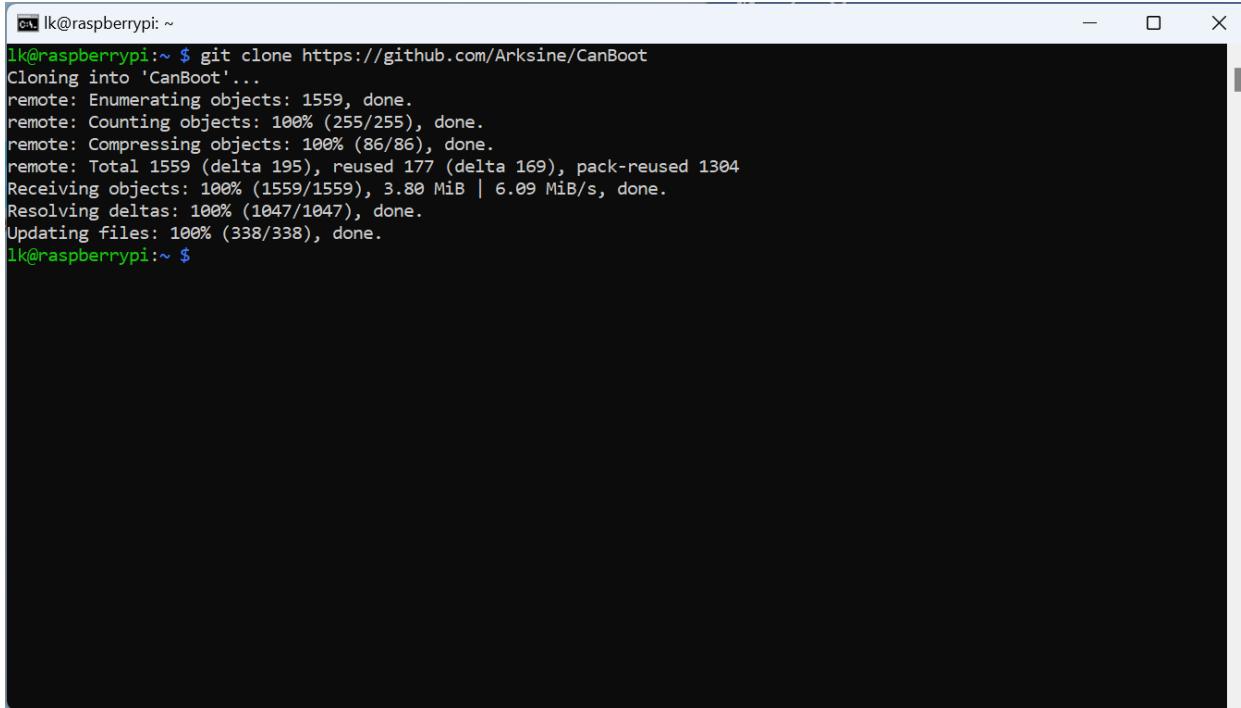


**The following document is a revised step-by-step guide, originally created for this video : <https://www.youtube.com/watch?v=Ho8dSqwX4-Q> (posted by Driver 732)**

## GENERATE CANBOOT FIRMWARE

Type: 'git clone <https://github.com/Arksine/CanBoot>' to download CanBoot by Arksine

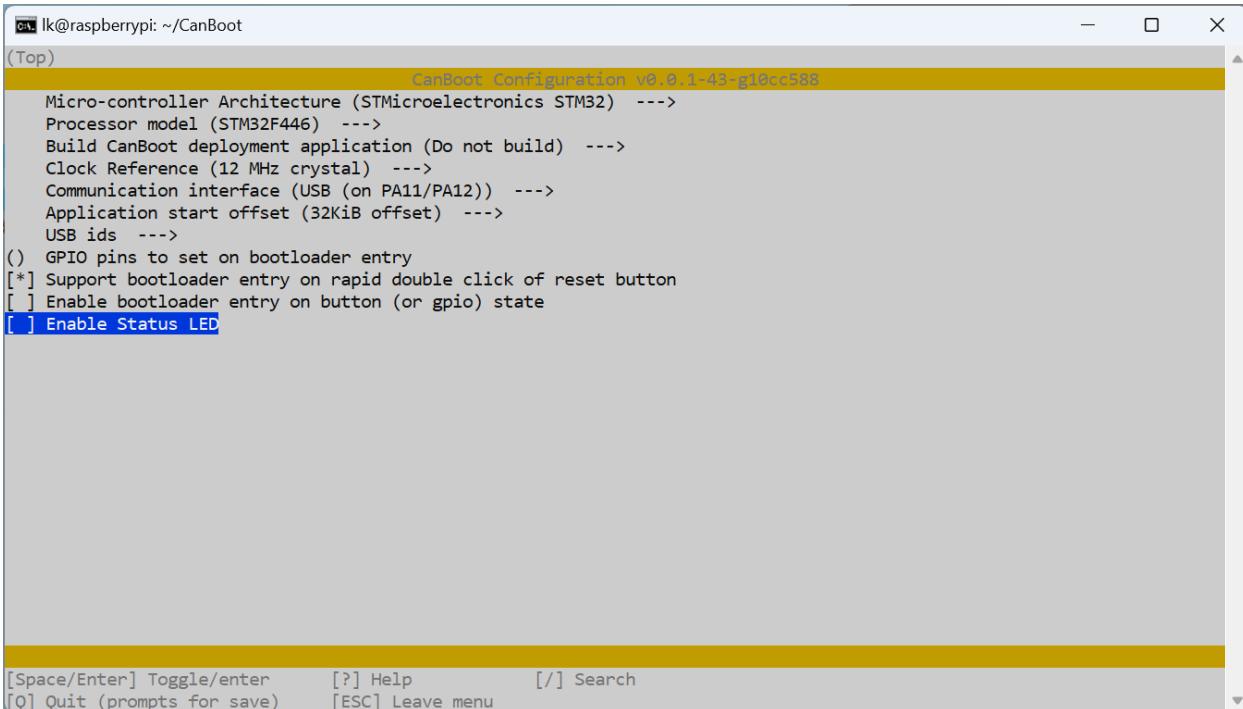


```
lk@raspberrypi:~ $ git clone https://github.com/Arksine/CanBoot
Cloning into 'CanBoot'...
remote: Enumerating objects: 1559, done.
remote: Counting objects: 100% (255/255), done.
remote: Compressing objects: 100% (86/86), done.
remote: Total 1559 (delta 195), reused 177 (delta 169), pack-reused 1304
Receiving objects: 100% (1559/1559), 3.80 MiB | 6.09 MiB/s, done.
Resolving deltas: 100% (1047/1047), done.
Updating files: 100% (338/338), done.
lk@raspberrypi:~ $
```

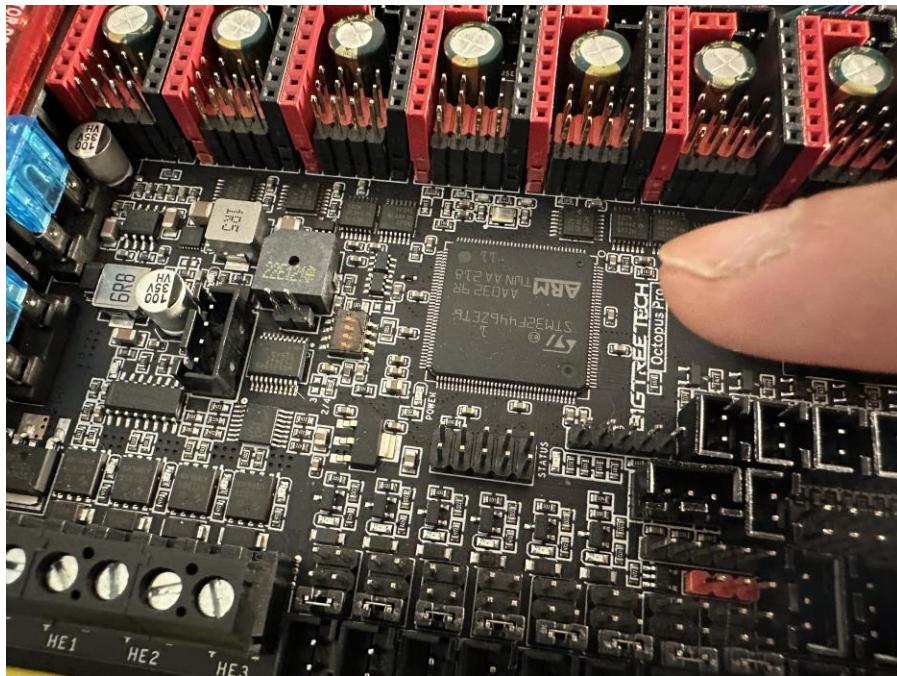
Type: 'cd *CanBoot*' and 'make menuconfig' to enter the CanBoot Configuration. Note that Linux is case sensitive and typing: 'canboot' will throw an error.

```
ca: lk@raspberrypi: ~/CanBoot
lk@raspberrypi:~ $ cd CanBoot
lk@raspberrypi:~/CanBoot $ make menuconfig
```

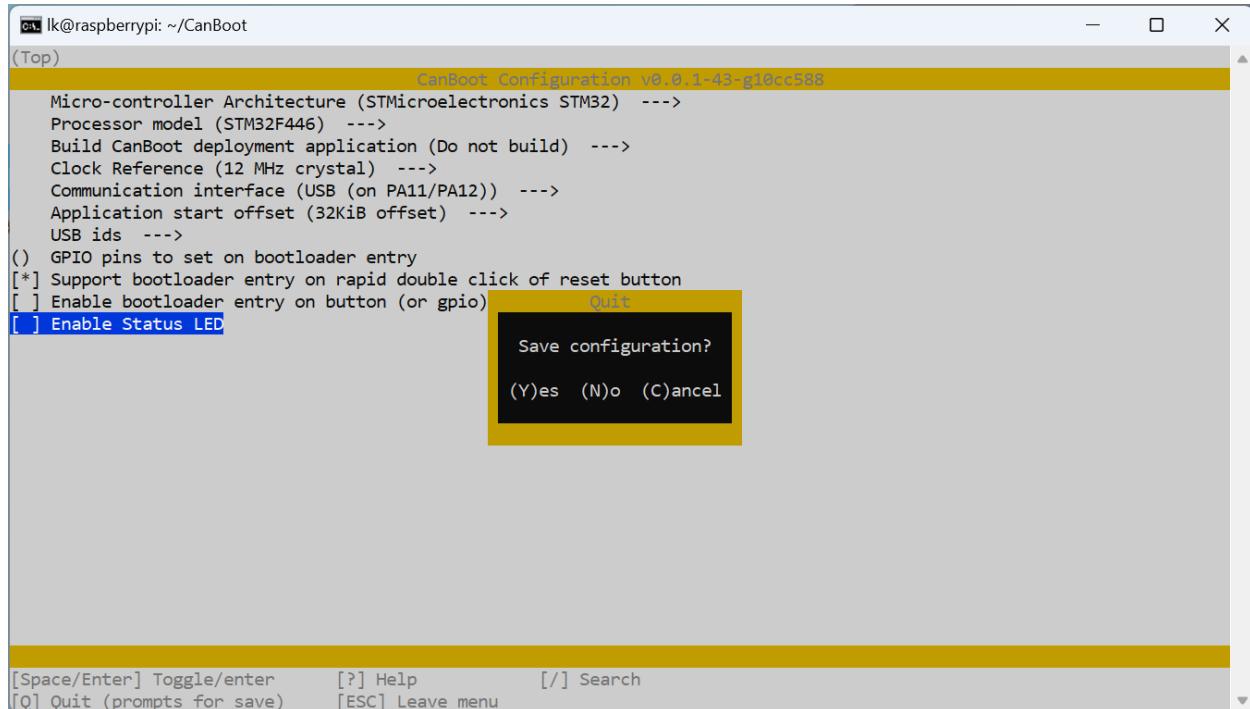
There are several lines to change. Micro-controller Architecture to 'STMicroelectronics STM32', Processor model to (controller chipset), and Clock Reference to '12 MHz crystal'. These are specifically for my Octopus Pro F446. Screenshot of my settings:



The processor model is labeled on the chip.



