot from System memory
- boot from embedded SRAM

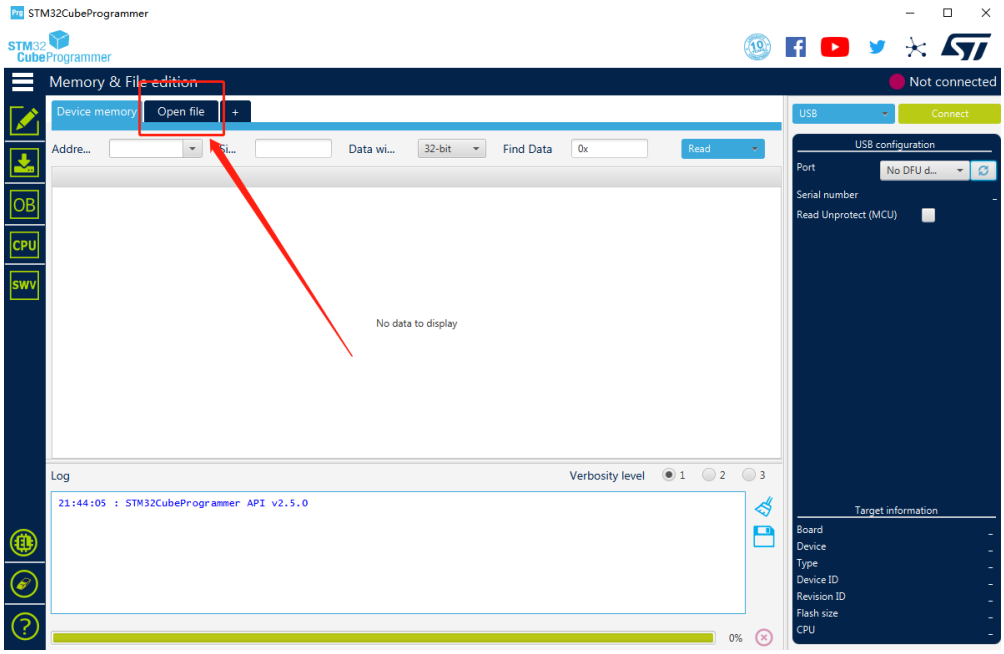
The boot pin is shared with a standard GPIO and can be enabled through the boot selector option bit. The boot loader is located in System memory. It manages the Flash memory reprogramming through one of the following interfaces:

- USART on pins PA9/PA10, PC10/PC11, or PA2/PA3
- I^2C-bus on pins PB6/PB7 or PB10/PB11
- SPI on pins PA4/PA5/PA6/PA7 or PB12/PB13/PB14/PB15
- USB on pins PA11/PA12
- FDCAN on pins PD0/PD1

