General Info

The following should be taken as an overall guide on what you are going to be achieving.

PLEASE DO NOT TAKE THE SCREENSHOTS/CONFIGURATIONS ON THIS PAGE EXACTLY AS WRTTEN AS THEY MAY NOT BE COMPATIBLE WITH YOUR PARTICULAR MAINBOARD

Before doing anything it is good to have some dependencies installed. Do this by running these on your Pi:

```
sudo apt update
sudo apt upgrade
sudo apt install python3 python3-serial
```

> If you get an error along the lines of “unable to locate package python3-serial” then you may be on an older version of linux.
>
> In that case, run:
>
> `sudo apt install python3-pip`
>
> then
>
> `pip3 install pyserial`

Installing Katapult

First you need to clone the Katapult repo onto your pi. Run the following commands to clone (or update) the repo:

```
test -e ~/katapult && (cd ~/katapult && git pull) || (cd ~ && git clone https://github.com/Arksine/katapul
```

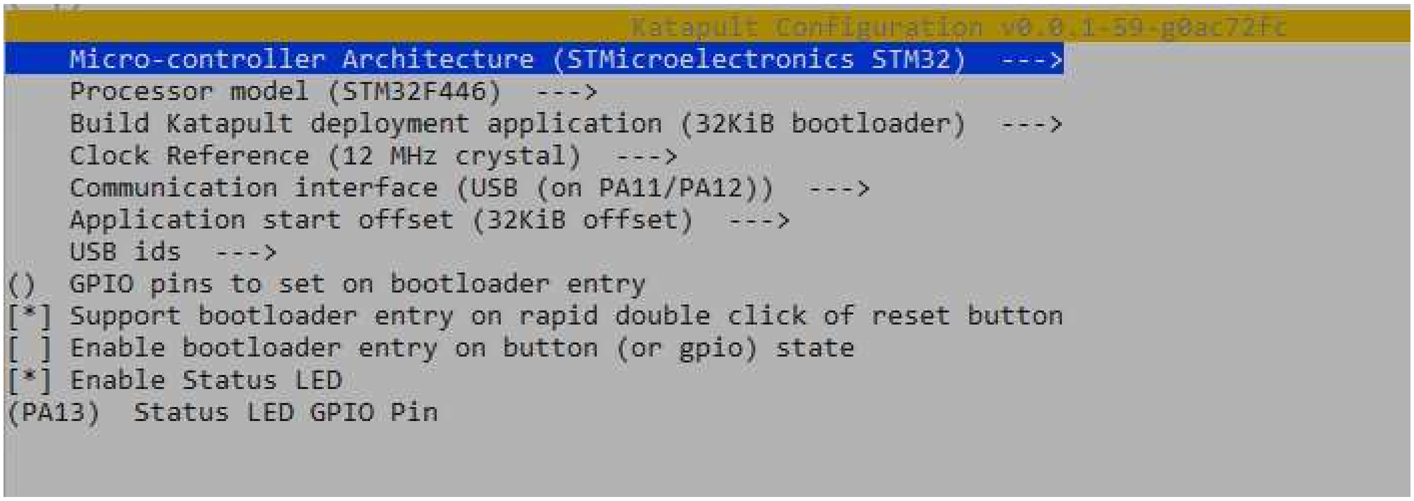
This will clone the Katapult repo into a new folder in your home directory called "katapult" if you don't already have it, or it will update your Katapult folder to the latest version if you did already have it.

To configure the Katapult firmware, run these commands to change into the katapult directory and then modify the firmware menu:

```
cd ~/katapult
make menuconfig
```

You will need to adapt the below instructions so they cover *your* board's specicific configuration. You can find screenshots of settings for common toolheads in the Common Mainboard Hardware section.

If your board doesn't exist in the common_hardware folder already, then you want the Processor, Clock Reference, and Application Start offset to be set as per whatever board you are running. Set the "Build Katapult deployment application" (this is really only for updating but doesn't hurt having it enabled at this stage), and make sure "Communication Interface" is set to USB. Also make sure the "Support bootloader entry on rapid double click of reset button" is marked. It makes it so a double press of the reset button will force the board into Katapult mode. Makes re-flashing after a mistake a lot easier. Lastly, setting the Status LED GPIO Pin won't affect how katapult functions, but it will give a nice visual indicator (of an LED flashing on and off once a second) on the toolhead to show the board is sitting in Katapult mode.



```
                                Katapult Configuration v0.0.1-59-g0ac72fc
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F446)  --->
    Build Katapult deployment application (32KiB bootloader)  --->
    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
(PA13)  Status LED GPIO Pin
```

Press Q to quit the menu (it will ask to save, choose yes).

