
Printed parts:

BTT HDMI 5 mount (you'll need some hardware but the kit has plenty enough)

Nozzle brush mount

PG7 mount gantry

Y endstop relocation (you'll need to use 2 of the bolts mounting the B motor and use 4mm longer bolts)
This one will give you a couple more mm in Y

PG7 mount SB

SB main body for SB2209 (specifically the STEP file, the STL is wrong as of 05/12/24)

Cable cover for PCB

TAP printed parts (You will NOT need to print the normal X gantry or SB mounting parts such as the main body)

M8P din rail mount (be sure to mount your din rails parallel to bed extrusions or the wires won't be long enough. This did NOT affect me on the 250mm)

Annex panel clips (You will still need the front and bottom panel mounts from Voron)

Nevermore V6 (If you print the v4 you wont need the additional wago clips)

Gantry installation hooks (you will need m3 nuts and m3 hardware but the kit has plenty)

You do want to print these optional files contained in the Voron parts:

Tools:
MGN9 rail guide x2
MGN12 rail guide x2
Pulley Jig

SB tools:
5015 cutting jig A and B

Tap:
MGN9 assembly tool

Things I needed that WEREN'T in the kit:

Ring terminals

WAGO Lever nut 2-connector - you don't need these if you go with the V4 nevermore

Mobilux EP2 - you *need* to grease your rails

Plastic syringes

Things I used that you don't NEED but I opted for:

Aluminum squares

Hex drivers

Calipers

Crimping tool

Rulers

Better magnets

Black fasteners for certain locations

24V LED strip

Docs you will need:

M8P doc
You'll mostly use this for the pinout in your printer.cfg

BTT SB2209 RP2040 doc
You'll mostly use this for the pinout in your printer.cfg

Formbot wiring diagram
There are a few errors - I made some corrections here

Voron tap instructions
Remember you won't need the traditional X-carriage and toolhead mounting