Type: 'q' then 'y' to save.



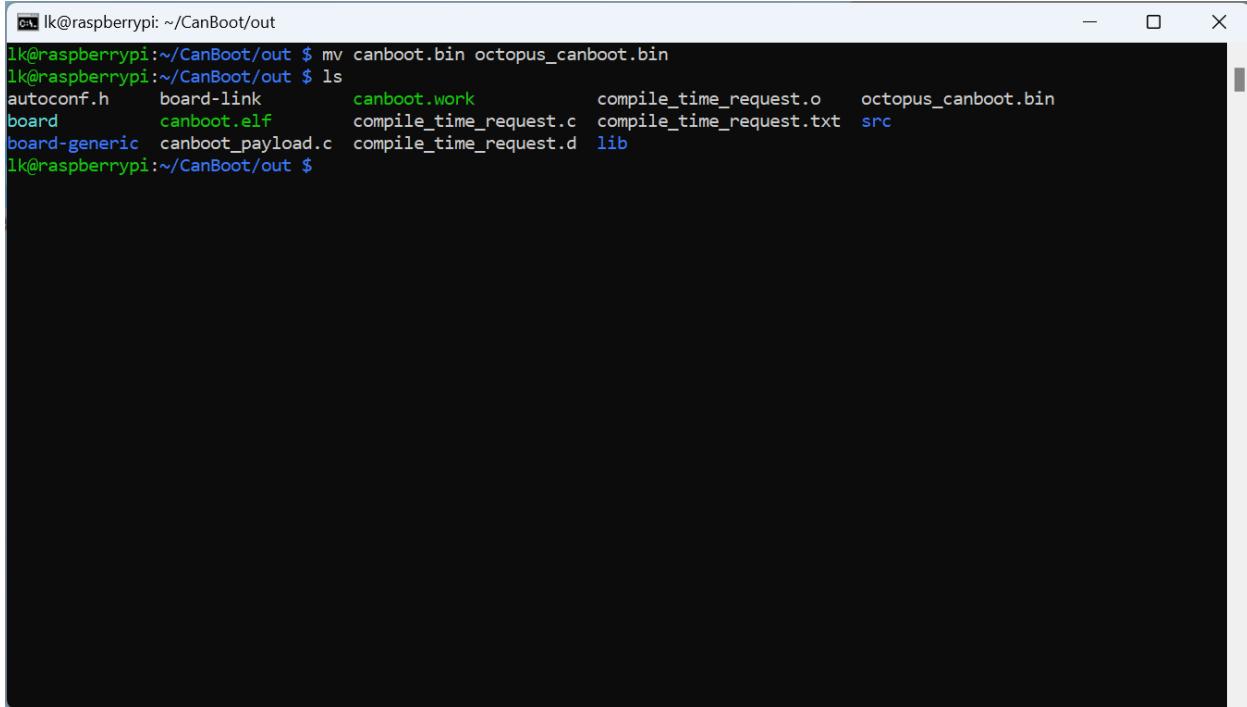
Type: 'make' which compiles the firmware into a subdirectory called 'out'.

```
lk@raspberrypi: ~/CanBoot
lk@raspberrypi:~/CanBoot $ make
Creating symbolic link out/board
Building out/autoconf.h
Compiling out/src/sched.o
Compiling out/src/bootentry.o
Compiling out/src/command.o
Compiling out/src/flashcmd.o
Compiling out/src/initial_pins.o
Compiling out/src/generic/armcm_canboot.o
Compiling out/src/stm32/gpio.o
Compiling out/src/stm32/flash.o
Compiling out/src/stm32/clockline.o
Compiling out/src/stm32/dfu_reboot.o
Compiling out/src/generic/armcm_irq.o
Compiling out/src/generic/crc16_ccitt.o
Compiling out/src/..../lib/stm32f4/system_stm32f4xx.o
Compiling out/src/stm32/stm32f4.o
Compiling out/src/generic/armcm_timer.o
Compiling out/src/stm32/gpioperiph.o
Compiling out/src/stm32/usbotg.o
Compiling out/src/stm32/chipid.o
Compiling out/src/generic/usb_cdc.o
Building out/compile_time_request.o
Preprocessing out/src/generic/armcm_link.ld
Linking out/canboot.elf
Creating bin file out/canboot.bin
lk@raspberrypi:~/CanBoot $ -
```

Type: 'cd out' which navigates to the subfolder and 'ls -l' lists the generated files. The 'canboot.bin' file is here and typing: 'pwd' lists the full path which can be used for file copying.

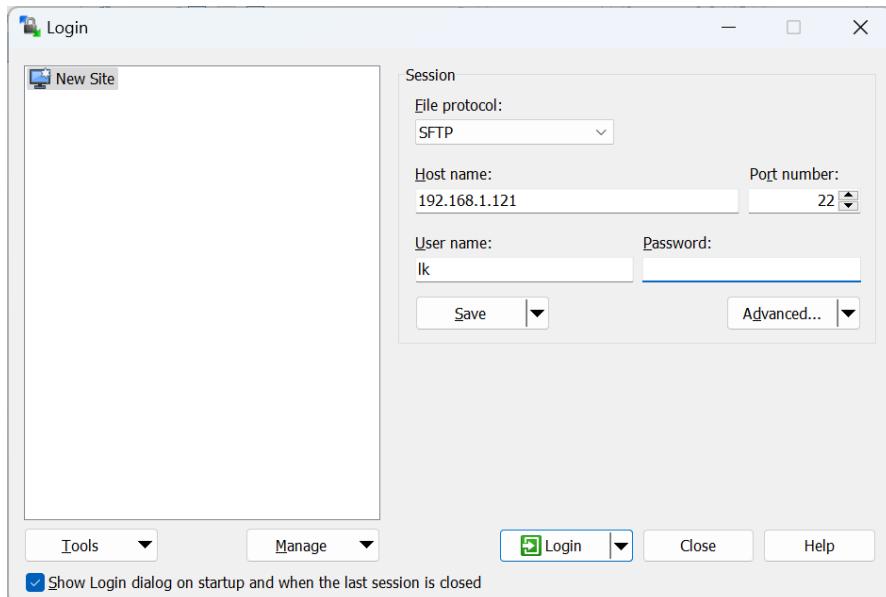
```
lk@raspberrypi: ~/CanBoot/out
lk@raspberrypi:~/CanBoot/out $ cd out
lk@raspberrypi:~/CanBoot/out $ ls -l
total 988
-rw-r--r-- 1 lk lk 6021 Apr 29 20:29 autoconf.h
lrwxrwxrwx 1 lk lk 26 Apr 29 20:29 board -> /home/lk/CanBoot/src/stm32
drwxr-xr-x 2 lk lk 4096 Apr 29 20:29 board-generic
-rw-r--r-- 1 lk lk 27 Apr 29 20:29 board-link
-rw-r--r-- 1 lk lk 4962 Apr 29 20:29 canboot.bin
-rwxr-xr-x 1 lk lk 919700 Apr 29 20:29 canboot.elf
-rw-r--r-- 1 lk lk 26838 Apr 29 20:29 canboot_payload.c
-rwxr-xr-x 1 lk lk 4962 Apr 29 20:29 canboot.work
-rw-r--r-- 1 lk lk 3040 Apr 29 20:29 compile_time_request.c
-rw-r--r-- 1 lk lk 208 Apr 29 20:29 compile_time_request.d
-rw-r--r-- 1 lk lk 67584 Apr 29 20:29 compile_time_request.o
-rw-r--r-- 1 lk lk 1124 Apr 29 20:29 compile_time_request.txt
drwxr-xr-x 5 lk lk 4096 Apr 29 20:29 lib
drwxr-xr-x 5 lk lk 4096 Apr 29 20:29 src
lk@raspberrypi:~/CanBoot/out $ pwd
/home/lk/CanBoot/out
lk@raspberrypi:~/CanBoot/out $ -
```

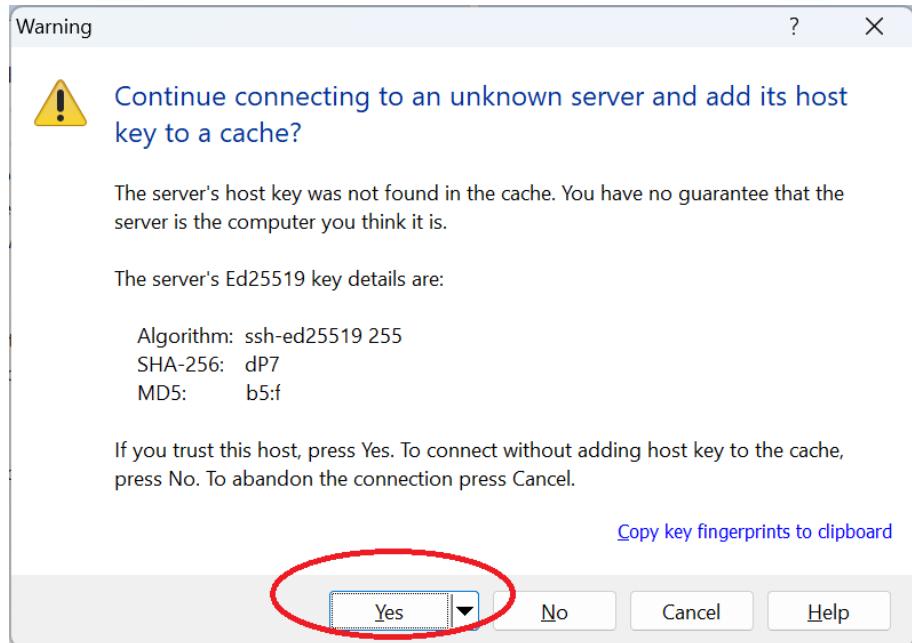
This entire directory gets overwritten once we re-run menuconfig, so rename 'canboot.bin' to 'octopus\_canboot.bin' or similar. Type: 'mv canboot.bin octopus\_canboot.bin' to rename the file.



```
lk@raspberrypi:~/CanBoot/out
lk@raspberrypi:~/CanBoot/out $ mv canboot.bin octopus_canboot.bin
lk@raspberrypi:~/CanBoot/out $ ls
autoconf.h      board-link          canboot.work        compile_time_request.o    octopus_canboot.bin
board          canboot.elf          compile_time_request.c  compile_time_request.txt  src
board-generic   canboot_payload.c  compile_time_request.d  lib
lk@raspberrypi:~/CanBoot/out $
```

From your Windows machine, download and run WinSCP. Connect using your Pi's IP address and simply move the 'octopus\_canboot.bin' file over to the Desktop. Reference the full file path from earlier to locate the .bin file.