Compile the firmware with `make`. You will now have a katapult.bin at ~/katapult/out/katapult.bin.

```
make clean
make
```

To flash, connect your mainboard to the Pi via USB then put the mainboard into DFU/BOOT mode (your mainboard user manual should have instructions on doing this).

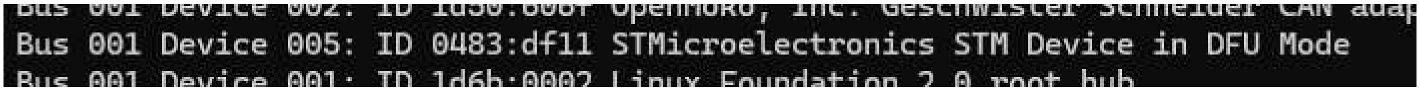
If your mainboard board uses an STM32 based MCU use these flashing steps

If your mainboard board uses an RP2040 MCU, use these flashing steps

STM32 based boards

To confirm it's in DFU mode you can run the command `lsusb` and look for an entry of "STMicroelectronics STM Device in DFU mode"

```
lsusb
```



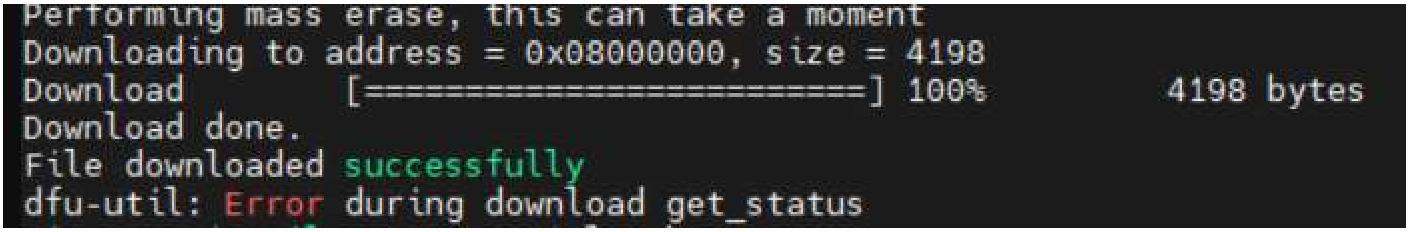
```
Bus 001 Device 002: ID 1d50:606f OpenMoko, Inc. Geschwister Schneider CAN adap
Bus 001 Device 005: ID 0483:df11 STMicroelectronics STM Device in DFU Mode
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

You can then flash the Katapult firmware to your mainboard by running

```
cd ~/katapult
make
sudo dfu-util -R -a 0 -s 0x08000000:mass-erase:force:leave -D ~/katapult/out/katapult.bin -d 0483:df11
```

If the result shows an "Error during download get_status" or something, but above it it still has "File downloaded successfully" then it still flashed OK and you can ignore that error.

> If you get a different error and do **not** see a "File downloaded successfully" then try:
>
> `sudo dfu-util -R -a 0 -s 0x08000000:leave -D ~/katapult/out/katapult.bin -d 0483:df11`



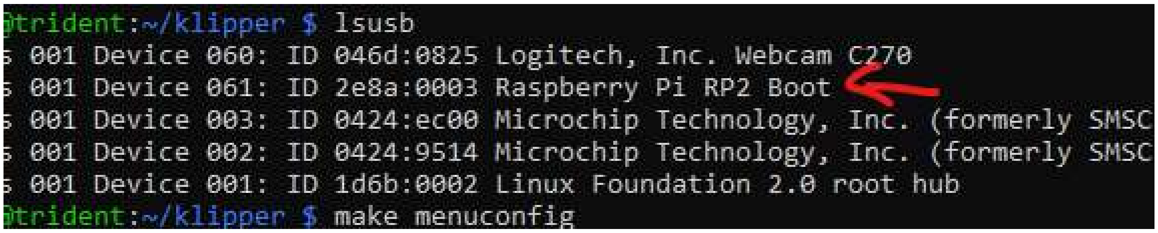
```
Performing mass erase, this can take a moment
Downloading to address = 0x08000000, size = 4198
Download        [=========================] 100%         4198 bytes
Download done.
File downloaded successfully
dfu-util: Error during download get_status
```

Katapult is now installed, click here for the next steps.

RP2040 based boards

To confirm it’s in BOOT mode, run an `lsusb` command and you should see the device as a “Raspberry Pi boot” device (or similar)

