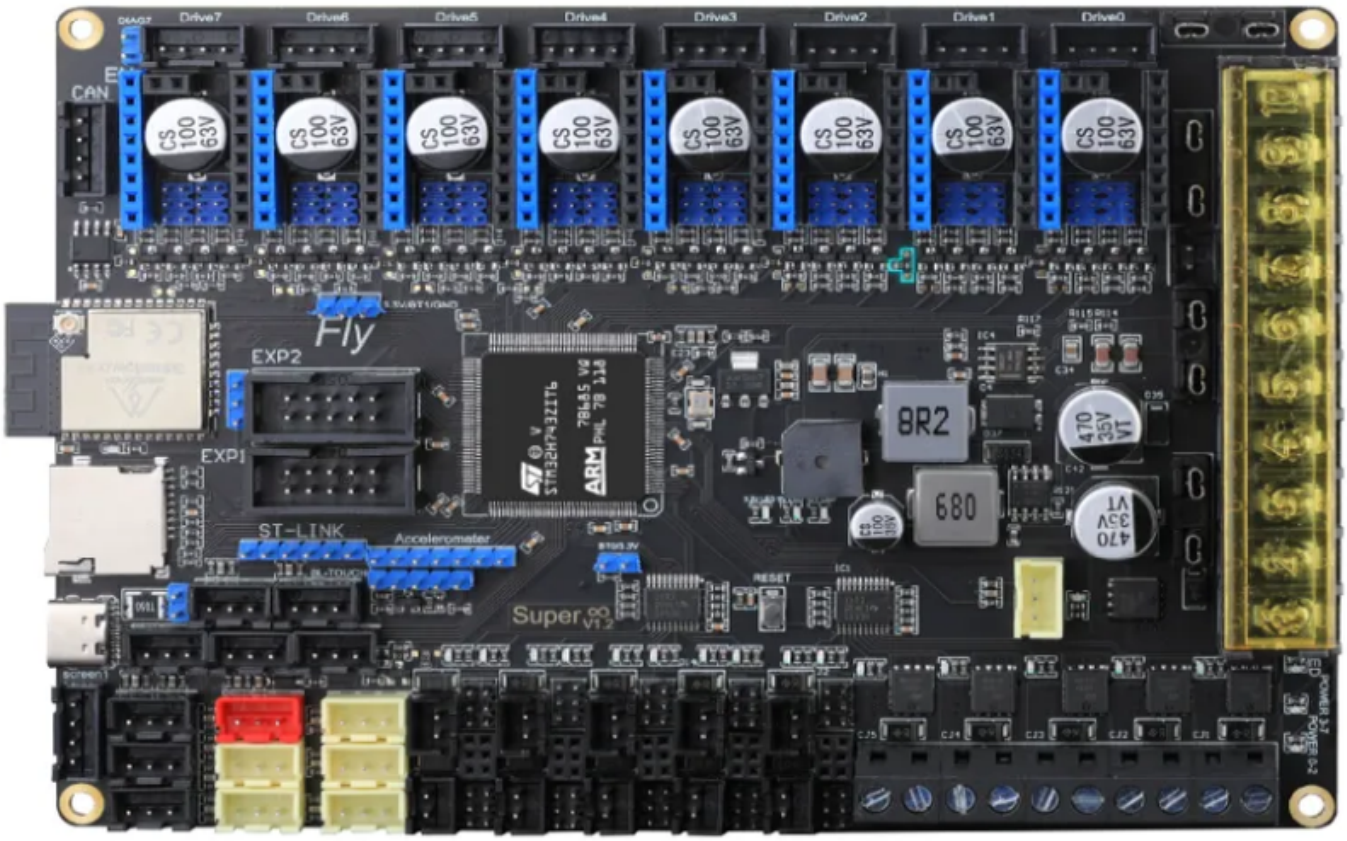
Mellow Fly-Super8Pro (CAN-Bridge)

Schrittweise Anleitung, um das Mellow **Fly-Super8Pro** Board über die **USB/CAN Bridge** in Betrieb zu nehmen.

