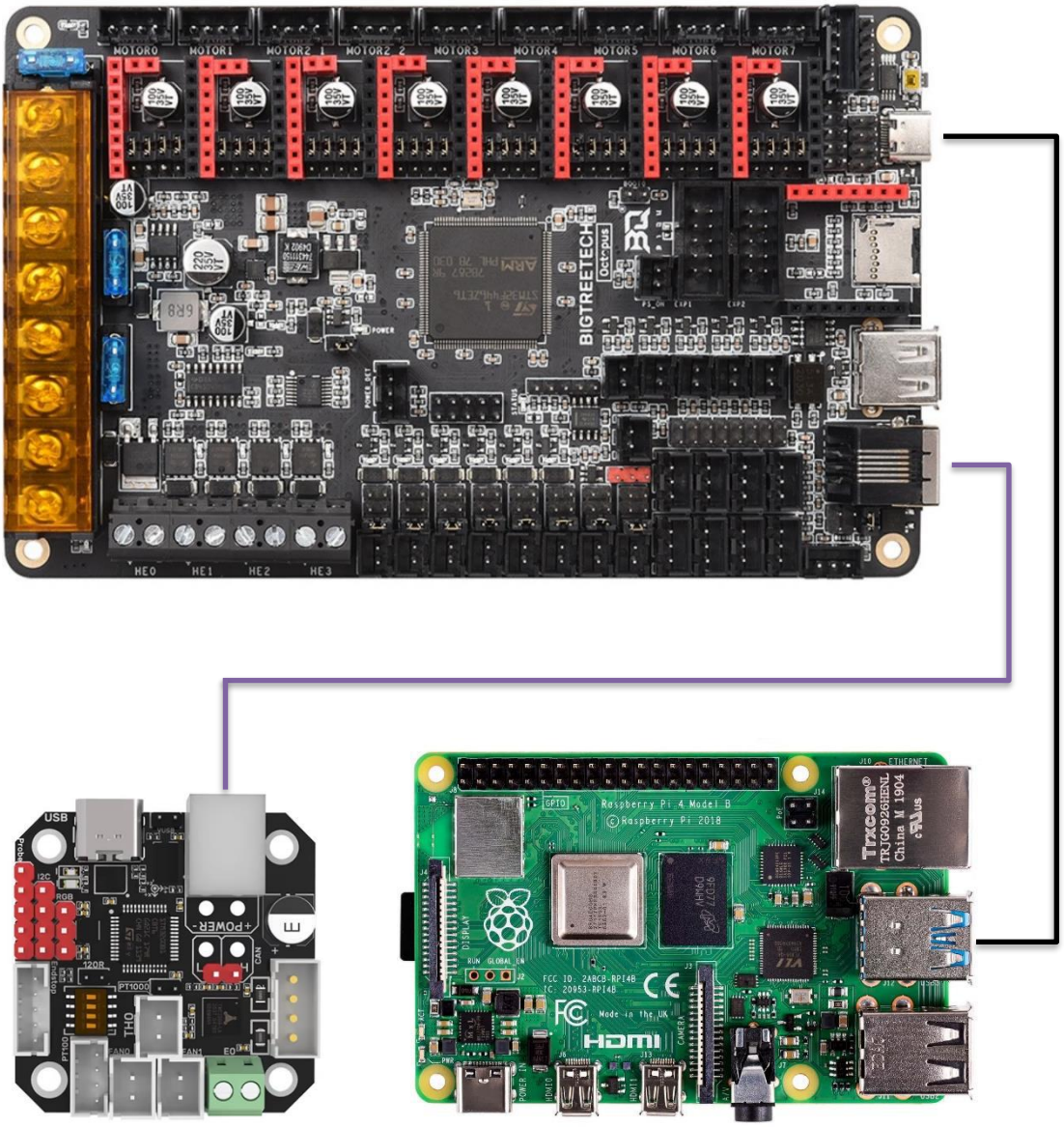
如何使用 Octopus / Octopus Pro 的CAN直连 小雨翻译修改版 V1.32

- Klipper 具有新的CAN总线功能，即用于Octopus的USB到CAN总线桥接通信板板（和其他兼容的MCU).
- 此功能允许Octopus板（和其他兼容MCU）充当USB到CAN总线适配器。这取代了对树莓派Pi CAN HAT、canable适配器或Bigtreetech U2C板等适配器的需求。



什么是 CanBoot?

CanBoot是加载到您的Octopus和EBB板上的自定义引导加载程序，允许用户通过USB、UART或CAN通信更新Klipper固件，而无需实际访问板重置按钮或BOOT跳线。

它使用相同的“make menuconfig”设置来配置和编译固件。

使用Klipper USB到CAN总线桥接通信不需要CanBoot。但我没有单独测试这个通信功能。

你将需要以下内容

- USB-A至USB-C电缆
- Octopus和EBB工具板（即24V）的电源
- RJ11或RJ12电话电缆，压接以将EBB板连接到Octopus
  - 参考指南末尾的接线信息。最好在安装软件之前完成接线。
- 树莓派或类似产品，安装并运行Klipper/Moonraker/UI
- 具有以下软件的计算机：
  - SSH终端软件， 或者Windows CMD （我使用的是PuTTY与WinSCP），用于在树莓派和计算机之间传输文件。
    - 还可以使用Windows CMD执行此操作
  - STM32CubeProgrammer
    - 首先为您的计算机下载STM32CubeProgrammer软件
    - https://www.st.com/en/development-tools/stm32cubeprog.html

说明，最好可以理解一下

1. 在Raspberry Pi上安装CanBoot，将固件写到主板上
   a、 为Octopus/Pro和EBB 配置和编译CanBoot固件。使用STM32CubePrograMer和计算机将CanBoot写到主板上
2. Klipper安装板
   a、 Octopus-设置用于USB到CAN总线桥的Klipper，带到EBB的CAN通信。EBB-设置用于CAN通信的Klipper
   c、 在树莓派上查找Octopus/Pro的串行设备
   d、 使用CanBoot串行命令将Klipper写到Octopus/Pro
3. 在树莓派上设置can0网络，关闭打印机
4. 查找板的CAN uuid
   a、 将EBB连接到Octopus/Pro，电源运行打印机
5. 使用CanBoot CAN命令将Klipper闪存到EBB板
6. 打印机配置更新和一般提示
7. 如何使用CanBoot更新主板，提示

让我们开始吧！

说明书

1. 在Raspberry Pi上安装CanBoot，并将固件闪存到板上

启动 SSH 终端软件并登录树莓派.

```
sudo apt-get install git -y   #如果您还没有安装，请执行这一步（执行完这一步请重启树莓派，我测试中不重启会造成下一步失败。但是，这一步好像可以省略。）
```

