

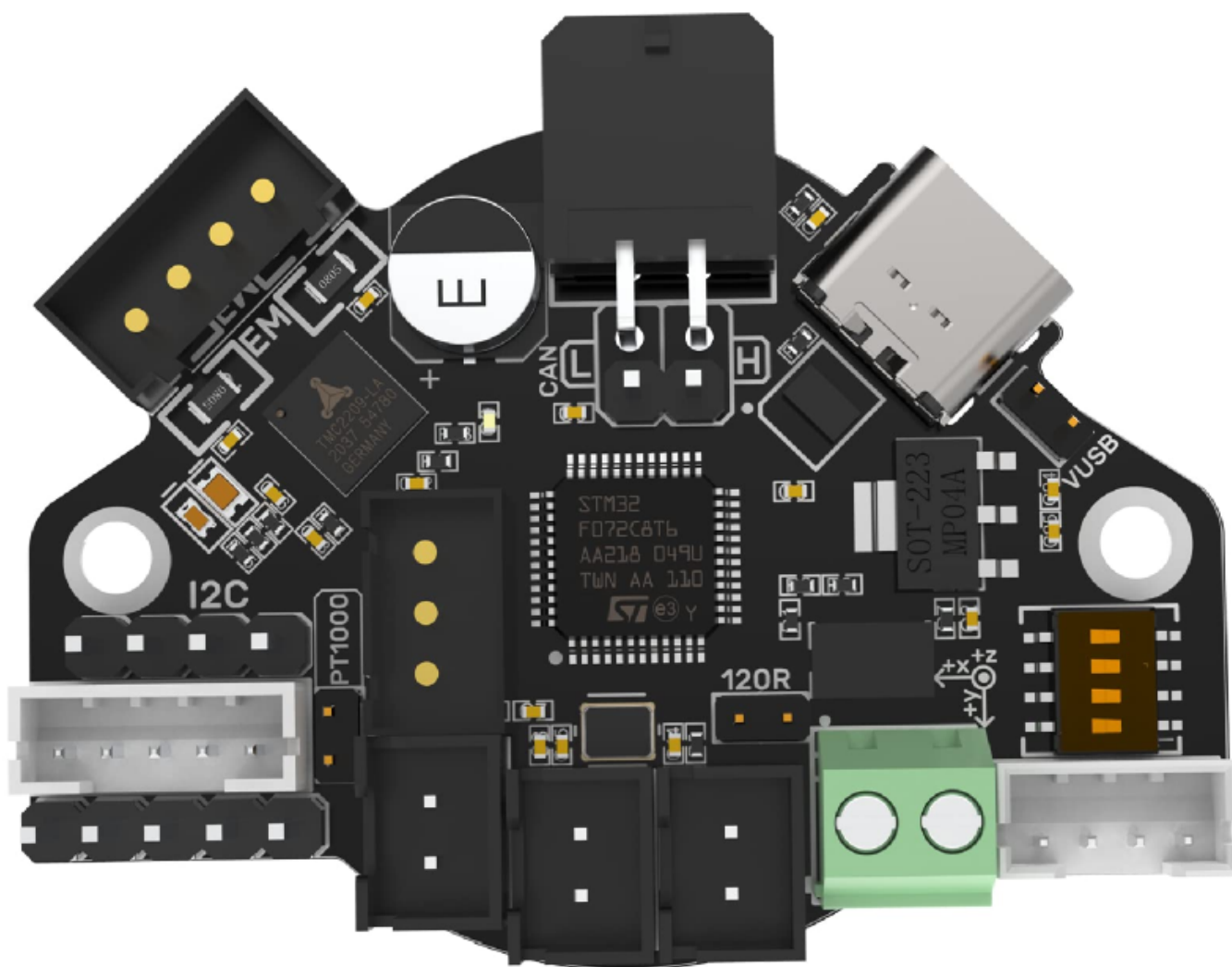
BTT EBB36 / EBB42 (CAN)

Schrittweise Anleitung, um die Bigtree Tech **EBB36 / EBB42** Kopfboards mit dem BTT Buskoppler **U2C** über **CAN** in Betrieb zu nehmen.

