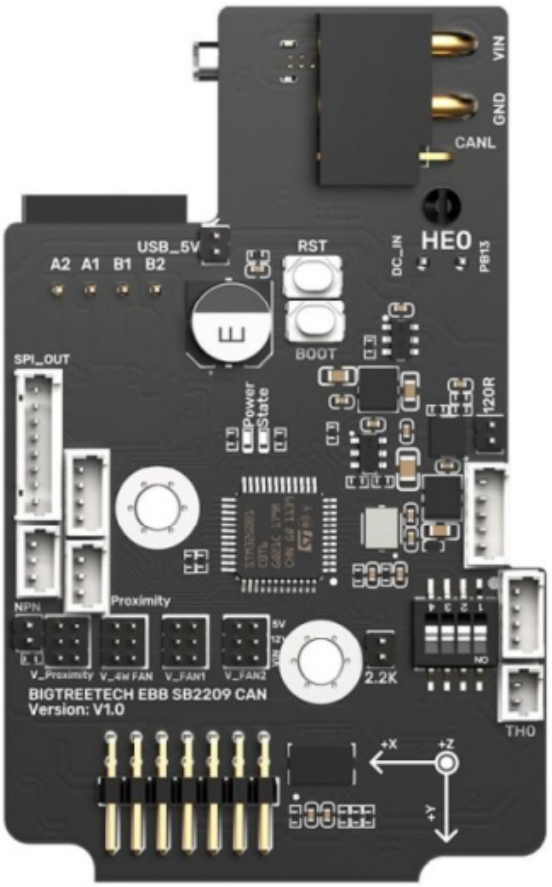
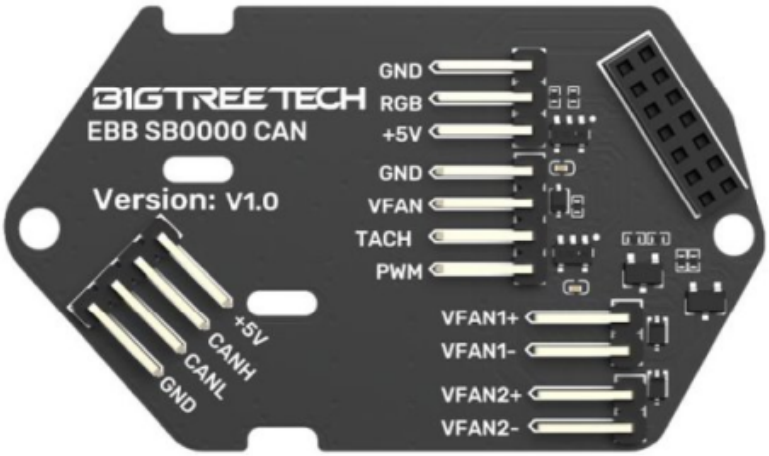
# EBB SB2209 RP (CAN)

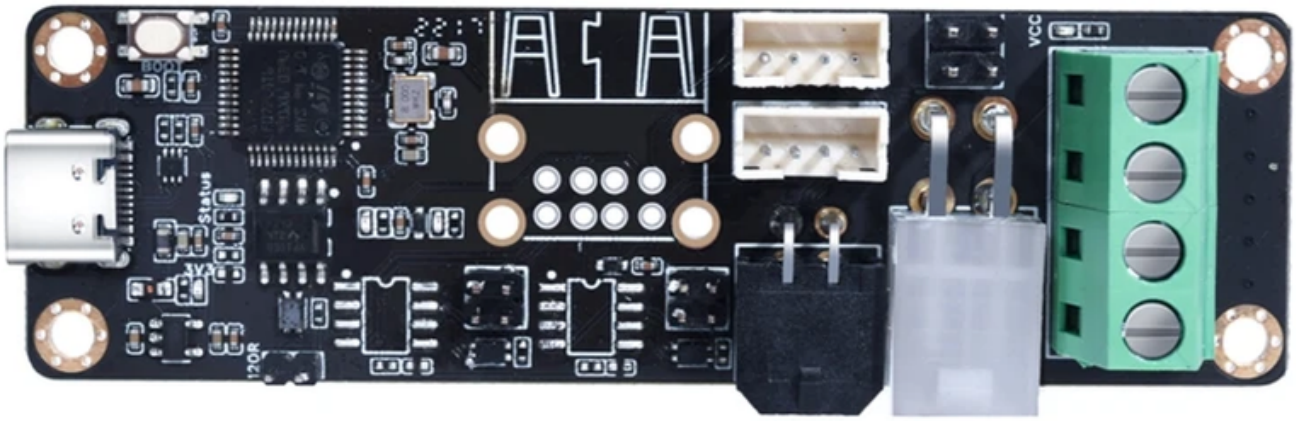
Schrittweise Anleitung, um das Bigtree Tech **EBB SB2209** Kopfboard mit dem Buskoppler **U2C** über **CAN** in Betrieb zu nehmen.