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

```
cd CanBoot
```

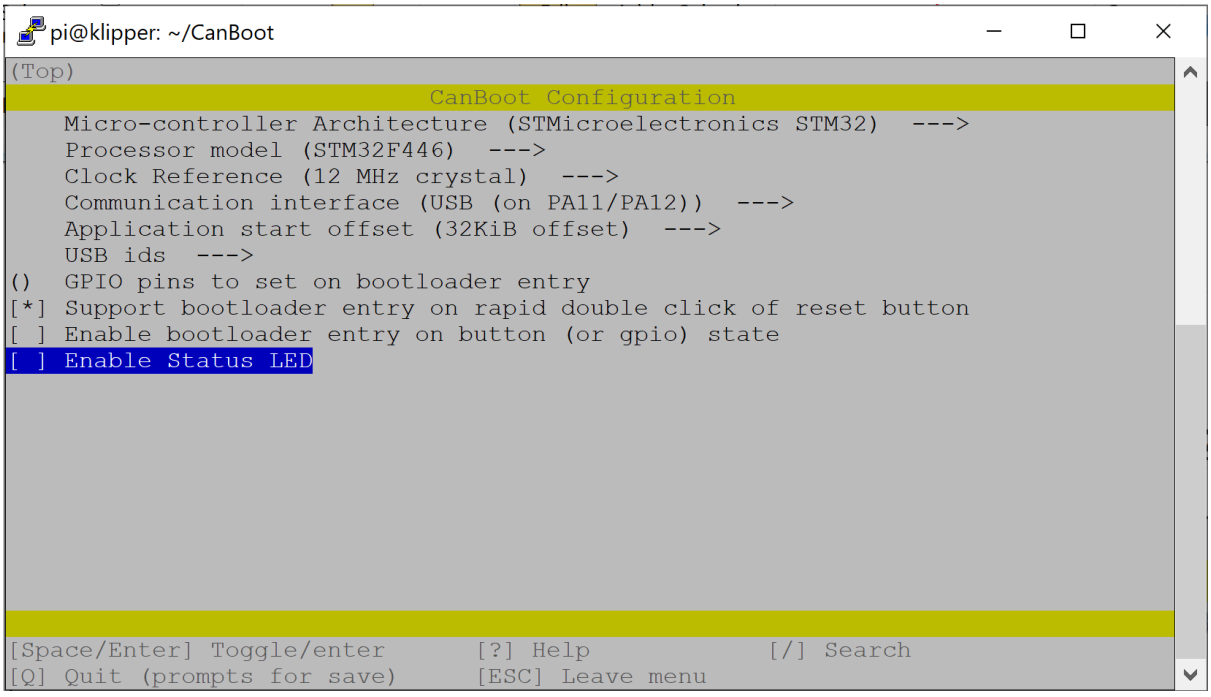
```
make menuconfig
```

1a. CanBoot 固件配置

- 这里列出了Octopus、OctopusPro、EBB36和EBB42的设置以及它们的所有类型。

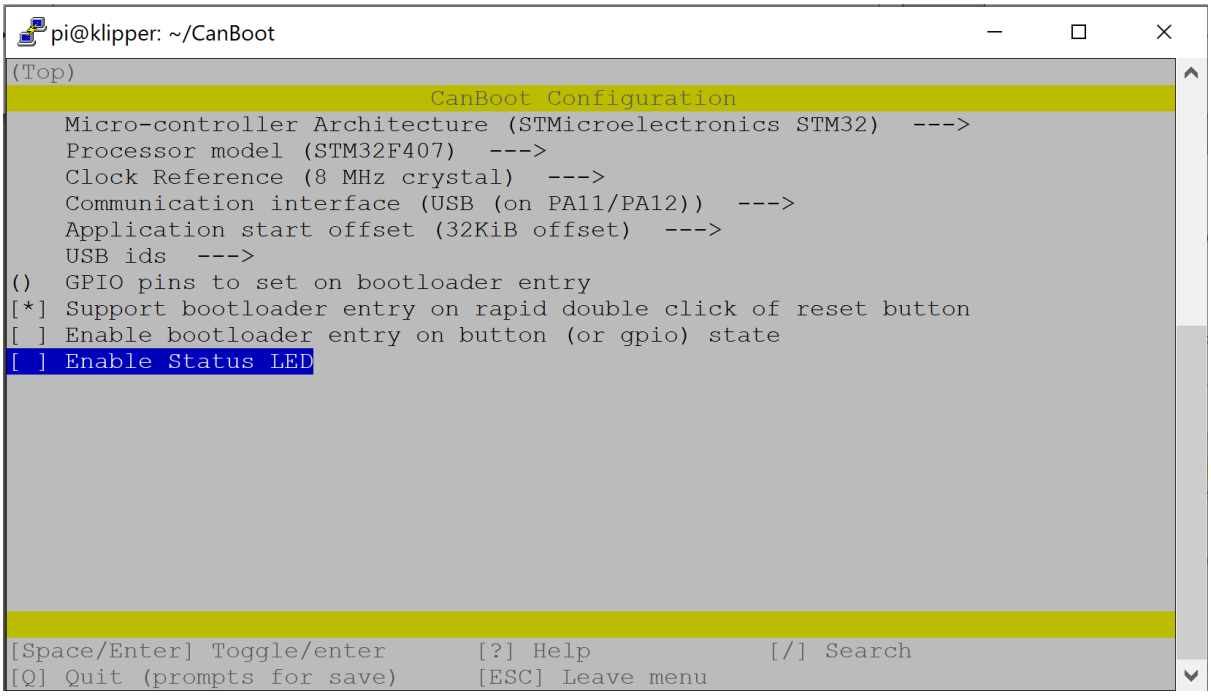
使用 F446 处理器的Octopus，Octopus Pro

- 12MHz crystal, USB on PA11/PA12, 32KiB offset



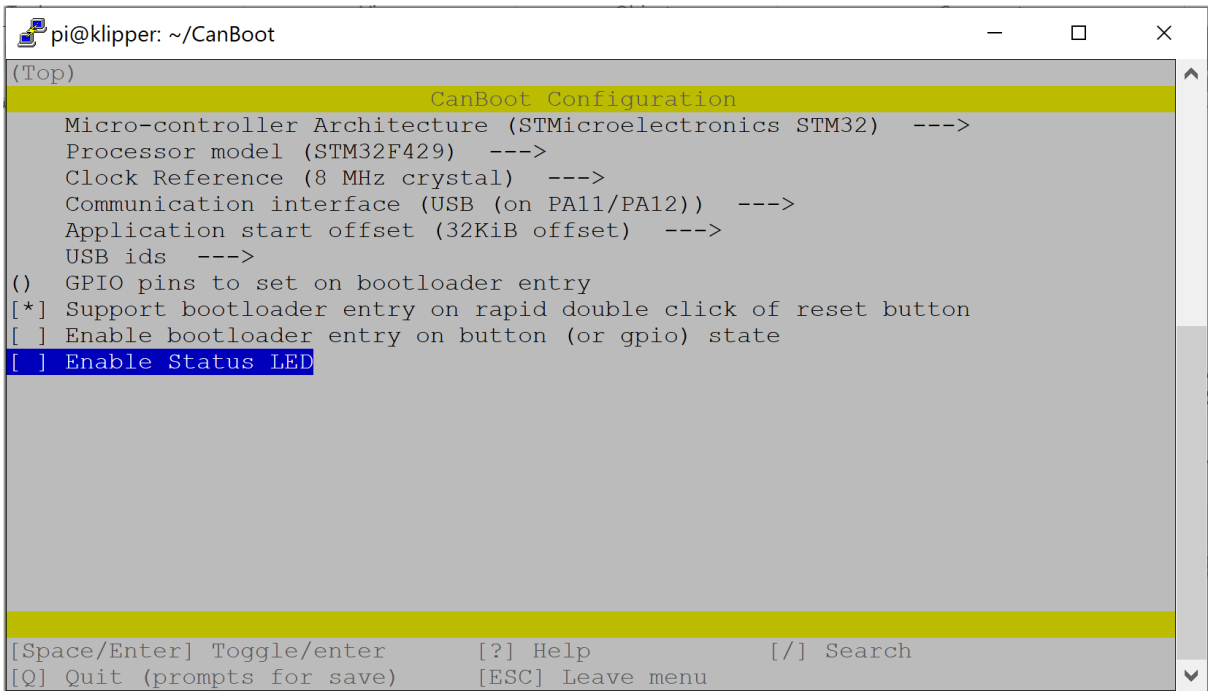
使用 F407 处理器的Octopus，Octopus Pro

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



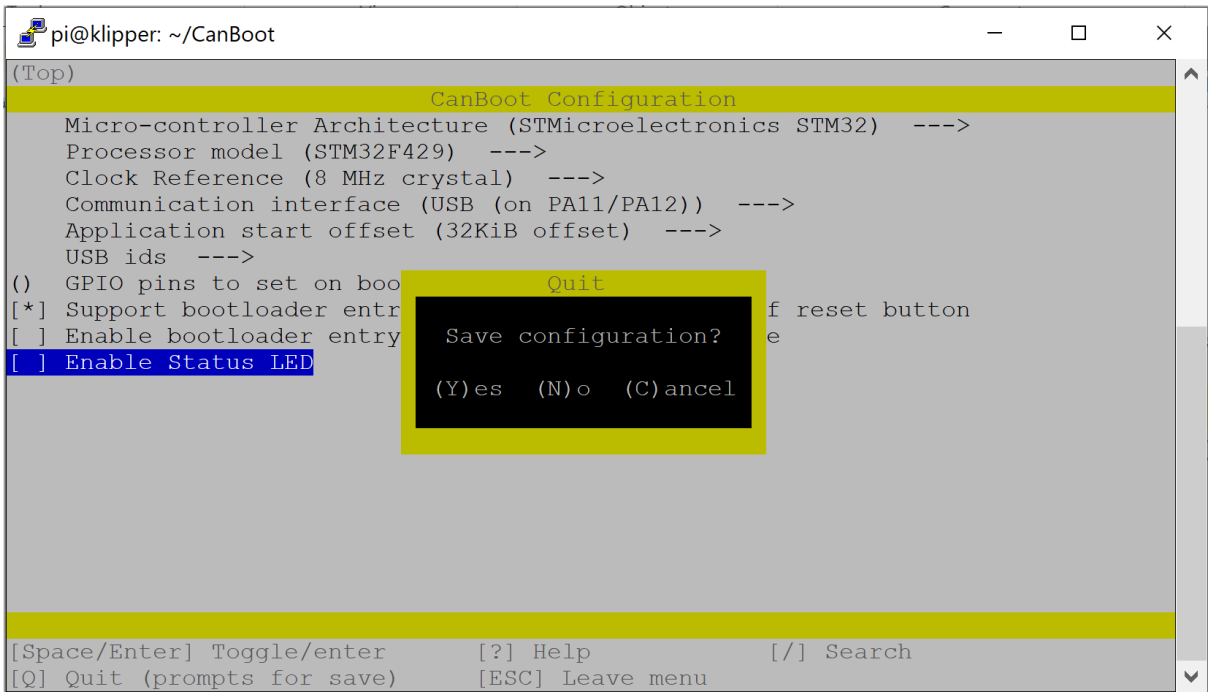
使用 F429 处理器的Octopus，Octopus Pro

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



结束OCTOPUS CONFIG 选项

点击 Q 然后点击 Y 键 提交 CanBoot 配置。



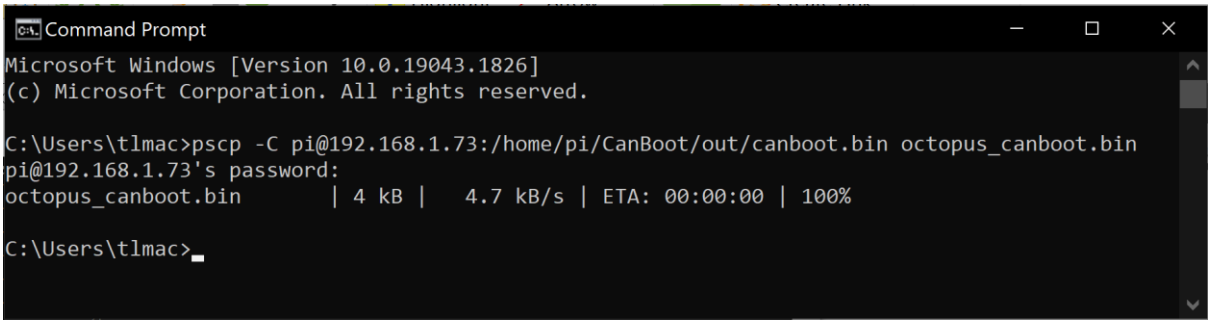

```
make
```

现在您已经创建了Octopus的canboot.bin固件文件。现在将要为EBB板制作CanBoot固件。但如果你现在这样做，它将覆盖你刚刚创建的Octopus固件。因此，请执行以下操作：