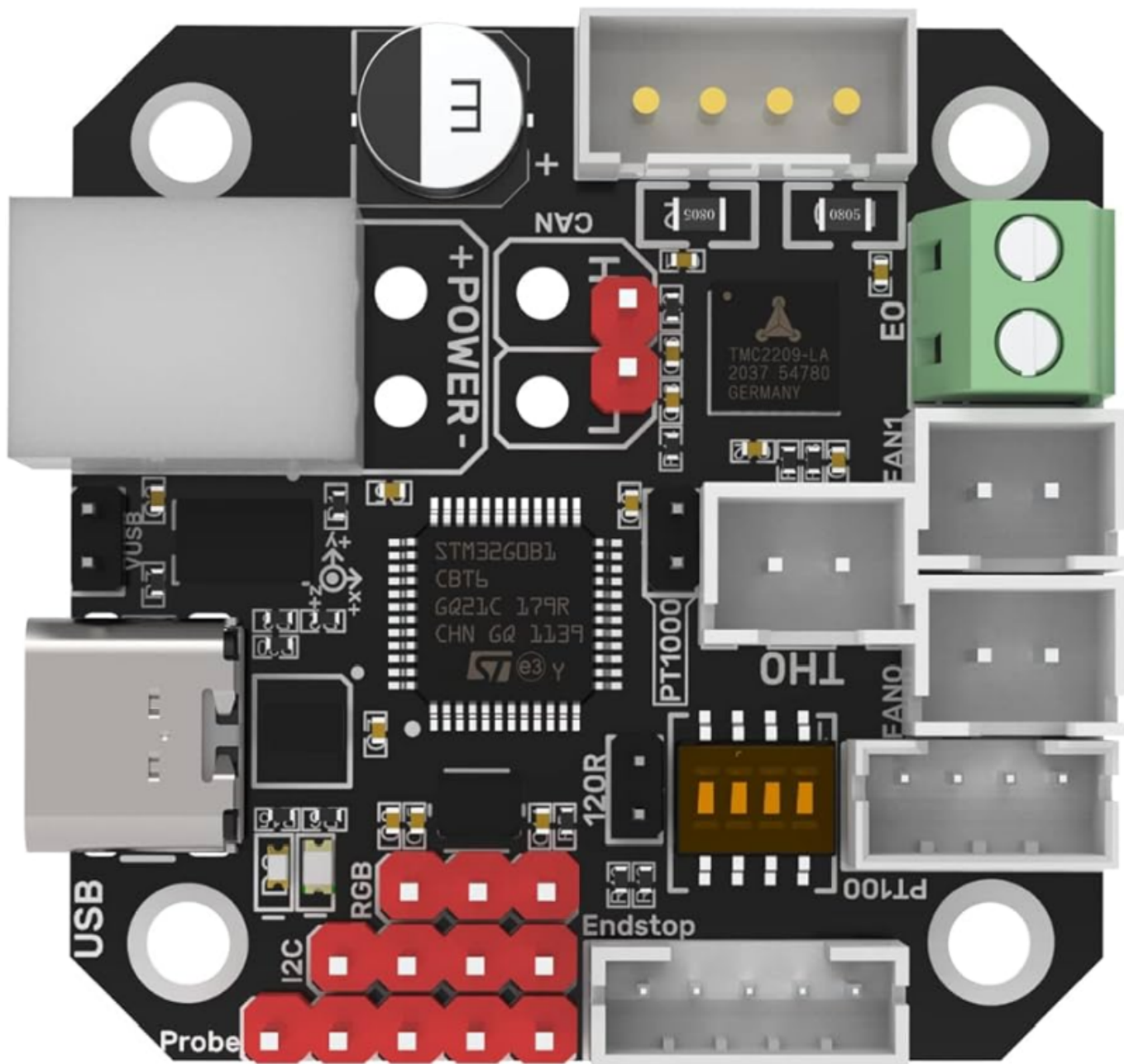
Hinweise

- 1) Diese Anleitung beschreibt beide EBB Boards (36 & 42) und das in allen derzeit verfügbaren Varianten (V1.0, V1.1, V1.2).
- 2) Variante 1.0 hat einen STM32F072C8T6 Controller. Die nachfolgenden Modelle einen STM32G0B1CBT6.