Mellow Fly-Super8Pro



YouTube Video #120

Hinweise

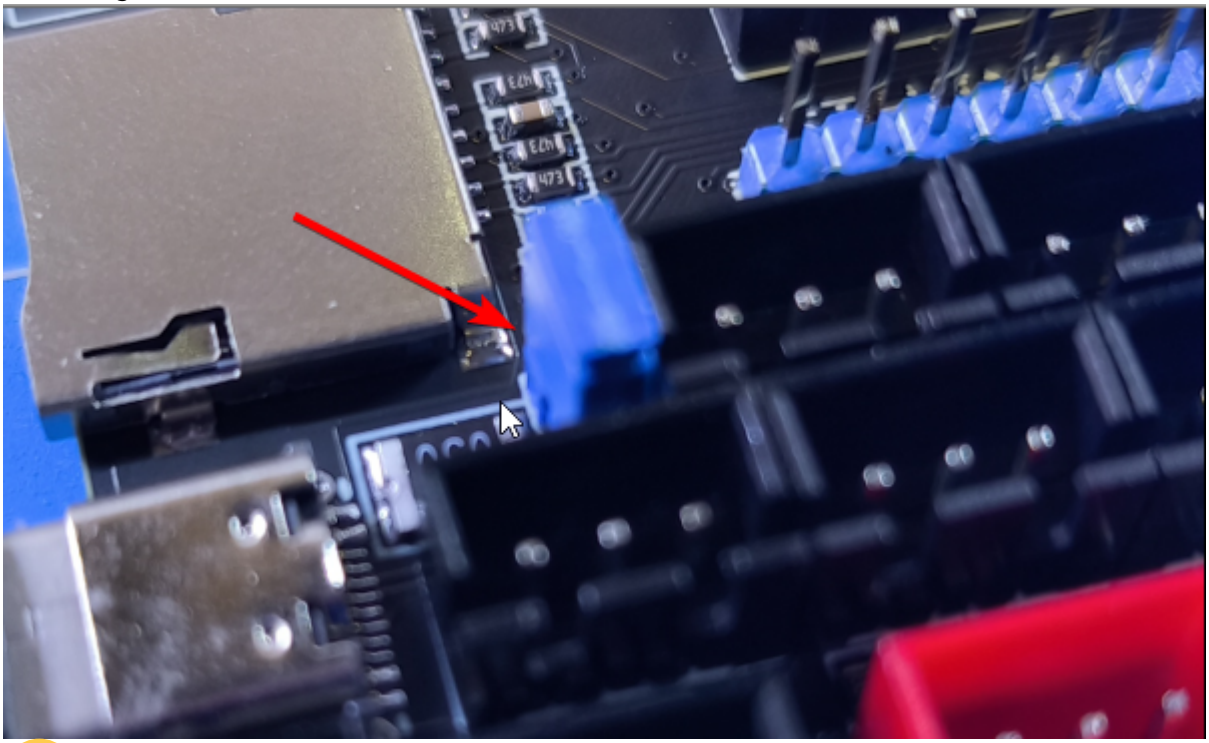
- **SBC** bedeutet in der Anleitung **S**ingle **B**oard **C**omputer. Also meistens wohl ein Raspberry Pi.
- Es wird davon ausgegangen das auf dem SBC Klipper und MainSail eingerichtet ist.
- Ein Zugang zum SBC über SSH ist notwendig!