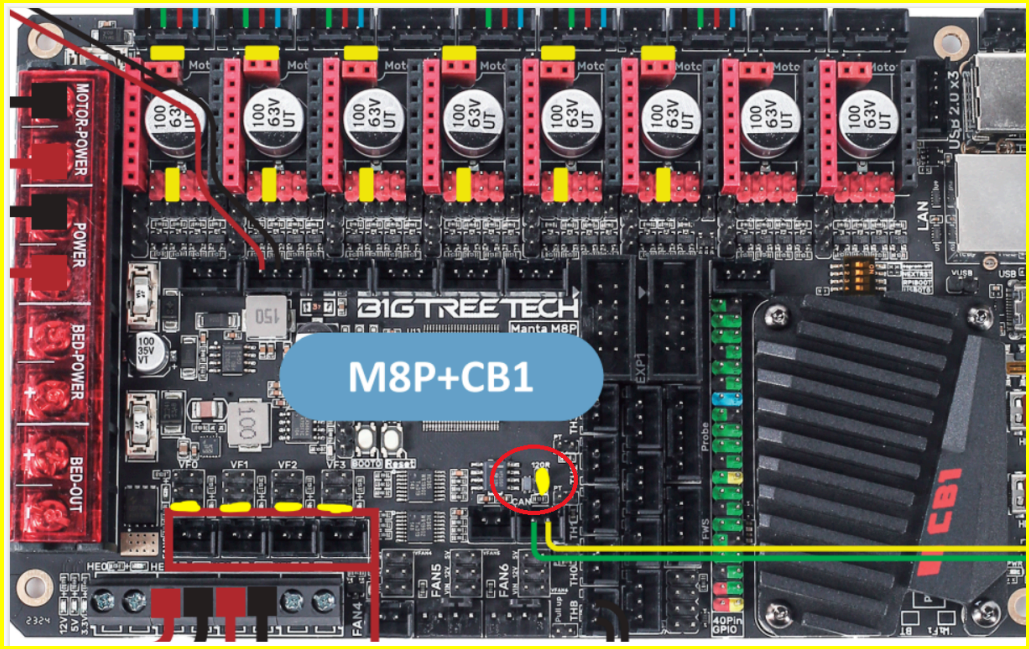
Voron 2.4 docs

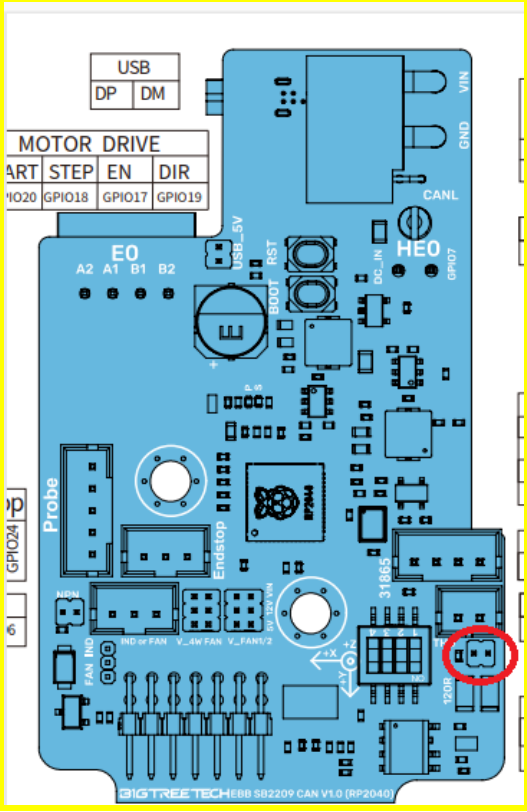
Stealthburner docs
You will need the printed parts above and won't need the default parts they replace

---

BEFORE YOU START

PLUG IN THE 120Ohm JUMPER ON THE M8P AND THE SB2209





Also, be sure to mount your din rails parallel to bed extrusions or the wires won't be long enough

Software setup including SB2209 (RP2040) CAN:

THIS IS ALL VALID FOR THE M8P V1.1 - THIS IS WHAT CAME WITH MY KIT

Install Amrbian

https://www.armbian.com/bigtreetech-cb1/

Use Etcher to flash SD card

https://etcher.balena.io/

Boot up the machine with the CAN cable disconnected

Setup Armbian with default password 1234

Reboot the machine and log into the account you created

Update dependencies

```
sudo apt update
sudo apt upgrade
sudo apt install python3 python3-pip python3-can
```

Install Klipper, Moonraker, Mainsail, and Klipperscreen with KIAUH

```
sudo apt-get update && sudo apt-get install git -y
cd ~ && git clone https://github.com/dw-0/kiauh.git
./kiauh/kiauh.sh
```

Install Katapult

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

Setup the network file - THE LAST TWO LINES SHOULD BE TABBED OVER

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

allow-hotplug can0
```

```
iface can0 can static
        bitrate 1000000
        up ip link set can0 txqueuelen 1024
```

May need to run this as well

```
sudo nano /etc/systemd/network/10-can.link

[Match]
Type=can

[Link]
TransmitQueueLength=1024

sudo nano /etc/systemd/network/25-can.network

[Match]
Name=can*

[CAN]
BitRate=1M
```

Create Katapult firmware file for M8P

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

```
(Top)
                    Katapult Configuration v0.0.1-64-g3e23332
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32G0B1)  --->
    Build Katapult deployment application (8KiB bootloader)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (CAN bus (on PD12/PD13))  --->
    Application start offset (8KiB offset)  --->
(1000000) 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
```

```
make clean
make
```

Create Klipper firmware file for M8P

```
cd ~/klipper
make menuconfig
```

```
(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 (USB to CAN bus bridge (USB on PA11/PA12))  --->
    CAN bus interface (CAN bus (on PD12/PD13))  --->
    USB ids  --->
(1000000) 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
```

```
make clean
```

```
make
```

Hold down boot0, hit reset, let go of boot0 - puts M8P in DFU mode. Replace the text in red with YOUR ID from lsusb

```
lsusb
```

```
sudo dfu-util -a 0 -D ~/katapult/out/katapult.bin --dfuse-address 0x08000000:force:leave -d 0483:df11
```

Hit reset on the M8P
Go back to DFU mode
Go to klipper directory - Replace the text in red with YOUR ID from lsusb