- 3) Die Modelle der Versionen (also als Beispiel EBB36 V1.0 & EBB42 V1.0) sind Pinkompatibel. Die verschiedenen Versionen sind nicht pinkompatibel!

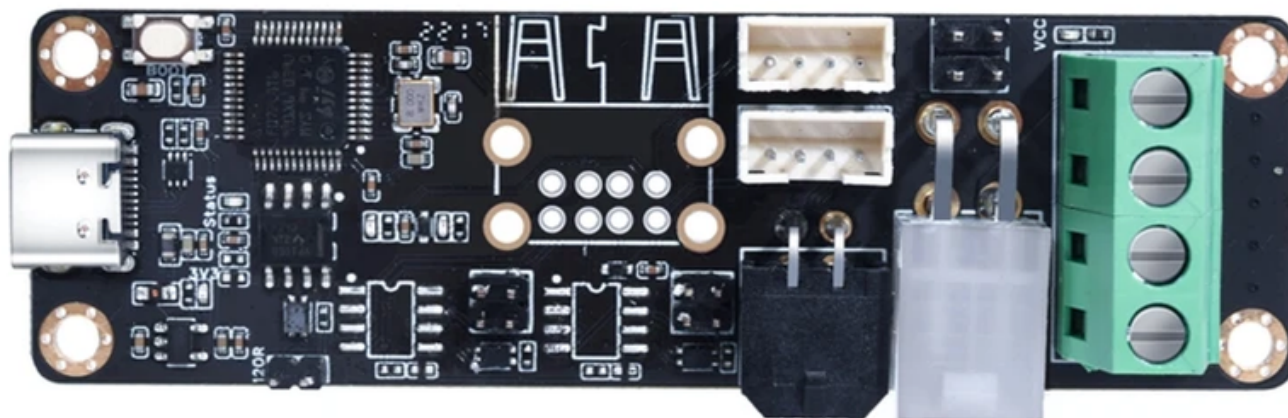
EBB36



EBB42



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!

