**How to Use CAN Toolhead Boards Connected Directly to Octopus / Octopus Pro on CanBoot**

- Klipper has a new CAN bus feature, the USB to CAN bus bridge communication for Octopus boards (and other compatible MCUs).
- This feature allows the Octopus boards (and other compatible MCUs) to act as a USB to CAN bus adapter. This replaces the need for an adapter like a Raspberry Pi CAN HAT, canable adapter, or the Bigtreetech U2C board.
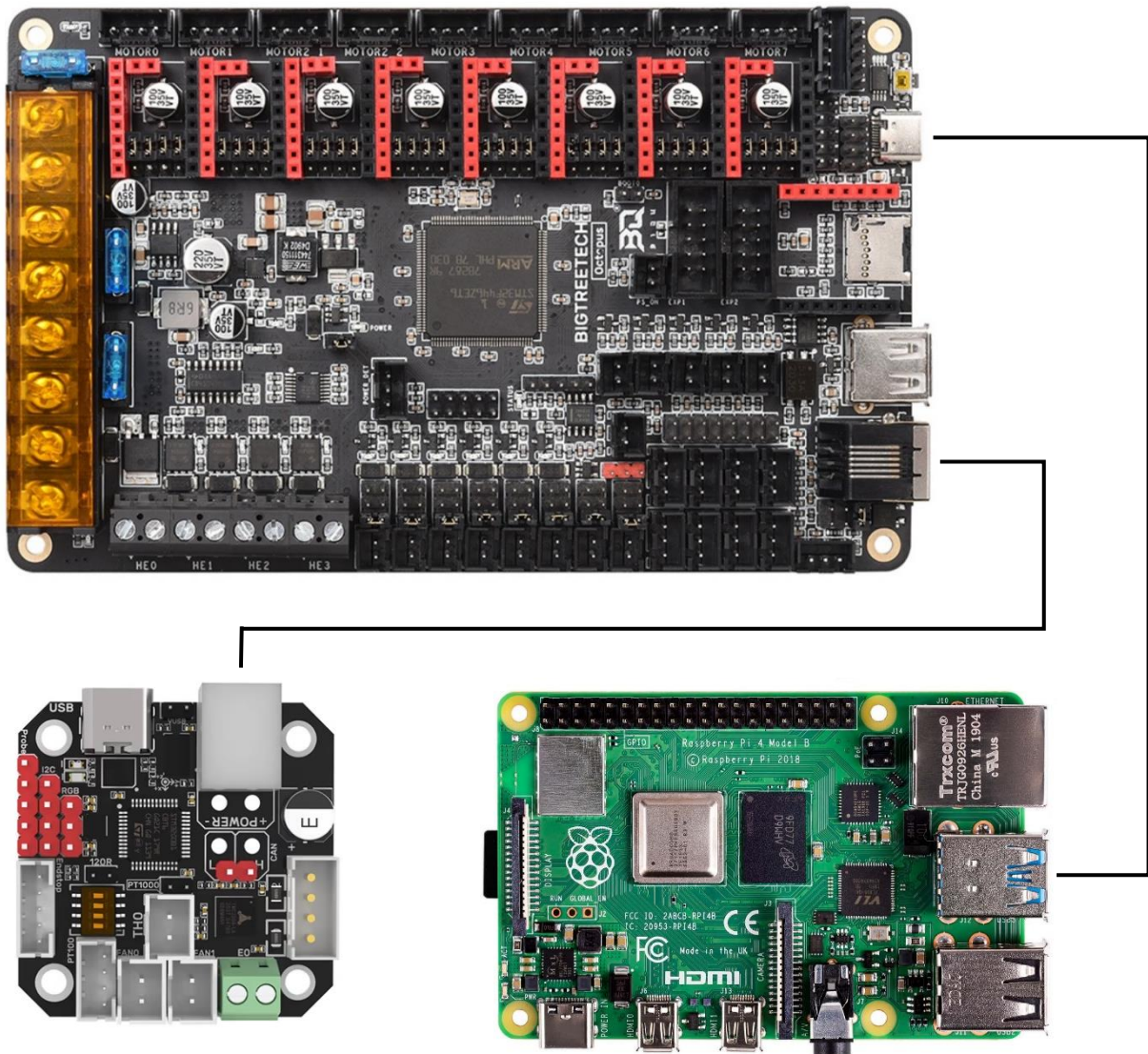
What is CanBoot?

- CanBoot is a custom bootloader loaded onto your Octopus and EBB board that allows users to update Klipper firmware over USB, UART, or CAN comms without physically having to access the board reset buttons or BOOT jumpers.
- It uses the same 'make menuconfig' setup to configure and compile firmware.
- **CanBoot is NOT required to use the Klipper USB to CAN bus bridge comms**. But I have not tested this comms feature separately.

You Will Need the Following

- USB-A to USB-C cable
- Power supply for Octopus and EBB toolboard (i.e. printer)
- RJ11 or RJ12 telephone cable, crimped to connect EBB board to Octopus
    - See wiring info at end of guide. Best to setup wiring before installing software.
- Raspberry Pi, or similar, with Klipper / Moonraker / UI installed and working
- Computer with following software:
    - SSH terminal software, I use PuTTY or Windows CMD
    - WinSCP, to transfer files between Raspberry Pi and computer
        - I also use Windows CMD to do this
    - STM32CubeProgrammer
        - Download for your computer OS and install
        - https://www.st.com/en/development-tools/stm32cubeprog.html#get-software

High Level Instructions, for Understanding

1. Install CanBoot on Raspberry Pi, flash firmware to boards
    a. Configure and compile CanBoot firmware for Octopus / Pro and EBB
    b. Flash CanBoot to boards using STM32CubeProgrammer and computer
2. Setup boards for Klipper
    a. Octopus - Setup Klipper for USB to CAN bus bridge, with CAN comms to EBB
    b. EBB - Setup Klipper for CAN comms
    c. Find serial device for Octopus / Pro on Raspberry Pi
    d. Flash Klipper to Octopus / Pro with CanBoot serial command
3. Setup can0 network on Raspberry Pi, power cycle printer
4. Find CAN uuid for Boards
    a. Connect EBB to Octopus / Pro, power cycle printer
5. Flash Klipper to EBB board with CanBoot CAN command
6. Printer Config Update and General Tips
7. How to Use CanBoot to update boards, Tips

Okay, enough talking! Let's get started!

Instructions _____

1. Install CanBoot on Raspberry Pi, flash firmware to boards

Start your SSH terminal.

⌨ sudo apt-get install git -y #Do this if you haven't installed git yet…

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

⌨ cd CanBoot
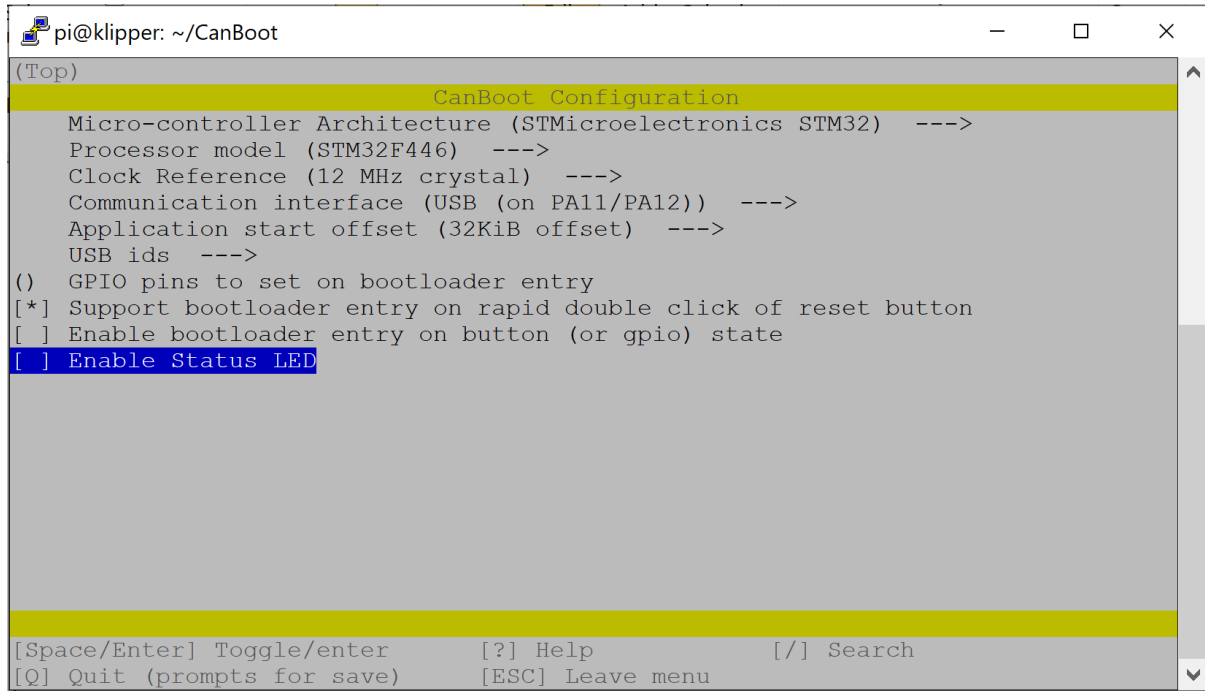
⌨ make menuconfig

1a. CanBoot Firmware configuration

- I listed settings for the Octopus, Octopus Pro, EBB36, and EBB42…and all their variants.