**EBB SB2209**




**U2C**




## YouTube Video #74




# 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`

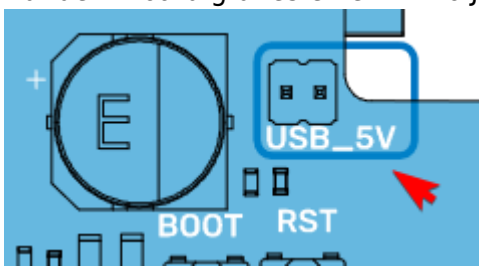
folgendes eintragen, speichern und mit STRG + x, dann Y, dann Enter beenden

```
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

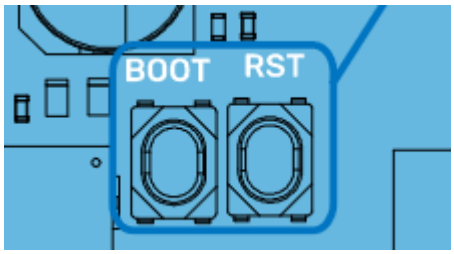
# Katapult aufspielen

- make menuconfig

```
    Micro-controller Architecture (Raspberry Pi RP2040)  --->
    Flash chip (W25Q080 with CLKDIV 2)  --->
    Build Katapult deployment application (Do not build)  --->
    Communication interface (CAN bus)  --->
(4) CAN RX gpio number
(5) CAN TX gpio number
(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
(gpio26) Status LED GPIO Pin
```

    - beenden mit Q und Y
- Katapult kompilieren
  `make -j4`
- Das Board in den Boot Loader 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 Board meldet sich mit **`Product: RP2 Boot`**

```
[Dec13 10:07] usb 1-1.4: new full-speed USB device number 44 using
dwc_otg
[  +0.132594] usb 1-1.4: New USB device found, idVendor=2e8a,
idProduct=0003, bcdDevice= 1.00
[  +0.000035] usb 1-1.4: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[  +0.000016] usb 1-1.4: Product: RP2 Boot
[  +0.000013] usb 1-1.4: Manufacturer: Raspberry Pi
[  +0.000012] usb 1-1.4: SerialNumber: E0C9125B0D9B
[  +0.003555] usb-storage 1-1.4:1.0: USB Mass Storage device
detected
[  +0.001922] scsi host2: usb-storage 1-1.4:1.0
[  +1.002900] scsi 2:0:0:0: Direct-Access     RPI      RP2
3    PQ: 0 ANSI: 2
[  +0.000671] sd 2:0:0:0: Attached scsi generic sg0 type 0
[  +0.000792] sd 2:0:0:0: [sdc] 262144 512-byte logical blocks:
(134 MB/128 MiB)
[  +0.000871] sd 2:0:0:0: [sdc] Write Protect is off
[  +0.000020] sd 2:0:0:0: [sdc] Mode Sense: 03 00 00 00
[  +0.000980] sd 2:0:0:0: [sdc] No Caching mode page found
[  +0.000022] sd 2:0:0:0: [sdc] Assuming drive cache: write
through
[  +0.013172]  sdc: sdc1
[  +0.000934] sd 2:0:0:0: [sdc] Attached SCSI removable disk
```

    - STRG+C drücken, um die Meldungen zu beenden
- Katapult aufspielen mit

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

Das Ergebnis sollte sein:

```
pi@Pi3Test:~/katapult $ make flash FLASH_DEVICE=2e8a:0003
  Flashing out/katapult.uf2
[sudo] password for pi:
Loaded UF2 image with 31 pages
Found rp2040 device on USB bus 1 address 44
Flashing...
Resetting interface
Locking
Exiting XIP mode
Erasing
Flashing
Rebooting device
```

- USB Kabel abziehen - ab jetzt wird das Board nur noch über den CAN Bus angesprochen!
- **RST Taste** drücken um das Board einmal zu resetten

## Verkabelung

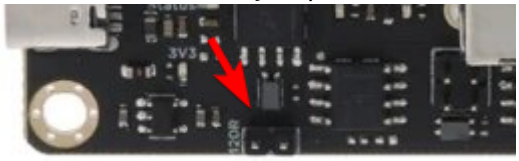
Ganz normale CAN Verkabelung:

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

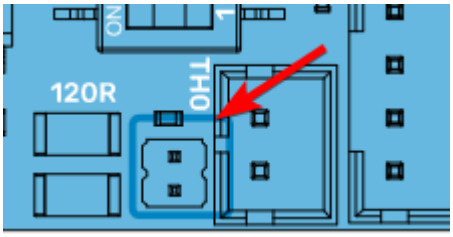



## CAN Bus Terminierung

- 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**
> Wenn Katapult gestartet wurde sollte die **rote LED** auf dem SB2209 blinken.
>
> **Wichtig**
> 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:

```
pi@Pi3Test:~ $ ~/klippy-env/bin/python
~/klipper/scripts/canbus_query.py can0
Found canbus_uuid=73be55aa184d, Application: CanBoot
Total 1 uuids found
```

  - Die **UUID** (canbus_uuid=**73be55aa184d**) 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 (Raspberry Pi RP2040)  --->
    Bootloader offset (16KiB bootloader)  --->
    Communication interface (CAN bus)  --->
(4) CAN RX gpio number
(5) CAN TX gpio number
(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 73be55aa184d`
    - 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`

# Klipper Test

- `~/klippy-env/bin/python ~/klipper/klippy/console.py -c can0 73be55aa184d`
    - Die **UUID** von eurem Board eintragen!
- Sollte unter anderem ein
  `==================== connected ====================`
  liefern
- Abbrechen kann man den Test mit STRG + C

# Konfig

- `cd ~/printer_data/config`
- `wget "https://raw.githubusercontent.com/bigtreetech/EBB/master/EBB%20SB2209%20CAN%20(RP2040)/sample-bigtreetech-ebb-sb-rp2040-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: 73be55aa184d
```

    - 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

- Vor dem Firmware Update natürlich erstmal Klipper in Mainsail updaten (Machine Seite)!
- Klipper Dienst stoppen
  `sudo systemctl stop klipper.service`
- Klipper Kompilieren (siehe Einstellungen)
- Klipper aufspielen
  `~/klippy-env/bin/python3 ~/katapult/scripts/flash_can.py -i can0 -f ~/klipper/out/klipper.bin -u` **`73be55aa184d`**
  - Die **UUID** von eurem Board eintragen!
  - Das sollte wieder mit `Flash Success` bestätigt werden.
- Klipper Dienst starten
  `sudo systemctl start klipper.service`

# Links

- https://github.com/bigtreetech/EBB/tree/master/EBB%20SB2209%20CAN%20(RP2040)
- Klipper Konfig
  https://github.com/bigtreetech/EBB/blob/master/EBB%20SB2209%20CAN%20(RP2040)/sample-bigtreetech-ebb-sb-rp2040-canbus-v1.0.cfg
- Schaltplan
  https://github.com/bigtreetech/EBB/blob/master/EBB%20SB2209%20CAN%20(RP2040)/Hardware/BIGTREETECH%20EBB%20SB2209%20CAN(RP2040)%20V1.0-SCH.pdf

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

Permanent link:
**https://www.drklipper.de/doku.php?id=klipper_faq:flash_guide:rp2040:ebb_sb2209_can&rev=1702460696**

Last update: **2023/12/13 10:44**