

Make SB2040 and the curious case of disappearing firmware

Some of the SB2040 boards behave weirdly – the symptoms are that after flashing Klipper image they reboot fine and Klipper is active (the gpio24 led is alight and the board communicates via USB/CAN). Still, after power-cycling the board reverts to behaving as if there was no klipper flashed.

A couple of people found out that the Klipper is actually there, one can get it to boot, if they are lucky, by inserting a USB-C cable at a specific angle or just by trying to insert it many times, but that's not very useful aside from proving the board is flashed correctly

Some people also found out that CANBOOT on RP2040 does boot every time without any hitch. Using the CANBOOT bootloader it is actually possible to flash the board so that it actually comes up reliably. The exact mechanism of why one way it does work and the other way doesn't is unknown at the time of writing this memo.

I'm documenting here the walkthrough for flashing the CANBOOT and the Klipper image. Note that I assume at least a cursory ability to work over SSH (putty) and be able to follow the walkthrough carefully

Updates

12/21/2023 – added clarification about BLTouch probe and gpio29 pin

01/27/2023 – fixed grammar and some paths. Thanks to everyone who reported the issues

10/31/2022 – RP2040 was merged to upstream CanBoot. The howto updated.

11/08/2022 – RP2040 mods were merged to upstream Klipper. 🎉 Howto will be updated.

Should still work for the time being.

11/12/2022 – Updated the instructions for using stock Klipper. Please report issues/questions to yenda#2094 on discord

Installing CANBOOT

Plug the SB2040 in the bootloader mode to USB: press the reset button and **while** holding it, plug the USB-C cable in. Only after you plug the USB-C in, release the reset button. Verify the SB2040 is properly connected in the bootloader mode, using the lsusb command. See the following pic

```
pi@tri:~/klipper $ lsusb
Bus 002 Device 002: ID 090c:1000 Silicon Motion, Inc. - Taiwan (formerly Feiya Technology Corp.) Flash Drive
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 004: ID 1d50:614e OpenMoko, Inc. stm32f446xx
Bus 001 Device 003: ID 1d50:606f OpenMoko, Inc. Geschwister Schneider CAN adapter
Bus 001 Device 005: ID 2e8a:0003 Raspberry Pi RP2 Boot
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

You need to see the line “Raspberry Pi RP2 Boot”. Please read the remarks if you see the same ID (2e8a:0003) but the text is empty.

Then we will compile and install CanBoot image

```
cd ~
git clone https://github.com/Arksine/CanBoot
cd CanBoot
make clean
make menuconfig
```

```
(top)
Micro-controller Architecture (Raspberry Pi RP2040) --->
Flash chip (W25Q080 with CLKDIV 2) --->
Build CanBoot deployment application (Do not build) --->
Communication interface (CAN bus) --->
(4) CAN RX gpio number
(5) CAN TX gpio number
(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
(!gpio29) Button GPIO Pin
[*] Enable Status LED
(gpio24) Status LED GPIO Pin
```

```
make -j 4
sudo make flash FLASH_DEVICE=2e8a:0003
```

```
pi@tri:~/CanBoot $ sudo make flash FLASH_DEVICE=2e8a:0003
[sudo] password for pi:
Flashing out/canboot.uf2
Loaded UF2 image with 31 pages
Found rp2040 device on USB bus 1 address 5
Flashing...
Resetting interface
Locking
Exiting XIP mode
Erasing
Flashing
Rebooting device
```

Now the CanBoot bootloader should be installed and active. An easy way to confirm that is that the gpio24 LED is slowly blinking. Unplug USB-C and plug it back again. The LED should start blinking again.

A couple of remarks/possible issues (read carefully!)

1. This example assumes the can0 interface is configured with a 500k baud rate. Please change the menuconfig value to reflect your actual setting. Be sure to use the chosen baud rate consistently, otherwise, the CAN network will not work
2. This howto does not deal with hardware setup, i.e. 120 Ohm termination, cables pinouts, and so on.
3. This example assumes you will want to use your board over CAN – you can change the connection to USB in menuconfig if you do not plan to use the board over CAN. You can reflash with a modified config at any time anyway, so you are not committing to anything
4. If your board does not come up again (i.e. your gpio24 LED won't start blinking), then this walkthrough will not be able to help you. Probably try to return the board).
5. The example menuconfig sets the two-pin endstop port (gpio29) as a button that can be used to force enter the CanBoot bootloader. To do so, short the GND and the gpio pin in the slot and replug USB. If you don't plan to use this feature, feel free to uncheck the “*Enable bootloader entry on button (gpio) state*” option. It is not strictly necessary to have this option enabled. The latest release of the RP2040 CanBoot initializes the internal flash in such a way that it will always boot in bootloader mode unless it has been already flashed.
6. If you are using BLtouch or similar probe using the gpio29 pin, you **must** remove the “*Enable bootloader entry on button (gpio) state*” check. It's not useful for normal operation of the board anyway.
7. I've been told some SB-2040 come up like this