Octopus / Pro with F446 processor
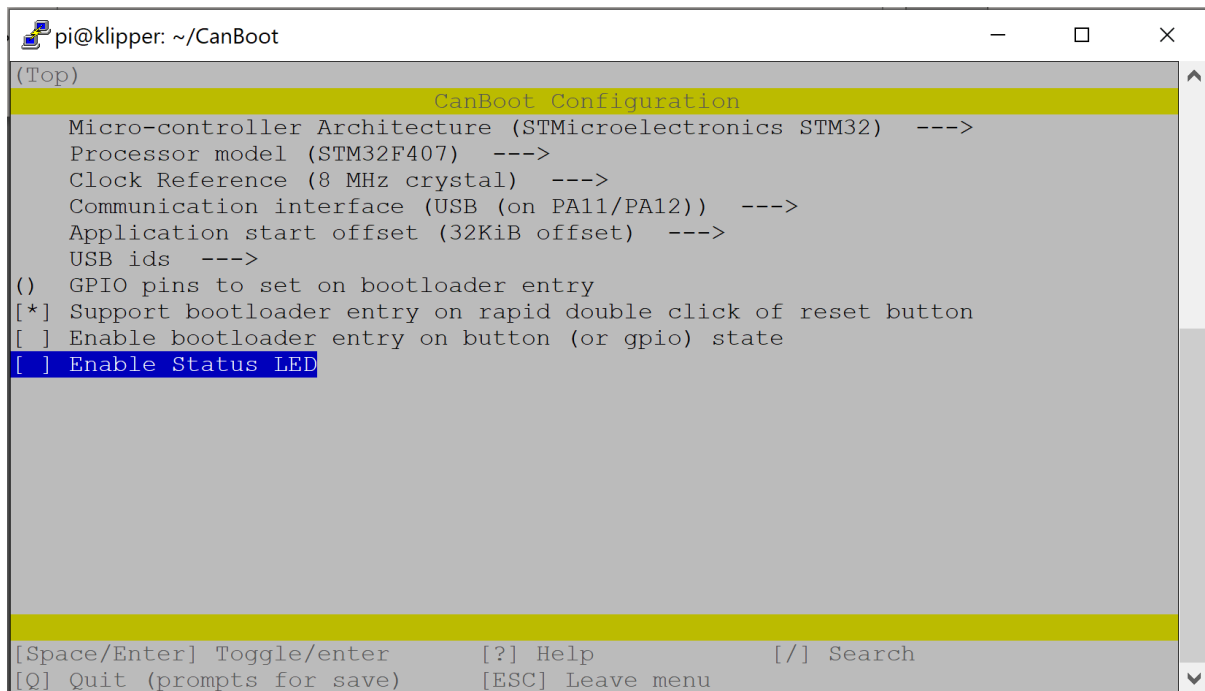
- 12MHz crystal, USB on PA11/PA12, 32KiB offset

```
pi@klipper: ~/CanBoot                                               —    □    ×
(Top)
                          CanBoot Configuration
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F446)  --->
    Clock Reference (12 MHz crystal)  --->
    Communication interface (USB (on PA11/PA12))  --->
    Application start offset (32KiB offset)  --->
    USB ids  --->
()  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




[Space/Enter] Toggle/enter        [?] Help           [/] Search
[Q] Quit (prompts for save)       [ESC] Leave menu
```

Octopus / Pro with F407 processor

- 8MHz crystal, USB on PA11/PA12, 32KiB offset

```
pi@klipper: ~/CanBoot                                               —    □    ×
(Top)
                          CanBoot Configuration
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F407)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (USB (on PA11/PA12))  --->
    Application start offset (32KiB offset)  --->
    USB ids  --->
()  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




[Space/Enter] Toggle/enter        [?] Help           [/] Search
[Q] Quit (prompts for save)       [ESC] Leave menu
```

Octopus Pro with F429 processor

- 8MHz crystal, USB on PA11/PA12, 32KiB offset

```
pi@klipper: ~/CanBoot                                          —    □    ✕
(Top)
                        CanBoot Configuration
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F429)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (USB (on PA11/PA12))  --->
    Application start offset (32KiB offset)  --->
    USB ids  --->
()  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



[Space/Enter] Toggle/enter        [?] Help              [/] Search
[Q] Quit (prompts for save)       [ESC] Leave menu
```

_____END OCTOPUS CONFIG OPTIONS_____

Press the Q then Y key to commit CanBoot config.

```
pi@klipper: ~/CanBoot                                          —    □    ✕
(Top)
                        CanBoot Configuration
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F429)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (USB (on PA11/PA12))  --->
    Application start offset (32KiB offset)  --->
    USB ids  --->
()  GPIO pins to set on boo          Quit
[*]  Support bootloader entr                      f reset button
[ ]  Enable bootloader entry    Save configuration?    e
[ ]  Enable Status LED
                                (Y)es  (N)o  (C)ancel



[Space/Enter] Toggle/enter        [?] Help              [/] Search
[Q] Quit (prompts for save)       [ESC] Leave menu
```
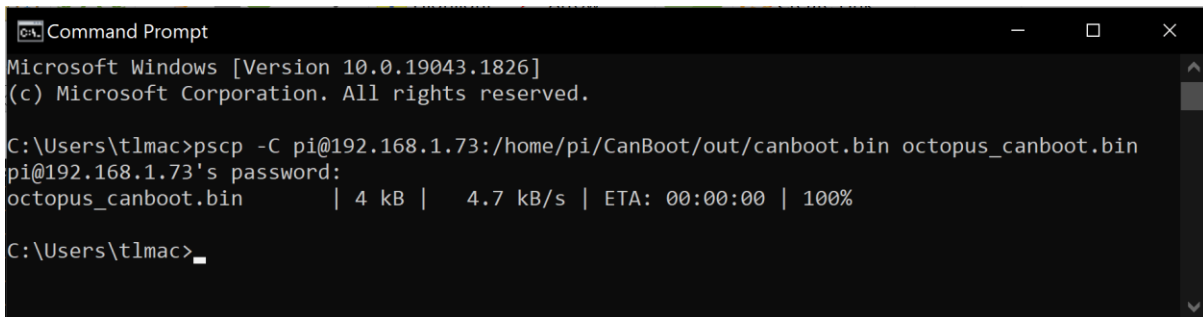
make

You have now created a canboot.bin firmware file for your Octopus board. My approach is to now make the CanBoot firmware for the EBB board. But if you do this now it will override the Octopus firmware you just created. So do the following:

On your computer, open a Windows CMD terminal

Type: pscp -C pi@<IP_ADDRESS>:/home/pi/CanBoot/out/canboot.bin octopus_canboot.bin

- Make sure to replace <IP_ADDRESS> with your Pi IP address or network name, like fluiddpi.local

```
Command Prompt                                             —    □    ×
Microsoft Windows [Version 10.0.19043.1826]
(c) Microsoft Corporation. All rights reserved.

C:\Users\tlmac>pscp -C pi@192.168.1.73:/home/pi/CanBoot/out/canboot.bin octopus_canboot.bin
pi@192.168.1.73's password:
octopus_canboot.bin          | 4 kB |   4.7 kB/s | ETA: 00:00:00 | 100%

C:\Users\tlmac>_
```

NOTE: You can also manually move the canboot.bin file to your computer with WinSCP. Make sure to name it octopus_canboot.bin (or similar). We will be moving an ebb_canboot.bin file next.
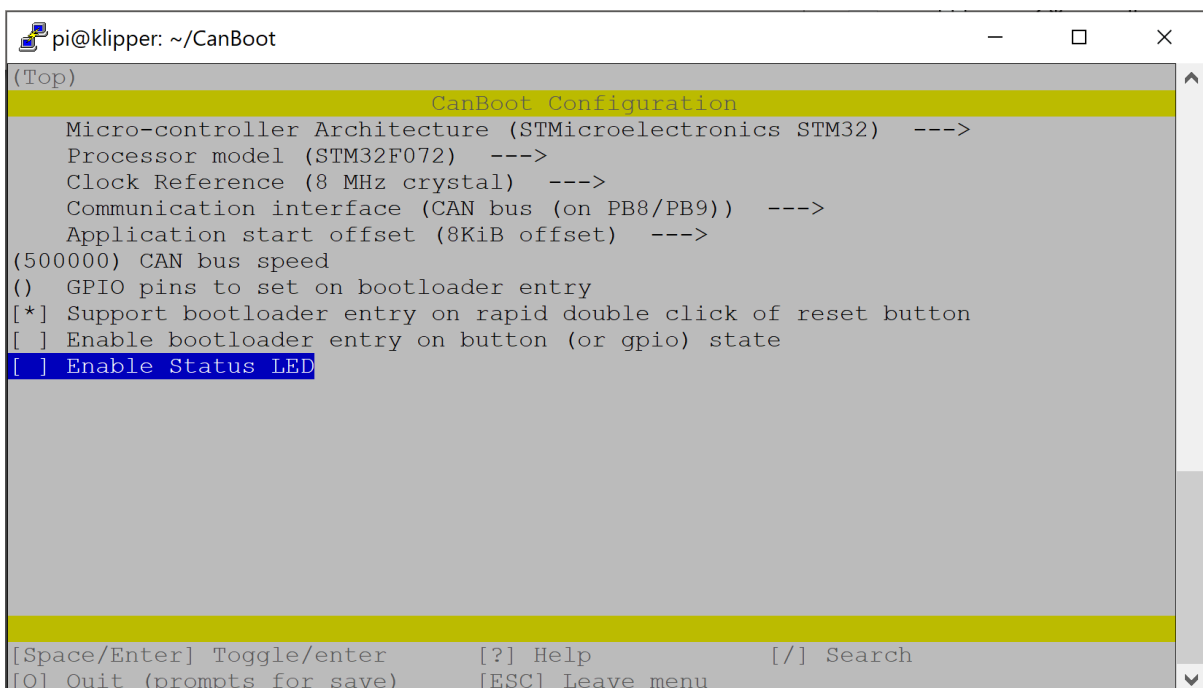
…Back to the Raspberry Pi SSH terminal

make clean

make menuconfig

EBB36 / EBB42 v1.0 with F072 processor

- 8MHz crystal, CAN bus on PB8/PB9, 8KiB offset, 500000 CAN bus speed

```
pi@klipper: ~/CanBoot                                       —    □    ×
(Top)
                    CanBoot Configuration
    Micro-controller Architecture (STMicroelectronics STM32)   --->
    Processor model (STM32F072)   --->
    Clock Reference (8 MHz crystal)   --->
    Communication interface (CAN bus (on PB8/PB9))   --->
    Application start offset (8KiB offset)   --->
(500000) 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




[Space/Enter] Toggle/enter        [?] Help              [/] Search
[Q] Quit (prompts for save)       [ESC] Leave menu
```