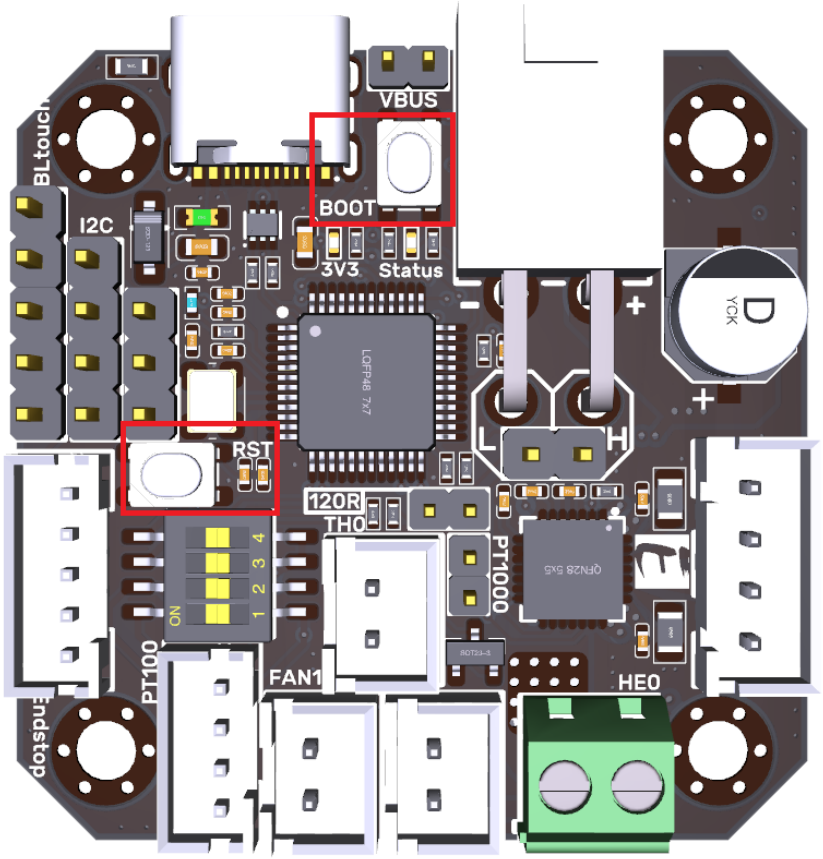
PA2 is used for the hotend MOSFET in **EBB36 CAN V1.1** and **EBB42 CAN V1.1**, The high level in the DFU mode change the hotend into heating state. Therefore, please pay attention to disconnect the main power VIN of the hotend when using the DFU of Type-C port to update the firmware, or ensure that the firmware update is completed soon and goto the normal working mode. Never keep MCU in DFU mode for a long time when the main power supply and hotend are connected.

Upgrade with STM32CubeProgrammer

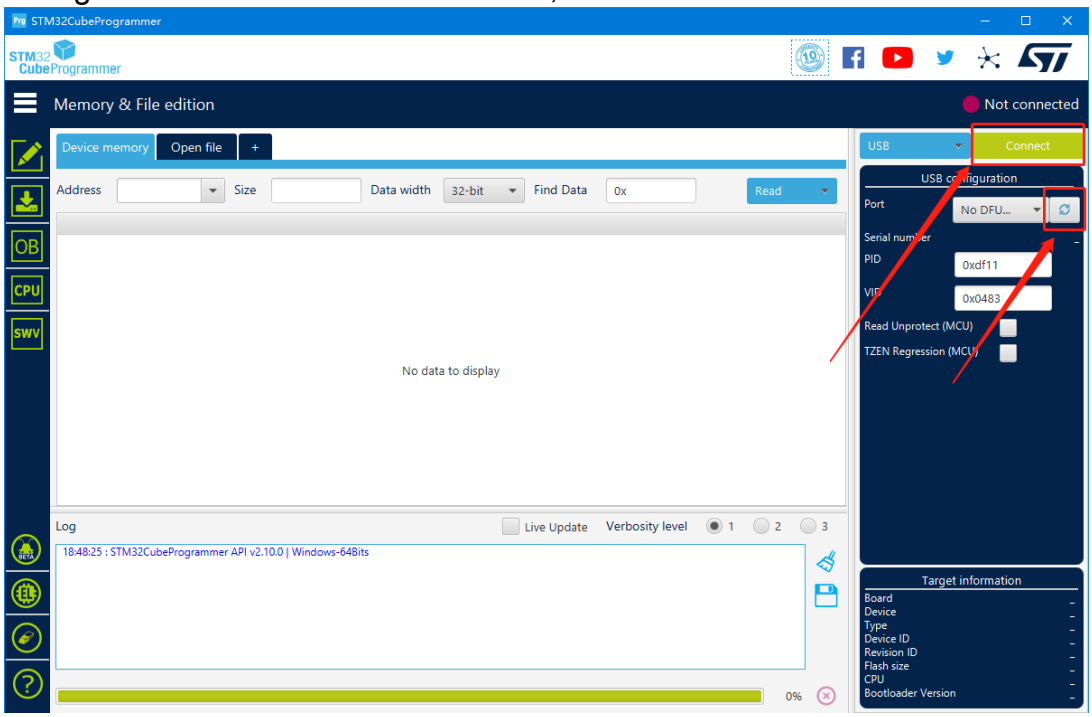
1. Open the installed STM32CubeProgrammer and select the firmware to download (klipper.bin).