- Wenn `dmesg -HW` einen Fehler bringt, einfach `dmesg -Hw` verwenden.
- Der SD-Slot ist bei diesem Controller komplett überflüssig 😎

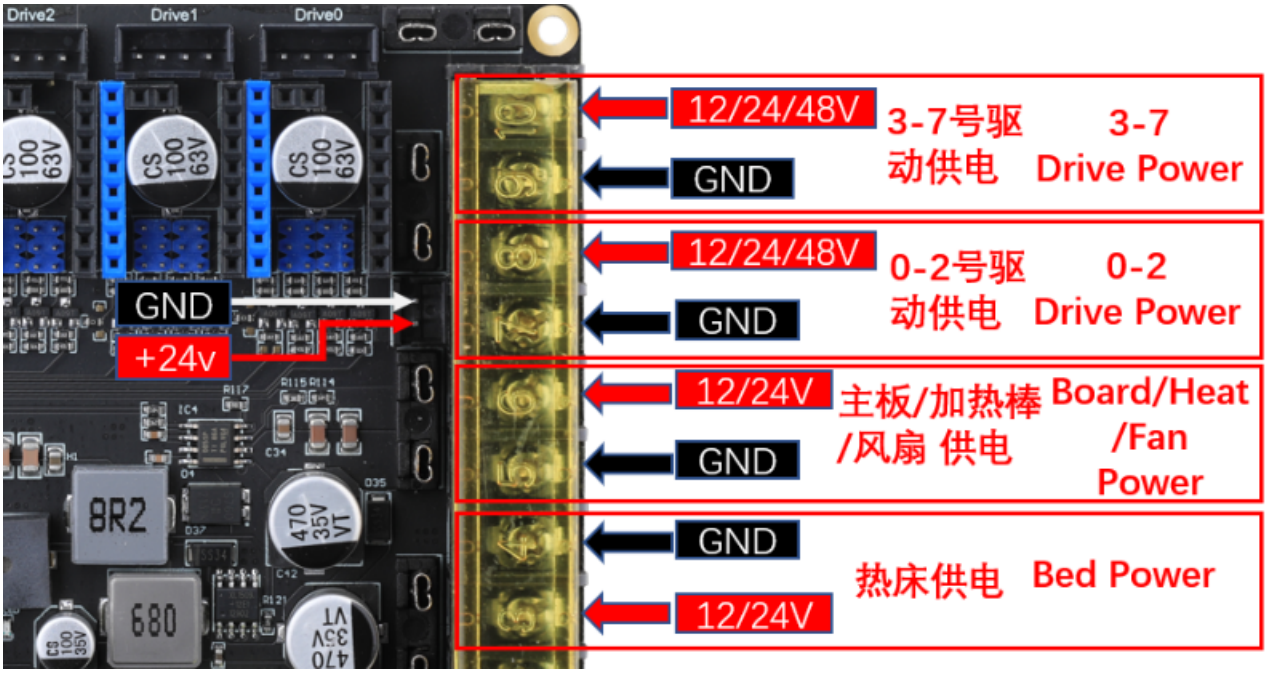
Verkabelung

Stromversorgung

- Der Jumper direkt hinter dem USB-C Anschluss muss gesetzt sein wenn das Board nur am USB Port hängt und **nicht an 24V**.



- ❗ Der Jumper muss im normalen Betrieb gezogen werden!
- **Betrieb**
  - Im Betrieb wird das Board mit 24V versorgt (Anschluss POWER Board / + -)
  - ❗ Der Jumper für 5V muss gezogen sein!
- **Firmware flashen**
  - Das Board wird **nicht** **mit 24V versorgt**.
    Den Jumper muss gesteckt sein!
  - Das Board wird **mit 24V betrieben**.
    Den Jumper muss gezogen sein!
- Anschluss