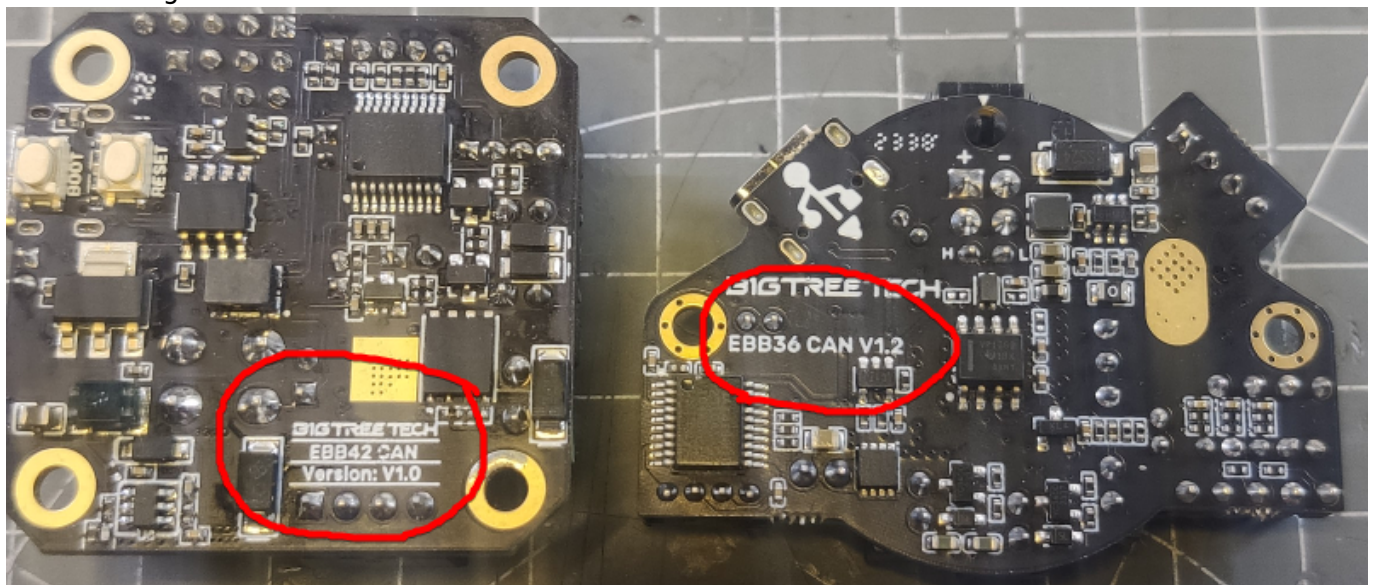


- **Board mit Version 1.1**

Hier wird der Pin PA2 beim Booten kurzfristig aktiviert. Da hängt in dieser Version die Ansteuerung für den Extruder Heizblock dran. Beim Flashen also lieber die 24V Versorgung abklemmen, um ein ungewolltes Aufheizen zu vermeiden!

Version ermitteln

Es ist wichtig die Version von dem Board zu ermitteln! Dazu einfach auf der Rückseite nachsehen:



Links ist das EBB42, rechts ist das EBB36.

Die Versionen haben unterschiedliche Controller verbaut und teilweise unterschiedliche Pinbelegungen!

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

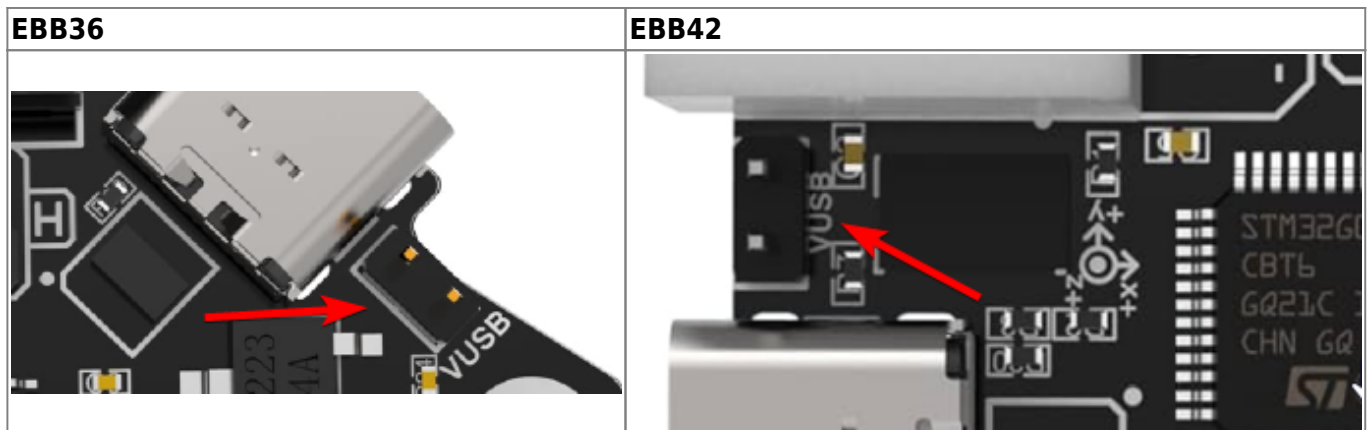
- Interface einrichten
Achtung : die Bitrate von 1000000 muss auch in der Board Firmware eingestellt werden!
`sudo nano /etc/network/interfaces.d/can0`
folgendes eintragen, speichern und mit STRG + x, dann Y, dann Enter beenden

```
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`
- Sollte das Interface auf DOWN stehen hilft meist ein
`sudo systemctl restart networking.service`

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

Katapult wird hier benötigt, um über den CAN Bus das Board mit Klipper flashen zu können.

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

Katapult aufspielen

ACHTUNG Auf die Version achten bei make menuconfig!

- make menuconfig für **Version 1.0**

