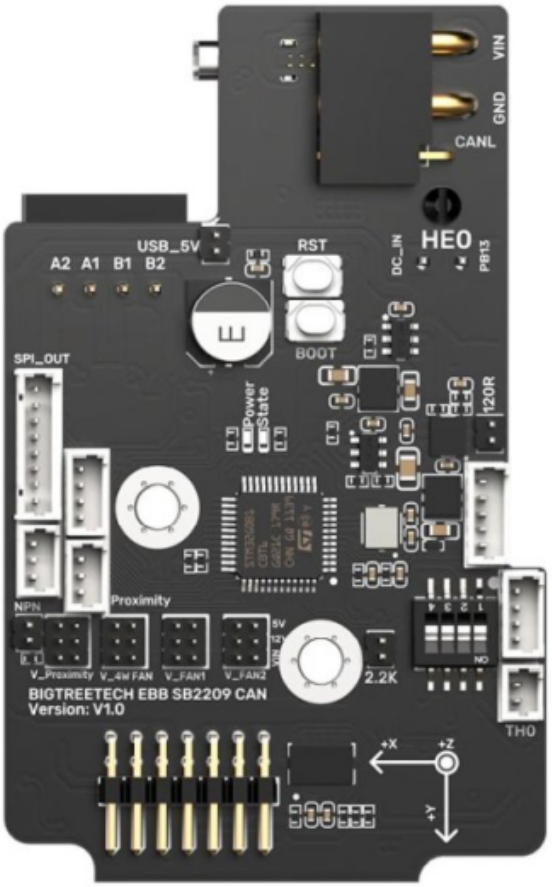
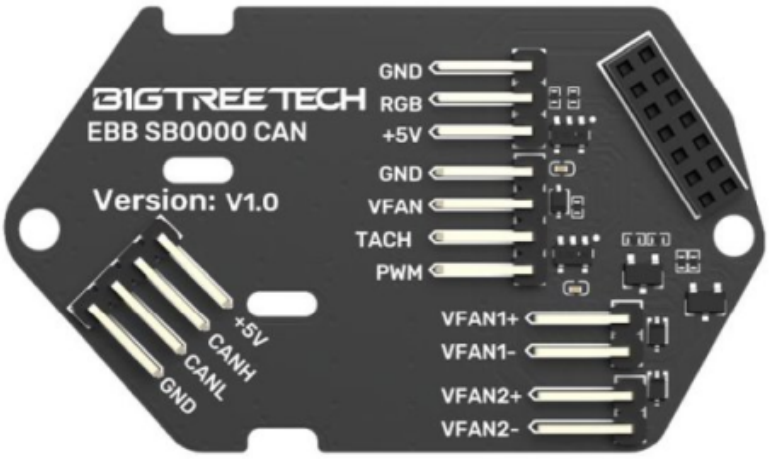
EBB SB2209 (CAN)

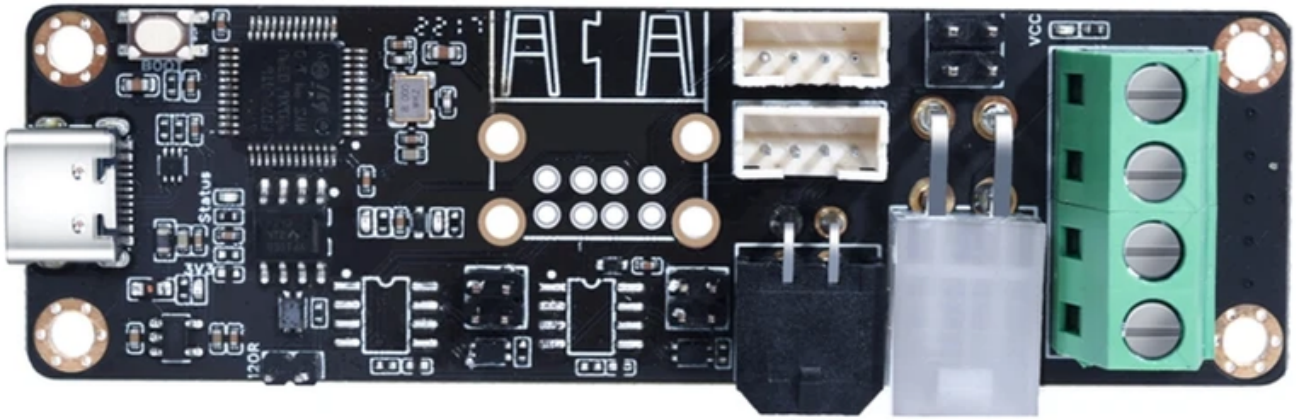
Schrittweise Anleitung, um das Bigtree Tech **EBB SB2209** Kopfboard mit dem Buskoppler **U2C** über **CAN** in Betrieb zu nehmen.