Desktop - lk@192.168.1.121 - WinSCP

Local Mark Files Commands Session Options Remote Help

Synchronize Queue Transfer Settings Default

lk@192.168.1.121 × New Session

Desktop Upload Edit Properties New

Download Edit Properties New

C:\Users\LK\Desktop\ /home/lk/CanBoot/out/

Name	Type	Size	Changed	Rights	Owner
..	Parent directory		4/29/2023 3:37:27 PM	rw-r--r--	lk
12_software	File folder		4/29/2023 1:54:47 AM	rw-r--r--	lk
board	File	6 KB	4/29/2023 3:29:26 PM	rw-r--r--	lk
board-generic	File	1 KB	4/29/2023 3:29:26 PM	rw-r--r--	lk
lib	File	899 KB	4/29/2023 3:29:50 PM	rw-r--r--	lk
src	File	5 KB	4/29/2023 3:29:50 PM	rw-r--r--	lk
autoconf.h	File	27 KB	4/29/2023 3:29:50 PM	rw-r--r--	lk
board-link	File	3 KB	4/29/2023 3:29:44 PM	rw-r--r--	lk
canboot.elf	File	66 KB	4/29/2023 3:29:44 PM	rw-r--r--	lk
canboot.work	File	2 KB	4/29/2023 3:29:44 PM	rw-r--r--	lk
canboot_payload.c	File	5 KB	4/29/2023 3:29:44 PM	rw-r--r--	lk
compile_time_request.c	File	1 KB	4/29/2023 3:29:44 PM	rw-r--r--	lk
compile_time_request.d	File	2 KB	4/29/2023 3:29:44 PM	rw-r--r--	lk
compile_time_request.o	File	5 KB	4/29/2023 3:29:50 PM	rw-r--r--	lk
compile_time_request	File				
octopus_canboot.bin	File				

0 B of 2.94 MB in 0 of 13      3 hidden      4.84 KB of 0.98 MB in 1 of 14

Session status: SFTP-3      Last update: 0:00:40

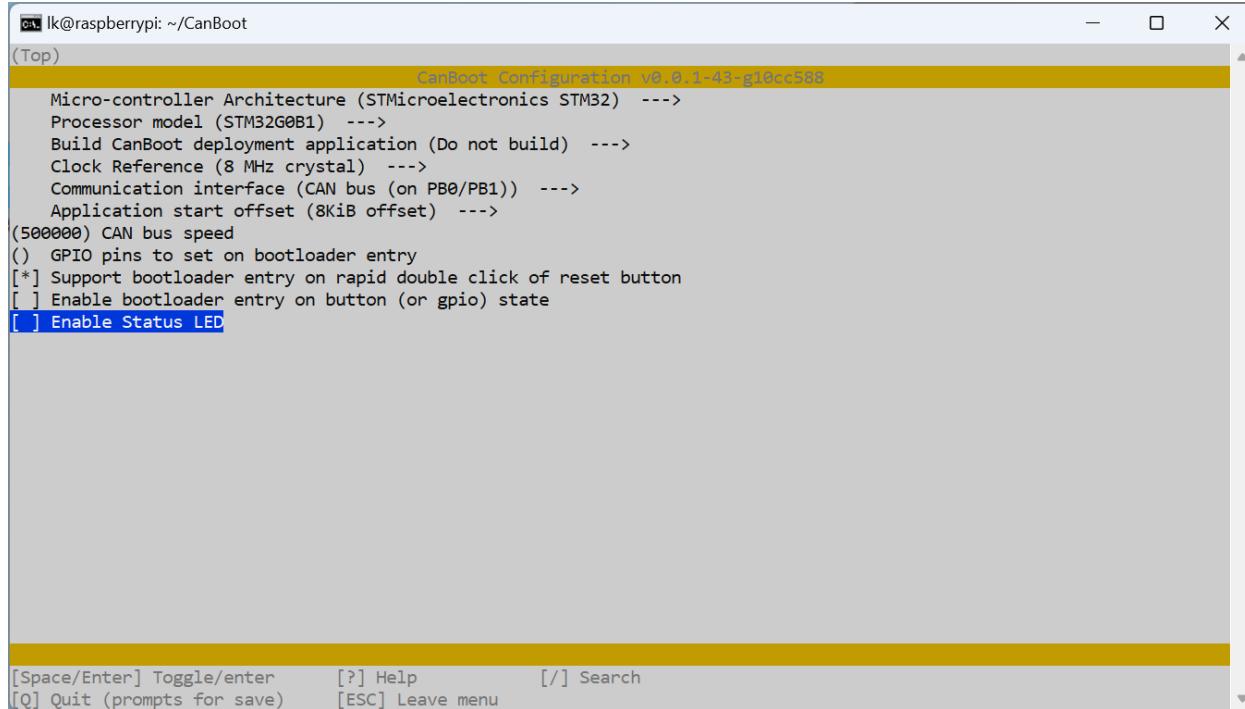
Back in CMD, make sure you are in the right directory. Type: 'cd ~/CanBoot'

```
lk@raspberrypi: ~/CanBoot
lk@raspberrypi:~/CanBoot/out $ cd ~/CanBoot
lk@raspberrypi:~/CanBoot $
```

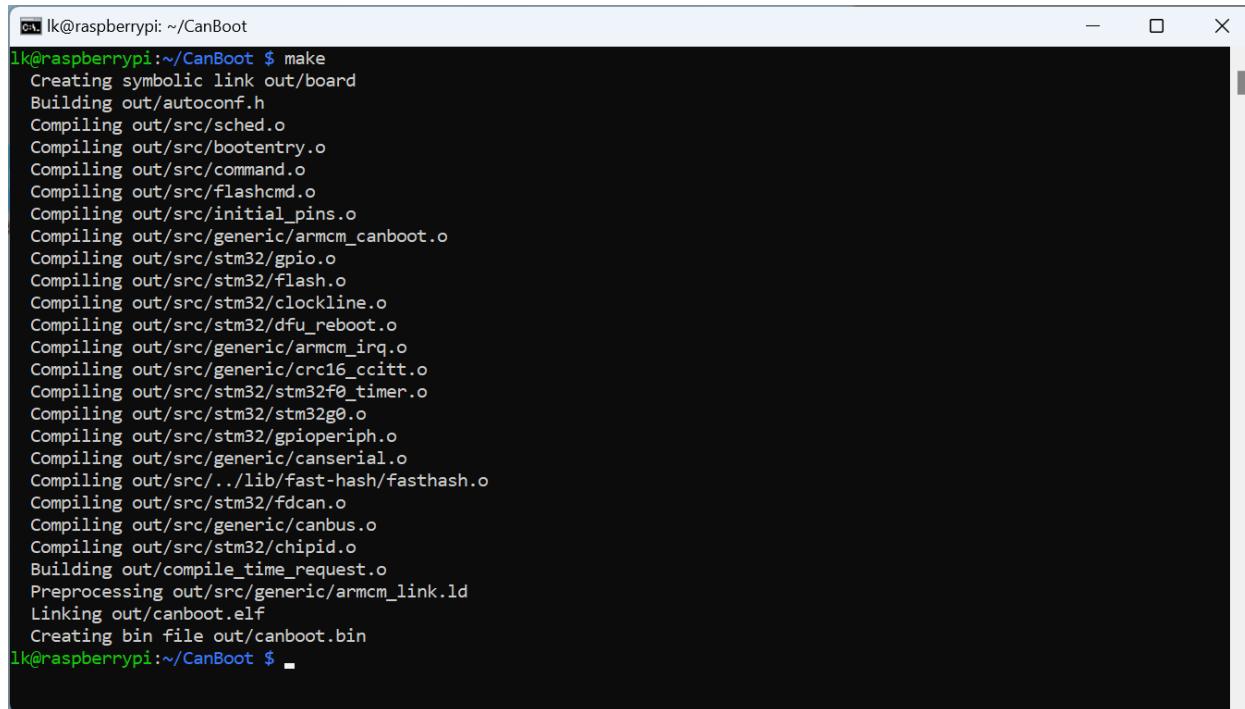
Type: '*make clean*' followed by '*make menuconfig*'

```
lk@raspberrypi: ~/CanBoot
lk@raspberrypi:~/CanBoot $ make clean
lk@raspberrypi:~/CanBoot $ make menuconfig
```

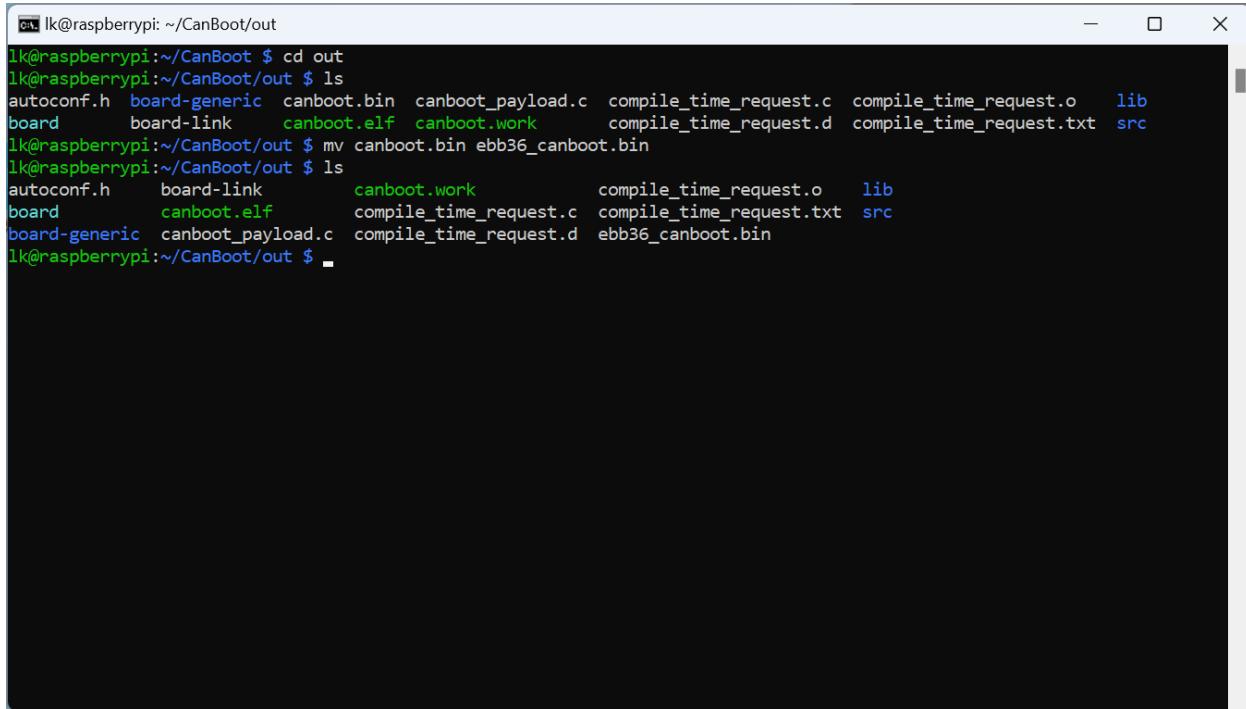
Repeat for the EBB42 v1.2 board. Note that I am using the default CAN bus speed of 500000. This number should not be set lower, especially if using ADXL input shaping. Some have had better luck increasing this to 1000000, which may resolve certain data communication issues.



Quit and save. Type: 'make' to generate a new firmware which saves to the same 'out' subdirectory.