```
lsusb
```



```
@trident:~/klipper $ lsusb
s 001 Device 060: ID 046d:0825 Logitech, Inc. Webcam C270
s 001 Device 061: ID 2e8a:0003 Raspberry Pi RP2 Boot
s 001 Device 003: ID 0424:ec00 Microchip Technology, Inc. (formerly SMSC
s 001 Device 002: ID 0424:9514 Microchip Technology, Inc. (formerly SMSC
s 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
@trident:~/klipper $ make menuconfig
```

> Note the address of the usb device => 2e8a:0003

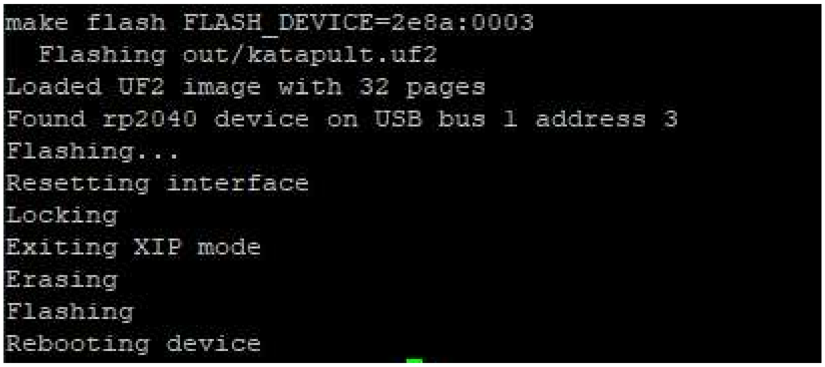
You can then flash the Katapult firmware to your toolhead board by running

```
cd ~/katapult

make flash FLASH_DEVICE=2e8a:0003
```

where the FLASH_DEVICE ID is what you noted down from the `lsusb` command.

It should look something like this if the download was successfull



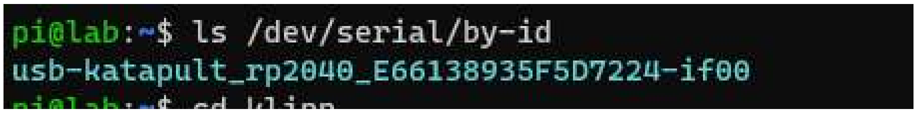
```
make flash FLASH_DEVICE=2e8a:0003
  Flashing out/katapult.uf2
Loaded UF2 image with 32 pages
Found rp2040 device on USB bus 1 address 3
Flashing...
Resetting interface
Locking
Exiting XIP mode
Erasing
Flashing
Rebooting device
```

Katapult is now installed, click here for the next steps.

Katapult is now installed

Katapult should now be successfully flashed. Take out any DFU jumpers on your mainboard (if it needed them) and double-click the reset button on your board. Check that the board is in Katapult mode by running