EBB36 / EBB42 v1.1 and v1.2 with G0B1 processor

- 8MHz crystal, CAN bus on PB0/PB1, 8KiB offset, 500000 CAN bus speed

```
pi@klipper: ~/CanBoot                                          —    □    ×
(Top)                                                                   ^
                        CanBoot Configuration
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32G0B1)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (CAN bus (on PB0/PB1))  --->
    Application start offset (8KiB offset)  --->
(500000) 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




[Space/Enter] Toggle/enter       [?] Help            [/] Search
[Q] Quit (prompts for save)      [ESC] Leave menu                      v
```

Press the Q then Y key to commit CanBoot config.

```
pi@klipper: ~/CanBoot                                          —    □    ×
(Top)                                                                   ^
                        CanBoot Configuration
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F429)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (USB (on PA11/PA12))  --->
    Application start offset (32KiB offset)  --->
    USB ids  --->
()   GPIO pins to set on boo┌──────── Quit ────────┐
[*] Support bootloader entr │                       │ f reset button
[ ] Enable bootloader entry │ Save configuration?   │ e
[ ] Enable Status LED       │                       │
                            │ (Y)es   (N)o   (C)ancel│
                            └───────────────────────┘




[Space/Enter] Toggle/enter       [?] Help            [/] Search
[Q] Quit (prompts for save)      [ESC] Leave menu                      v
```

make

On your computer, open a Windows CMD terminal (or use WinSCP to move file and rename)

Type: pscp -C pi@<IP_ADDRESS>:/home/pi/CanBoot/out/canboot.bin ebb_canboot.bin

- Make sure to replace <IP_ADDRESS> with your Pi IP address or network name, like fluiddpi.local

Your two CanBoot firmware files are now on your computer (in your user folder). Next step is to flash CanBoot to your boards.

1b. Flash CanBoot using STM32CubeProgrammer

You'll need to put your boards into DFU mode and connect to the computer using a USB-A to USB-C cable. Boot instructions for the Octopus and Octopus Pro are below:
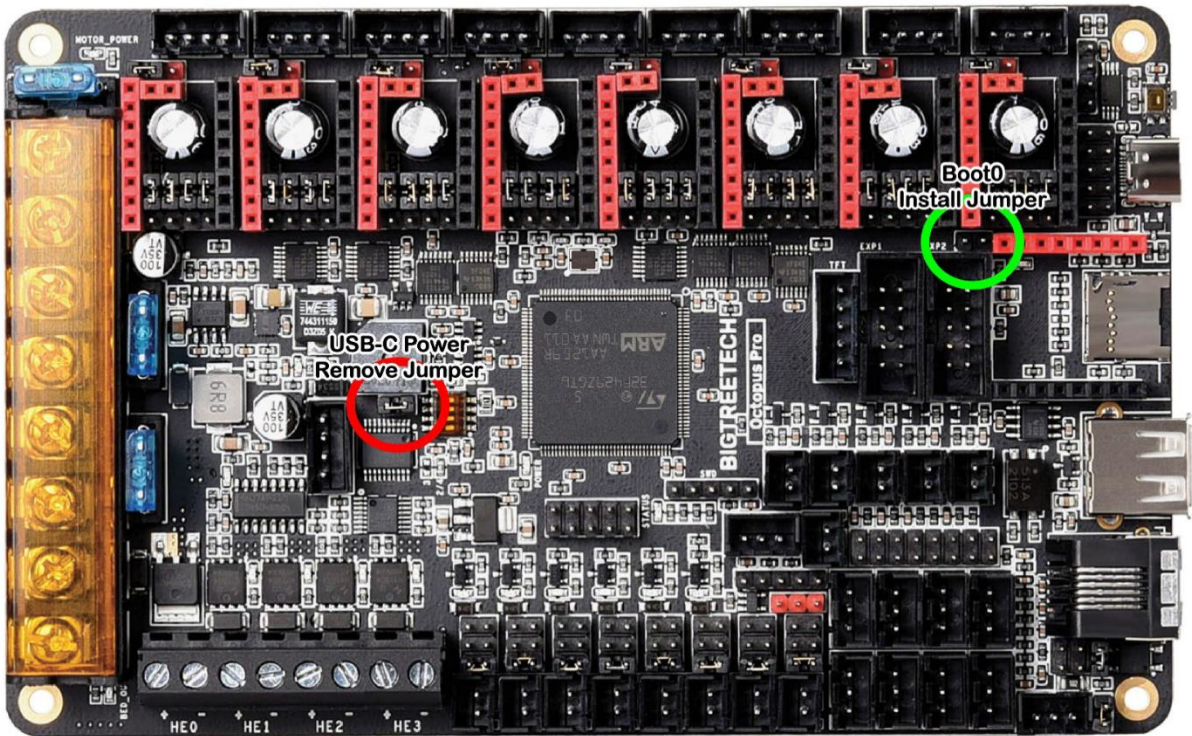
Make sure the power is off to the Octopus board

- To start have the USB-C to USB-A cable connected between the Windows PC and the Octopus Board.
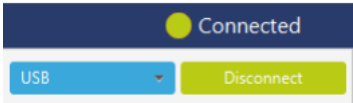- Remove the USB-C power jumper and place a jumper on the Boot0 header.
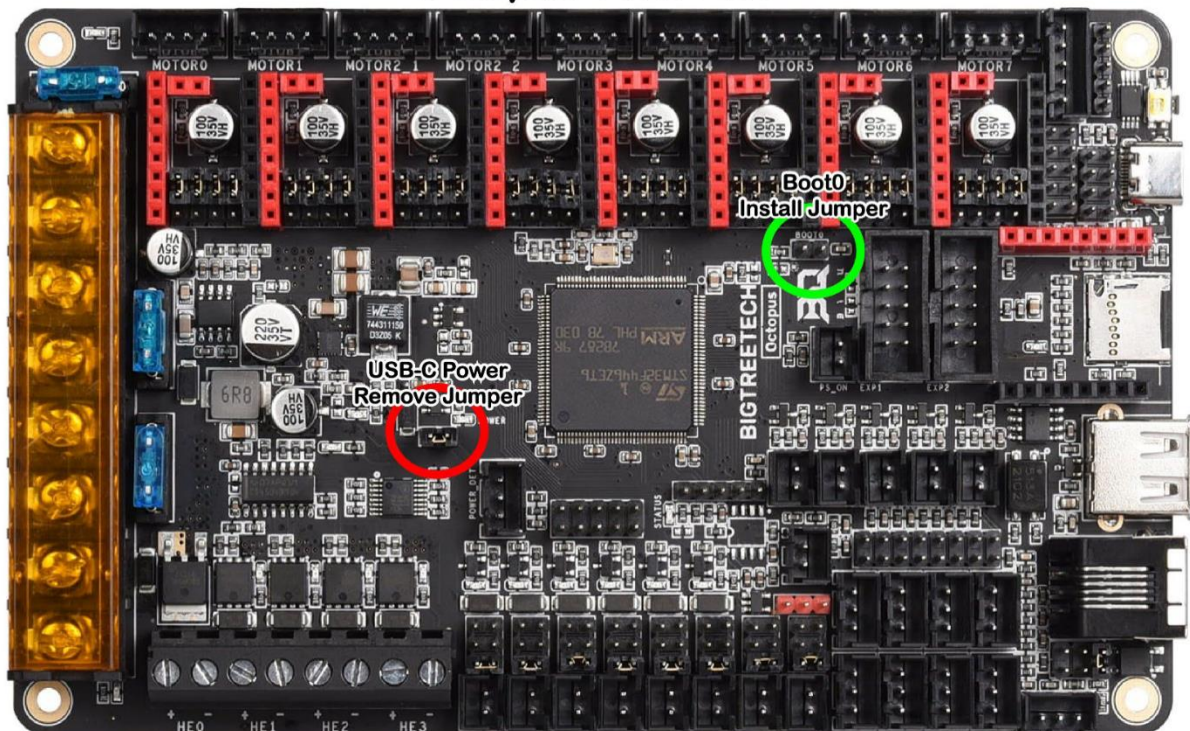
Octopus Pro
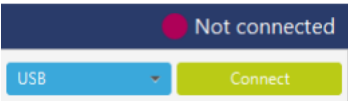
Apply power to the Octopus board.