Navigate to the subdirectory and rename this new 'canboot.bin' file to 'ebb42\_canboot.bin' or similar.  
Copy this file over to Windows Desktop using WinSCP.



```
lk@raspberrypi:~/CanBoot/out
lk@raspberrypi:~/CanBoot/out $ cd out
lk@raspberrypi:~/CanBoot/out $ ls
autoconf.h  board-generic  canboot.bin  canboot_payload.c  compile_time_request.c  compile_time_request.o  lib
board      board-link     canboot.elf   canboot.work    compile_time_request.d  compile_time_request.txt  src
lk@raspberrypi:~/CanBoot/out $ mv canboot.bin ebb36_canboot.bin
lk@raspberrypi:~/CanBoot/out $ ls
autoconf.h  board-link     canboot.work    compile_time_request.o  lib
board      canboot.elf   compile_time_request.c  compile_time_request.txt  src
board-generic  canboot_payload.c  compile_time_request.d  ebb36_canboot.bin
lk@raspberrypi:~/CanBoot/out $
```

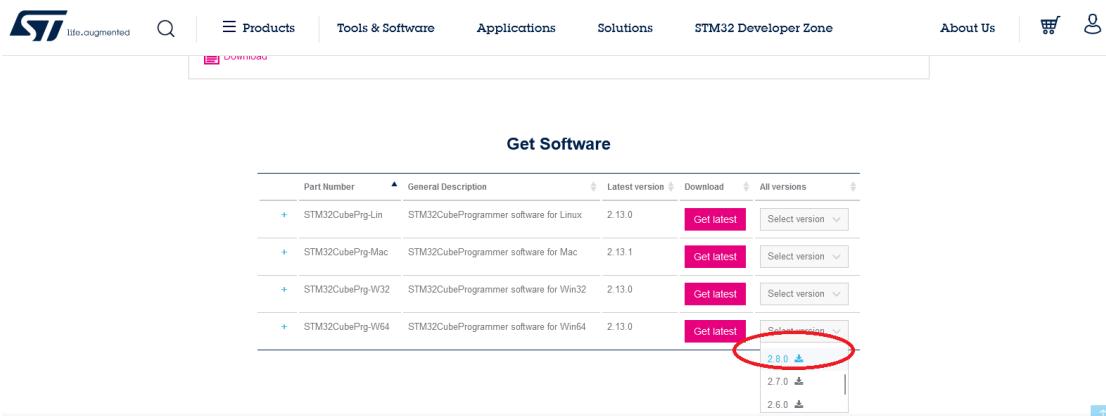
## FLASH CANBOOT FIRMWARE

If you did not read the disclaimer at the beginning, please do so before proceeding.

Download STM32CubeProgrammer:

<https://www.st.com/en/development-tools/stm32cubeprog.html#get-software>

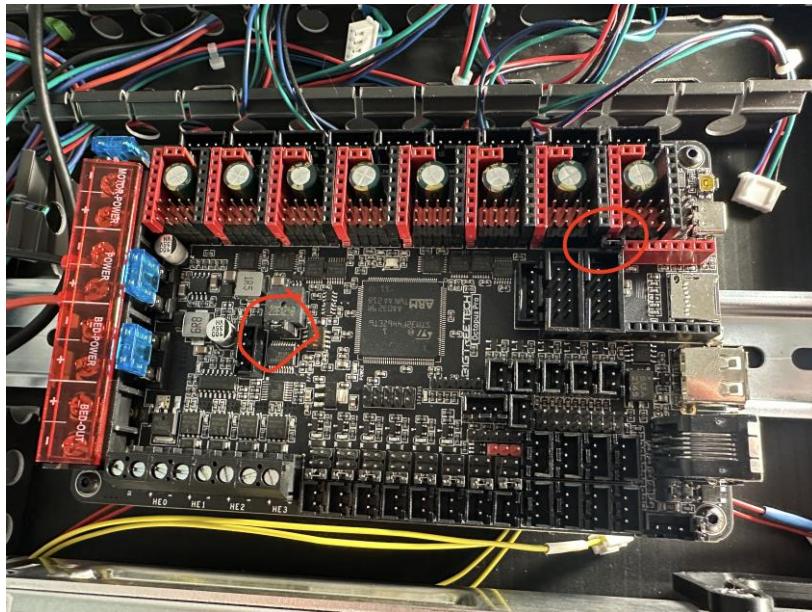
At this time, version 2.13.0 will *not* work and must be downgraded for the firmware flash to work. I am using version 2.8.0. Accept all installation defaults and dependency packages (not shown).



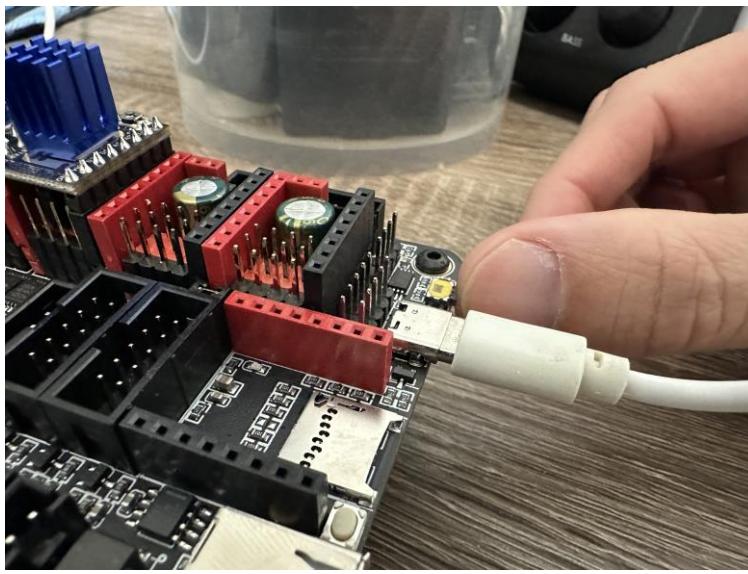
The screenshot shows the STM32 developer zone website with the 'Get Software' section for the STM32CubeProgrammer. The table lists four versions: Linux (2.13.0), Mac (2.13.1), Win32 (2.13.0), and Win64 (2.13.0). For each row, there is a 'Get latest' button and a 'Select version' dropdown. The dropdown for the Win64 row is circled in red, showing options 2.8.0, 2.7.0, and 2.6.0.

Part Number	General Description	Latest version	Download	All versions
+ STM32CubePrg-Lin	STM32CubeProgrammer software for Linux	2.13.0	<button>Get latest</button>	<button>Select version</button>
+ STM32CubePrg-Mac	STM32CubeProgrammer software for Mac	2.13.1	<button>Get latest</button>	<button>Select version</button>
+ STM32CubePrg-W32	STM32CubeProgrammer software for Win32	2.13.0	<button>Get latest</button>	<button>Select version</button>
+ STM32CubePrg-W64	STM32CubeProgrammer software for Win64	2.13.0	<button>Get latest</button>	<button>Select version</button>

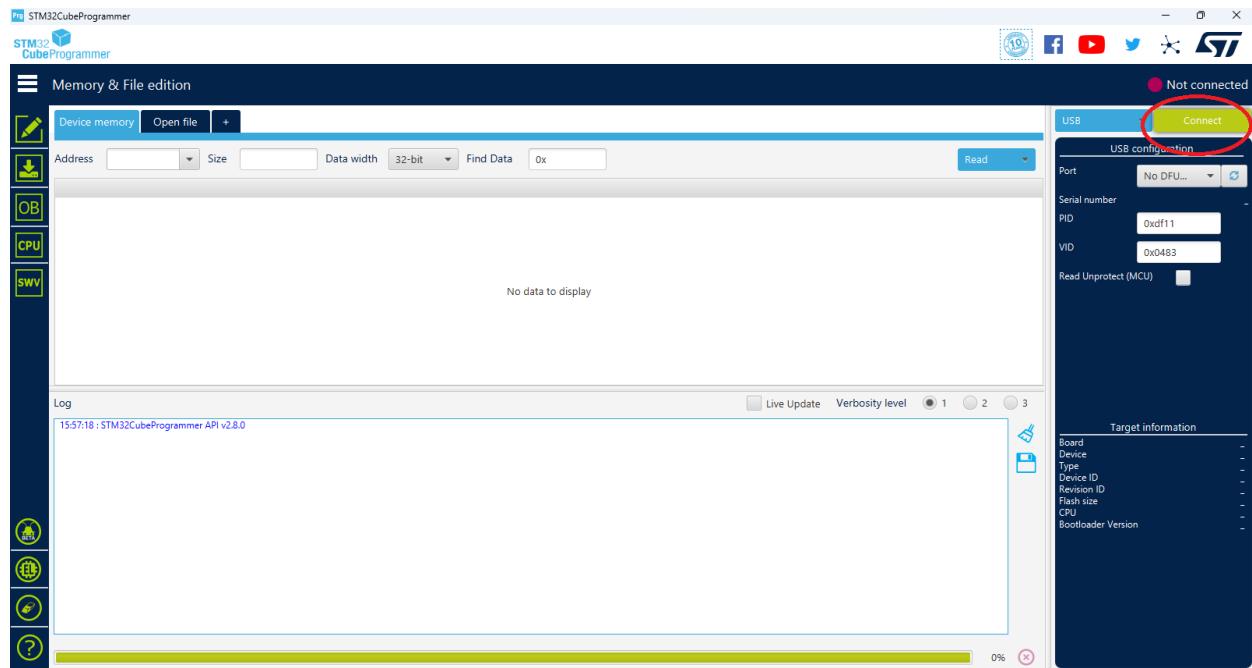
Unless the Octopus is powered by the 24V PSU, you need jumpers installed on BOTH the ‘USB-C Power’ AND ‘Boot0’ pins on the board. Otherwise, a USB A to C cable connected from the Windows computer is not powerful enough. This is contrary to many guides stating to remove the ‘USB-C Power’ jumper. Remove all other jumpers and motor drivers if you have difficulty powering the board.



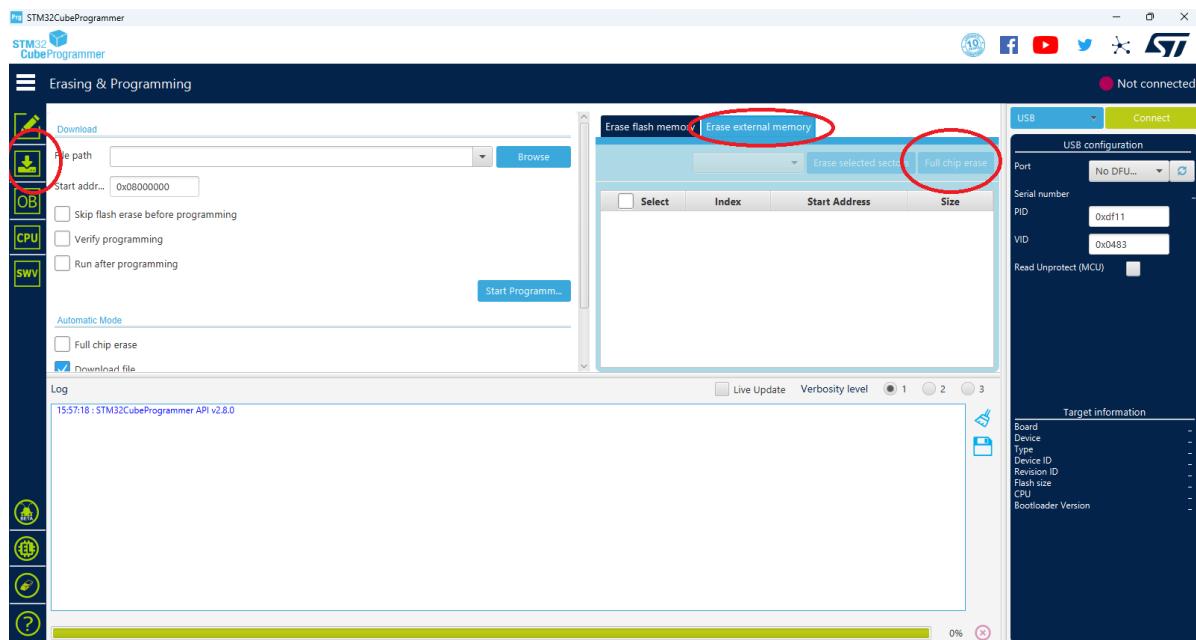
Press and hold the reset button for at least five seconds to set the Octopus to DFU mode.