```
ls /dev/serial/by-id
```



```
pi@lab:~$ ls /dev/serial/by-id
usb-katapult_rp2040_E66138935F5D7224-if00
pi@lab:~$ cd klipp
```

You should see a "usb-katapult_..." device there. If you don't, then double-click the RESET button on your board and `ls /dev/serial/by-id` again.

> If you keep seeing the board, but it shows as "usb-Klipper_..." instead of "usb-katapult_..." even after double-clicking the reset button then you may have a dodgy reset button or aren't getting the double-click timing correct. Either way, another method to kick the board into Katapult mode (assuming the Katapult install from further above was successful) is to run:
>
> ```
> cd ~/klipper/scripts
> ```
>
> then
>
> ```
> ~/klippy-env/bin/python -c 'import flash_usb as u; u.enter_bootloader("<serialID>")'
> ```
>
> replacing `<serialID>` with the usb_Klipper full path name that your board is showing up as. eg: `~/klippy-env/bin/python -c 'import flash_usb as u; u.enter_bootloader("/dev/serial/by-id/usb-Klipper_stm32h723xx_110032000D51323532393433-if00")'`

If you do *not* see a Katapult device listed in your /dev/serial/by-id, or if you get a `cannot access '/dev/serial/by-id': No such file or directory` then your mainboard *isn't* currently sitting in Katapult mode. Double-click the reset button on your mainboard then `ls /dev/serial/by-id`

> again. If you still don't see a Katapult device then either the flash didn't work or you had incorrect settings in the Katapult `make menuconfig` screen. Go back and try again.

As you are installing Katapult onto the mainboard that you are also going to use for USB-CAN-Bridge mode klipper, you still will *not* have a working CAN network at this stage. You can flash klipper to your mainboard via Katapult, but in reality it is flashing over USB and not flashing over CAN.

Flashing klipper via Katapult will be covered shortly.

Installing USB-CAN-Bridge Klipper

Move into the klipper directory on the Pi by running:

```
cd ~/klipper
```

Then go into the klipper configuration menu by running:

```
make menuconfig
```

Again, if your mainboard is already in Common Mainboard Hardware then you can copy the Klipper settings from there.

Otherwise, you want the Processor and Clock Reference to be set as per whatever board you are running. Set Communication interface to 'USB to CAN bus bridge' then set the CAN Bus interface to use the pins that are specific to your mainboard. Also set the CAN bus speed to the same as the speed in your can0 file. In this guide it is set to 1000000.

Once you have the firmware configured, run a `make clean` to make sure there are no old files hanging around, then `make` to compile the firmware. It will save the firmware to ~/klipper/out/klipper.bin

```
make clean
make
```

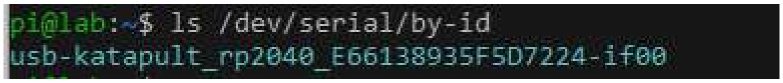
Using Katapult to flash Klipper

Stop the Klipper service on the Pi by running:

```
sudo service klipper stop
```

Run an `ls /dev/serial/by-id/` and take note of the Katapult device that it shows:

```
ls /dev/serial/by-id/
```



```
pi@lab:~$ ls /dev/serial/by-id
usb-katapult_rp2040_E66138935F5D7224-if00
```

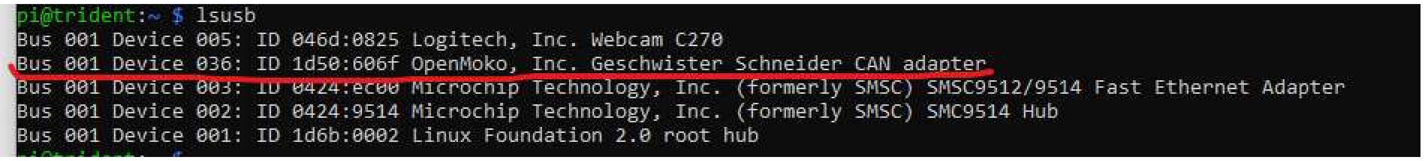
If the above command didn't show a 'katapult' device, or threw a "no such file or directory" error, then quickly double-click the RESET button on your mainboard and run the command again. Until you get a result from a `ls /dev/serial/by-id/` there is no point doing further steps below.

Run this command to install klipper firmware via Katapult via USB. Use the device ID you just retrieved in the above ls command.