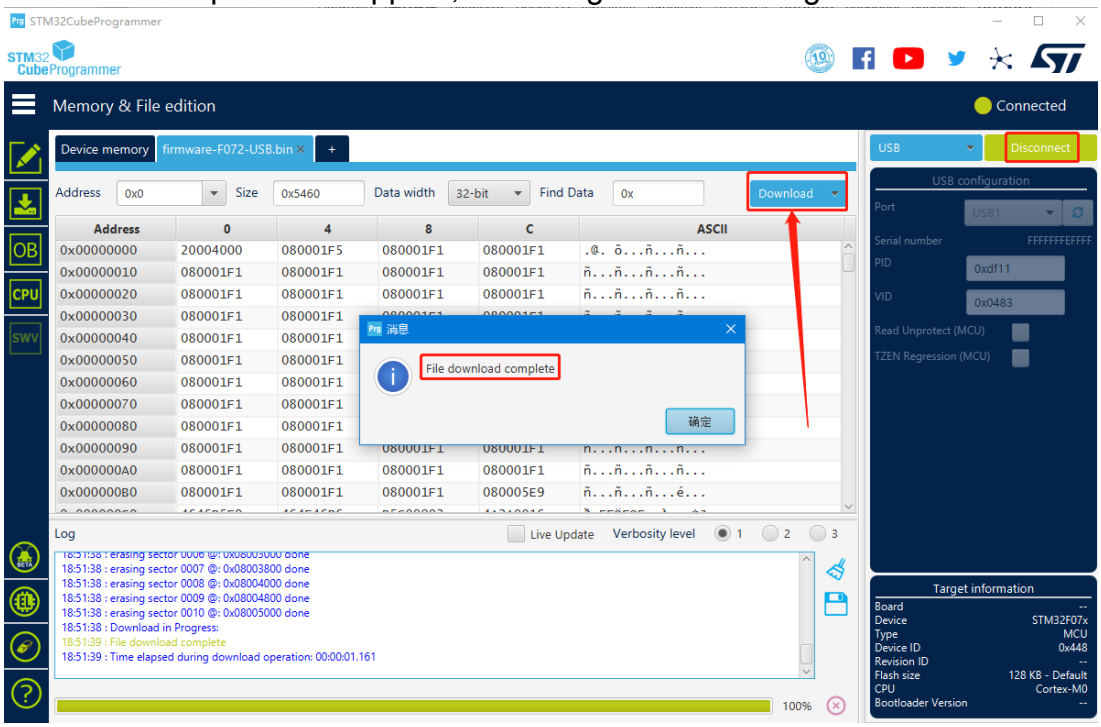
2. Press and hold the Boot button, then click the RST button to enter DFU mode.



3. Click the "Refresh" button in the STM32CubeProgrammer until the Port changes from "No DFU d..." to "USB1", then click "Connect" to connect the chip.