Versorgung Raspberry Pi

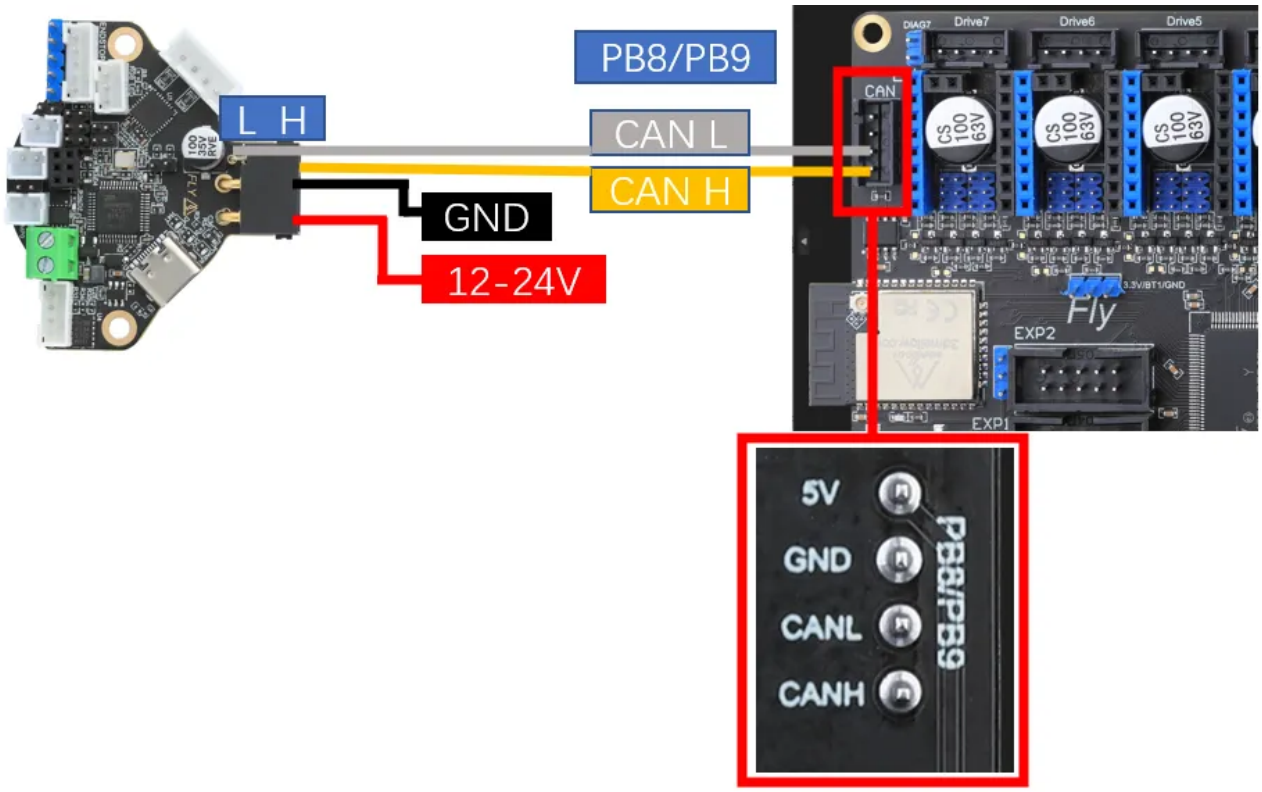
Auf dem Board ist unter dem USB Port ein 4 Port Connector mit 5V, GND und Tx/Rx. Hier könnte ein Raspberry Pi direkt versorgt und mit dem Board betrieben werden. Das führt aber fast immer zu Unterspannungswarnungen. Besser mit USB Verkabeln und den Pi extra versorgen!

48V Anschluss

Alle Treiber können mit 12V, 24V oder 48V betrieben werden.

CAN Bus Anschluss

- Wer den CAN Bus überprüfen will, kann im **ausgeschalteten Zustand** den Buswiderstand mit einem Ohmmeter messen. Es müsste zwischen CAN H und CAN L ca. 60Ω ergeben. Vorausgesetzt, es ist ein zweiter Busteilnehmer verkabelt und passend terminiert.
- Das Fly-Super8Pro Board hat einen Transceiver direkt verbaut.
- CAN H / CAN L kann direkt am Board angeschlossen werden:



CAN H ist hier links, CAN L mittig und Masse (GND) rechts.

- Der 120Ohm **Abschlusswiderstand ist nicht deaktivierbar**! Das Board muss also am Ende vom CAN Bus hängen!