```
cd ~/klipper
sudo dfu-util -a 0 -d 0483:df11 --dfuse-address 0x08002000 -D ~/klipper/out/klipper.bin
```

Hit reset on the board

Bring up CAN network

```
sudo ifup can0
```

Use

```
lsusb
```

Should see the M8P as a CAN adapter

```
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 011: ID 1d50:606f OpenMoko, Inc. Geschwister Schneider CAN adapte
r
Bus 002 Device 003: ID 04d9:8030 Holtek Semiconductor, Inc. BTT-HDMI5
Bus 002 Device 002: ID 1a40:0101 Terminus Technology Inc. Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Query to see if M8P shows up with a UUID

```
python3 ~/katapult/scripts/flash_can.py -q
```

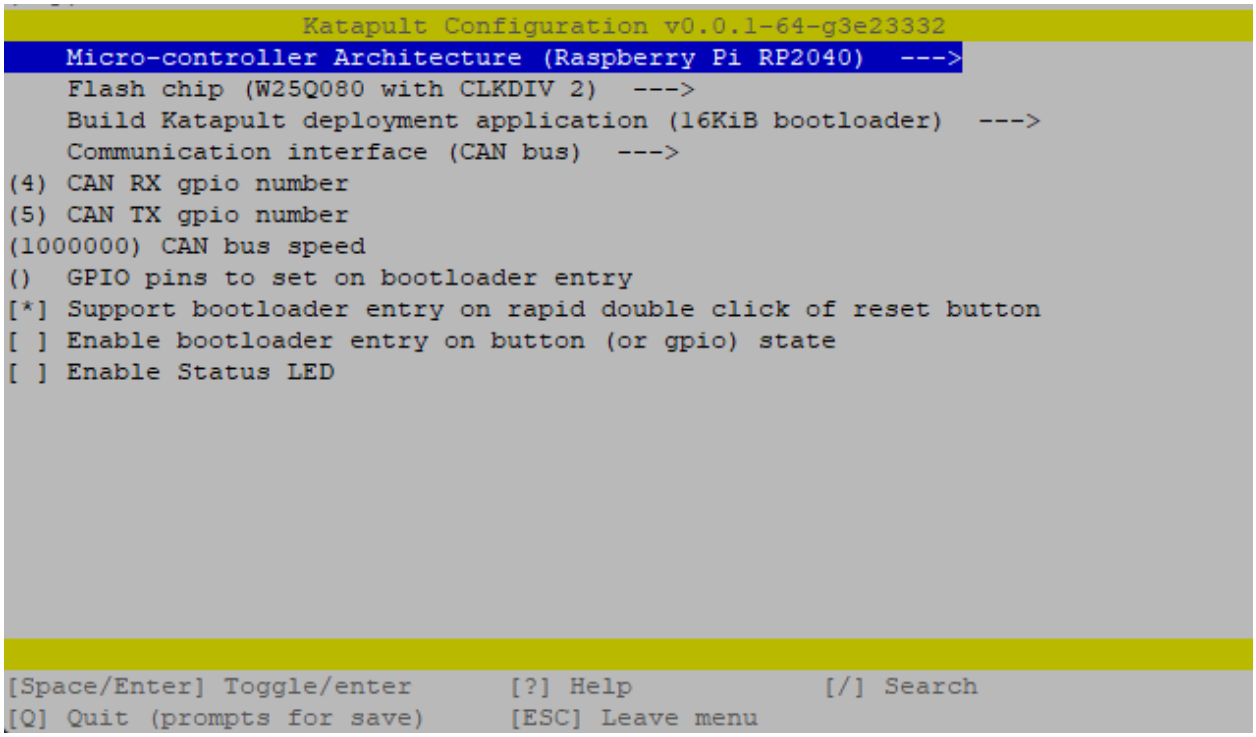
Should see the UUID to put into your printer.cfg file

Plug in the 5V jumper and the USB cable to EBB2209

Setup Katapult firmware file for EBB2209

```
cd ~/katapult
```

```
make menuconfig
```