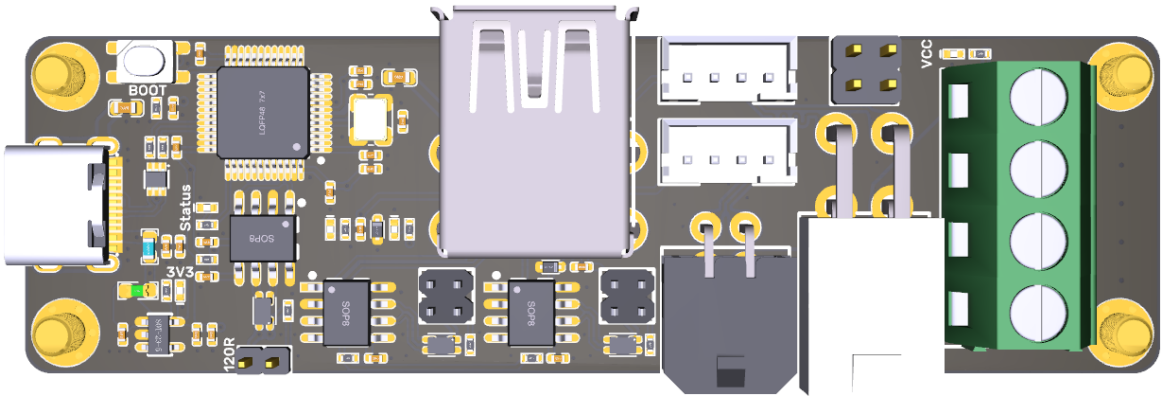


4. After the connection is successful, "Connect" will change into "Disconnect", and then click "Download", after the download is completed, a pop-up window of "File download complete" will appear, indicating that the writing is successful.



4.3 CANBus Configuration

4.3.1 Work with BIGTREETECH U2C



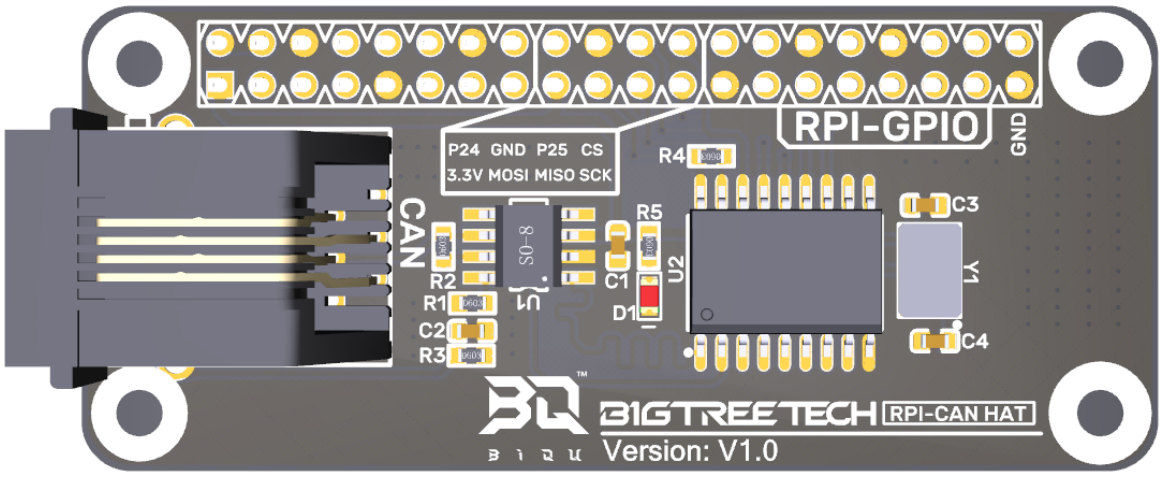
1. Type the following command `sudo nano /etc/network/interfaces.d/can0` in the SSH terminal and execute:

```
auto can0
iface can0 can static
    bitrate 250000
    up ifconfig $IFACE txqueuelen 1024
```

Set the speed for CANBus at 250K(must be the same as the speed set in the firmware **(250000) CAN bus speed**), Save(Ctrl + S)after modification and exit(Ctrl + X), Type sudo rebootto to reboot Raspberry Pi.

2. Every device on CANBus will generate a canbus_uuid based on MCU's UID. If users want to find the ID for every microcontroller, please make sure the hardware is powered on and wired correctly, then run the following command:
   ~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0
3. If detected an uninitialized CAN device, the above command will report the device's canbus_uuid:
   Found canbus_uuid=0e0d81e4210c
4. If Klipper operates normally and is connected to the device, then it won't report canbus_uuid, which is also normal.