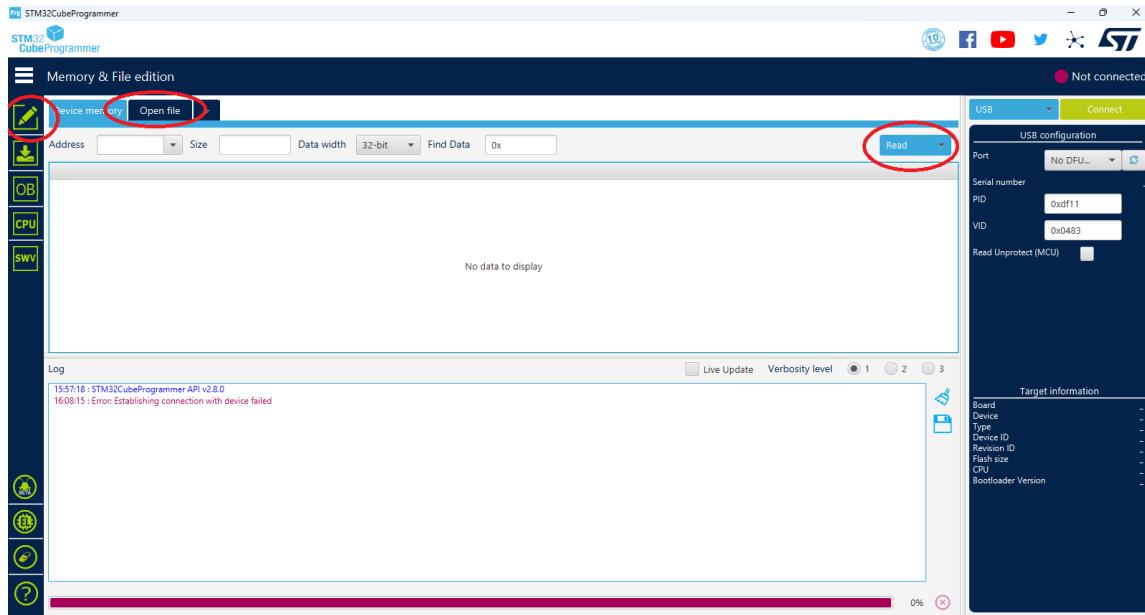
Start the STM32CubeProgrammer software. The following steps do not work if the software is started before setting the board in DFU mode. Click ‘Connect’. Status should change from ‘Not connected’ to ‘Connected’ which reads the memory contents.



Erase by clicking the icon on the left, clicking 'Erase external memory' and then 'Full chip erase'. I could not grab proper screenshots, so you will see much more information. It will prompt to 'erase full chip flash memory' and once finished, notify that 'Mass erase command correctly executed'. Be aware of error messages that may indicate problems with the process or a botched attempt. You may need to repeat this several times.



Once successfully erased, click the write icon on the left and click ‘Open File’. Choose the octopus\_canboot.bin file from earlier, and click ‘Download’. Again, my screenshot is slightly off since I wrote this documentation after the fact 😊



If everything goes well, you should see success messages. If this is the case, click the ‘Disconnect’ button and unplug the board from computer.

Log

```

01:51:03 : File      : OctoPus-F446-bootloader-32KB.hex
01:51:03 : Size      : 32464 Bytes
01:51:03 : Address   : 0x8000000
01:51:03 : Erasing memory corresponding to segment 0:
01:51:03 : Erasing internal memory sectors [0 1]
01:51:04 : erasing sector 0000 @: 0x08000000 done
01:51:04 : erasing sector 0001 @: 0x08004000 done
01:51:04 : Download in Progress:
01:51:05 : File download complete
01:51:05 : Time elapsed during download operation: 00:00:01.362
01:52:18 : MASS ERASE ...
01:52:25 : Mass erase command correctly executed. Note: if there's any flash protection, it will not be erased.
01:52:25 : UPLOADING ...
01:52:25 : Size      : 1024 Bytes
01:52:25 : Address   : 0x8000000
01:52:25 : Read progress:
01:52:25 : Data read successfully
01:52:25 : Time elapsed during the read operation is: 00:00:00.003
01:52:48 : Read File: C:\Users\LK\Desktop\octopus_canboot.bin
01:52:48 : Number of segments: 1
01:52:48 : segment[0]: address= 0x0, size= 0x1362
01:52:51 : Memory Programming ...
01:52:51 : Opening and parsing file: octopus_canboot.bin
01:52:51 : File      : octopus_canboot.bin
01:52:51 : Size      : 4962 Bytes
01:52:51 : Address   : 0x8000000
01:52:51 : Erasing memory corresponding to segment 0:
01:52:51 : Erasing internal memory sector 0
01:52:51 : erasing sector 0000 @: 0x08000000 done
01:52:51 : Download in Progress:
01:52:51 : File download complete
01:52:51 : Time elapsed during download operation: 00:00:00.461
01:53:26 : Disconnected from device.

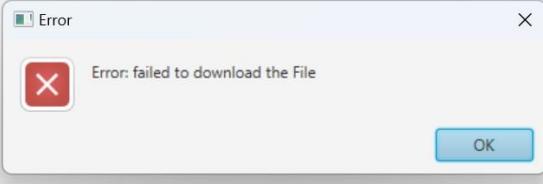
```

Be aware of errors such as this. You may have to re-attempt or investigate why it is happening. In this scenario, it was due to the buggy 2.13 version of software, and the reason I rolled back to 2.8

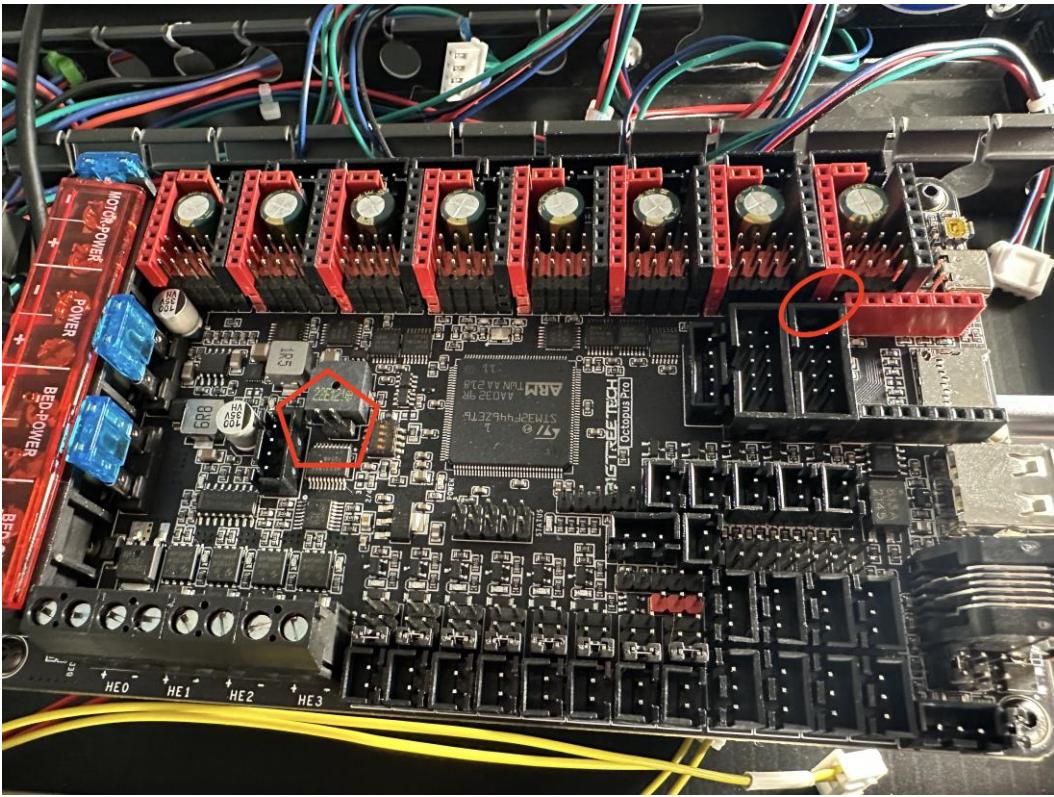
```

23:07:02 : Bank      : 0x01
23:07:02 : Address   : 0x1fff7870
23:07:02 : Size      : 4 Bytes
23:07:02 : UPLOADING ...
23:07:02 : Size      : 1024 Bytes
23:07:02 : Address   : 0x8000000
23:07:02 : Read progress:
23:07:02 : Data read successfully
23:07:02 : Time elapsed during the read operation is: 00:00:00.005
23:07:12 : Read File: C:\Users\LK\Desktop\ebb_canboot.bin
23:07:12 : Number of segments: 1
23:07:12 : segment[0]: address= 0x0, size= 0x1094
23:07:13 : Memory Programming ...
23:07:13 : Opening and parsing file: ebb_canboot.bin
23:07:13 : File      : ebb_canboot.bin
23:07:13 : Size      : 4.14 KB
23:07:13 : Address   : 0x8000000
23:07:13 : Erasing memory corresponding to segment 0:
23:07:13 : Not flash Memory : No erase done
23:07:13 : File segment @0x08000000 is not 255-bytes aligned. It will be aligned to @0x07FFFFF8
23:07:13 : Download in Progress:
23:07:14 : Error: failed to download Segment[0]
23:07:14 : Error: failed to download the File

```



**REMOVE BOTH JUMPERS FROM 5V USB C AND BOOT0 PINS ON THE MCU**



Repeat for the EBB42 board. Start by placing a jumper on the VBUS (near usb-c socket) header pins and plug in the same USB cable for power. Press and hold BOOT0, press and release RESET, release BOOT0 to enter DFU mode.

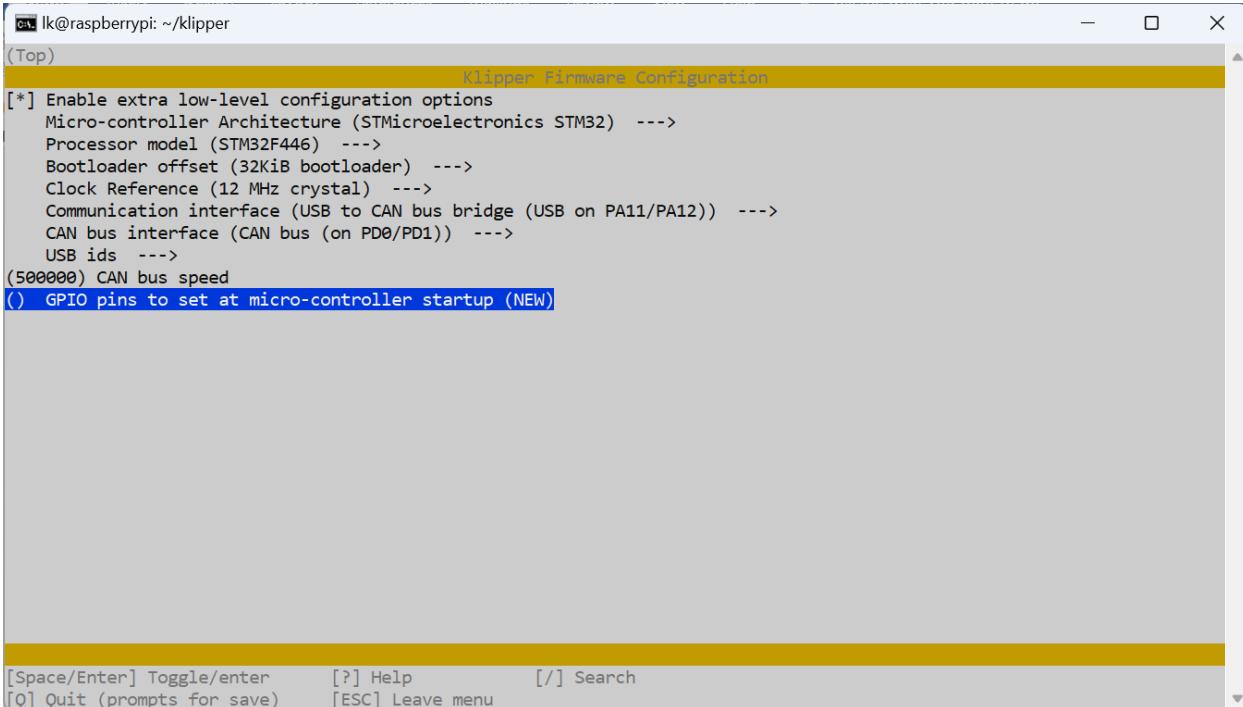
Start STM32CubeProgrammer as before and follow the exact same steps of erasing and re-writing the memory, only this time using the ebb42\_canboot.bin file. Once completed, remove the jumper from VBUS and place it on the 120R pins.