First press and release the reset button on the Octopus. This will set the Octopus into DFU mode.
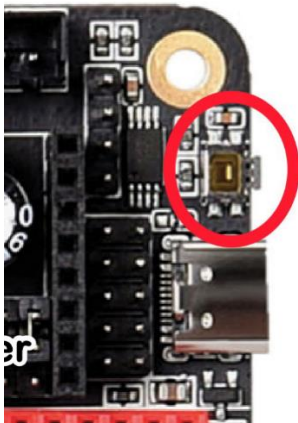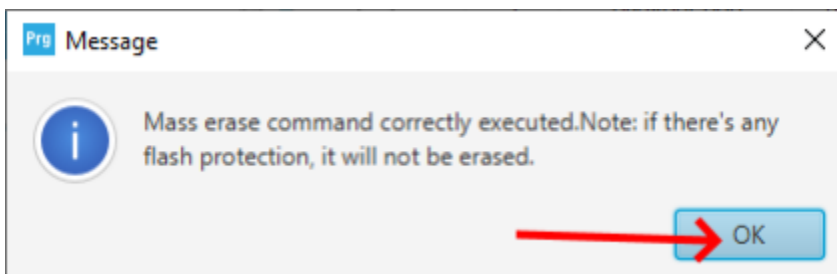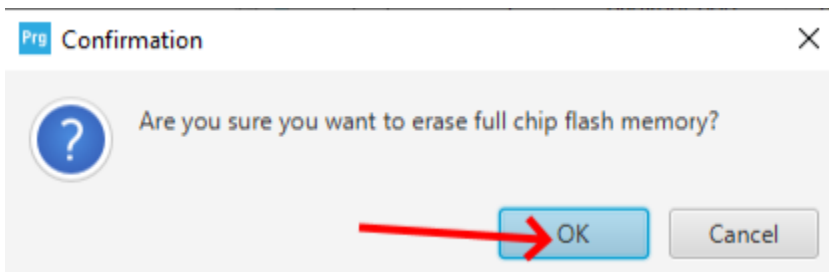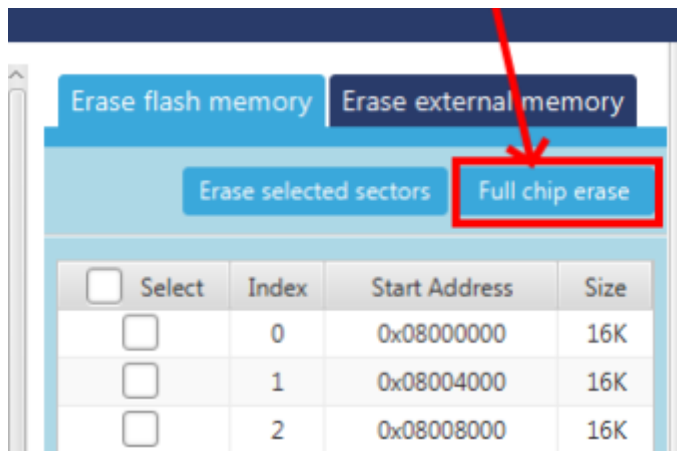

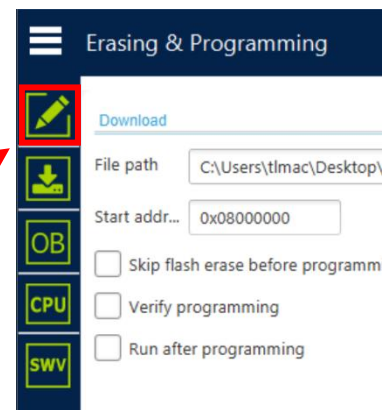Now start the STM32CubeProgramer on your Windows system.

- Top on the right you should see make sure to set for USB then click Connect.

- You should now see if yes then on the left find and click this button. (see next page)
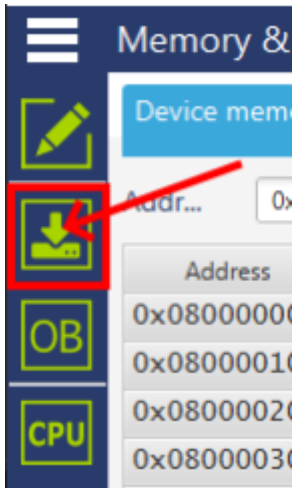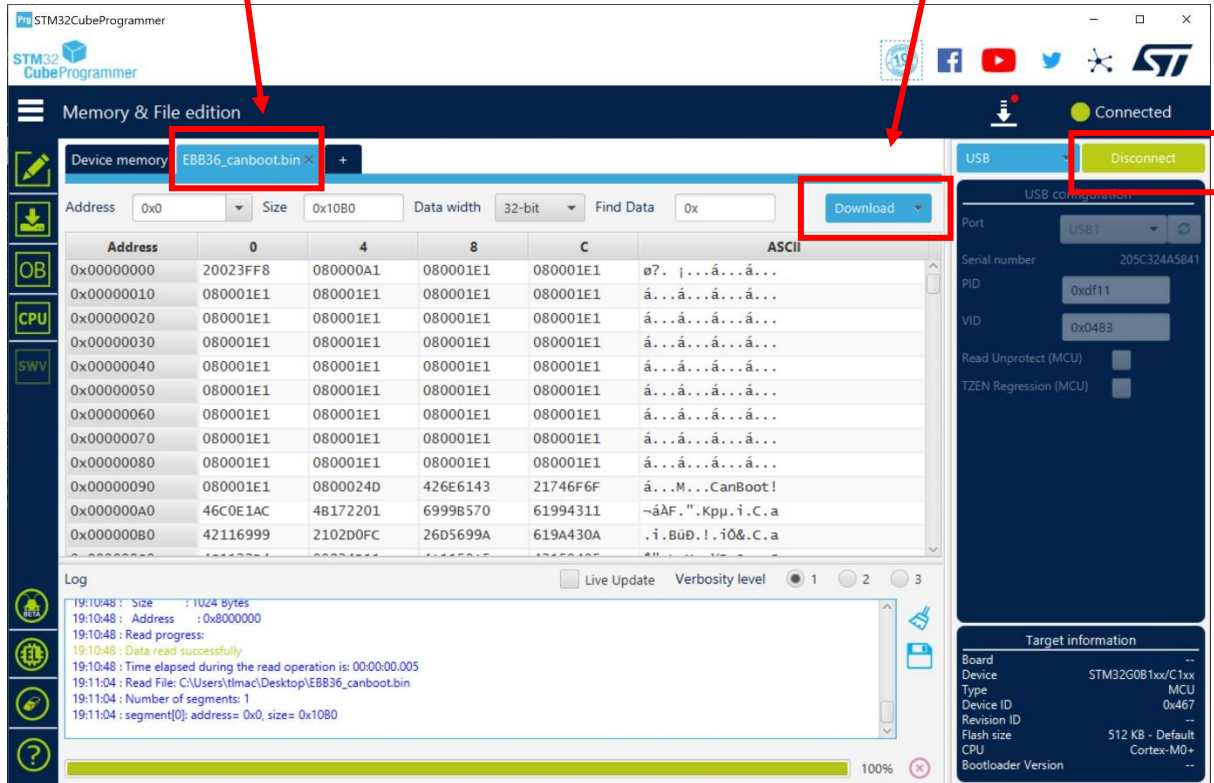
- For CanBoot and Klipper to work correctly we need to wipe the flash memory by clicking this button.

- Go back to main page by clicking on this button

- Select (1) "Open File" tab, find the **octopus_canboot.bin file**, and (2) download to your Octopus board. (3) Click Disconnect when done.
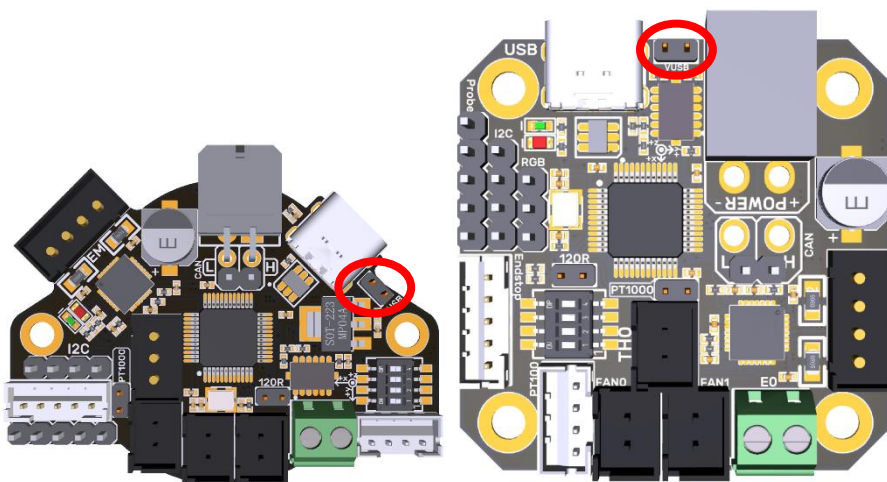

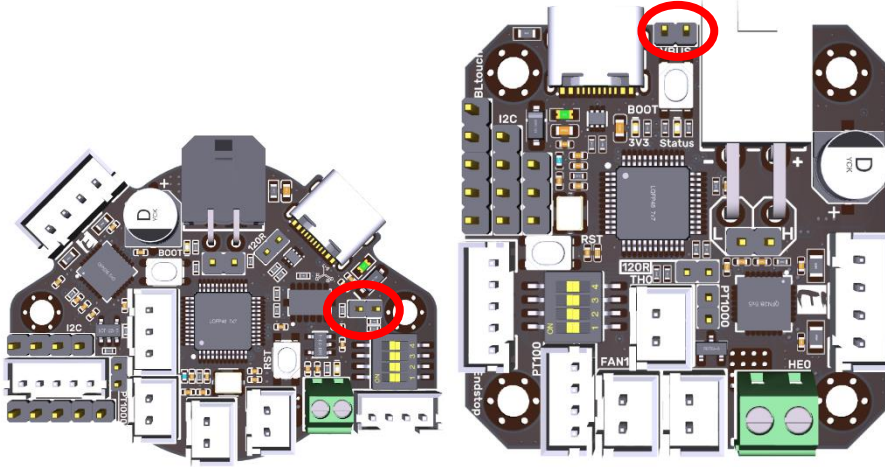**Power off your Octopus board**. Remove the jumper from BOOT0.