在你的电脑上打开并登录 CMD 。

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

- 确保将＜IP_ADDRESS＞替换为您的Pi IP地址或网络名称，如fluiddpi.local



NOTE: 您也可以手动用WinSCP移动canboot.bin文件到您的计算机。确保将其命名为octopus_canboot.bin。

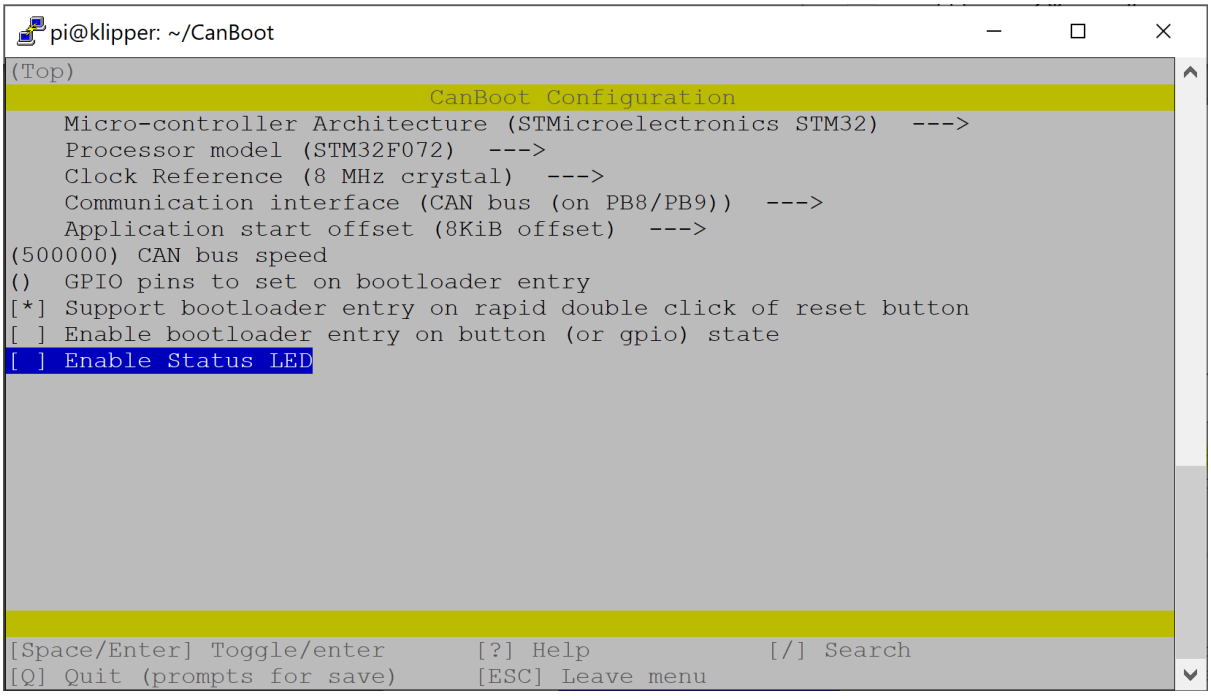
…返回树莓派SSH终端

`make clean`

`make menuconfig`

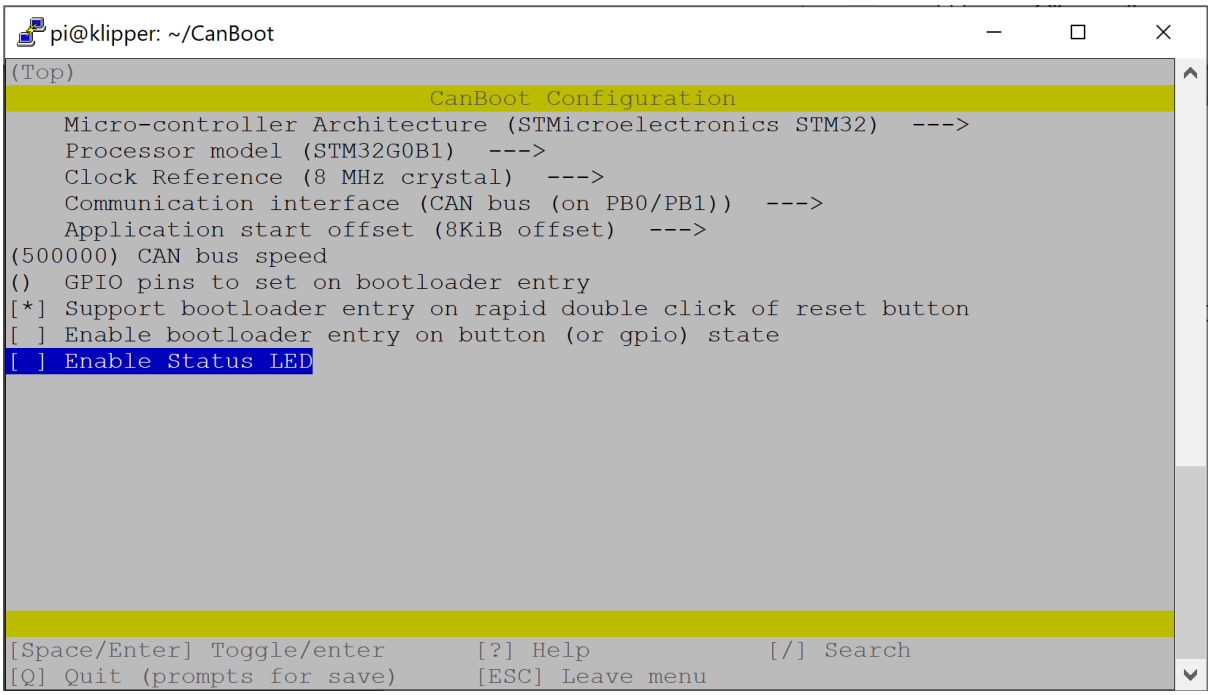
EBB36 / EBB42 v1.0 with F072 processor

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

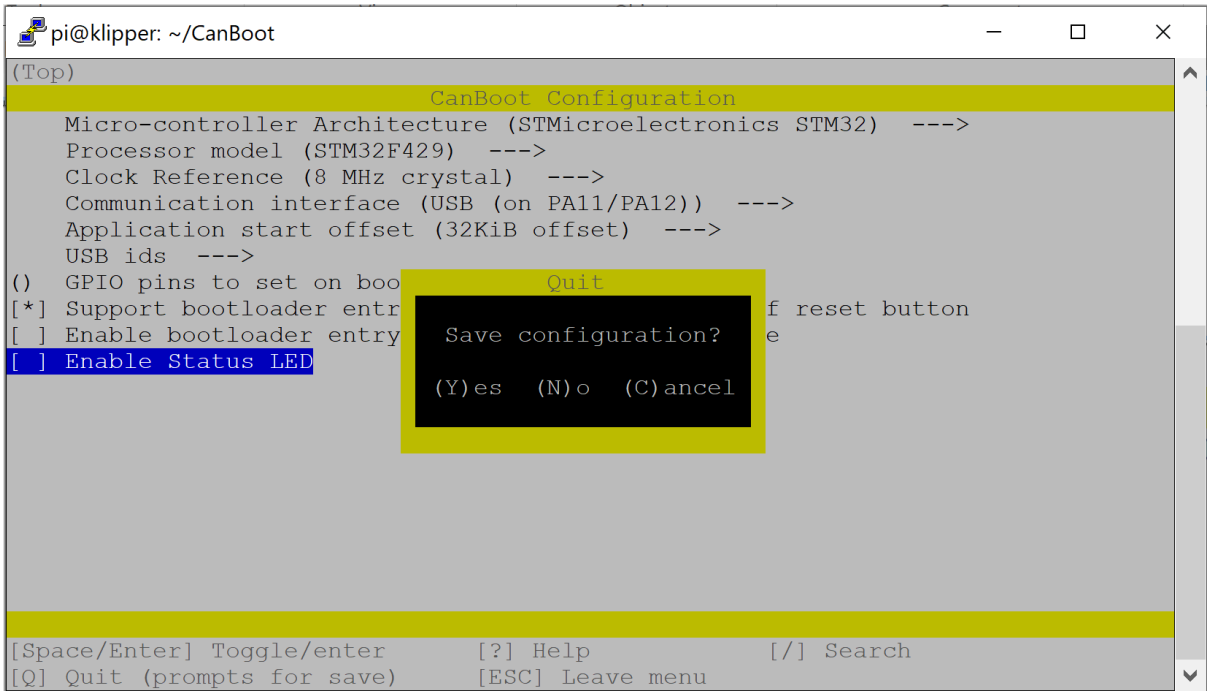


EBB36 / EBB42 v1.1 and v1.2 with G0B1 processor

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



同样点击 Q 然后点击 Y 键 提交 CanBoot 配置。

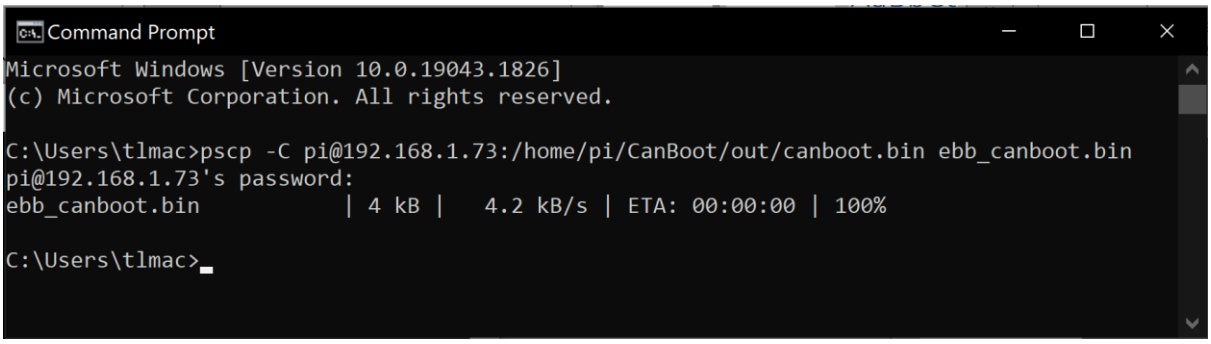


make

在计算机上，打开Windows CMD终端（或使用WinSCP移动文件并重命名ebb_canboot.bin）

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

- 确保将＜IP_ADDRESS＞替换为您的Pi IP地址或网络名称，如fluiddpi.local



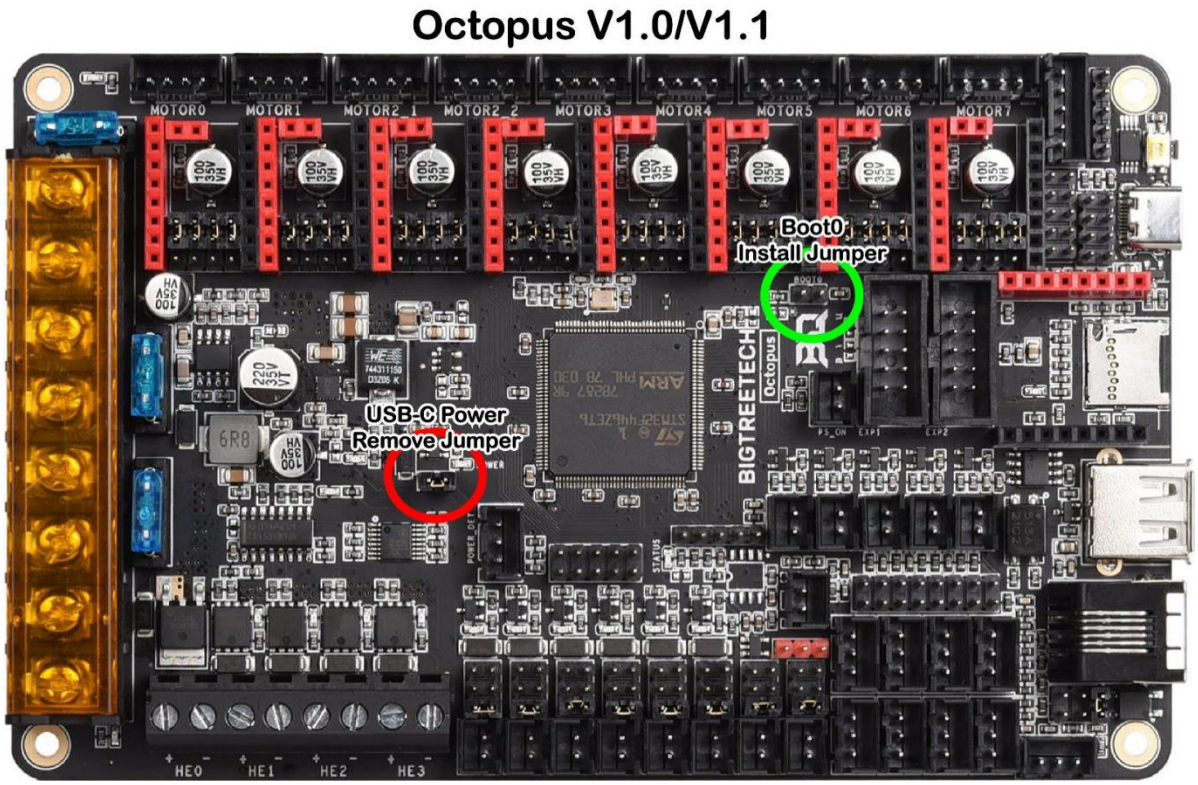
好了现在您有两个CanBoot固件文件现在在您的计算机上。下一步是将CanBoot写入到您Octopus和EBB 主板。

1b. 使用 STM32CubeProgrammer 写入 CanBoot

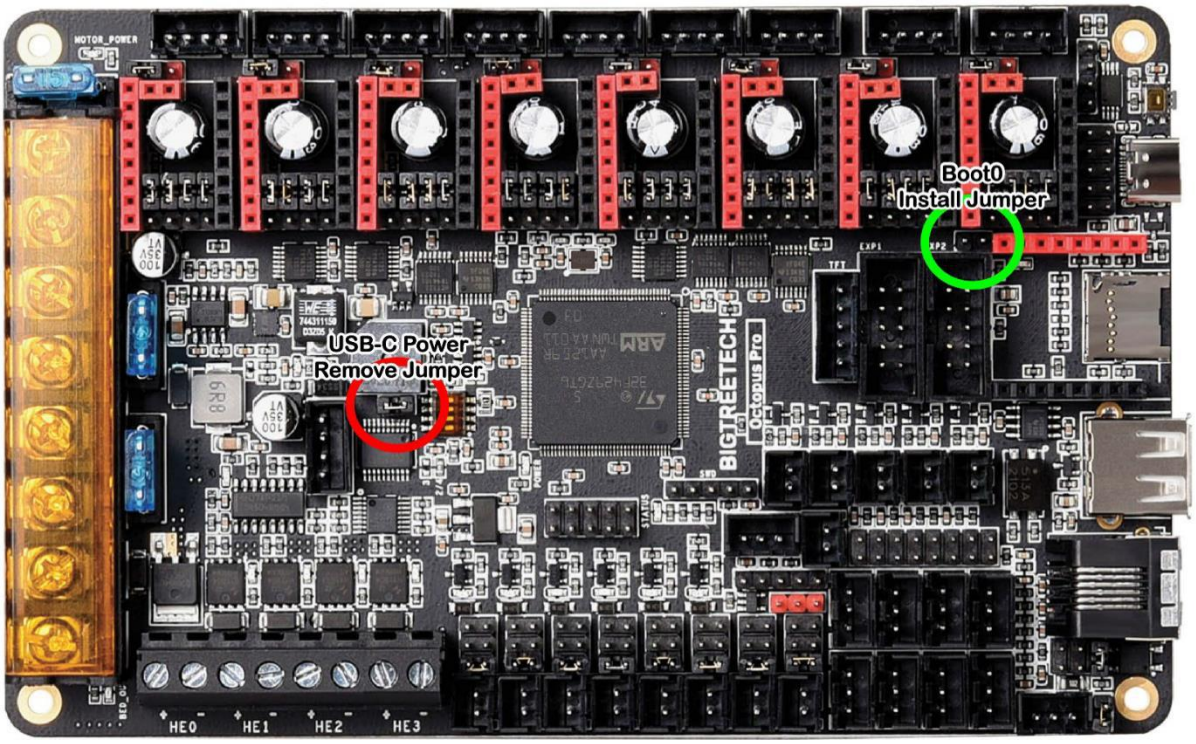
您需要将Octopus主板和EBB主板置于DFU模式，并使用USB-a到USB-C电缆连接到计算机。Octopus，Octopus Pro的引导说明如下:

确保Octopus主板已经关闭

首先，将USB-C至USB-A电缆将Windows PC和Octopus连接。
拆下USB-C电源跳线并将跳线帽放置在Boot0跳线上。



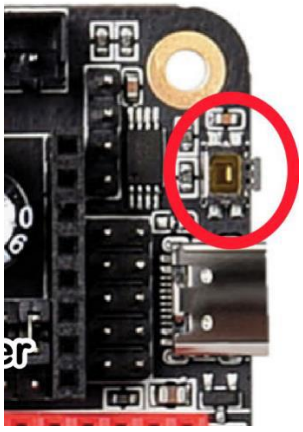
Octopus Pro



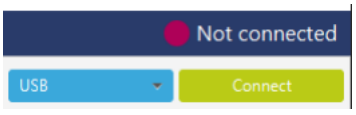
给Octopus板通电24V。

（我的操作方法不使用24V直接用USB，先插好BOOT0跳线，连线后插入USB-C跳线）

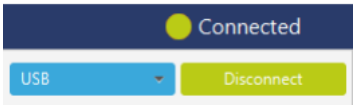
按下并释放Octopus主板上的重置按钮，Octopus会进入DFU模式。



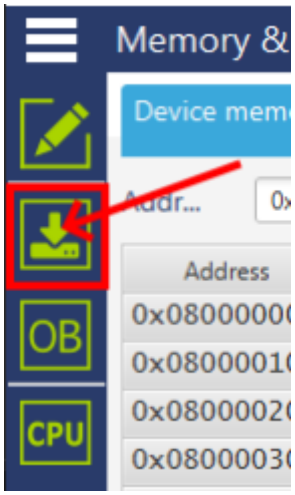
现在在Windows系统上启动STM32CubePrograrmer。

- 软件右上角可以看到 

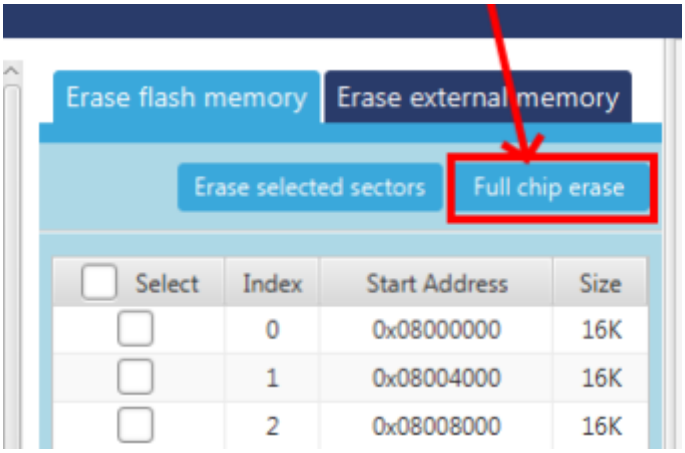
请确认选择USB连接后点击Connect 。

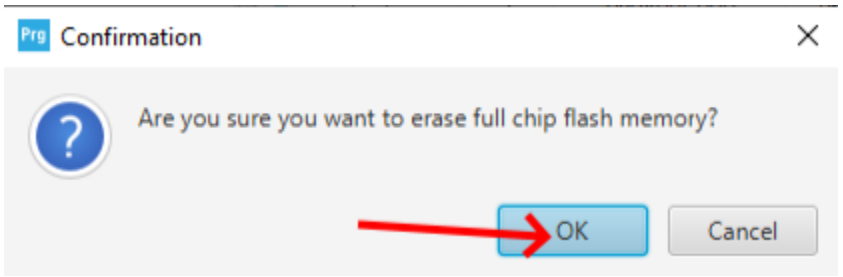
- 这时您可以看见 

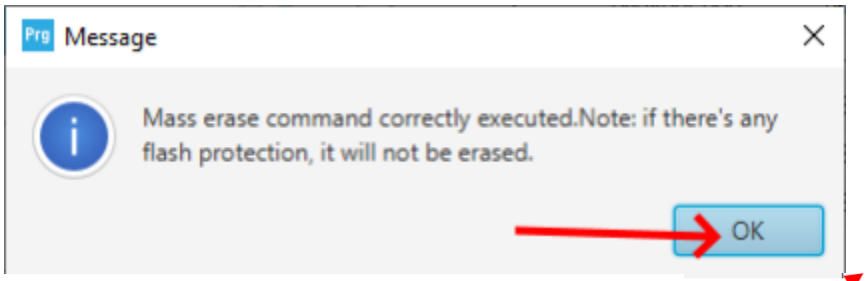
如果相同，恭喜您，可以进行下一步操作了（请见下一页）



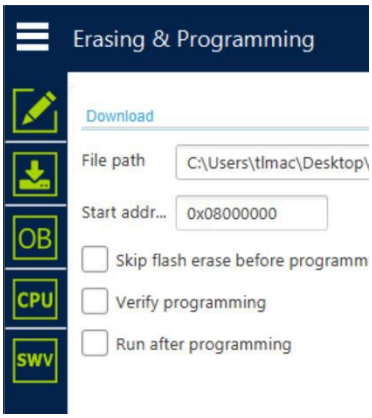
- 为了使CanBoot和Klipper正常工作，我们需要通过单击此按钮擦除闪存。



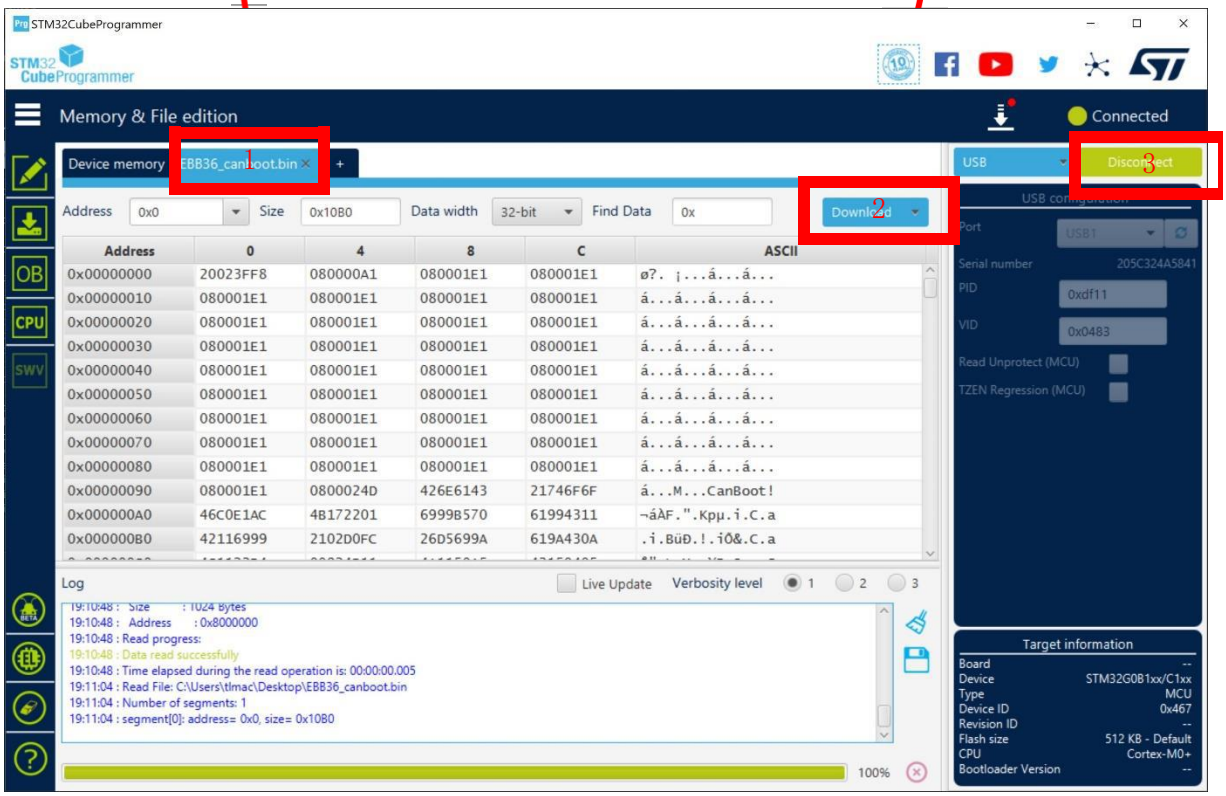




- 然后单击此按钮，返回主页



- 选择（1）“打开文件”选项卡，找到您之前保存的octopus_canboot.bin文件，然后点击（2）下载到您的Octopus。 完成后单击（3）“断开连接”。

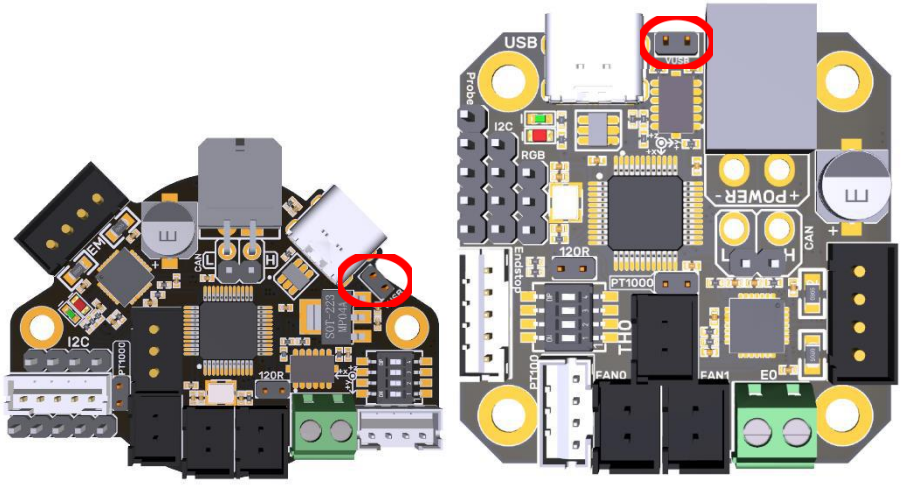


关闭您的Octopus 主板. 将 BOOT0上的跳线拆除.

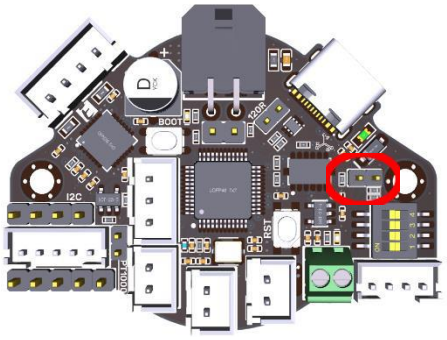
现在您需要刷 EBB 主板.

