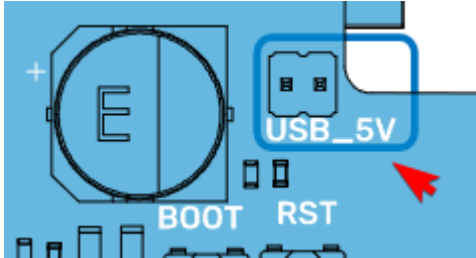


- Ein Zugang zum SBC über SSH ist notwendig!

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

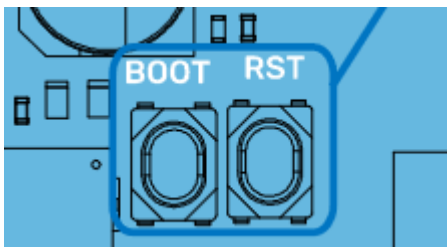
Als Faustregel im Betrieb den 5V Jumper immer abziehen!

Verkabelung

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

Boot Modus

- 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 20:07] 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

Klipper flashen

- Das Board in den [Boot Modus](#) bringen
- Klipper Dienst stoppen
sudo systemctl stop klipper.service
- cd ~/klipper
- make menuconfig

```
[*] Enable extra low-level configuration options
Micro-controller Architecture (Raspberry Pi RP2040) --->
Bootloader offset (No bootloader) --->
Flash chip (W25Q080 with CLKDIV 2) --->
Communication interface (USB) --->
USB ids --->
() GPIO pins to set at micro-controller startup
```

- beenden mit Q und Y
- Klipper kompilieren
make -j4
- Klipper flashen
make flash FLASH_DEVICE=2e8a:0003

```
pi@Pi3Test:~/klipper $ make flash FLASH_DEVICE=2e8a:0003
Flashing out/klipper.uf2 to 2e8a:0003
sudo lib/rp2040_flash/rp2040_flash out/klipper.uf2

[sudo] password for pi:
```

```
Loaded UF2 image with 101 pages
Found rp2040 device on USB bus 1 address 5
Flashing...
Resetting interface
Locking
Exiting XIP mode
Erasing
Flashing
Rebooting device
```

- Klipper Dienst wieder starten
`sudo systemctl start klipper.service`

Port ermitteln

- Den USB Stecker abziehen
- `dmesg -HW` starten

```
pi@Pi3Test:~/klipper $ dmesg -HW
[ +3.885051] usb 1-1.4: new full-speed USB device number 7 using
dwc_otg
[ +0.134266] usb 1-1.4: New USB device found, idVendor=1d50,
idProduct=614e, bcdDevice= 1.00
[ +0.000037] usb 1-1.4: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[ +0.000016] usb 1-1.4: Product: rp2040
[ +0.000012] usb 1-1.4: Manufacturer: Klipper
[ +0.000012] usb 1-1.4: SerialNumber: 455035712913A168
[ +0.001607] cdc_acm 1-1.4:1.0: ttyACM0: USB ACM device
```

- 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 Dec 13 22:03 usb-Klipper_rp2040_455035712913A168-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_rp2040_455035712913A168-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_rp2040_455035712913A168-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) 8.3.1
+rp1+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/bigtreotech/EBB/master/EBB%20SB2209%20CAN%20(RP2040)/sample-bigtreotech-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_rp2040_455035712913A168-if00
#canbus_uuid: 73be55aa184d
```

- Die Zeile mit `serial` eintragen oder ggf. die `#` am Anfang der Zeile entfernen (Port ermitteln siehe [hier](#))!
- Die UUID mit `canbus_uuid` auskommentieren oder löschen
- 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 flash -j4 FLASH_DEVICE=/dev/ttyACM0`
 - Wie man an den Port (hier `ttyACM0`) kommt, ist [hier](#) beschrieben
 - Das Ergebnis sollte so aussehen:

```
Entering bootloader on /dev/ttyACM0
Device reconnect on
/sys/devices/platform/soc/3f980000.usb/usb1/1-1/1-1.4
sudo lib/rp2040_flash/rp2040_flash out/klipper.uf2 1 8
```

```
Loaded UF2 image with 101 pages
Found rp2040 device on USB bus 1 address 8
Flashing...
Resetting interface
Locking
Exiting XIP mode
Erasing
Flashing
Rebooting device
```

- Klipper Dienst starten
`sudo systemctl start klipper.service`

Links

- [https://github.com/bigtreetech/EBB/tree/master/EBB%20SB2209%20CAN%20\(RP2040\)](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](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](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_usb&rev=1702572371

Last update: **2023/12/14 17:46**