Now you'll flash your EBB toolboard.

This step will power the toolboard from USB only. Place a jumper on VBUS header.

EBB36 / EBB42 v1.0

EBB36 / EBB42 v1.1 and v1.2

WARNING! For **EBB36/42 v1.1** when you enter DFU mode and flash new firmware it turns on the hotend heater pin. If you have voltage supplied (12/24V) this will heat up the hotend with no thermal protection. This can cause damage and is why I use USB power. I have had no issues with CanBoot / Klipper flashing outside of using STM32CubeProgramer

Enter DFU mode, you need to press and hold the BOOT button then press the RESET button. Release both buttons.

EBB36 / EBB42 v1.0

- Both buttons are on the back of the boards
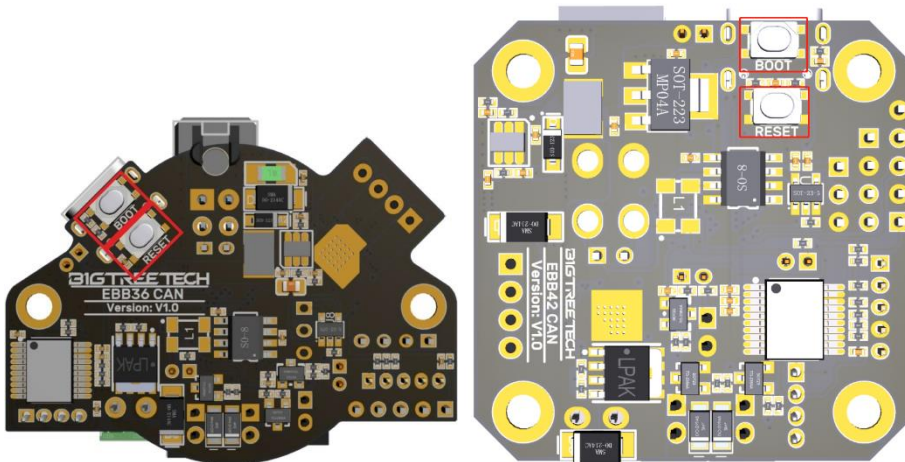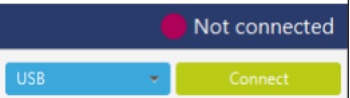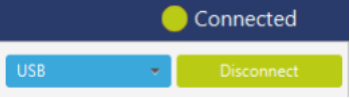
EBB36 / EBB42 v1.1 and v1.2

- Both buttons are on the front of the boards
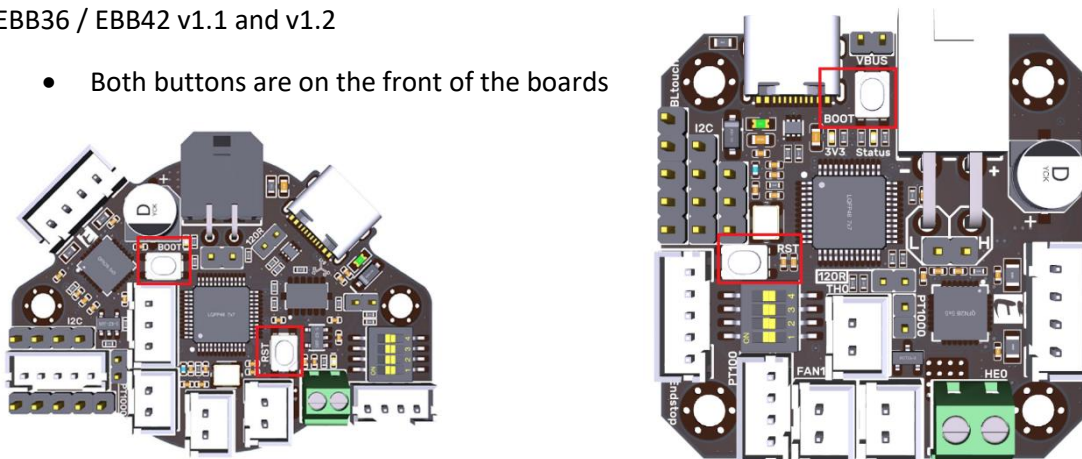
Now start the STM32CubeProgramer on your Windows system.

- Top on the right you should see make sure to set for USB then click Connect.

- You should now see if yes then on the left find and click this button.
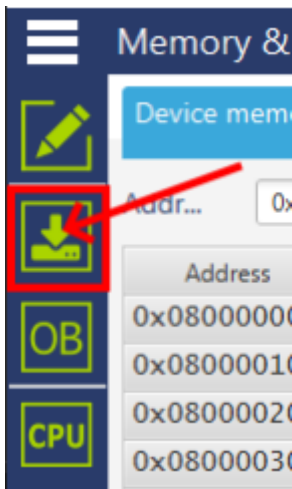
- For CanBoot and Klipper to work correctly we need to wipe the flash memory by clicking this button.
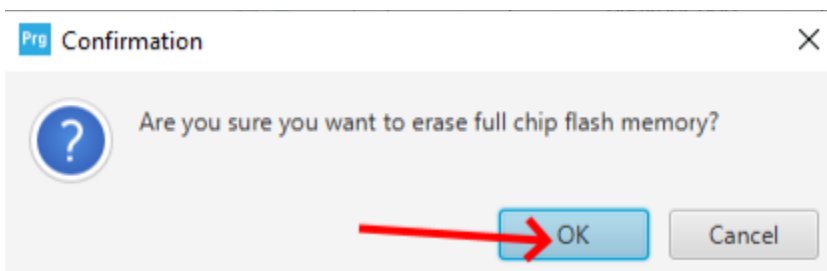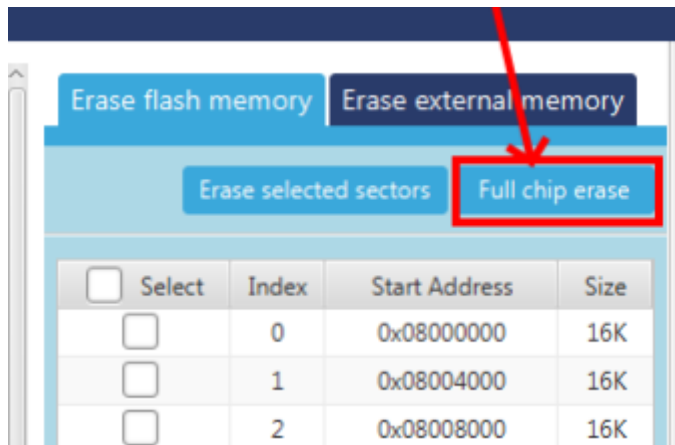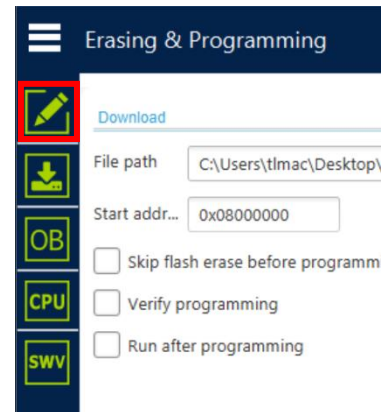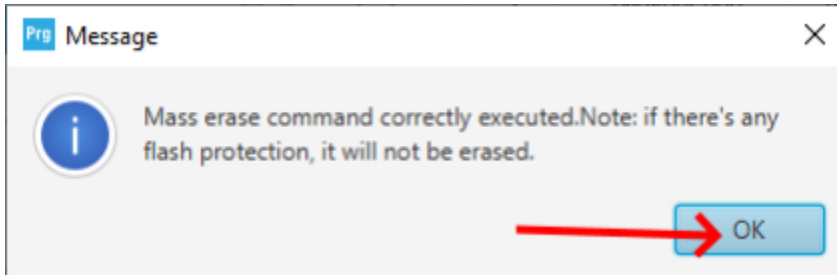
- Go back to main page by clicking on this button