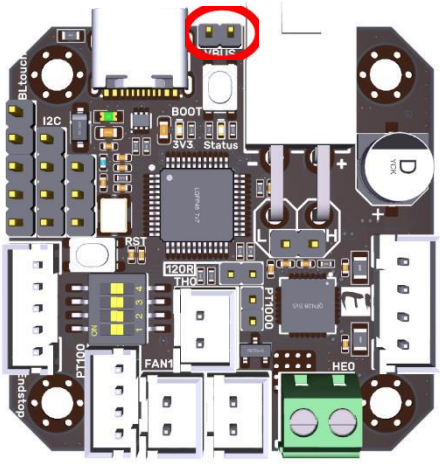
此步骤尽量从USB为工具板供电。需要在VBUS插针上放置跳线。

EBB36 / EBB42 v1.0



EBB36 / EBB42 v1.1 and v1.2





警告！

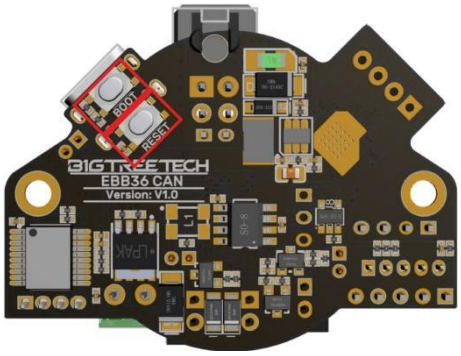
对于EBB36/42 v1.1，当您进入DFU模式时，模式默认固件是输出hotend加热器。如果您供电（12/24V）这将在没有热保护的情况下自动打开Hotend。这会造成损坏，因此尽量使用USB供电的刷机方式。

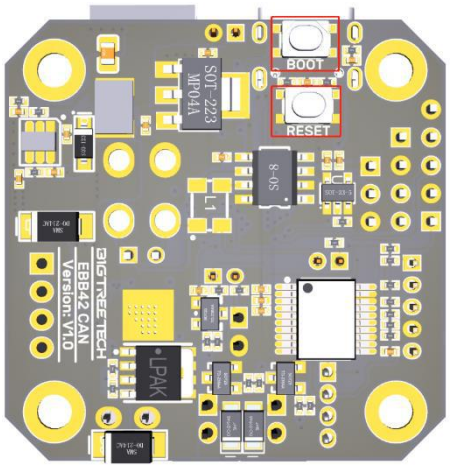
STM32CubeProgramer

进入DFU模式，USB线与电脑连接之后，您需要按住启动按钮，然后按下复位按钮。同时松开两个按钮。

EBB36 / EBB42 v1.0

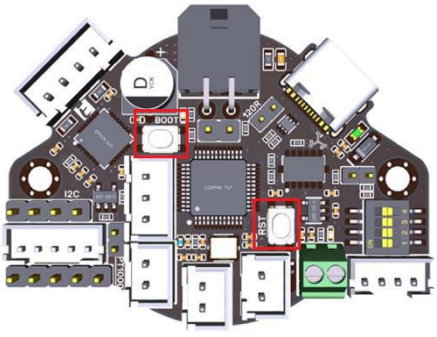
- 两个按钮都在背面

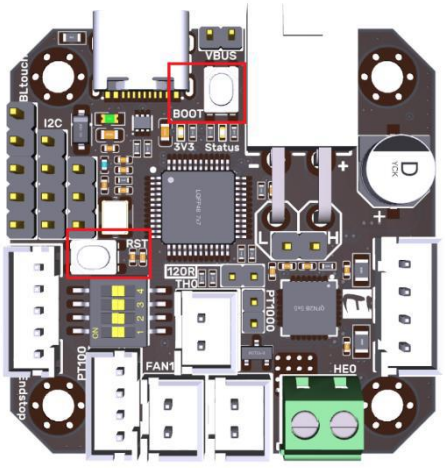




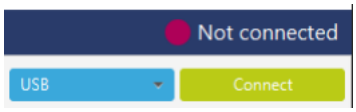
EBB36 / EBB42 v1.1 and v1.2

- 两个按钮都在前面

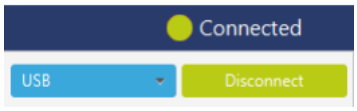




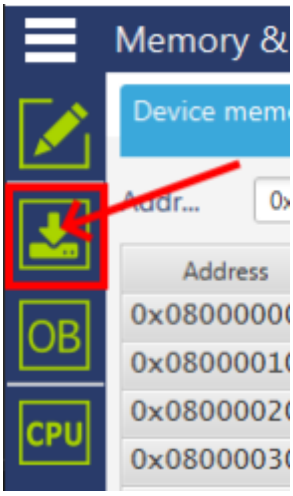
现在在Windows系统上启动 STM32CubeProgramer。

- 软件右上角可以看到 

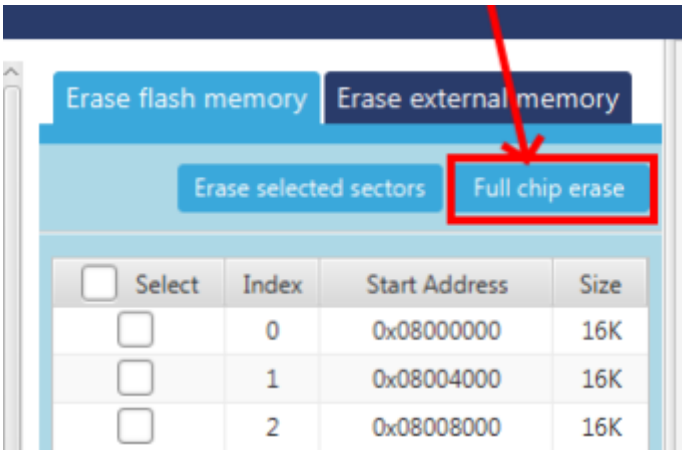
请确认选择USB连接后点击Connect 。

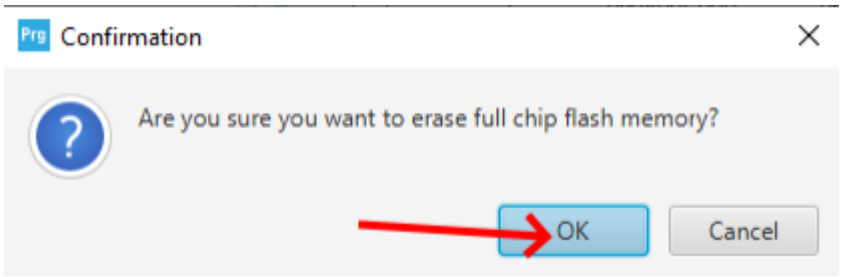
- 这时您可以看到 

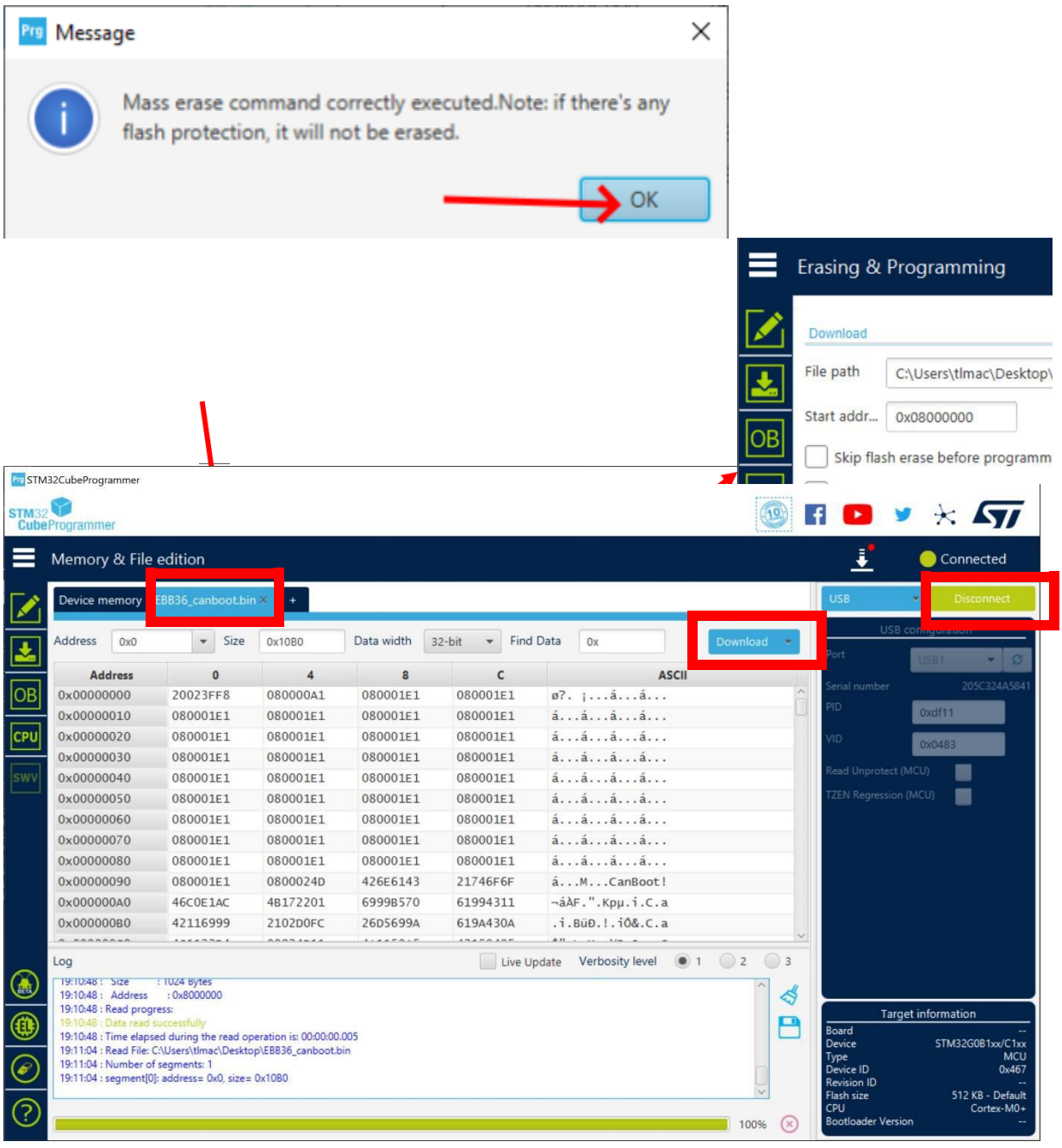
如果相同，恭喜您，可以进行下一步操作了



- 同样为了使CanBoot和Klipper正常工作，我们需要通过单击此按钮擦除闪存



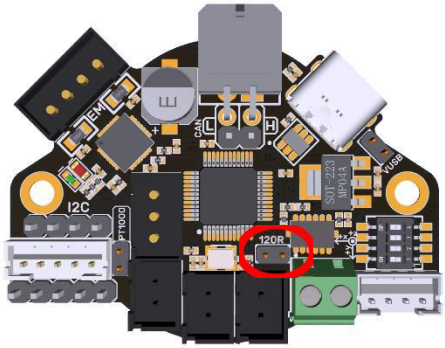


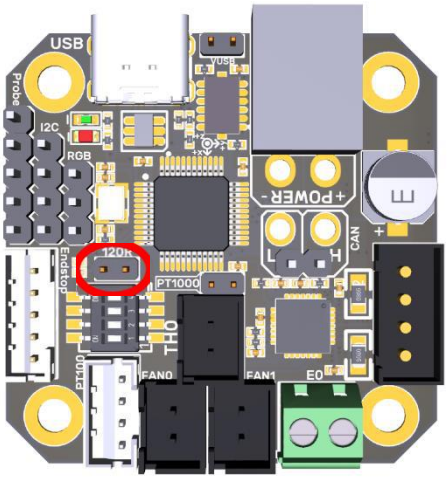


P断开USB与计算机的连接，关闭EBB的电源. 将VBUS的跳线帽拆除，

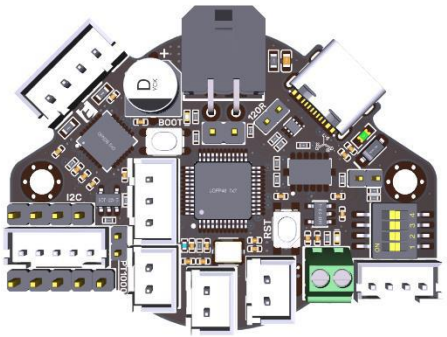
并将跳线帽子装至120R 的跳线针上. （请看下一页）

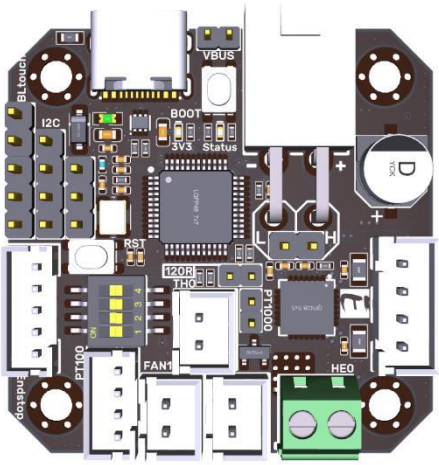
EBB36 / EBB42 v1.0





EBB36 / EBB42 v1.1 and v1.2





好了您的EBB板可以暂时搁置了.