```
python3 ~/katapult/scripts/flashtool.py -f ~/klipper/out/klipper.bin -d /dev/serial/by-id/usb-katapult_you
```

Klipper is now installed

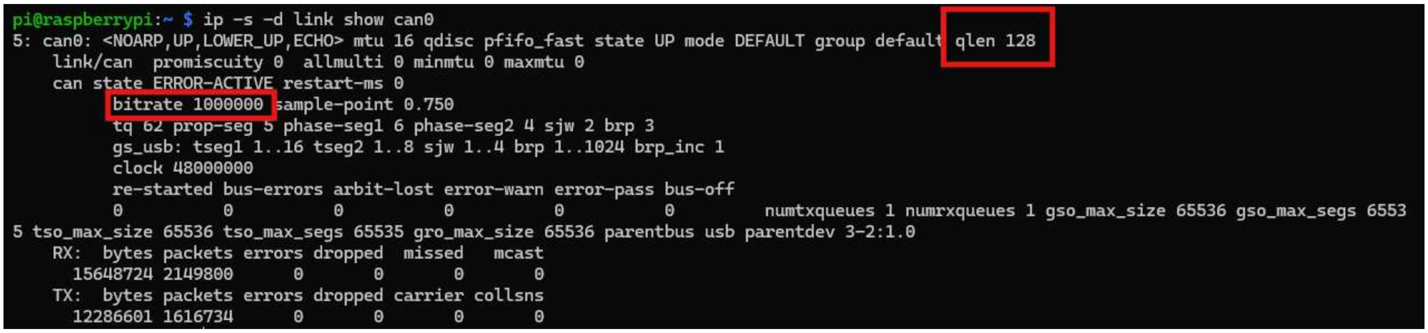
This should have now installed klipper firmware to your mainboard. You can verify by running `lsusb` and you should see a "Geschwister Schneider CAN adapter" or similar device.



```
pi@trident:~ $ lsusb
Bus 001 Device 005: ID 046d:0825 Logitech, Inc. Webcam C270
Bus 001 Device 036: ID 1d50:606f OpenMoko, Inc. Geschwister Schneider CAN adapter
Bus 001 Device 003: ID 0424:ec00 Microchip Technology, Inc. (formerly SMSC) SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Microchip Technology, Inc. (formerly SMSC) SMC9514 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Check that the can0 interface is up by running `ip -s -d link show can0`. If everything is correct you will see something like this:

```
ip -s -d link show can0
```



```
pi@raspberrypi:~ $ ip -s -d link show can0
5: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode DEFAULT group default qlen 128
    link/can  promiscuity 0  allmulti 0 minmtu 0 maxmtu 0
    can state ERROR-ACTIVE restart-ms 0
          bitrate 1000000 sample-point 0.750
          tq 62 prop-seg 5 phase-seg1 6 phase-seg2 4 sjw 2 brp 3
          gs_usb: tseg1 1..16 tseg2 1..8 sjw 1..4 brp 1..1024 brp_inc 1
          clock 48000000
          re-started bus-errors arbit-lost error-warn error-pass bus-off
          0          0          0          0          0          0         numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 6553
5 tso_max_size 65536 tso_max_segs 65535 gro_max_size 65536 parentbus usb parentdev 3-2:1.0
    RX:  bytes packets errors dropped  missed   mcast
      15648724 2149800      0       0       0       0
    TX:  bytes packets errors dropped carrier collsns
      12286601 1616734      0       0       0       0
```

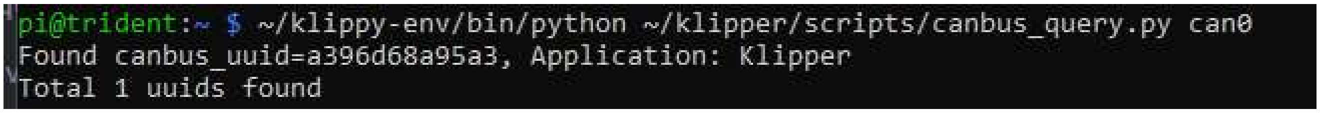
You see a can0 interface, the “qlen” will be 128, and the bitrate will be 1000000

If the `ip -s -d link show can0` command returns an error (eg. “Device can0 does not exist) then reboot your Pi with `sudo reboot now` and once the Pi is back up check `ip -s -d link show can0` again. If you still get the error then your mainboard isn’t showing as a CAN adapter and you need to go back to the Installing USB-CAN-Bridge Klipper and try again, making sure the Klipper `make menuconfig` settings are absolutely correct.

If the can0 network shows up, but the qlen *isn’t* 128 or the bitrate *isn’t* 1000000 then go back to Getting_Started and check the can0 file settigns in both the ifupdown section and the netplan section.

