

EBB SB2209 (USB)

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

YouTube Video #69



Hinweise

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

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!

Verkabelung

- Ganz normale USB Verkabelung
- GND & 12 bzw. 24V extra zum Board führen

DFU Modus

- 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 Board 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

Klipper flashen

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

```
[*] Enable extra low-level configuration options
Micro-controller Architecture (STMicroelectronics STM32) --->
Processor model (STM32G0B1) --->
Bootloader offset (No bootloader) --->
Clock Reference (8 MHz crystal) --->
Communication interface (USB (on PA11/PA12)) --->
USB ids --->
() GPIO pins to set at micro-controller startup
```

- beenden mit Q und Y
- Klipper kompilieren
`make -j4`
- Klipper flashen
`dfu-util -R -a 0 -s 0x08000000:mass-erase:force -D`

~/klipper/out/klipper.bin

Das Ergebnis sollte sein File downloaded successfully

Port ermitteln

- Den USB Stecker abziehen
- dmesg -HW starten

```
pi@Pi3Test:~/klipper $ dmesg -HW
[Nov11 15:15] usb 1-1.4: new full-speed USB device number 8 using
dwc_otg
[ +0.134623] usb 1-1.4: New USB device found, idVendor=1d50,
idProduct=614e, bcdDevice= 1.00
[ +0.000034] usb 1-1.4: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[ +0.000016] usb 1-1.4: Product: stm32g0b1xx
[ +0.000012] usb 1-1.4: Manufacturer: Klipper
[ +0.000012] usb 1-1.4: SerialNumber: 3F003A000E50425539393020
[ +0.075794] cdc_acm 1-1.4:1.0: ttyACM0: USB ACM device
[ +0.000134] usbcore: registered new interface driver cdc_acm
[ +0.000010] cdc_acm: USB Abstract Control Model driver for USB modems
and ISDN adapters
```

- Wir brauchen die Information mit **tty...** also in diesem Fall **ttyACM0**
- STRG+C drücken, um die Meldungen zu beenden
- Den zugehörigen Link ermitteln


```
ls -lR /dev/ | grep -v '\->\s../tty' | grep -e 'tty[[:alpha:]]' -e
serial
```

 - Wir brauchen die Info unter /dev/serial/by-id: :


```
lrwxrwxrwx 1 root root 13 Nov 11 15:15 usb-
Klipper_stm32g0b1xx_3F003A000E50425539393020-if00 → ../../ttyACM0
```
 - Achte darauf das am Ende die gleiche tty Bezeichnung steht wie sie im vorherigen Schritt ermittelt wurde (hier also ttyACM0)
 - Was für für die Konfig brauchen ist dann am Ende:


```
/dev/serial/by-id/usb-Klipper_stm32g0b1xx_3F003A000E50425539393020-
if00
```

kurzer Test

Ob das Board korrekt mit Klipper läuft, lässt sich mit folgendem Befehl schnell testen:

```
~/klippy-env/bin/python ~/klipper/klippy/console.py /dev/serial/by-id/usb-
Klipper_stm32g0b1xx_3F003A000E50425539393020-if00
```

Der Pfad am Ende muss natürlich mit dem übereinstimmen was ihr im vorherigen Schritt ermittelt habt!

Wenn ihr ein **connected** am Anfang des Textes seht, ist das Board richtig geflasht.

```
INFO:root:Starting serial connect
Loaded 105 commands (v0.11.0-205-g5f0d252b / gcc: (15:8-2019-q3-1+b1) 8.3.0
+rpil+14) 2.34)
MCU config: ADC_MAX=4095 BUS_PINS_i2c1_PA9_PA10=PA9,PA10 BUS_PINS_i2c1_PB6
10,PB11 BUS_PINS_i2c2_PB13_PB14=PB13,PB14 BUS_PINS_i2c3_PB3_PB4=PB3,PB4 BUS
,PB15,PB13 BUS_PINS_spi2a=PC2,PC3,PB10 BUS_PINS_spi3=PB4,PB5,PB3 CLOCK_FREQ
ERVE_PINS_crystal=PF0,PF1 STATS_SUMSQ_BASE=256 STEPPER_BOTH_EDGE=1
=====
                                connected
=====
003.909: stats count=373 sum=462028 sumsq=4662247
```

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_stm32g0b1xx_3F003A000E50425539393020-if00
#canbus_uuid: 539892be834d
```

- Die Zeile mit `serial` entsprechend mit dem ermittelten Pfad von oben anpassen
- Die Zeile mit `canbus_uuid` auskommentieren
- In der `printer.cfg` noch die neue Konfig einbetten
`[include BTT_EBB_SB2209.cfg]`

Klipper Update

- Klipper Dienst stoppen
`sudo systemctl stop klipper.service`
- `cd ~/klipper && make menuconfig`
 - Die Einstellungen sind genauso wie im Kapitel [Klipper flashen](#)
- `make -j4 flash FLASH_DEVICE=`
- 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://www.drklipper.de/> - **Dr. Klipper Wiki**

Permanent link:
https://www.drklipper.de/doku.php?id=klipper_faq:flash_guide:stm32g0b1:ebb_sb2209_usb&rev=1699713641

Last update: **2023/11/11 15:40**