2. 为Octopus / Pro 设置 Klipper

返回您的SSH 终端窗口.

SSH `cd ~/klipper`

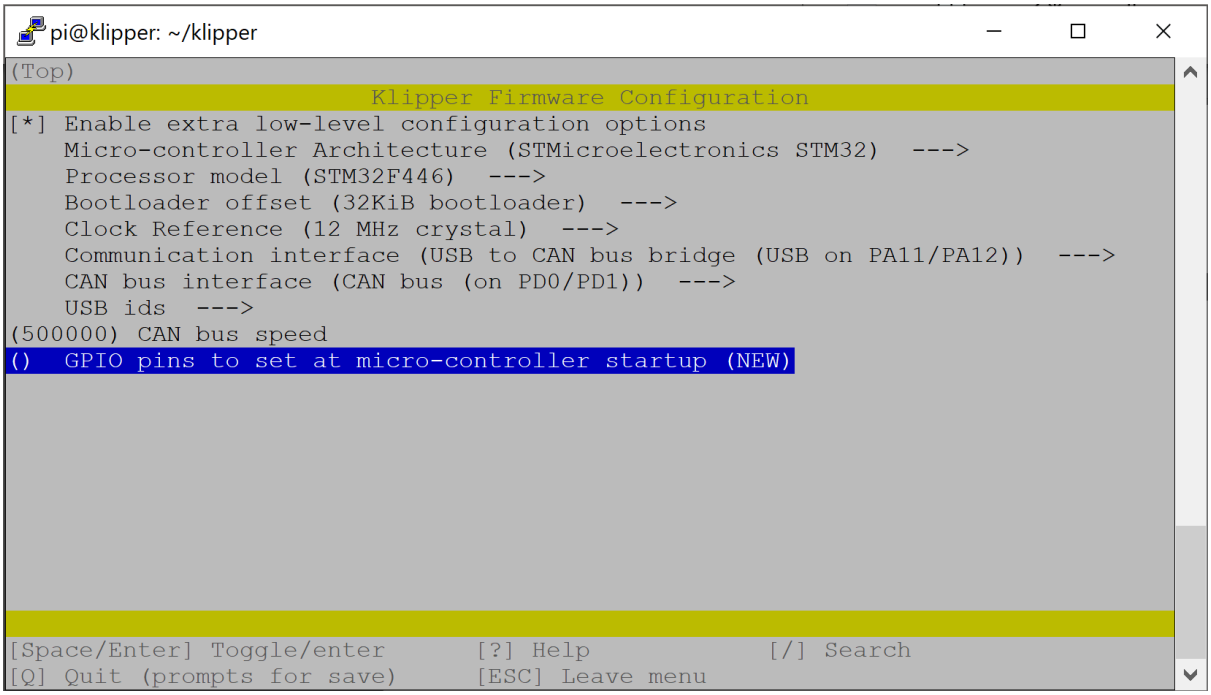
SSH `make menuconfig`

2a. Octopus - 设置用于USB到CAN总线桥的Klipper，与EBB的CAN通信

现在，您将在Octopus/Pro板上设置新的USB到CAN总线桥。下面是每个Octopus板的Klipper配置选项。确保选择了“Enable extra low-level configuration options”。

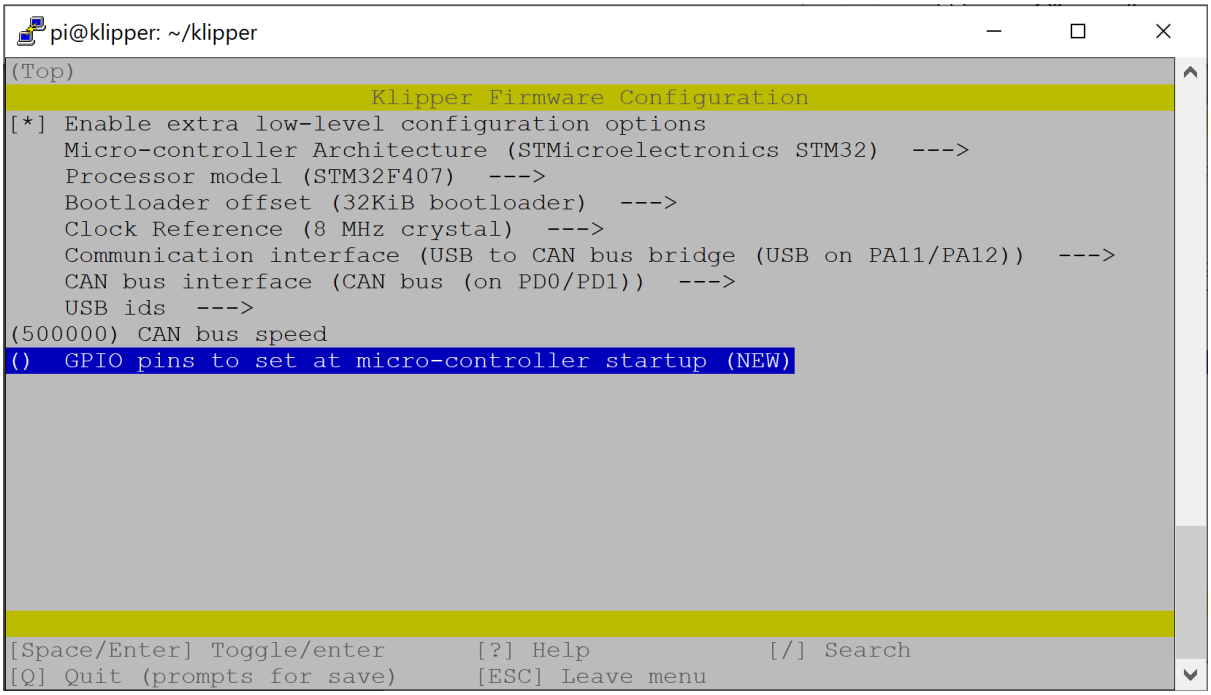
Octopus / Pro with F446 processor

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



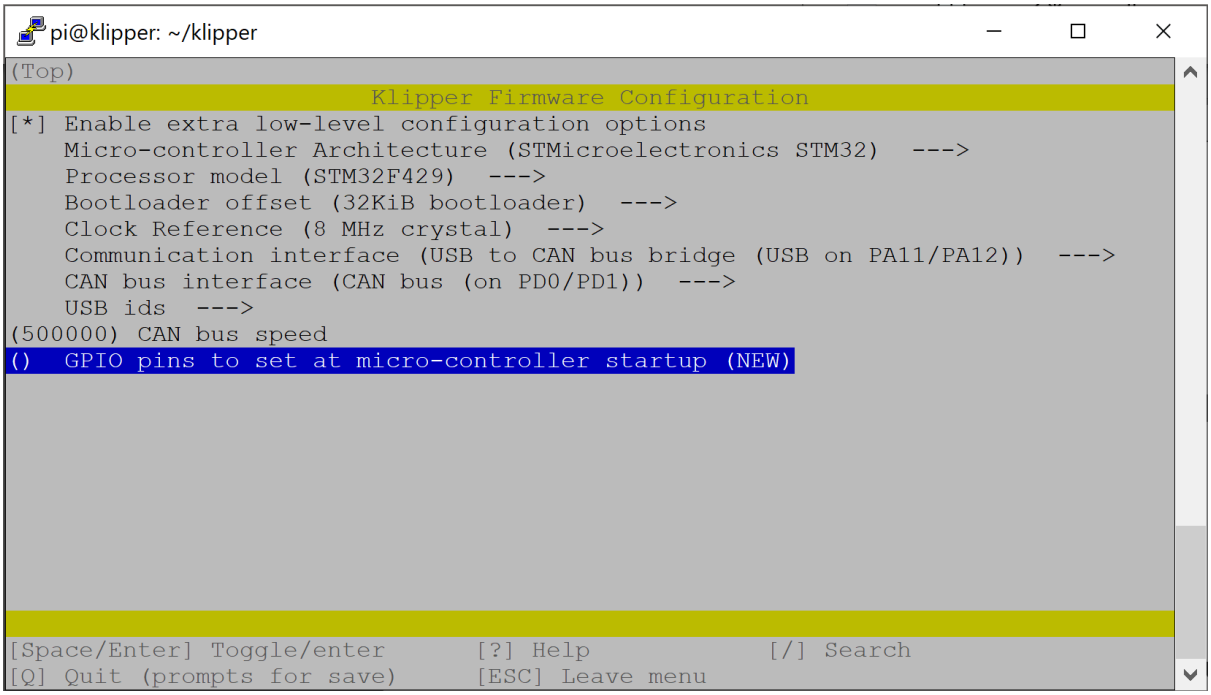
Octopus / Pro with F407 processor

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



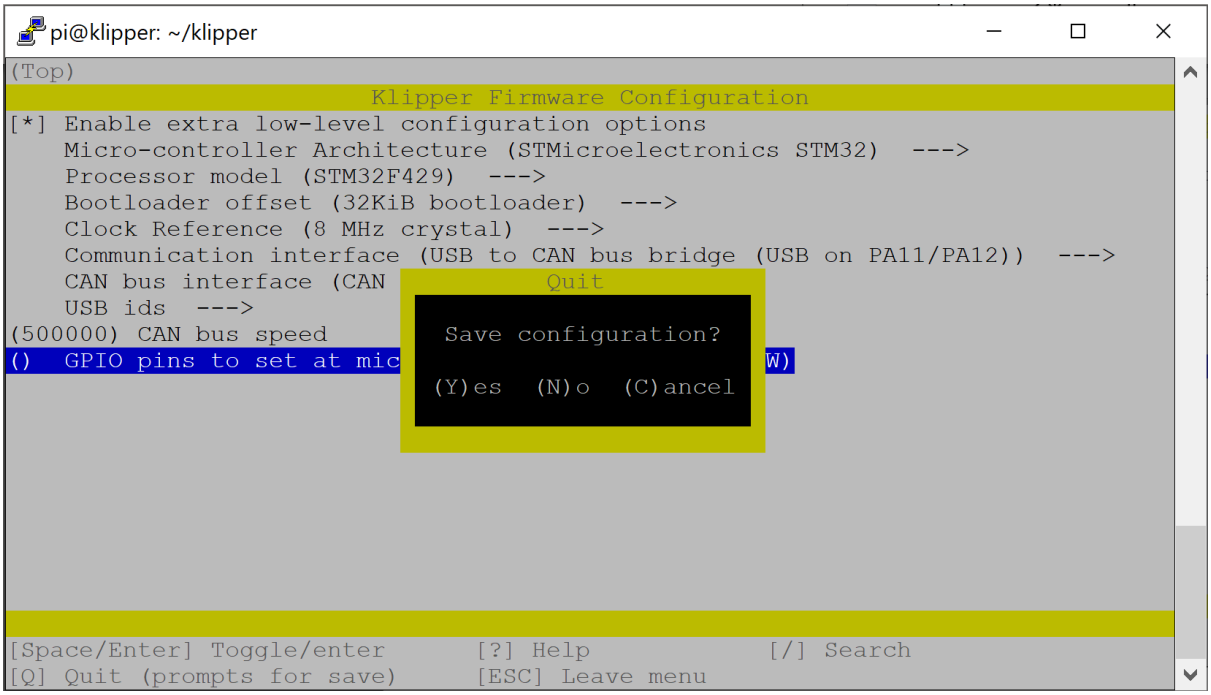
Octopus Pro with F429 processor

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



END OCTOPUS CONFIG OPTIONS

Press the Q then Y key to commit Klipper config.