#### **SET UP KLIPPER FIRMWARE**

Return to CMD and navigate to the Klipper directory, type: 'cd ~/klipper' and 'make menuconfig'

```
lk@raspberrypi: ~/klipper
lk@raspberrypi:~ $ cd ~/klipper
lk@raspberrypi:~/klipper $ make menuconfig
```

Here are the configuration settings for the F446 Octopus Pro. Note the CAN bus speed must match what was specified earlier. Some other settings are copied over from earlier.



Type: 'make' to compile the Klipper firmware

```
lk@raspberrypi: ~/klipper
lk@raspberrypi:~/klipper $ make
Building out/autoconf.h
Compiling out/src/sched.o
Compiling out/src/command.o
Compiling out/src/basecmd.o
Compiling out/src/debugcmds.o
Compiling out/src/initial_pins.o
Compiling out/src/gpiocmds.o
Compiling out/src/stepper.o
Compiling out/src/endstop.o
Compiling out/src/trsync.o
Compiling out/src/adccmds.o
Compiling out/src/spicmds.o
Compiling out/src/thermocouple.o
Compiling out/src/sdioscmds.o
Compiling out/src/i2ccmds.o
Compiling out/src/pwmcmds.o
Compiling out/src/spi_software.o
Compiling out/src/sensor_adxl345.o
Compiling out/src/sensor_angle.o
Compiling out/src/sensor_mpu9250.o
Compiling out/src/lcd_st7920.o
Compiling out/src/lcd_hd44780.o
Compiling out/src/buttons.o
Compiling out/src/tmcuart.o
Compiling out/src/neopixel.o
Compiling out/src/pulse_counter.o
Compiling out/src/stm32/watchdog.o
Compiling out/src/stm32/gpio.o
Compiling out/src/stm32/clockline.o
```

As before, rename the Klipper file to prevent overwriting in the next step. Type: 'cd out' followed by 'mv *klipper.bin* *octopus\_klipper.bin*'. Move the file by typing: 'mv *octopus\_klipper.bin* ~\klipper'

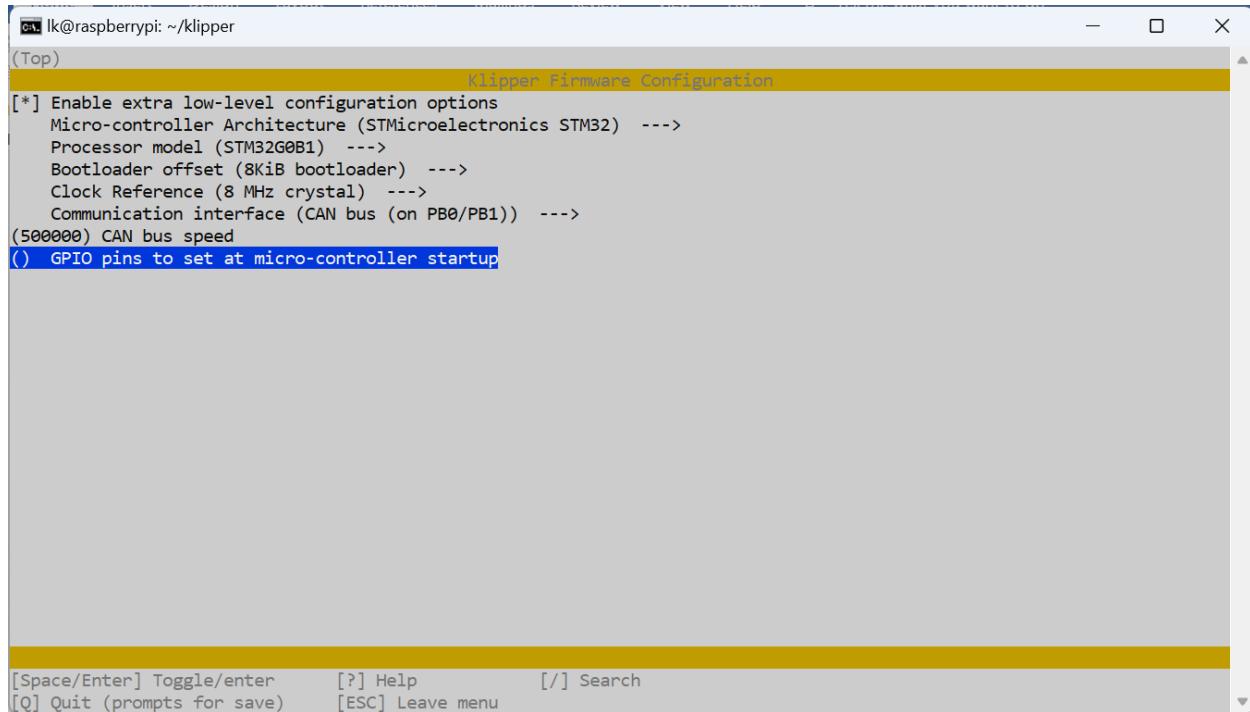
```
lk@raspberrypi: ~/klipper/out
lk@raspberrypi:~/klipper $ cd out
lk@raspberrypi:~/klipper/out $ ls
autoconf.h  board-generic  compile_time_request.c  compile_time_request.o  klipper.bin  klipper.elf  src
board      board-link     compile_time_request.d  compile_time_request.txt  klipper.dict  lib
lk@raspberrypi:~/klipper/out $ mv klipper.bin octopus_klipper.bin
lk@raspberrypi:~/klipper/out $ mv octopus_klipper.bin ~\klipper
```

Type: 'cd ..' and 'ls -l' to make sure renamed file exists in the parent Klipper directory.

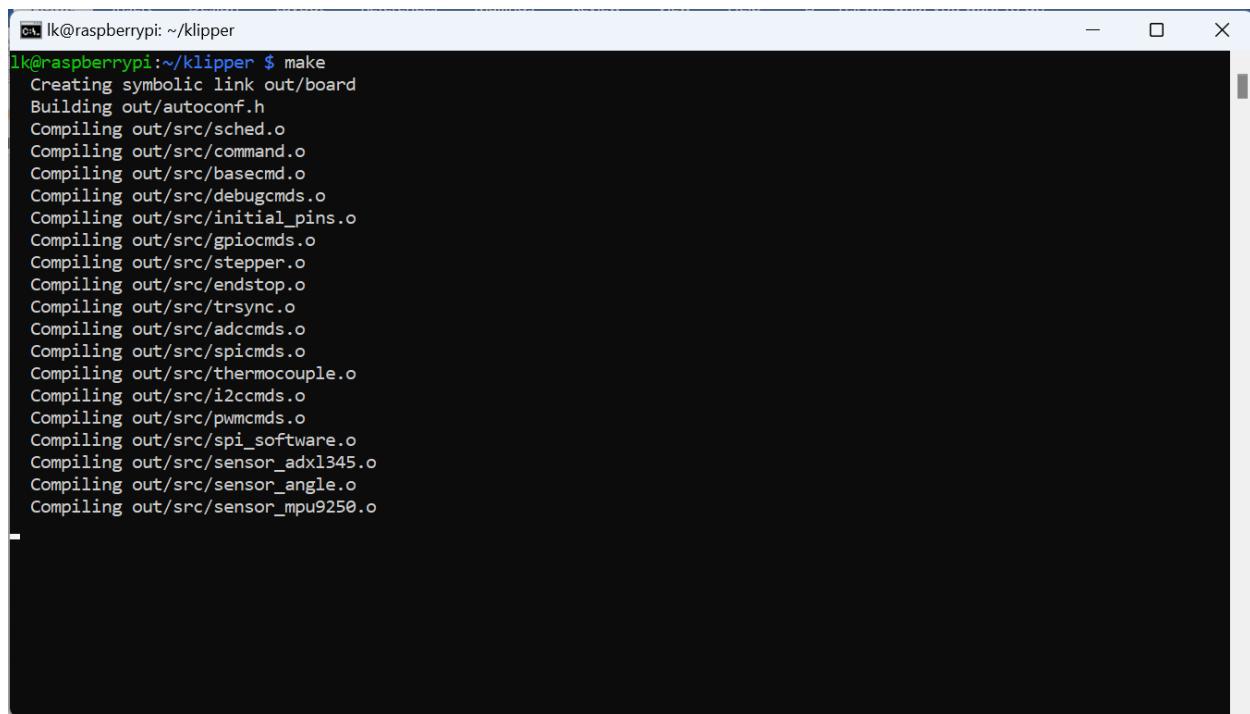
```
lk@raspberrypi: ~/klipper
lk@raspberrypi:~/klipper $ cd ..
lk@raspberrypi:~/klipper $ ls -l
total 152
drwxr-xr-x  2 lk lk 16384 Apr 28 21:30 config
-rw-r--r--  1 lk lk 35149 Apr 28 21:30 COPYING
drwxr-xr-x  5 lk lk 4096 Apr 28 21:30 docs
-rw-r--r--  1 lk lk 30092 Apr 29 21:49 klipperbinextra delete
drwxr-xr-x  6 lk lk 4096 Apr 28 21:35 klippy
drwxr-xr-x 34 lk lk 4096 Apr 28 21:30 lib
-rw-r--r--  1 lk lk 3801 Apr 28 21:30 Makefile
-rwxr-xr-x  1 lk lk 30092 Apr 29 21:57 octopus_klipper.bin
drwxr-xr-x  5 lk lk 4096 Apr 29 22:00 out
-rw-r--r--  1 lk lk 694 Apr 28 21:30 README.md
drwxr-xr-x  4 lk lk 4096 Apr 28 21:30 scripts
drwxr-xr-x 14 lk lk 4096 Apr 28 21:30 src
drwxr-xr-x  4 lk lk 4096 Apr 28 21:30 test
lk@raspberrypi:~/klipper $
```