- Select (1) "Open File" tab, find the **ebb_canboot.bin file**, and (2) download to your EBB board. (3) Click Disconnect when done.
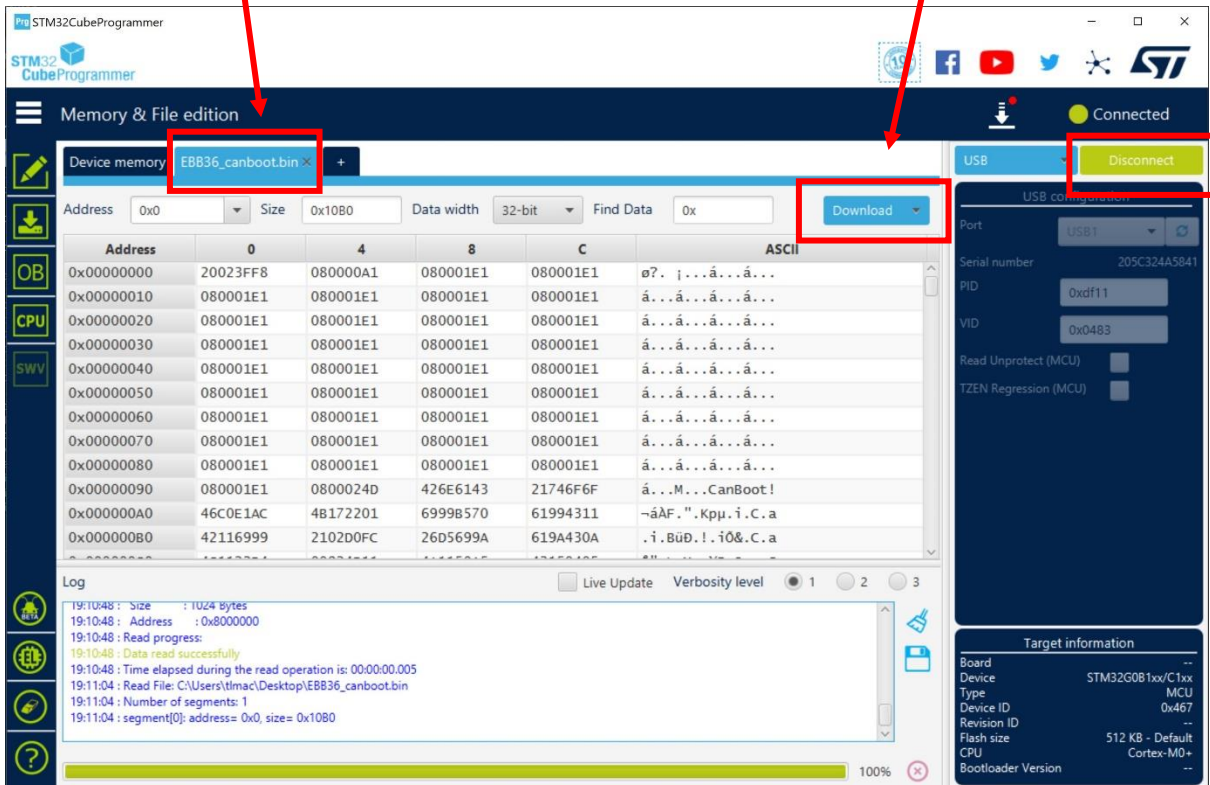
Power off your EBB board by disconnecting USB from computer. Remove the jumper from VBUS

Place the jumper on the 120R pins. (see next page)

EBB36 / EBB42 v1.0
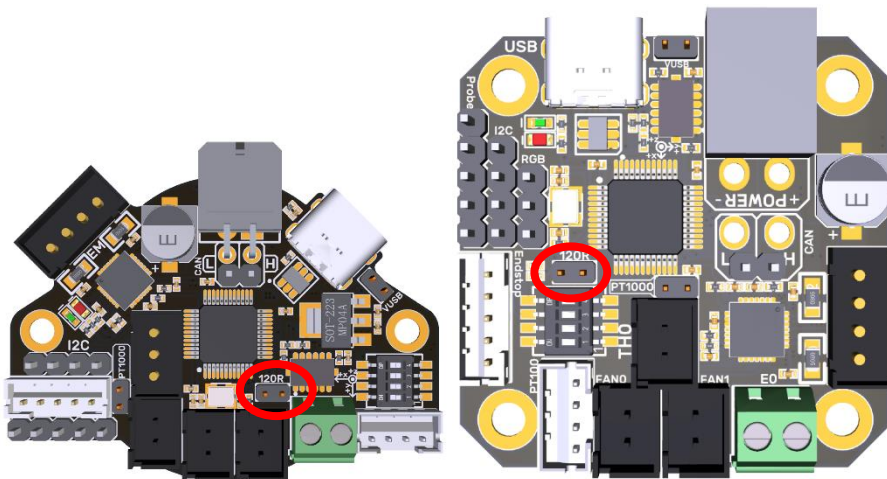
EBB36 / EBB42 v1.1 and v1.2
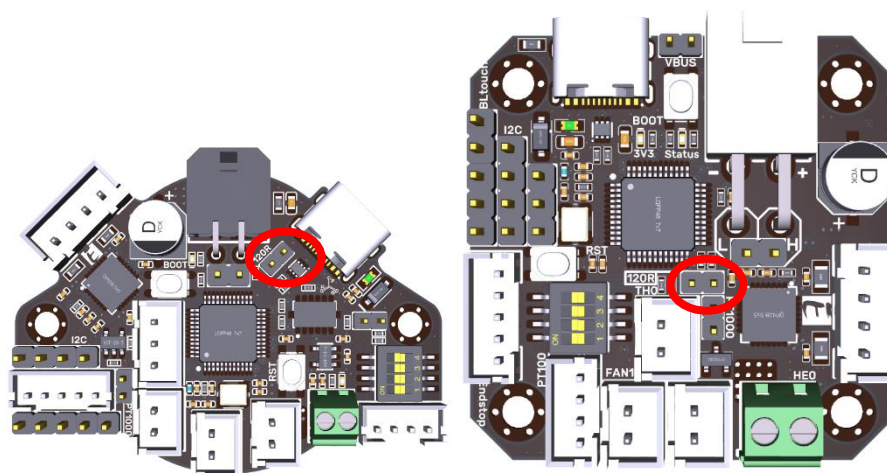
Set aside the EBB toolboard for now.

2. Setup Octopus / Pro for Klipper

Go back to your SSH terminal window.

🖥️ cd ~/klipper

🖥️ make menuconfig

2a. Octopus - Setup Klipper for USB to CAN bus bridge, with CAN comms to EBB

You will now setup the new USB to CAN bus bridge on your Octopus / Pro board. Below are the Klipper config options for each board. Make sure to have "Enable extra low-level configuration options" selected.

Octopus / Pro with F446 processor

- 32KiB offset, 12MHz crystal, USB to CAN bus bridge (USB on PA11/PA12), CAN bus on PD0/PD1, 500000 CAN bus speed

```
pi@klipper: ~/klipper                                          —    □    ×
(Top)
                    Klipper Firmware Configuration
[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F446)  --->
    Bootloader offset (32KiB bootloader)  --->
    Clock Reference (12 MHz crystal)  --->
    Communication interface (USB to CAN bus bridge (USB on PA11/PA12))  --->
    CAN bus interface (CAN bus (on PD0/PD1))  --->
    USB ids  --->
(500000) CAN bus speed
()    GPIO pins to set at micro-controller startup (NEW)




[Space/Enter] Toggle/enter        [?] Help            [/] Search
[Q] Quit (prompts for save)       [ESC] Leave menu
```

Octopus / Pro with F407 processor

- 32KiB offset, 8MHz crystal, USB to CAN bus bridge (USB on PA11/PA12), CAN bus on PD0/PD1, 500000 CAN bus speed

```
pi@klipper: ~/klipper                                          —    □    ×
(Top)
                    Klipper Firmware Configuration
[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F407)  --->
    Bootloader offset (32KiB bootloader)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (USB to CAN bus bridge (USB on PA11/PA12))  --->
    CAN bus interface (CAN bus (on PD0/PD1))  --->
    USB ids  --->
(500000) CAN bus speed
()    GPIO pins to set at micro-controller startup (NEW)




[Space/Enter] Toggle/enter        [?] Help            [/] Search
[Q] Quit (prompts for save)       [ESC] Leave menu
```

Octopus Pro with F429 processor

- 32KiB offset, 8MHz crystal, USB to CAN bus bridge (USB on PA11/PA12), CAN bus on PD0/PD1, 500000 CAN bus speed

```
pi@klipper: ~/klipper                                           —    □    ×
(Top)
                    Klipper Firmware Configuration
[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F429)  --->
    Bootloader offset (32KiB bootloader)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (USB to CAN bus bridge (USB on PA11/PA12))  --->
    CAN bus interface (CAN bus (on PD0/PD1))  --->
    USB ids  --->
(500000) CAN bus speed
()   GPIO pins to set at micro-controller startup (NEW)




[Space/Enter] Toggle/enter      [?] Help              [/] Search
[Q] Quit (prompts for save)     [ESC] Leave menu
```

END OCTOPUS CONFIG OPTIONS

Press the Q then Y key to commit Klipper config.

```
pi@klipper: ~/klipper                                           —    □    ×
(Top)
                    Klipper Firmware Configuration
[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F429)  --->
    Bootloader offset (32KiB bootloader)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (USB to CAN bus bridge (USB on PA11/PA12))  --->
    CAN bus interface (CAN              Quit
    USB ids  --->
(500000) CAN bus speed        Save configuration?
()   GPIO pins to set at mic                             W)
                              (Y)es   (N)o   (C)ancel




[Space/Enter] Toggle/enter      [?] Help              [/] Search
[Q] Quit (prompts for save)     [ESC] Leave menu
```

make

NOTE:

Just like with the CanBoot firmware we will be making Klipper firmware for the Octopus /
Pro board and the EBB board. Since we are already compiling Klipper firmware, I like to
compile the EBB Klipper firmware now. But if we do, then the Octopus Klipper firmware we
just made will be overwritten. So:

>_SSH mv ~/klipper/out/klipper.bin octopus_klipper.bin

>_SSH make clean

NOTE:

This moves the klipper firmware file to the klipper folder and renames it octopus_klipper.bin
then deletes the compile files and folders, including /klipper/out