SSH make

NOTE:

就像CanBoot固件一样，我们将为Octopus/Pro板和EBB板制作Klipper固件。由于我们已经编译Klipper固件，所以我现在想编译EBB Klipper固件。但如果我们这样做，那么我们刚刚制作的Octopus Klipper固件将被覆盖。因此请执行以下指令：

```
mv ~/klipper/out/klipper.bin octopus_klipper.bin
```

```
make clean
```

NOTE: （这步一定要看懂，一定要做对！也可以忽略以上命令，手动用winSCP完成。）

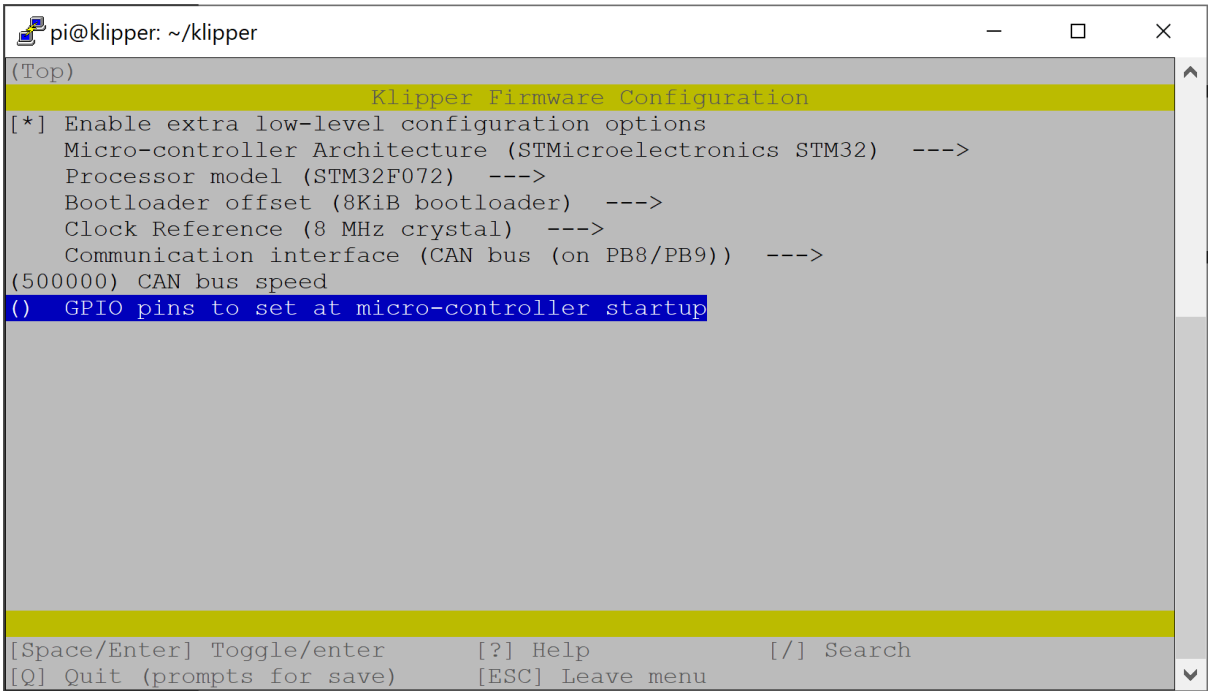
以上命令会将klipper固件文件移动到klipper文件夹并将其重命名为octopus_klippert.bin，然后删除编译文件和文件夹，包括/klippers/out

2b. EBB - 为EBB设置CAN通讯的 Klipper

```
make menuconfig
```

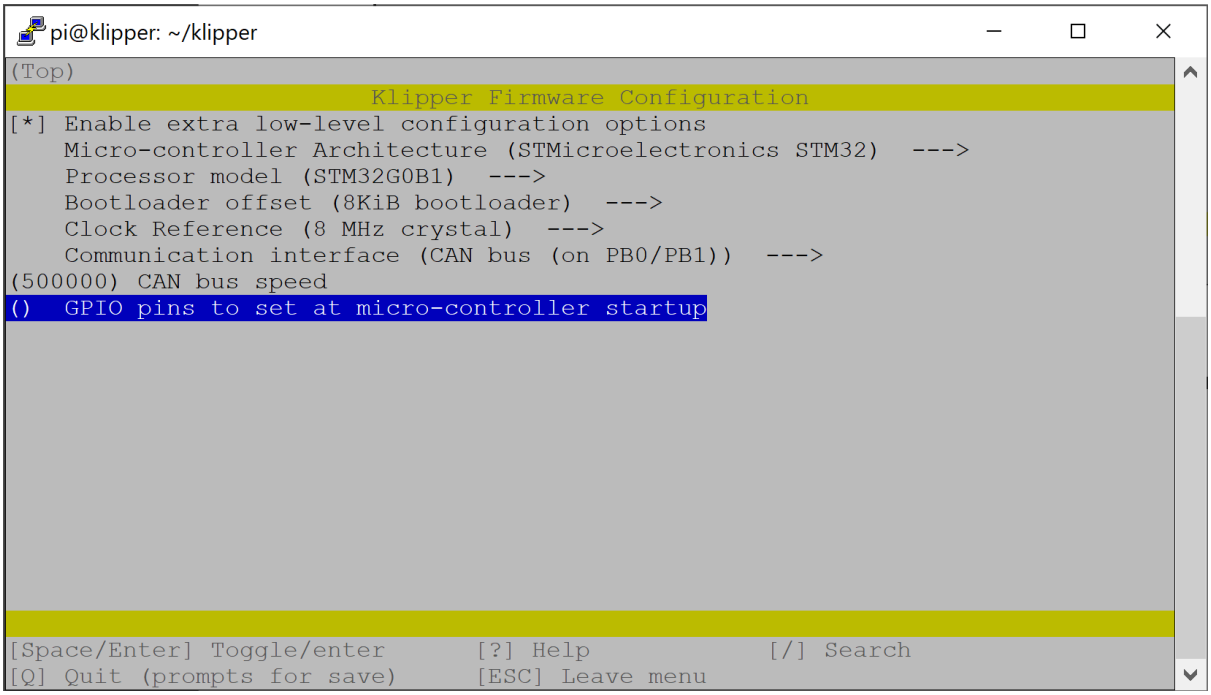
EBB36 / EBB42 v1.0 with F072 processor

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

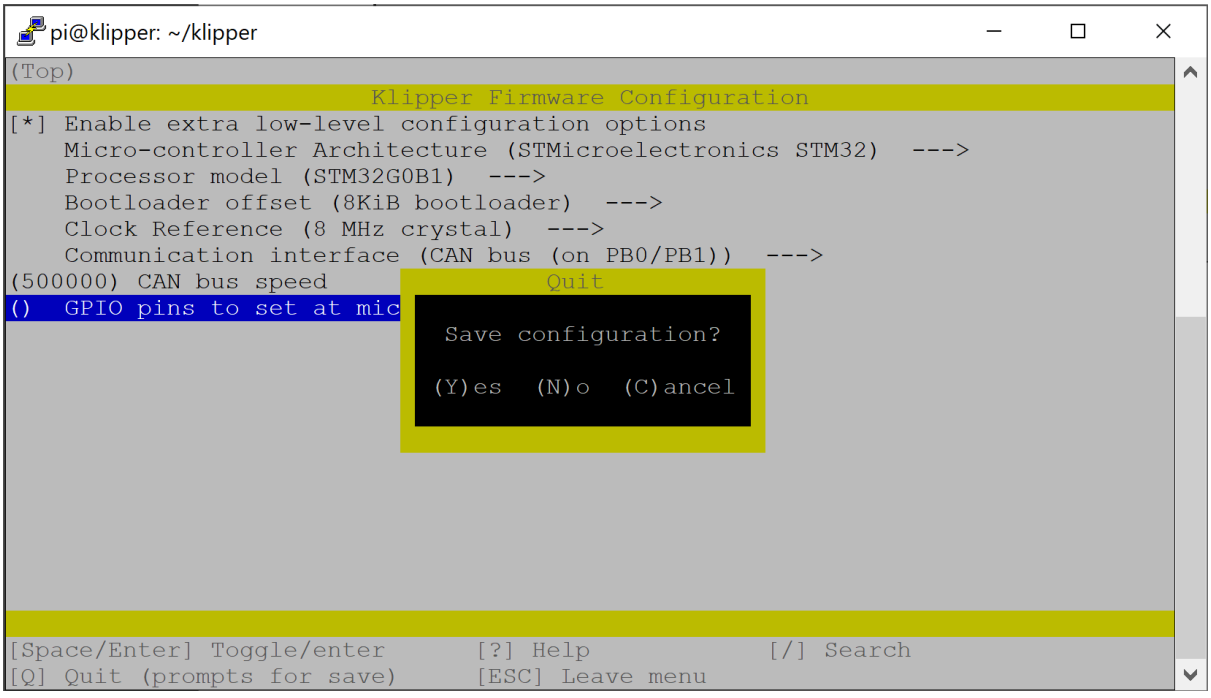


EBB36 / EBB42 v1.1 / v1.2 with G0B1 processor

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



Press the Q then Y key to commit CanBoot config.