```

Micro-controller Architecture (STMicroelectronics STM32) --->
Processor model (STM32F072) --->
Build Katapult deployment application (Do not build) --->
Clock Reference (8 MHz crystal) --->
Communication interface (CAN bus (on PB8/PB9)) --->
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
(PA14) Status LED GPIO Pin

```

- make menuconfig für **Version 1.1 & 1.2**

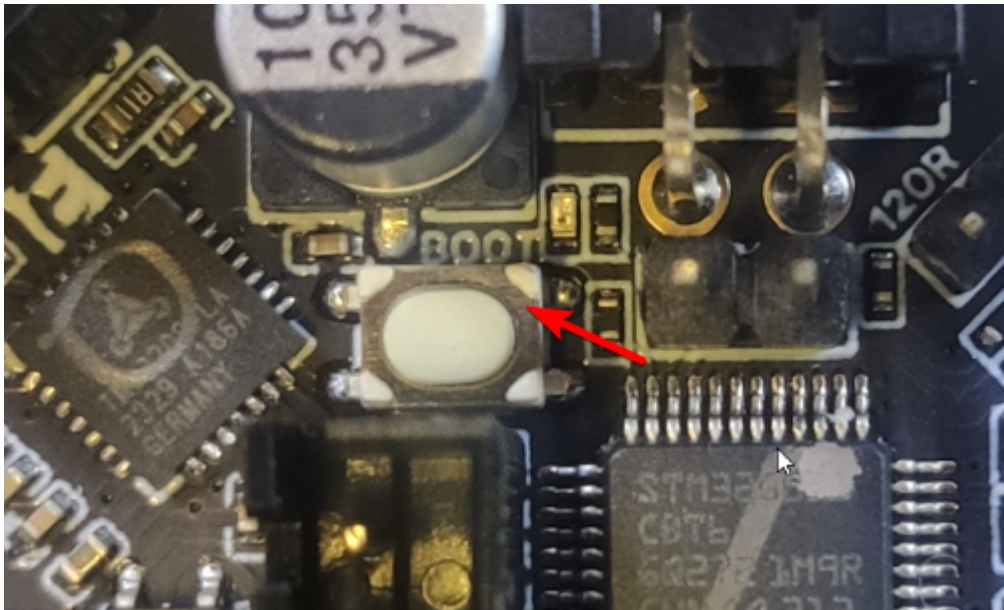
```