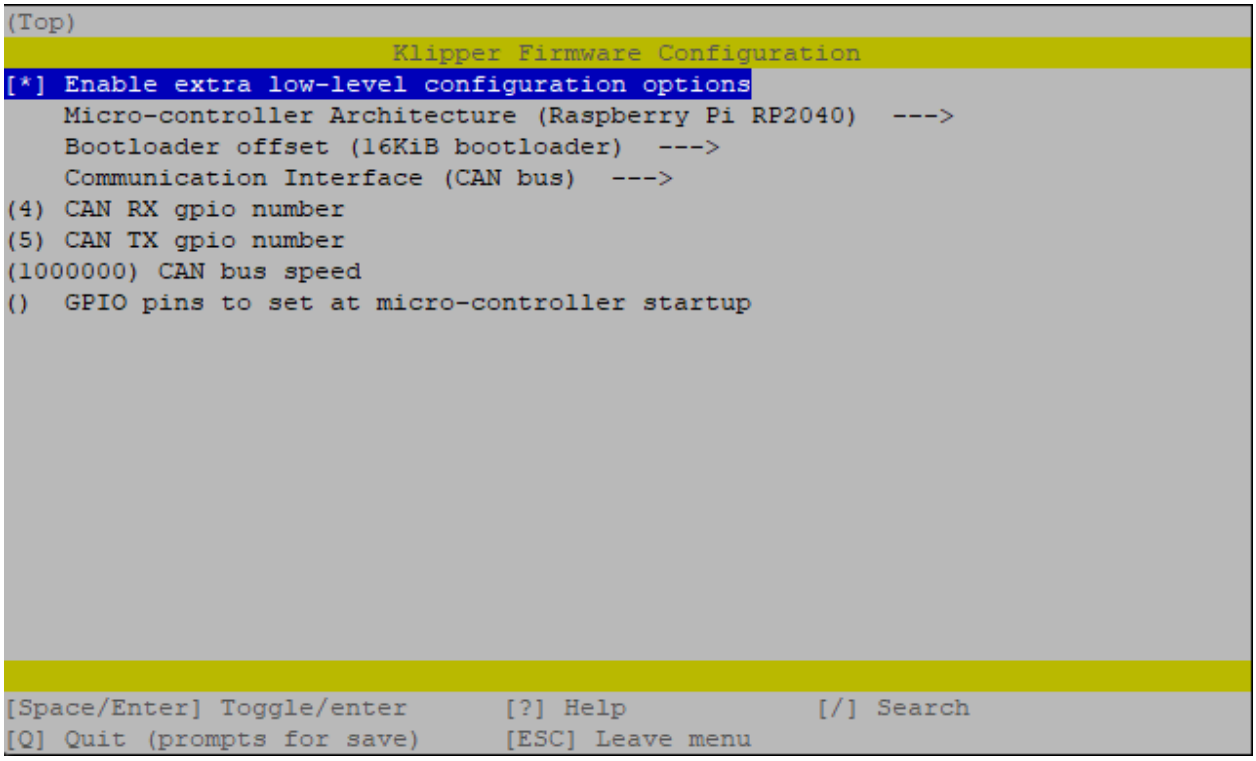

```
make clean
```

```
make
```

Setup Klipper firmware for EBB2009

```
cd ~/klipper
```

```
make menuconfig
```




```
make clean
```

```
make
```

Put the board into DFU mode with “boot0” and “reset” buttons

```
lsusb
```

Should now see the device in boot mode - use the ID you get with “lsusb” in red below

```
cd ~/katapult
```

```
make flash FLASH_DEVICE=2e8a:0003
```

Shutdown the printer

```
sudo shutdown now
```

Remove jumper and USB cable - install the CAN cable

Boot the machine back up

```
python3 ~/katapult/scripts/flashtool.py -i can0 -q
```

Should see a device with klipper and katapult

The Katapult device is the toolhead board and the UUID we want - use the UUID YOU GET in red

```
cd ~/klipper
```