SSH `make`

SSH `mv ~/klipper/out/klipper.bin ebb_klipper.bin`

SSH `make clean`

这步与上一步同样重要！！！！一定要做对！！

2c. 在树莓派上查找Octopus/Pro的串行设备

使用USB-a至USB-C电缆将您的Octopus/Pro板连接到树莓派……连接到Octopus板上的USB-C端口。**并打开打印机电源。**

SSH cd

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

这将显示类似于下面的内容。每个设备ID都会不同！

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

复制此内容，例如：

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

2d. 使用CanBoot串行命令将Klipper使能到Octopus/Pro

SSH cd CanBoot/scripts

SSH pip3 install pyserial

…这步只需要做一次

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

NOTE: 将＜serial_device＞替换为先前复制的路径…

```
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: Octopus 最初通过USB进行通信。使用现在固件后（以及Klipper更新，稍后会详细介绍）我们需要使用串行设备路径。Octopus 现在有一个CAN uuid… 让我们找到它！

3. 设置can0的树莓派网络，通电打印 （重要步骤）

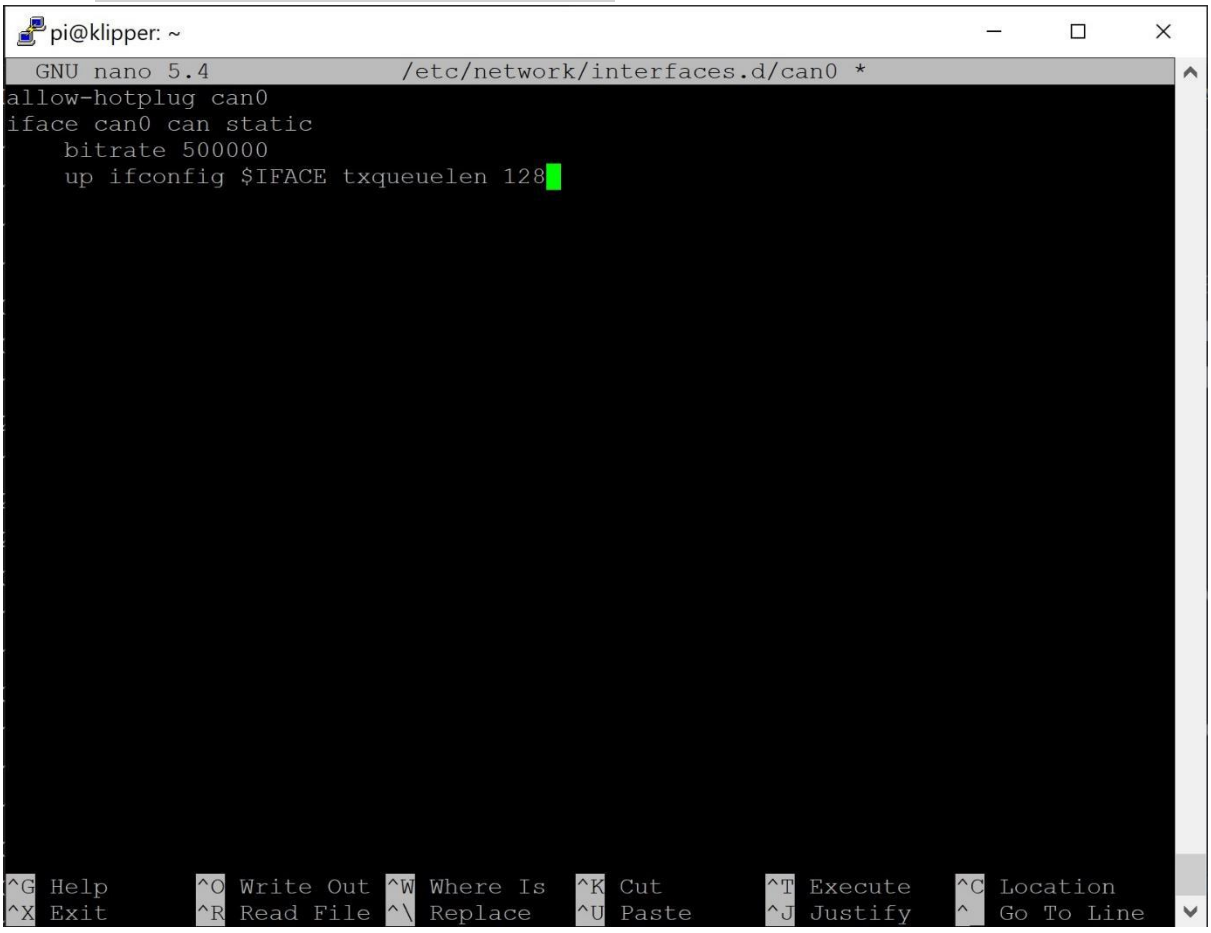
现在我们需要启用can0网络并设置它，以便CAN总线工作。

```
cd
```

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

- 复制并粘贴以下内容：

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



Ctrl - X 退出.

输入 Y 然后回车.

现在关闭打印机电源。是时候插入您的EBB工具板了。将电源和can数据电缆插入EBB。将RJ11或RJ12连接器插入Octopus/Pro板。

打开打印机电源。

NOTE:

我们刚刚将can0接口创建为热插拔网络。这意味着当设备发生变化或连接了新设备时，网络会做出反应。重启打印机是can0网络检测并配置Octopus作为CAN设备工作的一种变化。此外，此热插拔选项似乎可以使“firmware_restarts”更流畅。r.

4. 查找Octopus/Pro的CAN uuid

现在can0已经设置好了，让我们确保它已经启动并运行。

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>
```

如果你看到类似的结果就表明已经成功了。

NOTE: 如果这不起作用，请尝试关闭打印机电源并重新启动复盆子Pi。无法启动备份，请打开SSH终端并重试ifconfig。Can0不会显示。打开打印机电源，然后尝试ifconfig。应该有can0启动并运行。

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 $
```

在这种情况下，EBB和Octopus都被检测到。EBB只有CanBoot，而Octopus有Klipper。请将两个ID都保存以备以后使用。（config.cfg里已经写过UUID的，这里将没有显示。但是也表示您的通讯已经成功了。）

5. 使用CanBoot CAN命令将Klipper配置EBB板

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

NOTE: 将＜ebb_uuid＞替换为之前找到的CAN UUID。EBB标有“CanBoot”。在这种情况下，UUID是127081e7e3c66

```
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. 打印机配置更新和常规提示

编辑printer.cfg 修改您的新MCU信息CAN uuid。例如:

```
[mcu]
canbus_uuid: 463b35222d7b

[mcu EBBCan]
canbus_uuid: 127081e7e3c6
```

保存后重启.

您如果严格按照以上操作，则已经完成了所有配置。

如果您在这里遇到问题，请尝试第二次“重新启动”（重新启动Klipper）和/或“固件重新启动”

问题当重新启动导致打印机断电时，无法断开网络连接。时间可能太短，can0网络无法正常恢复。有时需要等待较长时间。

根据Klipper CANBUS文档，您可能需要运行SSH SSH `sudo ip up` 检查并

重新启动can0网络。可能没必要这么做。但这在初始设置期间可能有用。

您也可以使用 `sudo ip link set up can0 type can bitrate 500000` 在SSH里启动
can0如果它没有出现在ifconfig中。我使用过这个，但发现如果按照安装说明操作，就不需要它了。
如有问题，请参阅CANBUS上的Klipper文档
https://www.klipper3d.org/CANBUS.html#usb-to-can-bus-bridge-mode

以下部分就不全翻译了，萌新严格按照以上教程操作已经完全没有问题了。

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:

- Make Klipper firmware for Octopus board, cd to
  - Note I am still moving and renaming firmware files. If you this then leave out: `-f ~/klipper/octopus_klipper.bin` and CanBoot defaults to using default location of Klipper firmware [~/klipper/out/klipper.bin]
- CanBoot throws an error. If I try `ifconfig`, the can0 network is down
- Now if I try `ls /dev/serial/by-id/*` I get a serial ID:
- /dev/serial/by-id/usb-CanBoot_stm32f429xx_350026000B50314B33323220-if00 o Same as earlier…
  `CanBoot_stm32f429xx_350026000B50314B33323220-if00`
- And CanBoot works and flashes Klipper update!
- I can now update the EBB with: `python3 flash_can.py -f ~/klipper/ebb_klipper.bin -u 127081e7e3c6`

Long story short;

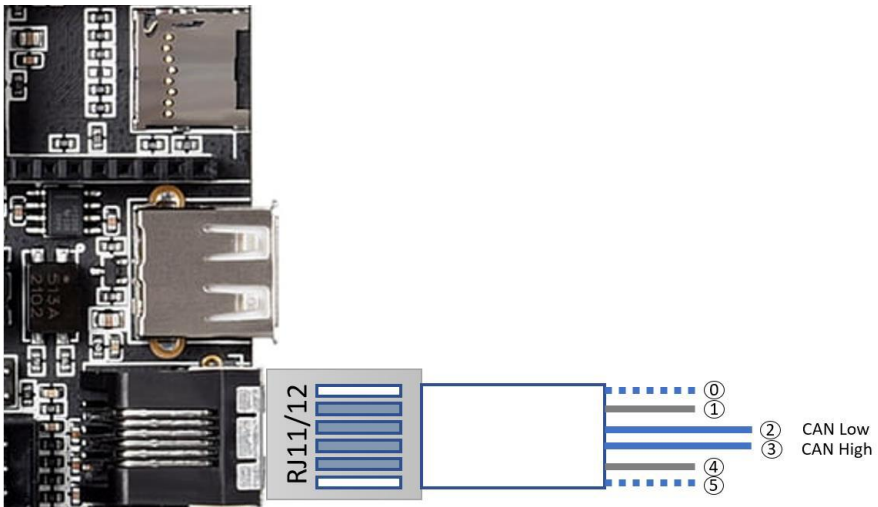
1. Try to flash Octopus board with CAN uuid, it will fail and can0 network is down. 2. Now you can flash Octopus board with serial ID.
3. Lastly you can flash EBB board with CAN uuid.

CAN Bus Wiring on next page…

RJ11 / RJ12 CAN bus 接线方式

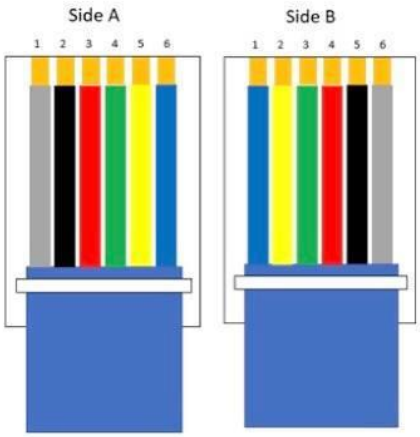
要将EBB36/42或其他支持CAN的工具板直接连接到Octopus板，您需要使用RJ11/RJ12电话端口。最简单的选择是使用一根2-3米长的电话线并切断一端。压接Molex MiniFit或Molex MicroFit连接器，具体取决于您的CAN工具板。RJ11/12连接器连接到Octopus板.

- 不允许使用普通绞合线，这是建议用于CAN总线规格的线材的。
- 作者有4线RJ11电话线直接连接到EBB36，没有绞合线。在作者这边没有问题。



(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.

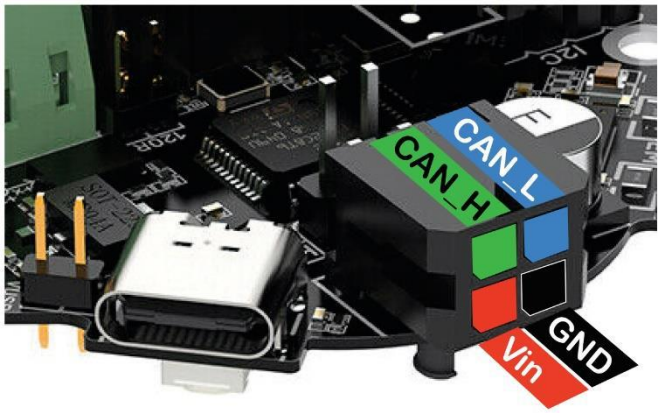
If plugging in a RJ12 cable with SIDE B your CAN Low wire is RED and CAN High wire is GREEN.

The colour of your wiring setup will be situational.

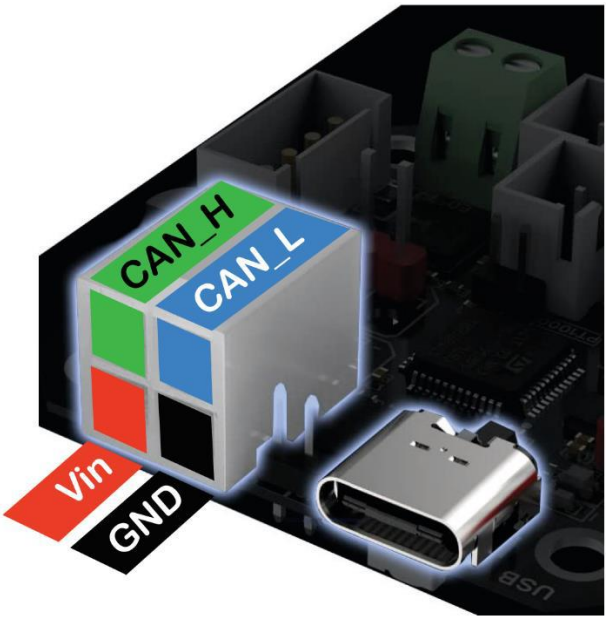
Take note of the wiring that you have plugged into the
Octopus board. Wire 2 is your CAN Low connection. Note the colour of the wire and make sure it goes to CAN_L on the EBB end. Same thing for the CAN

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



申明

以上翻译由小雨根据原版材料与自己的经验编辑翻译，仅供参考。

为避免纠纷，请大家谨慎使用此教程，小雨不承担任何负责！

特此申明！