Type: '*make clean*' and '*make menuconfig*' to repeat for the EBB36 board

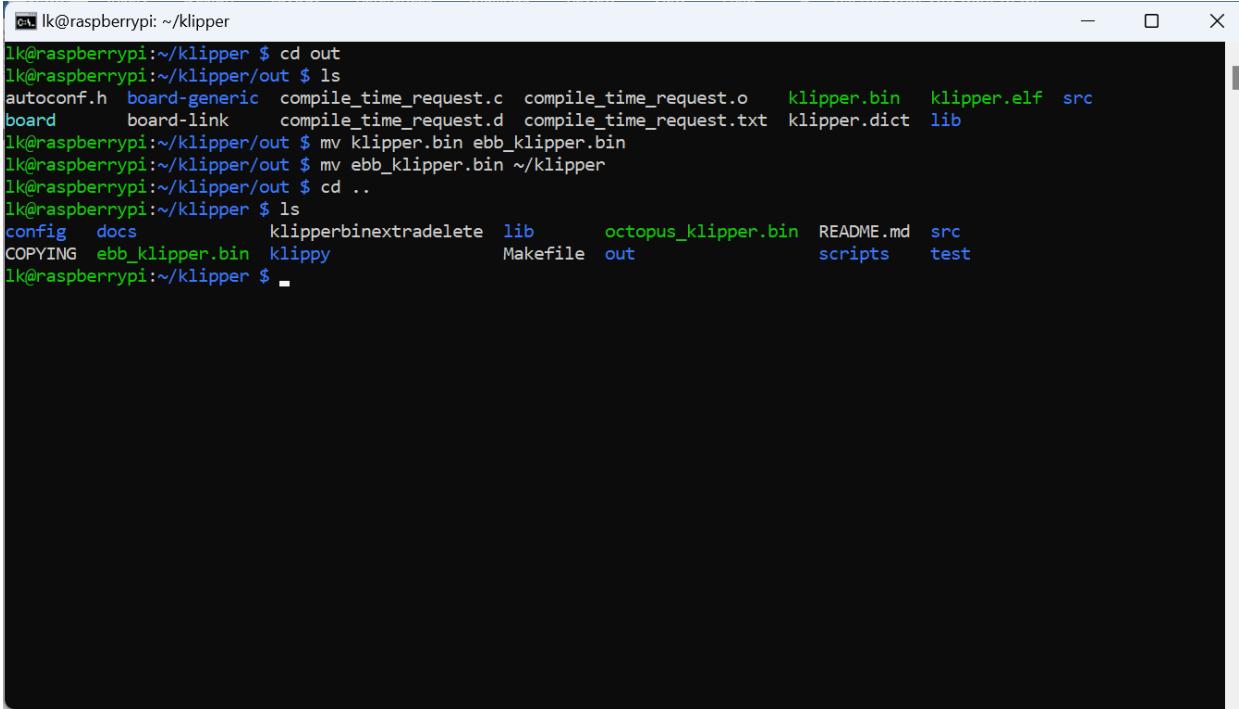
```
lk@raspberrypi: ~/klipper
lk@raspberrypi:~/klipper $ make clean
lk@raspberrypi:~/klipper $ make menuconfig
```



Quit, Save, and type: '*make*' to generate another firmware for the EBB42 board.

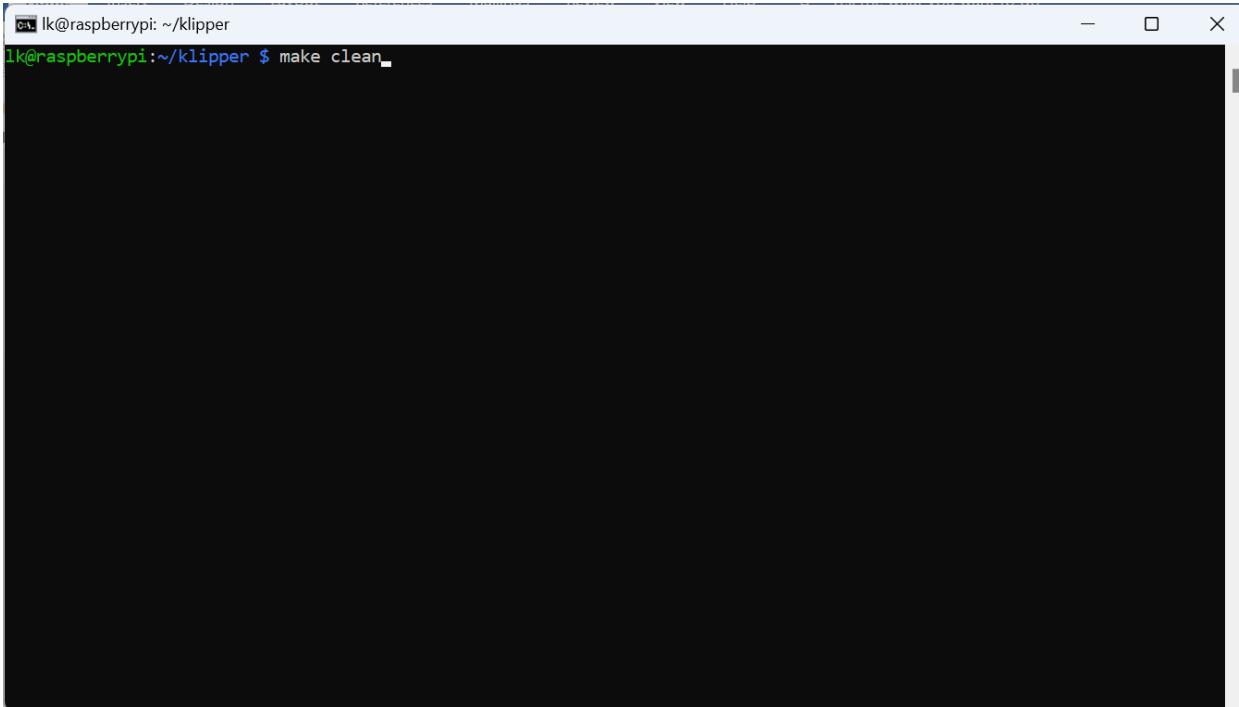


Repeat previous steps to rename and move ebb\_klipper .bin file to parent Klipper directory.



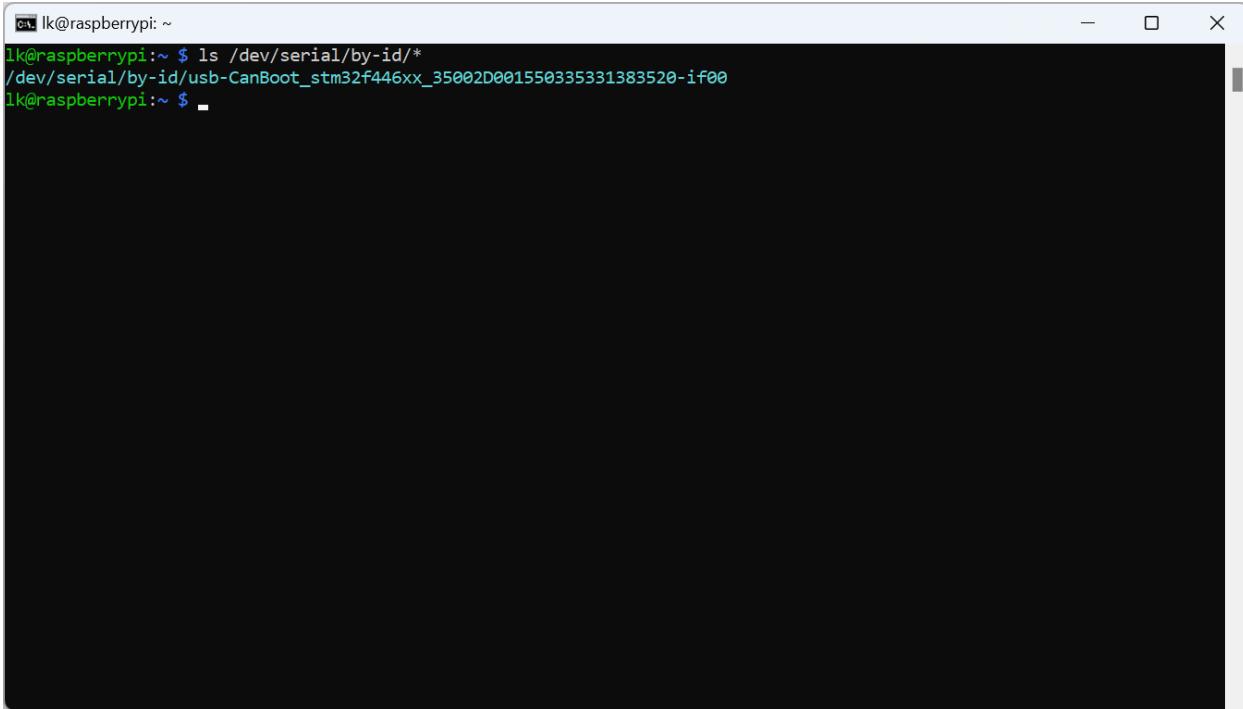
```
lk@raspberrypi:~/klipper $ cd out
lk@raspberrypi:~/klipper/out $ ls
autoconf.h  board-generic  compile_time_request.c  compile_time_request.o  klipper.bin  klipper.elf  src
board      board-link     compile_time_request.d  compile_time_request.txt  klipper.dict  lib
lk@raspberrypi:~/klipper/out $ mv klipper.bin ebb_klipper.bin
lk@raspberrypi:~/klipper/out $ mv ebb_klipper.bin ~/klipper
lk@raspberrypi:~/klipper/out $ cd ..
lk@raspberrypi:~/klipper $ ls
config  docs          klipperbinextra delete  lib      octopus_klipper.bin  README.md  src
COPYING  ebb_klipper.bin  klippy           Makefile  out                  scripts  test
lk@raspberrypi:~/klipper $ -
```

Type: 'make clean'



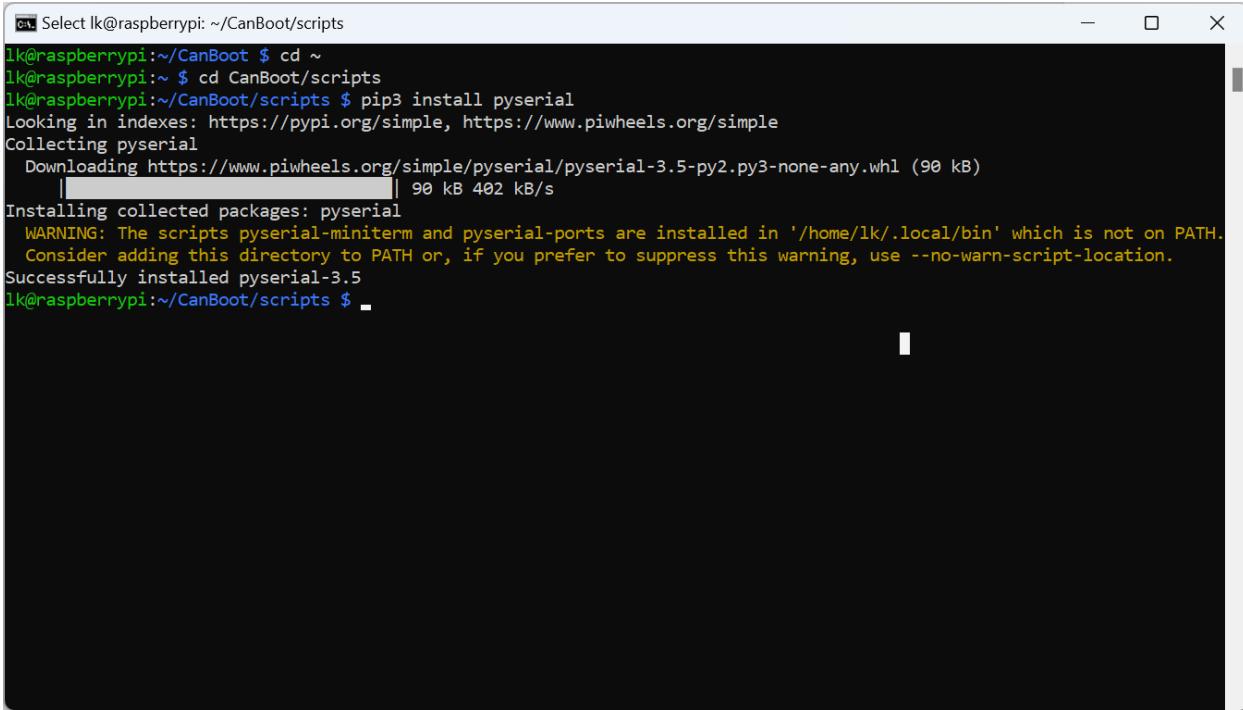
```
lk@raspberrypi:~/klipper $ make clean
```

Plug in Octopus to Pi via USB A to C cable and power printer on. Use a proper cable with data capability, some USB cables only provide power. From CMD, type: 'ls /dev/serial/by-id/\*' which returns the serial device ID of the Octopus board. Copy this down.