```
steven@Voron2:~/CanBoot $ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 004: ID 1d50:606f OpenMoko, Inc.
Bus 001 Device 003: ID 1d50:614e OpenMoko, Inc.
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
steven@Voron2:~/CanBoot $ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 004: ID 1d50:606f OpenMoko, Inc.
Bus 001 Device 003: ID 1d50:614e OpenMoko, Inc.
Bus 001 Device 007: ID 2e8a:0003
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
steven@Voron2:~/CanBoot $ |
```

I.e. the name is empty but the ID is the same. These seem to work OK as well. Try to flash. If you get a blinking diode, you should be good to go.

Time to CAN!

Unplug the board from USB and (with the printer turned off) plug the board to CAN. Turn on the printer. You should see the gpio24 LED starting to blink again.
SSH to the printer.

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

Ignore the paths (~/CanBoot) and the differences in command paths, those

```
Rebooting device
pi@tri:~/CanBoot $ ../klippy-env/bin/python ../klipper/scripts/canbus_query.py can0
Found canbus_uuid=af7f3ba501ee, Application: CanBoot
Total 1 uuids found
```

are inconsequential.

If there is no UUID found then something is wrong and this walkthrough won't be able to help you. If everything is right, you should see (at least one) UUID with Application:CanBoot
Time for the next step!

Klipper configuration and flashing

With the merge of the needed changes into klipper upstream, we can just use the latest klipper to compile firmware for the canboot-enabled board. Before running make menuconfig, make sure you backup the previous config (.config). Also you might need to update firmware in the other boards if you didn't update klipper for some time.

```
cd ~
cd klipper
git pull
make menuconfig
make clean
```

```
[*] Enable extra low-level configuration options
    Micro-controller Architecture (Raspberry Pi RP2040) --->
    Bootloader offset (16KiB bootloader) --->
    Communication interface (CAN bus) --->
(4) CAN RX gpio number
(5) CAN TX gpio number
(500000) CAN bus speed
(gpio24) GPIO pins to set at micro-controller startup
```

(save the config and quit. The Bootloader offset is the one that is important. Also, make sure your CAN bus speed matches your settings in RPI and in CANboot)

```
make -j 4
```

```
version: 1221c2d7
Linking out/klipper.elf
Creating hex file out/klipper.bin
pi@tri:~/klipper-sb2040 $ █
```

(you should see the message “Creating hex file out/klipper.bin”. It is important that it’s klipper.bin, not klipper.uf2! If you see uf2, go back to menuconfig and enable “Use CanBoot bootloader”)

```
cd ~/klipper
```

```
./lib/canboot/flash_can.py -u <UUID> -v -f ./out/klipper.bin
```

```
klipper:bin klipper:file klipper:cli
pi@tri:~/CanBoot $ ./scripts/flash_can.py -u af7f3ba501ee -v -f ../klipper/out/klipper.bin
Sending bootloader jump command...
Resetting all bootloader node IDs...
Checking for canboot nodes...
Detected UUID: af7f3ba501ee, Application: CanBoot
Attempting to connect to bootloader
CanBoot Connected
Protocol Version: 1.0.0
Block Size: 64 bytes
Application Start: 0x10004000
MCU type: rp2040
Verifying canbus connection
Flashing '/home/pi/klipper/out/klipper.bin'...

[#####]

Write complete: 113 pages
Verifying (block count = 452)...

[#####]

Verification Complete: SHA = 7629CEDEA0F6EDA3A8D2EE1EFA6C3D7C35242AE
CAN Flash Success
```

(note that the path in the picture is a bit different and includes the UUID of `_my_` system but that's ok, it's just a picture from a different revision of the document. The output and behavior will be the same.)

Let's verify if the klipper is up!

One indication is that the gpio24 LED stays lit constantly (no blinking).

Let's scan the CAN bus

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

```
pi@tri:~/klipper $ ../klippy-env/bin/python ./klipper/scripts/canbus_query.py can0
Found canbus_uuid=af7f3ba501ee, Application: Klipper
Total 1 uuids found
```

(notice the output is a bit different. The UUID is the same, but now it says Application: Klipper)

Power down the printer. Power up the printer. SSH in. Repeat the drill:

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

```
pi@tri:~/klipper $ ../klippy-env/bin/python ./klipper/scripts/canbus_query.py can0
Found canbus_uuid=af7f3ba501ee, Application: Klipper
Total 1 uuids found
```

(if you don't see the UUID available again, something is wrong and you should ask in the #CAN thread in #CAN bus stop thread on the voron discord. It's possible your board is faulty in some other, additional, exciting, way.)

Now you should be able to use the UUID (canbus_uuid) in the klipper config!