EBB SB2209





U2C



YouTube Video #68

Hinweise

- **SBC** bedeutet in der Anleitung **S**ingle **B**oard **C**omputer. Also meistens wohl ein Raspberry Pi.
- **U2C** ist der BTT Buskoppler U2C
- Es wird davon ausgegangen das auf dem SBC Klipper und MainSail eingerichtet ist.
- Ein Zugang zum SBC über SSH ist notwendig!

U2C

- Bei neuen U2C Buskopplern wäre zu überlegen ein Firmware Update einzuspielen.
  siehe dazu USB Buskoppler
- Im Terminal folgendes starten
  `dmesg -HW`
- Jetzt den U2C anstecken

```
pi@Pi3Test:~/printer_data/config $ dmesg -HW
[Nov11 14:09] usb 1-1.4: new full-speed USB device number 5 using
dwc_otg
[  +0.133404] usb 1-1.4: New USB device found, idVendor=1d50,
idProduct=606f, bcdDevice= 0.00
[  +0.000037] usb 1-1.4: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[  +0.000016] usb 1-1.4: Product: candleLight USB to CAN adapter
[  +0.000013] usb 1-1.4: Manufacturer: bytewerk
[  +0.000011] usb 1-1.4: SerialNumber: 0050004A5542501720393839
[  +0.002168] gs_usb 1-1.4:1.0: Configuring for 1 interfaces
```

  - Es sollte etwas mit `candleLight` gefunden werden
- STRG+C drücken, um die Meldungen zu beenden

SBC

- Interface einrichten
  Achtung : die Bitrate von 1000000 muss auch in der Board Firmware eingestellt werden!
  `sudo nano /etc/network/interfaces.d/can0`

folgendes eintragen, speichern und mit STRG + x, dann Y, dann Enter beenden

```
auto can0
iface can0 can static
    bitrate 1000000
    up ifconfig $IFACE txqueuelen 1024
```

- Testen mit ip a
  can0: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state **UP** group default qlen 1024

Stromversorgung

Auf dem Board gibt es einen Mikro Jumper, den man setzen kann:

Jumper	Stromversorgung Controller	Verwendung
gesetzt	5V Stromversorgung über USB Anschluss	Board liegt “auf dem Tisch”
NICHT gesetzt	5V Stromversorgung über 12/24V Stecker	Board ist im Kopf eingebaut

Als Faustregel im Betrieb den 5V Jumper immer abziehen!

Katapult download

- cd ~ && ls
- Wenn kein katapult Ordner vorhanden ist
  git clone https://github.com/Arksine/katapult
- cd katapult

Katapult aufspielen

- make menuconfig

```
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32G0B1)  --->
    Build Katapult deployment application (Do not build)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (CAN bus (on PB0/PB1))  --->
    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
```

    - beenden mit Q und Y
- Katapult kompilieren
  `make -j4`
- Das Board in den DFU Modus bringen
    - `dmesg -HW`
    - Die **Boot Taste** am Board gedrückt halten und dabei das Board über USB mit dem SBC verbinden

      Wenn das Board schon angeschlossen ist geht auch folgendes
      Boot Taste gedrückt halten, **RST Taste** einmal drücken und loslassen, dann die **Boot Taste** loslassen.
    - Das Boiard meldet sich mit **`Product: DFU in FS Mode`**

```
pi@Pi3Test:~/katapult $ dmesg -HW
[Nov11 07:34] usb 1-1.4: new full-speed USB device number 17 using
dwc_otg
[  +0.133427] usb 1-1.4: New USB device found, idVendor=0483,
idProduct=df11, bcdDevice= 2.00
[  +0.000037] usb 1-1.4: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[  +0.000016] usb 1-1.4: Product: DFU in FS Mode
[  +0.000013] usb 1-1.4: Manufacturer: STMicroelectronics
[  +0.000012] usb 1-1.4: SerialNumber: 206A39785542
```

    - STRG+C drücken, um die Meldungen zu beenden
- Katapult aufspielen mit
  `dfu-util -R -a 0 -s 0x08000000:mass-erase:force -D ~/katapult/out/katapult.bin`
  Das Ergebnis sollte sein `File downloaded successfully`
- ggf. USB Kabel abziehen
- **RST Taste** drücken um das Board einmal zu resetten

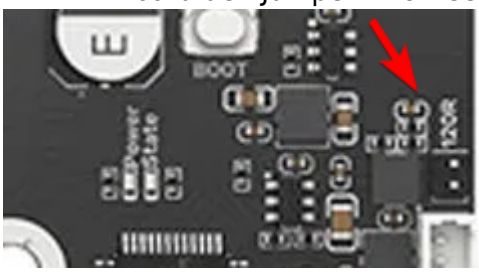
Verkabelung

Ganz normale CAN Verkabelung:

- CAN H auf CAN H
- CAN L auf CAN L
- GND & 12 bzw. 24V

CAN Bus Terminierung

- Am U2C Board den Jumper **120R** setzen

- Am EBB Board den Jumper **120R** setzen

- Wer will, kann im **ausgeschalteten Zustand** den Buswiderstand mit einem Ohmmeter messen. Es müsste zwischen CAN H und CAN L ca. 60 Ω ergeben.