2b. EBB - Setup Klipper for CAN comms

>_SSH make menuconfig

EBB36 / EBB42 v1.0 with F072 processor

- 8KiB bootloader, 8MHz crystal, CAN bus on PB8/PB9, 500000 CAN bus speed

```
pi@klipper: ~/klipper                                        —    □    ×
(Top)                                                                  ^
                    Klipper Firmware Configuration
[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F072)  --->
    Bootloader offset (8KiB bootloader)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (CAN bus (on PB8/PB9))  --->
(500000) CAN bus speed
()   GPIO pins to set at micro-controller startup




[Space/Enter] Toggle/enter          [?] Help            [/] Search
[Q] Quit (prompts for save)         [ESC] Leave menu                 v
```

EBB36 / EBB42 v1.1 / v1.2 with G0B1 processor

- 8KiB bootloader, 8MHz crystal, CAN bus on PB0/PB1, 500000 CAN bus speed

```
pi@klipper: ~/klipper                                    —    □    ✕
(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 PB0/PB1))  --->
(500000) CAN bus speed
()  GPIO pins to set at micro-controller startup



[Space/Enter] Toggle/enter        [?] Help           [/] Search
[Q] Quit (prompts for save)       [ESC] Leave menu
```

Press the Q then Y key to commit CanBoot config.

```
pi@klipper: ~/klipper                                    —    □    ✕
(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 PB0/PB1))  --->
(500000) CAN bus speed        Quit
()  GPIO pins to set at mic
                          Save configuration?

                          (Y)es  (N)o  (C)ancel



[Space/Enter] Toggle/enter        [?] Help           [/] Search
[Q] Quit (prompts for save)       [ESC] Leave menu
```

⊟ make

⊟ mv ~/klipper/out/klipper.bin ebb_klipper.bin

⊟ make clean

2c. Find serial device for Octopus / Pro on Raspberry Pi

Connect your Octopus / Pro board to the Raspberry Pi with a USB-A to USB-C cable…connected to the USB-C port on the Octopus board. **Power on your printer.**

[SSH] cd

[SSH] ls /dev/serial/by-id/*

This will give you something similar to below. YOUR DEVICE WILL BE DIFFERENT!

```
pi@klipper:~/CanBoot $ cd
pi@klipper:~ $ ls /dev/serial/by-id/*
/dev/serial/by-id/usb-CanBoot_stm32f429xx_350026000B50314B33323220-if00
pi@klipper:~ $
```

Copy this, for example:

/dev/serial/by-id/usb-CanBoot_stm32f429xx_350026000B50314B33323220-if00

2d. Flash Klipper to Octopus / Pro with CanBoot **serial** command

[SSH] cd CanBoot/scripts

[SSH] pip3 install pyserial

…this only has to be done once

[SSH] python3 flash_can.py -f ~/klipper/octopus_klipper.bin -d <serial_device>

NOTE: Replace <serial_device> with the path you copied earlier…

```
pi@klipper:~/CanBoot/scripts $ python3 flash_can.py -f ~/klipper/octopus_klipper
.bin -d /dev/serial/by-id/usb-CanBoot stm32429xx 350026000B50314B33323220-if00
Attempting to connect to bootloader
CanBoot Connected
Protocol Version: 1.0.0
Block Size: 64 bytes
Application Start: 0x8008000
MCU type: stm32f429xx
Flashing '/home/pi/klipper/octopus_klipper.bin'...

[#################################################]

Write complete: 2 pages
Verifying (block count = 441)...

[#################################################]

Verification Complete: SHA = 43EF7BD0026EC6B76C97E2692F2FB3DDE590ADDA
CAN Flash Success
pi@klipper:~/CanBoot/scripts $
```

NOTE: The Octopus board communicates over USB initially. Only this flash (and Klipper updates, more on that later) use the serial device path. The Octopus board now has a CAN uuid…let's find it!
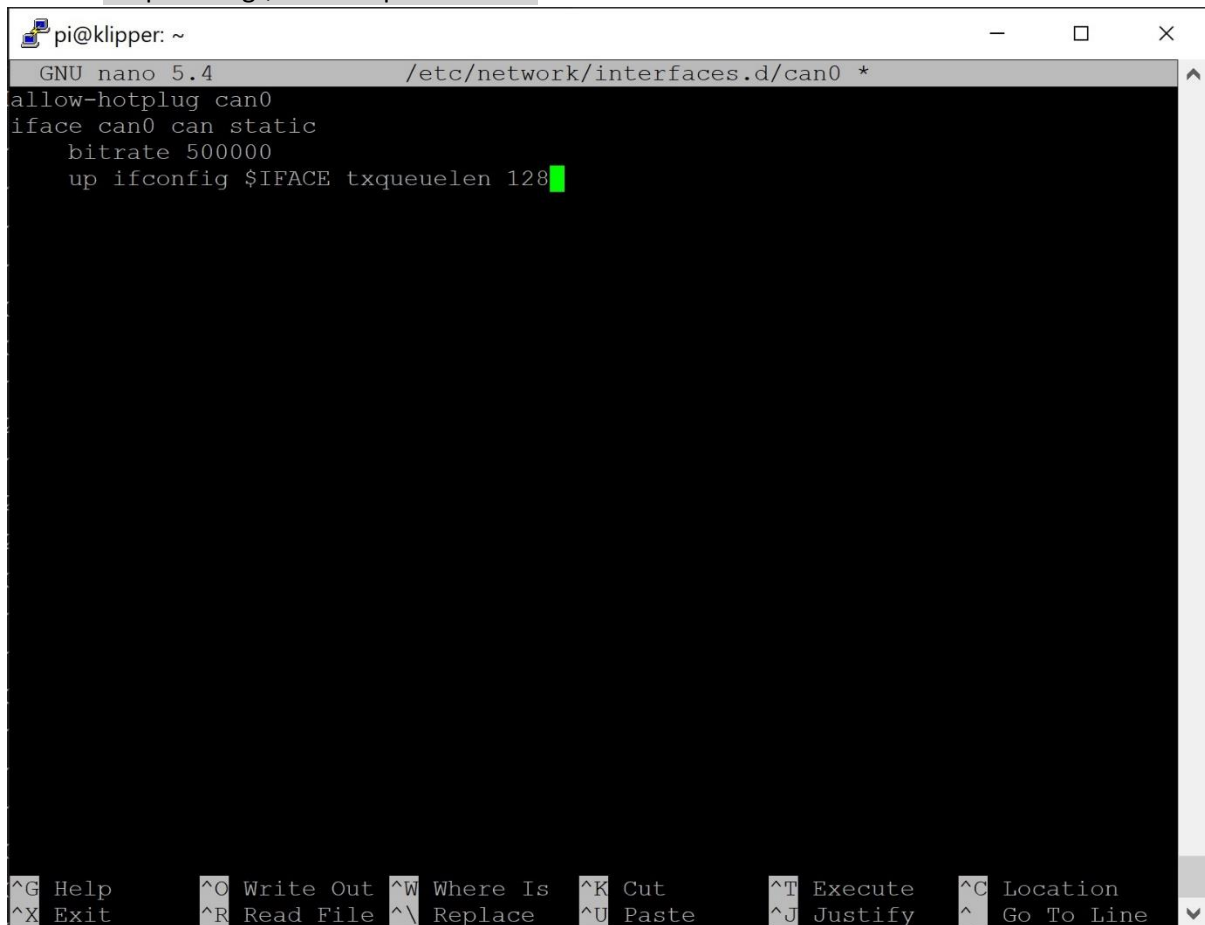
3. Setup can0 Network on Raspberry Pi, Power Cycle Printer

Now we need to enable the can0 network and set it up so that we can have CAN bus working

>_SSH cd

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

- Copy and paste the following:

allow-hotplug can0
iface can0 can static
    bitrate 500000
    up ifconfig $IFACE txqueuelen 128

```
pi@klipper: ~                                                    —    □    ×
  GNU nano 5.4              /etc/network/interfaces.d/can0 *
allow-hotplug can0
iface can0 can static
    bitrate 500000
    up ifconfig $IFACE txqueuelen 128

^G Help        ^O Write Out ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit        ^R Read File ^\ Replace   ^U Paste    ^J Justify  ^  Go To Line
```

Ctrl – X to EXIT.

Key Y for YES, and ENTER.

Power off your printer. Time to plug in your EBB toolboard. Plug power and can data cables into EBB. Plug RJ11 or RJ12 connector into Octopus / Pro board.

Power on your printer.

NOTE:

We just created the can0 interface as a hot-plug network. This means that when something changes to devices, or new devices are connected, the network reacts. Power cycling the printer is a change that the can0 network detects and configures the Octopus to work as a CAN device. Also, this hot-plug option seems to make "firmware_restarts" smoother.

4. Find CAN uuid for Octopus / Pro

Now that can0 is setup, let's make sure it is up and running.

[SSH] ifconfig

```
pi@klipper:~ $ ifconfig
can0: flags=193<UP,RUNNING,NOARP>  mtu 16
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 128
(UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether b8:27:eb:64:17:e1  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 5393  bytes 1389761 (1.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5393  bytes 1389761 (1.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.73  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::2652:fc24:eb09:c7ef  prefixlen 64  scopeid 0x20<link>
```

You should get results similar to this. We are looking for the can0 on top. Success!

NOTE: If this doesn't work try to turn off power to your printer and reboot your Raspberry Pi. Wont booted back up, open a SSH terminal and try ifconfig again. Can0 won't show up. Power on your printer and then try ifconfig. Should have can0 up and running.