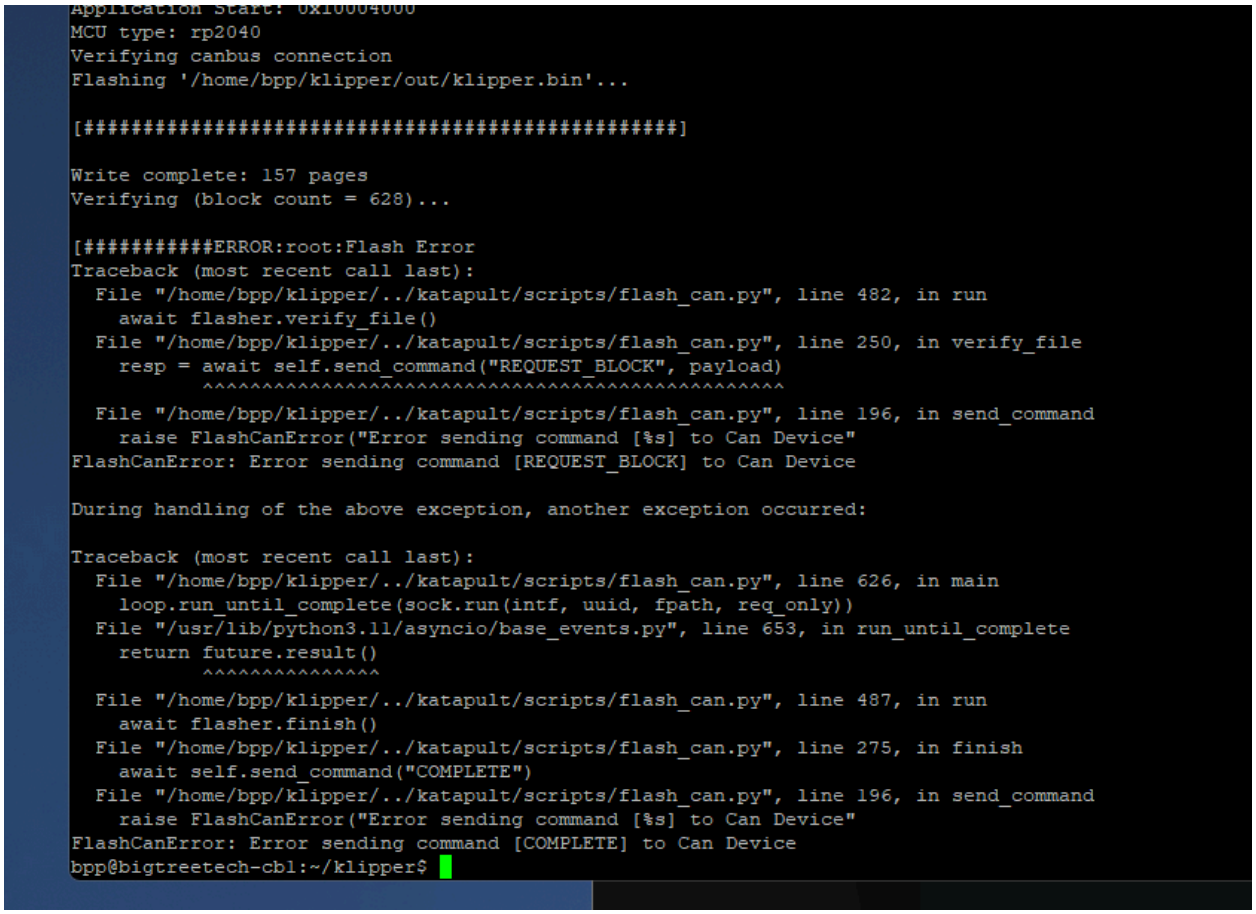
Now we flash klipper

```
sudo service klipper stop

python3 ~/katapult/scripts/flashtool.py -i can0 -q

python3 ~/katapult/scripts/flashtool.py -i can0 -u b6d9de35f24f -f ~/klipper/out/klipper.bin

python3 ~/katapult/scripts/flashtool.py -i can0 -q
```

Should see the device in klipper mode

****

You might see this error when you flash klipper onto the SB2209



That shouldn’t be an issue

Get the UUIDs

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

```
sudo service klipper start
```

Use the UUIDs in your Printer.cfg