Can Query

> **Hinweis**
> Die folgenden Schritte setzen natürlich voraus, das der CAN Bus korrekt im Vorfeld eingerichtet wurde!

Wenn das Board über CAN verbunden ist, dann kann man mit den folgenden Schritten prüfen, ob Katapult geflasht wurde:

- Klipper Dienst stoppen
  `sudo systemctl stop klipper.service`
- `~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0`
  Wenn ein Board gefunden wird, dann sollte folgende Ausgabe erscheinen:

```
biqu@BTT-PI12:~/katapult$ ~/klippy-env/bin/python
~/klipper/scripts/canbus_query.py can0
Found canbus_uuid=539892be834d, Application: CanBoot
Total 1 uuids found
```

  - Die **UUID** (canbus_uuid=**539892be834d**) notieren !
  - Wird bei diesem Schritt kein Board gefunden, hilft oft ein Reset am Board (entweder über Reset Taster oder 1x Strom weg und wieder dran)

Klipper per Katapult

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

```
[*] 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 (CAN bus (on PB0/PB1))  --->
(1000000) CAN bus speed
()  GPIO pins to set at micro-controller startup
```

  - beenden mit Q und Y
- Klipper kompilieren
  `make -j4`
- Klipper flashen
  `~/klippy-env/bin/python3 ~/katapult/scripts/flash_can.py -i can0 -f ~/klipper/out/klipper.bin -u <uuid>`
  In diesem Beispiel wäre das dann
  `~/klippy-env/bin/python3 ~/katapult/scripts/flash_can.py -i can0 -f ~/klipper/out/klipper.bin -u` **`539892be834d`**
  - Das sollte mit einem `Flash Success` beendet werden!
- kurzer Test
  `~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0`
  Sollte jetzt `… Application: Klipper` liefern
- Klipper Dienst wieder starten
  `sudo systemctl start klipper.service`

Konfig

- `cd ~/printer_data/config`
- `wget https://raw.githubusercontent.com/bigtreetech/EBB/master/EBB%20SB2240_2209%20CAN/sample-bigtreetech-ebb-sb-canbus-v1.0.cfg -O BTT_EBB_SB2209.cfg`
- `nano ~/printer_data/config/BTT_EBB_SB2209.cfg`

```
[mcu EBBCan]
#serial: /dev/serial/by-id/usb-Klipper_Klipper_firmware_12345-if00
canbus_uuid: 539892be834d
```

  - Die Zeile mit `serial` auskommentieren oder löschen
  - Die UUID bei `canbus_uuid` eintragen und ggf. die # am Anfang der Zeile entfernen (UUID ermitteln siehe CAN Query)!
- In der `printer.cfg` noch die neue Konfig einbetten
  `[include BTT_EBB_SB2209.cfg]`

Klipper Update

Das Update erfolgt genauso wie das erste Bespielen (Klipper per Katapult) mit Klipper.

- Vor dem Firmware Update natürlich erstmal Klipper in Mainsail updaten (Machine Seite)!
- Klipper Dienst stoppen
  `sudo systemctl stop klipper.service`
- Klipper Kompilieren
  siehe Klipper per Katapult
- Klipper aufspielen
  `~/klippy-env/bin/python3 ~/katapult/scripts/flash_can.py -i can0 -f ~/klipper/out/klipper.bin -u <uuid>`
  - Die UUID von eurem Board eintragen!
- Klipper Dienst starten
  `sudo systemctl start klipper.service`

Links

- https://github.com/bigtreetech/EBB
- Klipper Konfig
  https://github.com/bigtreetech/EBB/blob/master/EBB%20SB2240_2209%20CAN/sample-bigtreetech-ebb-sb-canbus-v1.0.cfg

From:
https://drklipper.de/ - **Dr. Klipper Wiki**

Permanent link:
https://drklipper.de/doku.php?id=klipper_faq:flash_guide:stm32g0b1:ebb_sb2209_can&rev=1699732572

Last update: **2023/11/11 20:56**