You can now run the Klipper canbus query to retrieve the canbus_uuid of your mainboard:

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



```
pi@trident:~ $ ~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0
Found canbus_uuid=a396d68a95a3, Application: Klipper
Total 1 uuids found
```

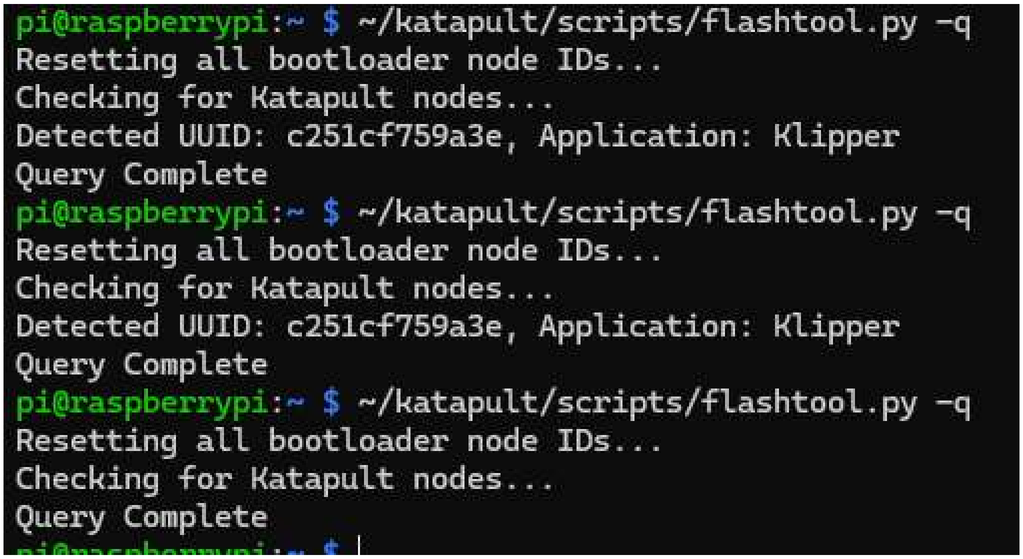
Use this UUID in the [mcu] section of your printer.cfg in order for Klipper (on Pi) to connect to the mainboard.

Start the Klipper service on the Pi again by running:

```
sudo service klipper start
```

Dissappearing Mainboard UUID

There is a quirk with USB-CAN-Bridge klipper if you don't have any second/other CAN nodes connected up (either due to not having done that step yet, or if there is something wrong with the toolhead/other CAN device or wiring) where the Mainboard UUID will stop showing up to a query and can cause your web interface/klipper logs to show connection errors to 'mcu'.



```
pi@raspberrypi:~ $ ~/katapult/scripts/flashtool.py -q
Resetting all bootloader node IDs...
Checking for Katapult nodes...
Detected UUID: c251cf759a3e, Application: Klipper
Query Complete
pi@raspberrypi:~ $ ~/katapult/scripts/flashtool.py -q
Resetting all bootloader node IDs...
Checking for Katapult nodes...
Detected UUID: c251cf759a3e, Application: Klipper
Query Complete
pi@raspberrypi:~ $ ~/katapult/scripts/flashtool.py -q
Resetting all bootloader node IDs...
Checking for Katapult nodes...
Query Complete
pi@raspberrypi:~ $
```

If you are still at this early stage of your flashing journey where you haven't yet set up/connected the toolhead, all it means is if you keep doing a CAN query the mainboard UUID will stop showing. Don't be alarmed, it's normal. Once you have the second CAN device connect you won't see this behaviour, and it won't stop the mainboard from still acting as a USB-CAN-Bridge for your Pi.

However, if you already *had* a working CAN toolhead setup and you see this problem, then the dissappearing UUID is **not** the cause of any errors/issues with your system. It can be a *symptom* of a problem, **not** the cause. Don't get lost trying to troubleshoot why the mainboard UUID dissappears. It's fine. Just hit the reset button on your mainboard (once), or power cycle the printer, and the mainboard UUID will return for at least a bit.

Next Step

Now that your mainboard is fully flashed, and you have a working can0 CANBus network, the next step is to flash your toolhead board.