4.3.2 Work with BIGTREETECH RPI-CAN HAT



1. Type and run the following command sudo nano /boot/config.txt, and added below contents on file config.txt.
   dtparam=spi=on
   dtoverlay=mcp2515-can0,oscillator=12000000,interrupt=25,spimaxfrequency=1000000
   After modification, save(Ctrl + S)and exist(Ctrl + X), type sudo reboot to reboot Raspberry Pi.
2. Type and run commands dmesg | grep -i '\(can\|spi\)' to test if RPI-CAN HAT module is normally connected. The normal response should be as below:

[8.680446] CAN device driver interface
[8.697558] mcp251x spi0.0 can0: MCP2515 successfully initialized.
[9.482332] IPv6: ADDRCONF(NETDEV_CHANGE): can0: link becomes ready

```
pi@fluiddpi:~ $ dmesg | grep -i '\(can\|spi\)'
[    8.426216] CAN device driver interface
[    8.470380] mcp251x spi0.0 can0: MCP2515 successfully initialized.
[    9.330545] IPv6: ADDRCONF(NETDEV_CHANGE): can0: link becomes ready
[   25.441341] can: controller area network core
[   25.467933] can: raw protocol
```

3. Type `sudo nano /etc/network/interfaces.d/can0` on the SSH terminal and run the command.
   ```
   auto can0
   iface can0 can static
      bitrate 250000
      up ifconfig $IFACE txqueuelen 1024
   ```
   Set the speed for CANBus at 250K(must be the same as the speed set in the firmware **(250000) CAN bus speed**). Save (Ctrl + S) after modification and exit (Ctrl + X). Type command `sudo reboot` to reboot Raspberry Pi.

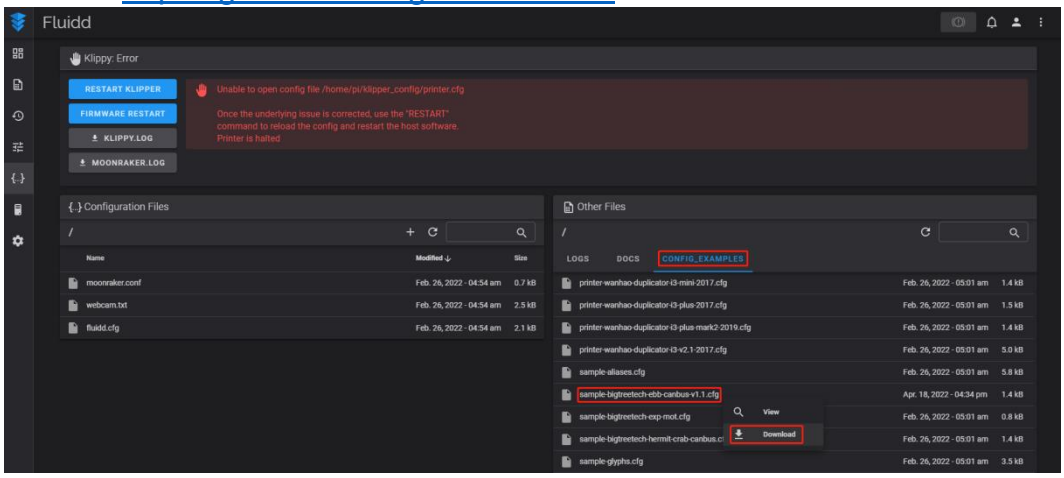
4. Every device on CANBus will generate a canbus_uuid base on MCU's UID. If users want to find the ID for every microcontroller, please make sure the hardware is powered on and wired correctly, then run the following command:
   `~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0`

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

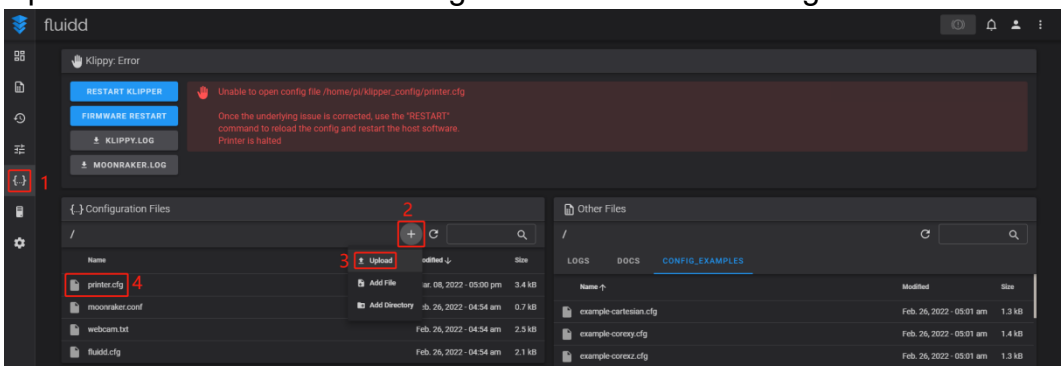
6. If Klipper operates well and is connected to the device, then it won't report the canbus_uuid, which is normal.