Bootloader sichern

Das Board wird mit RepRap Firmware ausgeliefert (Stand 29.11.2024).

```
pi@TestPi4:~ $ dmesg -HW
[Nov29 17:53] usb 1-1.1: new full-speed USB device number 13 using xhci_hcd
[  +0.111909] usb 1-1.1: New USB device found, idVendor=16c0,
idProduct=27dd, bcdDevice= 1.00
[  +0.000017] usb 1-1.1: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[  +0.000004] usb 1-1.1: Product: RepRapFirmware
[  +0.000004] usb 1-1.1: Manufacturer: RepRapFirmware
[  +0.000003] usb 1-1.1: SerialNumber: 2F00150009513134303238 35
[  +0.005831] cdc_acm 1-1.1:1.0: ttyACM1: USB ACM device
```

Es ist ein Bootloader im Flasch und die Firmware startet ab 0x20000h (128k).
Ein Abzug (inkl. Bootloader) kann hier geladen werden: orgfirmware_29_11_2024.zip
Das Backup kann mittels ST-Link oder DFU Mode wieder aufgespielt werden. Es muss nur an Adresse 0x0 geschrieben werden!

Vorgehen Flashen

- Wer sein Board das erste mal mit Klipper einrichtet muss die folgenden Schritte durchgehen:
  - DFU-Modus aktivieren
  - Katapult flashen
  - Port ermitteln
  - Klipper flashen
  - SBC einrichten
- Wer das Board schon nach dieser Anleitung eingerichtet hat kann das Klipper Update so durchführen ...
  - Update

DFU Modus

Das Board in den DFU Modus bringen:

- Im Terminal folgendes eingeben
  `dmesg -HW`
- Auf dem Board ist direkt am Controller ein 2 Pin Header mit der Beschriftung “BT0/3.3V”. Hier muss ein Jumper gesetzt werden. Dann das Board 1x stromlos machen oder einfach die Reset Taste drücken.
- Das Board meldet sich mit **`Product: STM32 BOOTLOADER`** oder **`Product: DFU in FS Mode`**

```
pi@Pi4Test:~ $ dmesg -HW
[Nov29 17:59] usb 1-1.1: new full-speed USB device number 15 using
xhci_hcd
[  +0.101997] usb 1-1.1: not running at top speed; connect to a high
speed hub
[  +0.006042] usb 1-1.1: New USB device found, idVendor=0483,
idProduct=df11, bcdDevice= 2.00
[  +0.000024] usb 1-1.1: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[  +0.000013] usb 1-1.1: Product: DFU in FS Mode
[  +0.000010] usb 1-1.1: Manufacturer: STMicroelectronics
[  +0.000010] usb 1-1.1: SerialNumber: 354D325F3431
```

  - STRG+C drücken, um die Meldungen zu beenden

Katapult flashen

> **Hinweis:**
> Katapult wird **über USB** (DFU-Mode) eingerichtet!

- Katapult laden wenn noch nicht vorhanden, sonst in den Katapult Ordner wechseln
  `[ ! -d "$HOME/katapult/" ] && cd ~ && git clone https://github.com/Arksine/katapult && cd katapult || cd ~/katapult`
- `make menuconfig`

```
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32H723)  --->
    Build Katapult deployment application (Do not build)  --->
    Clock Reference (25 MHz crystal)  --->
    Communication interface (USB (on PA11/PA12))  --->
    Application start offset (128KiB offset)  --->
    USB ids  --->
()  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
```

    - **Wichtig:** Hier wird als Communication interface USB ausgewählt, nicht CAN!
    - Sonst ist später kein Update möglich!
- Katapult kompilieren
  `make -j4`
- Katapult flashen (das Board muss im DFU Mode sein !)
  `dfu-util -R -a 0 -s 0x08000000:mass-erase:force -D ~/katapult/out/katapult.bin`
    - Wichtig ist am Ende `File downloaded successfully` bei der Ausgabe im Terminal
- Das Board einmal resetten
    - Reset-Taste (oberhalb vom USB-C Anschluss) drücken
    - oder das Board einmal stromlos machen
- Es gibt keine Status LED. Man muss als mittels `dmesg` auf Katapult prüfen. Siehe nächster Schritt ...