Micro-controller Architecture (STMicroelectronics STM32) --->

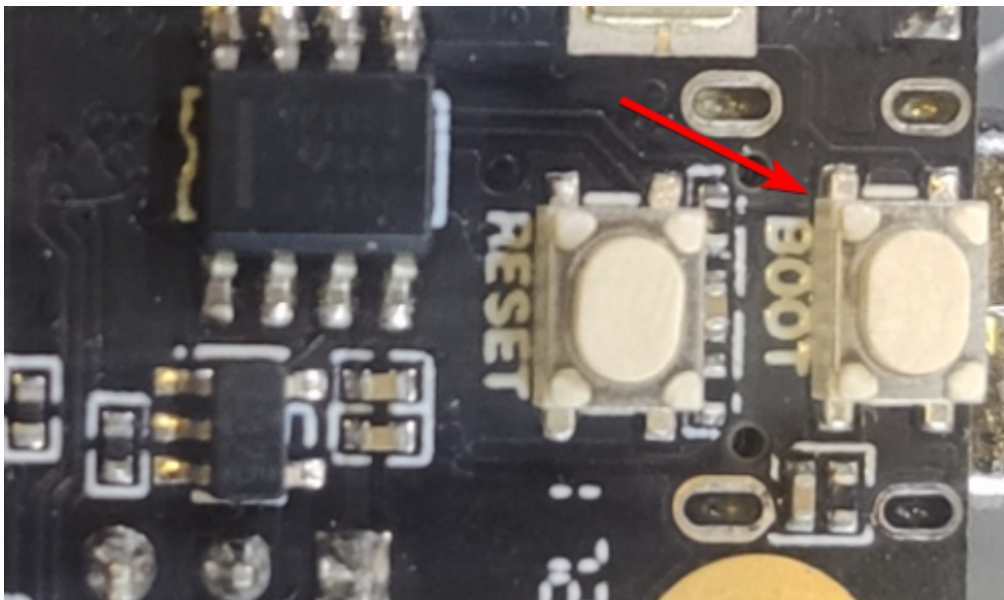
```

```
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
(PA13) Status LED GPIO Pin
```

- 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



EBB36 :



EBB42 :

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
- 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

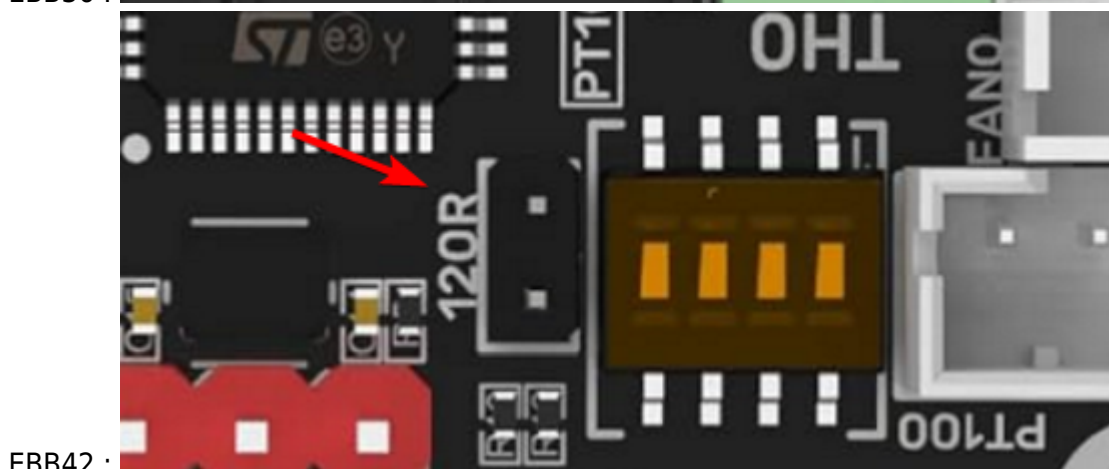
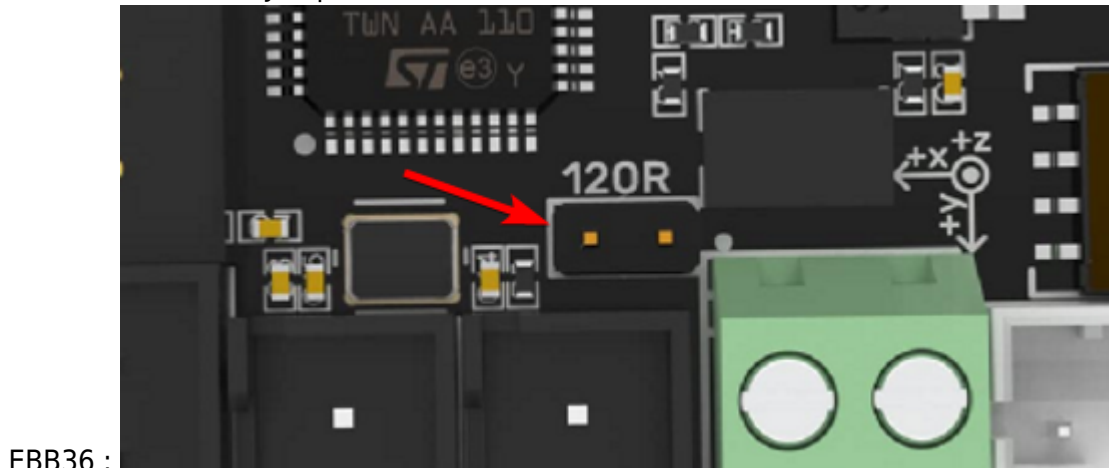
- Am U2C Board den Jumper **120R** setzen



Hinweis

Das U2C hat mehrere Jumper für 120Ohm. Es ist der Jumper der am nächsten zum USB Port sitzt!

- 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 flashen per Katapult

ACHTUNG Auf die Version achten bei make menuconfig!

- cd ~/klipper
- make menuconfig für **Version 1.0**