[SSH] cd CanBoot/scripts

[SSH] python3 flash_can.py -i can0 -q

```
pi@klipper:~/CanBoot/scripts $ python3 flash_can.py -i can0 -q
Resetting all bootloader node IDs...
Checking for canboot nodes...
Detected UUID: 127081e7e3c6, Application: CanBoot
Detected UUID: 463b35222d7b, Application: Klipper
Query Complete
pi@klipper:~/CanBoot/scripts $
```

In this instance both EBB and Octopus are detected. The EBB only has CanBoot on it while the Octopus has Klipper. It is obvious which is which. Save both for later.

5. Flash Klipper to EBB board with CanBoot **CAN** command

`[SSH]` python3 flash_can.py -f ~/klipper/ebb_klipper.bin -u <ebb_uuid>

NOTE: Replace <ebb_uuid> with the CAN uuid found before. The EBB is labelled with "CanBoot". In this case mine is 127081e7e3c6

```
pi@klipper:~/CanBoot/scripts $ python3 flash_can.py -f ~/klipper/ebb_klipper.bin
 -u 127081e7e3c6
Sending bootloader jump command...
Resetting all bootloader node IDs...
Checking for canboot nodes...
Detected UUID: 463b35222d7b, Application: Klipper
Detected UUID: 127081e7e3c6, 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/pi/klipper/ebb_klipper.bin'...

[#################################################]

Write complete: 12 pages
Verifying (block count = 375)...

[#################################################]

Verification Complete: SHA = DF6960C31BC813C14F1FDF111B65D258772002FB
CAN Flash Success
pi@klipper:~/CanBoot/scripts $ ▮
```

`[SSH]` cd ~/klipper

`[SSH]` rm octopus_klipper.bin ebb_klipper.bin

6. Printer Config Update and General Tips

Edit your printer.cfg to now include your new CAN uuid. For example:

    [mcu]
    canbus_uuid: 463b35222d7b

    [mcu EBBCan]
    canbus_uuid: 127081e7e3c6

Save and restart.

If you have issues here, try a "Restart" (restarts Klipper) and/or a "Firmware Restart" a 2nd time.

- The issue is can0 network disconnects when the restart causes a power cycle on your printer. The timing can be too fast and the can0 network doesn't properly come back up. Another restart command is sometimes needed.

- According to Klipper CANBUS docs, you might need to issue `[SSH]` sudo ip up in a SSH terminal to restart the can0 network. I haven't had to do this. But this might be useful during initial setup.

- You can also use [SSH] sudo ip link set up can0 type can bitrate 500000 in a SSH terminal to start can0 if it isn't coming up with ifconfig. I have used this, but find that if the install instructions are followed then I don't need it.
- See Klipper docs on CANBUS for official information
  https://www.klipper3d.org/CANBUS.html#usb-to-can-bus-bridge-mode

7. How to Use CanBoot to update boards, Tips

With CanBoot and Klipper now properly flashed to your Octopus / Pro board and EBB toolboard you now have two devices with unique CAN uuid on can0 network. But how to use CanBoot to upgrade Klipper firmware!?

When you run python3 flash_can.py ……. the python script should trigger the board to go into DFU mode, to access CanBoot on the board and flash the Klipper update. I can't cleanly get this to work with my Octopus Pro with F429 processor.

- Flashing with CAN uuid gives an error. Trying to use the original serial ID also gives an error.
- How I get it to work:
  o Make Klipper firmware for Octopus board, cd to CanBoot/scripts
  o I run: python3 flash_can.py -f ~/klipper/octopus_klipper.bin -u 463b35222d7b
    ▪ Note I am still moving and renaming firmware files. If you don't want to do this then leave out: -f ~/klipper/octopus_klipper.bin and CanBoot defaults to using default location of Klipper firmware [~/klipper/out/klipper.bin]
  o CanBoot throws an error. If I try ifconfig, the can0 network is down
  o Now if I try ls /dev/serial/by-id/* I get a serial ID:
  o /dev/serial/by-id/usb-CanBoot_stm32f429xx_350026000B50314B33323220-if00
  o Same as earlier…
  o Now I run:
  o python3 flash_can.py -f ~/klipper/octopus_klipper.bin -d /dev/serial/by-id/usb-CanBoot_stm32f429xx_350026000B50314B33323220-if00
  o And CanBoot works and flashes Klipper update!
  o I can now update the EBB with: python3 flash_can.py -f ~/klipper/ebb_klipper.bin -u 127081e7e3c6

Long story short;

1. Try to flash Octopus board with CAN uuid, it will fail and can0 network is down.
2. Now you can flash Octopus board with serial ID.
3. Lastly you can flash EBB board with CAN uuid.

CAN Bus Wiring on next page…

RJ11 / RJ12 CAN bus Wiring Info

To connect your EBB36/42 or other CAN enabled toolboard directly to the Octopus boards you need to use the RJ11 / RJ12 telephone port.

The easiest option is to use a 2-3 meter telephone cable and cut off one end. Crimp for Molex MiniFit or Molex MicroFit connector, depending on your CAN toolboard. The RJ11/12 connector goes to the Octopus board.

- This does not allow for twisted wire, which is recommended for CAN bus wiring.
- I have the 4 wire RJ11 telephone cable running directly to my EBB36, no twisted wire. No issues on my end but I will work to get twisted wire setup in the future.
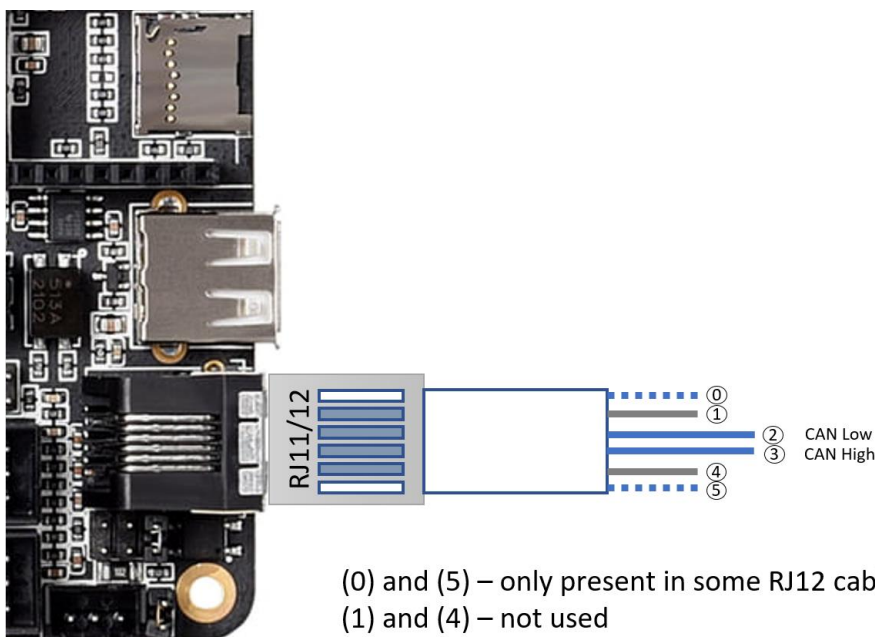
(0) and (5) – only present in some RJ12 cables, not used
(1) and (4) – not used
(2) – CAN Low
(3) – CAN High

When you plug in the RJ11/12 cable to the Octopus board, the wiring (from top down) is as above. **DO NOT RELY ON WIRE COLOURS.** For example:
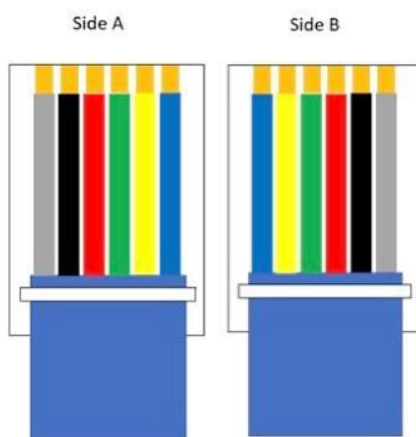
If plugging in a RJ12 cable with SIDE A your CAN Low wire is GREEN and CAN High wire is RED.
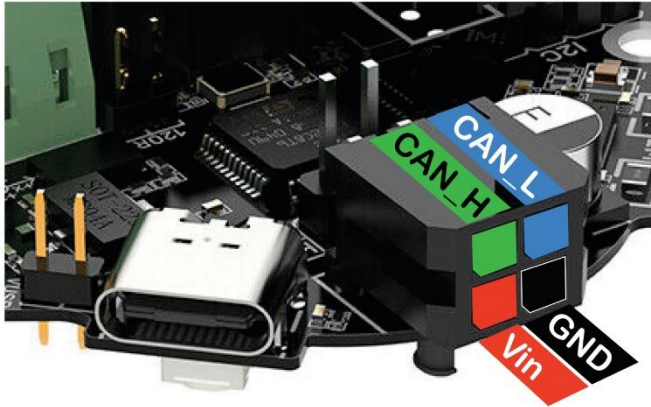
If plugging in a RJ12 cable with SIDE B your CAN Low wire is RED and CAN High wire is GREEN.

The colour of your wiring setup will be situational.

Take note of the wiring that you have plugged into the Octopus board. Wire 2 is your CAN Low connection. Note the colour of the wire and make sure it goes to CAN_L on the EBB end. Same thing for the CAN High on Wire 3. That colour wire goes to CAN_H on the EBB end.

You must also supply 12V or 24V directly to the EBB toolboard. Run 18awg directly from your power supply. Example EBB connections are below. Again, colours on the diagrams do not indicate wire colours.

EBB36

EBB42