4.4 Klipper Configuration

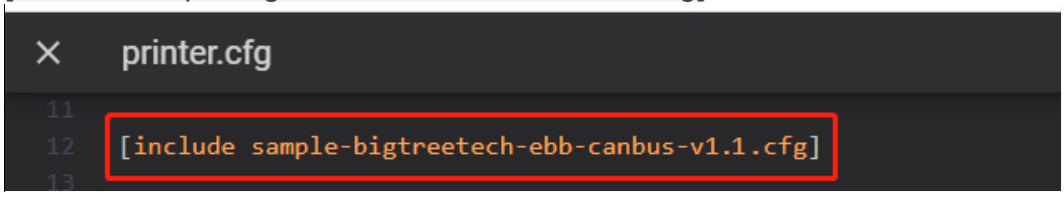
1. Enter the IP access of Raspberry Pi in the browser of the computer, and download the reference configuration of the motherboard from the file path shown in the figure below. If you cannot find this file, please update the Klipper firmware source code to the latest version, or download it from GitHub: https://github.com/bigtreetech/EBB



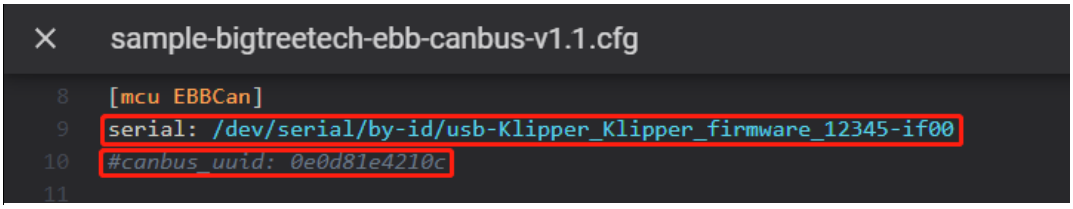
2. Upload the motherboard's configuration files to the Configuration Files.



3. Add the motherboard's configuration to the file "printer.cfg".
[include sample-bigtreetech-ebb-canbus-v1.1.cfg]



4. Revise the ID number of the configuration files as the actual ID of the motherboard (USB serial or CANBus).



5. Configure the specific functions of the module as instructed below:

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

5 Cautions

1. When the TH0 interface doesn't work with PT1000, you can't plug a jump cap on it, otherwise 100K NTC won't work.
2. When using CAN communication, you need to see whether it is used as a terminal. If it is, you must plug a jumper cap on the 120R position.
3. When DIY crimping, wire according to the silkscreen, and DIY according to the Pin and Schematic diagrams so as to avoid the power line from being reversely connected or connected to the CAN signal, which will get the module burned.
4. If there's no external power supply during programming via USB port, you need to short the VUSB by jumper cap so as to supply the module with working voltage.
5. The load current of the heater cartridge and the fan interfaces shall not exceed the maximum withstand current to prevent the MOS tube from being burned out.
6. Please pay attention to the precautions in 4.2 Firmware Update to avoid keeping MCU in DFU mode for a long time when the main power supply and hotend are connected.

6 FAQ

Q: What's the maximum current for the heater cartridge and fan interface?

A: The maximum output current of the heater cartridge interface: 5A.

The maximum output current of the fan interface: 1A.

The total current for the heater cartridge, driver and fan needs to be less than 9A.

Q: Cannot update firmware via USB port?

A: You need to make sure that the jump cap is plugged on VUSB, and the indicator light on the board is on.