```
[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32) --->
    Processor model (STM32F072) --->
    Bootloader offset (8KiB bootloader) --->
    Clock Reference (8 MHz crystal) --->
    Communication interface (CAN bus (on PB8/PB9)) --->
(1000000) CAN bus speed
() GPIO pins to set at micro-controller startup
```

- make menuconfig für **Version 1.1 & 1.2**

```
[*] 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

kurzer Test

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

```
~/klippy-env/bin/python ~/klipper/klippy/console.py -c can0 44d860c9632b
```

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.

```
==== attempting to connect =====
INFO:root:Starting CAN connect
INFO:can.interfaces.socketcan.socketcan:Created a socket
Loaded 114 commands (v0.12.0-61-gb50d6669 / gcc: (15:8-2019-q3-1+b1) 8.3
MCU config: ADC_MAX=4095 BUS_PINS_i2c1_PA9_PA10=PA9,PA10 BUS_PINS_i2c1_PA
14 BUS_PINS_i2c3_PB3_PB4=PB3,PB4 BUS_PINS_spi1=PA6,PA7,PA5 BUS_PINS_spi1
NCY=1000000 CLOCK_FREQ=64000000 MCU=stm32g0b1xx PWM_MAX=255 RECEIVE_WINDO
WARNING:root:got {'oid': 6, 'next_clock': 515819151, 'value': 31272, '#n
====
                        connected
=====
001.393: analog_in_state oid=6 next_clock=535019151 value=31275
```

Konfig

- `cd ~/printer_data/config`
- **Version 1.0**
 - `wget 'https://raw.githubusercontent.com/bigtreotech/EBB/master/EBB%20CAN%20V1.0%20(STM32F072)/sample-bigtreotech-ebb-canbus-v1.0.cfg' -O BTT_EBB.cfg`
- **Version 1.1**
 - `wget 'https://raw.githubusercontent.com/bigtreotech/EBB/master/EBB%20CAN%20V1.1%20(STM32G0B1)/sample-bigtreotech-ebb-canbus-v1.1.cfg' -O BTT_EBB.cfg`
- **Version 1.2**
 - `wget 'https://raw.githubusercontent.com/bigtreotech/EBB/master/EBB%20CAN%20V1.1%20(STM32G0B1)/sample-bigtreotech-ebb-canbus-v1.2.cfg' -O BTT_EBB.cfg`
- `nano ~/printer_data/config/BTT_EBB.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.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/bigtreotech/EBB>

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

Permanent link:
https://www.drklipper.de/doku.php?id=klipper_faq:flash_guide:stm32g0b1:btt_ebb36_ebb42_can&rev=1704622771

Last update: **2024/01/07 11:19**