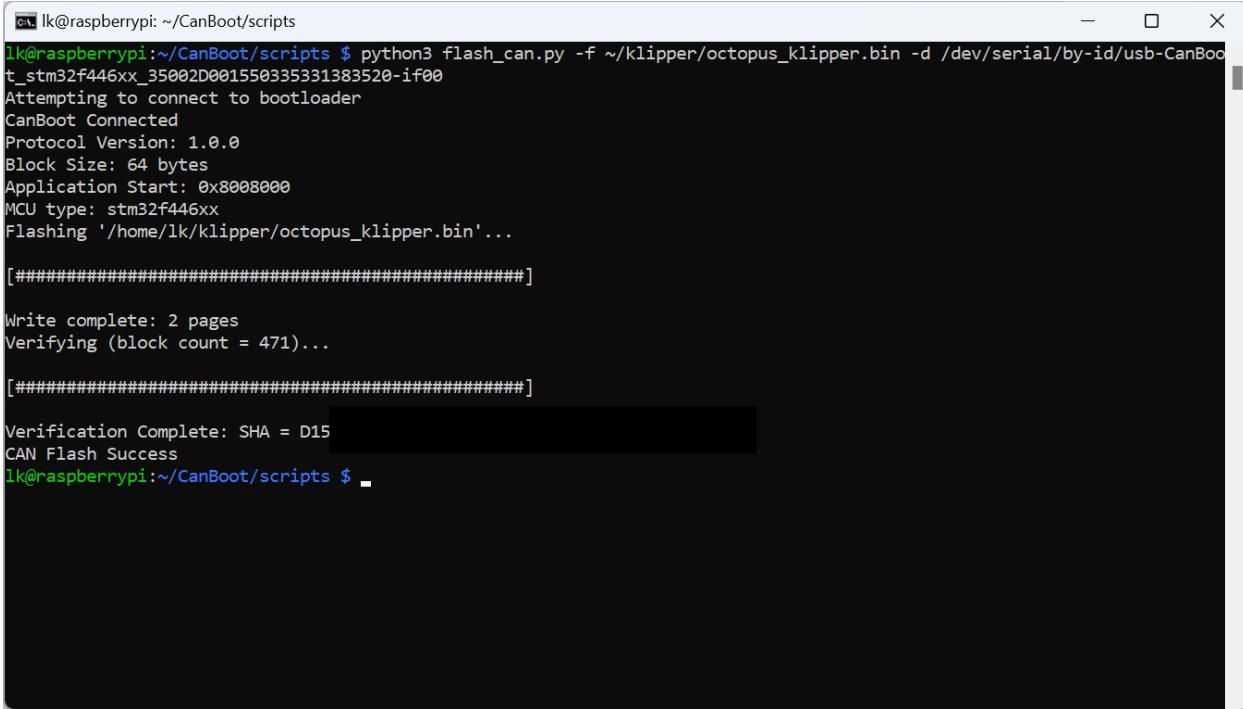
```
lk@raspberrypi: ~
lk@raspberrypi:~ $ ls /dev/serial/by-id/*
/dev/serial/by-id/usb-CanBoot_stm32f446xx_35002D001550335331383520-if00
lk@raspberrypi:~ $
```

Type: 'cd *CanBoot/scripts*' then 'pip3 install pyserial'



```
Select lk@raspberrypi: ~/CanBoot/scripts
lk@raspberrypi:~/CanBoot $ cd ~
lk@raspberrypi:~ $ cd CanBoot/scripts
lk@raspberrypi:~/CanBoot/scripts $ pip3 install pyserial
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting pyserial
  Downloading https://www.piwheels.org/simple/pyserial/pyserial-3.5-py2.py3-none-any.whl (90 kB)
    |██████████| 90 kB 402 kB/s
Installing collected packages: pyserial
  WARNING: The scripts pyserial-miniterm and pyserial-ports are installed in '/home/lk/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pyserial-3.5
lk@raspberrypi:~/CanBoot/scripts $
```

Type: '*python3 flash\_can.py -f ~/octopus\_klipper.bin -d (serial number)*' to flash Octopus over serial.



```
lk@raspberrypi:~/CanBoot/scripts
lk@raspberrypi:~/CanBoot/scripts $ python3 flash_can.py -f ~/klipper/octopus_klipper.bin -d /dev/serial/by-id/usb-CanBoo
t_stm32f446xx_35002D001550335331383520-if00
Attempting to connect to bootloader
CanBoot Connected
Protocol Version: 1.0.0
Block Size: 64 bytes
Application Start: 0x8008000
MCU type: stm32f446xx
Flashing '/home/lk/klipper/octopus_klipper.bin'...
[#####
Write complete: 2 pages
Verifying (block count = 471)...
[#####
Verification Complete: SHA = D15
CAN Flash Success
lk@raspberrypi:~/CanBoot/scripts $
```

Type: '*cd*' then '*sudo nano /etc/network/interfaces.d/can0*'

Paste the following, once again noting that bitrate matches the speed specified earlier if y.

allow-hotplug can0

iface can0 can static

bitrate 500000

up ifconfig \$IFACE txqueuelen 128

```
lk@raspberrypi: ~
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   M-U Undo   M-A Set Mark
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify   ^_ Go To Line M-E Redo   M-6 Copy
```

Press CTRL X to save and quit.

From CMD, type: 'sudo reboot'

Connect octopus with Ebb42 via RJ11 cable, you can power the ebb42 via USB or via 24v pins.

From CMD, type: 'ifconfig'. Look for the can0 entry.

```
Select lk@raspberrypi: ~
permitted by applicable law.
Last login: Sat Apr 29 22:39:38 2023 from [REDACTED]
lk@raspberrypi:~ $ 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
    [REDACTED] 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 10 bytes 1600 (1.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 1600 (1.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Type: ‘cd CanBoot/scripts’ and ‘python3 flash\_can.py -i can0 -q’ which now lists the UUID for both the Octopus and EBB42. The Klipper label references Octopus, and CanBoot references the EBB42. Copy both down.

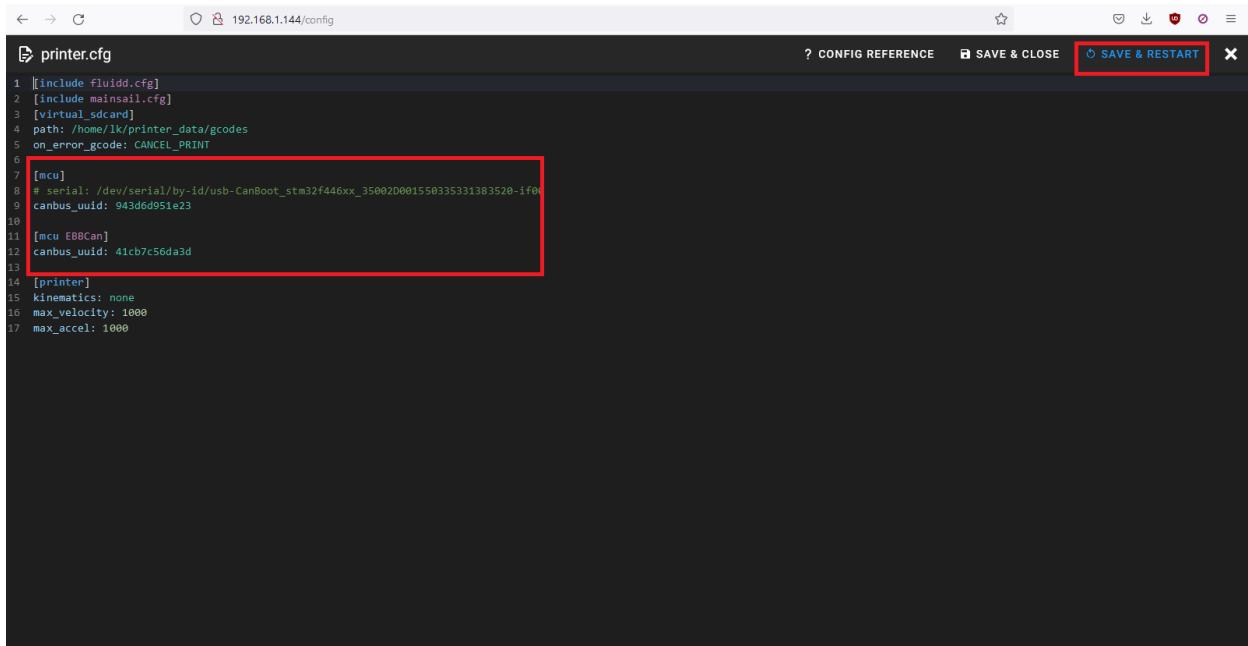
```
lk@raspberrypi:~$ cd CanBoot/scripts  
lk@raspberrypi:~/CanBoot/scripts $ python3 flash_can.py -i can0 -q  
Resetting all bootloader node IDs...  
Checking for canboot nodes...  
Detected UUID: 943d6d951e23, Application: Klipper  
Detected UUID: 41cb7c56da3d, Application: CanBoot  
Query Complete  
lk@raspberrypi:~/CanBoot/scripts $
```

Type: `'python3 flash_can.py -f ~/klipper/ebb_klipper.bin -u (ebbID canboot)'` to flash Klipper to the EBB36 board via CAN.

```
lk@raspberrypi: ~/CanBoot/scripts
Resetting all bootloader node IDs...
Checking for canboot nodes...
Detected UUID: 943d6d951e23, Application: Klipper
Detected UUID: 41cb7c56da3d, Application: CanBoot
Query Complete
lk@raspberrypi:~/CanBoot/scripts $ python3 flash_can.py -f ~/klipper/ebb_klipper.bin -u 41cb7c56da3d
Sending bootloader jump command...
Resetting all bootloader node IDs...
Checking for canboot nodes...
Detected UUID: 943d6d951e23, Application: Klipper
Detected UUID: 41cb7c56da3d, 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/lk/klipper/ebb_klipper.bin'...
[########################################]
Write complete: 14 pages
Verifying (block count = 418)...

[########################################]
Verification Complete: SHA = [REDACTED]
CAN Flash Success
lk@raspberrypi:~/CanBoot/scripts $
```

Load Mainsail, which may still show errors. Add both UUID from previous step into the printer.cfg for each cfg file where you find “canbus\_uuid:”. Click ‘Save and Restart’. You may also need to do a firmware restart.



Click ‘Machine’, check that both CAN UUIDs are present and you are done! As with non-CAN installs, time to configure printer.cfg and continue with set up/tuning. Best of luck!