Port ermitteln

- Den USB Stecker abziehen
- `dmesg -HW` starten und den USB Stecker wieder anstecken

```
pi@TestPi4:~/katapult $ dmesg
....
[1220850.458432] usb 1-1.2: new full-speed USB device number 31 using xhci_hcd
[1220850.570330] usb 1-1.2: New USB device found, idVendor=1d50, idProduct=6177, bcdDevice= 1.00
[1220850.570346] usb 1-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[1220850.570350] usb 1-1.2: Product: stm32h723xx
[1220850.570354] usb 1-1.2: Manufacturer: katapult
[1220850.570357] usb 1-1.2: SerialNumber: 2F0015000951313430323835
[1220850.576180] cdc_acm 1-1.2:1.0: ttyACM1: USB ACM device
```

    - Wir brauchen die Information mit **tty...** also in diesem Fall **ttyACM1**
    - 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 (STM32H723)  --->
    Bootloader offset (128KiB bootloader)  --->
    Clock Reference (25 MHz crystal)  --->
    Communication interface (USB to CAN bus bridge (USB on PA11/PA12))
--->
    CAN bus interface (CAN bus (on PB8/PB9))  --->
    USB ids  --->
(1000000) CAN bus speed
()  GPIO pins to set at micro-controller startup
```

- Klipper kompilieren und flashen (über USB / seriell!)
  make -j4 flash FLASH_DEVICE=/dev/ttyACM0

```
pi@TestPi4:~/klipper $ make -j4 flash FLASH_DEVICE=/dev/ttyACM1
  Creating symbolic link out/board
  Building out/autoconf.h
  Compiling out/src/sched.o
...
  Compiling out/src/stm32/hard_pwm.o
  Preprocessing out/src/generic/armcm_link.ld
  Building out/compile_time_request.o
Version: v0.12.0-377-g9bd0d4757-dirty-20241201_182541-TestPi4
  Linking out/klipper.elf
  Creating hex file out/klipper.bin
  Flashing out/klipper.bin to /dev/ttyACM1
Entering bootloader on /dev/ttyACM1
Device reconnect on
/sys/devices/platform/scb/fd500000.pcie/pci0000:00/0000:00:00.0/0000:01
:00.0/usb1/1-1/1-1.2/1-1.2:1.0
/usr/bin/python3 lib/canboot/flash_can.py -d /dev/serial/by-
path/platform-fd500000.pcie-pci-0000:01:00.0-usb-0:1.2:1.0 -f
out/klipper.bin

Attempting to connect to bootloader
CanBoot Connected
Protocol Version: 1.1.0
Block Size: 64 bytes
Application Start: 0x8020000
MCU type: stm32h723xxv0.0.1-76-g081918a
Flashing '/home/pi/klipper/out/klipper.bin'...

[##################################################]
```

```
Write complete: 1 pages
Verifying (block count = 650)...

[##################################################]

Verification Complete: SHA = CAA178A1B0A1FF924141FB489B3F82D091BF15B3
CAN Flash Success
```

- kurzer Test mit `lsusb` → Geschwister Schneider CAN adapter sollte erscheinen

```
pi@TestPi5:~/klipper $ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 010: ID 1d50:606f OpenMoko, Inc. Geschwister Schneider
CAN adapter
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

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

```
allow-hotplug 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`
  oder ein
  `sudo ip link set can0 up type can bitrate 1000000`

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:

```
pi@TestPi4:~/klipper $ ~/klippy-env/bin/python
~/klipper/scripts/canbus_query.py can0
Found canbus_uuid=c39b99373fbc, Application: Klipper
Total 1 uuids found
```

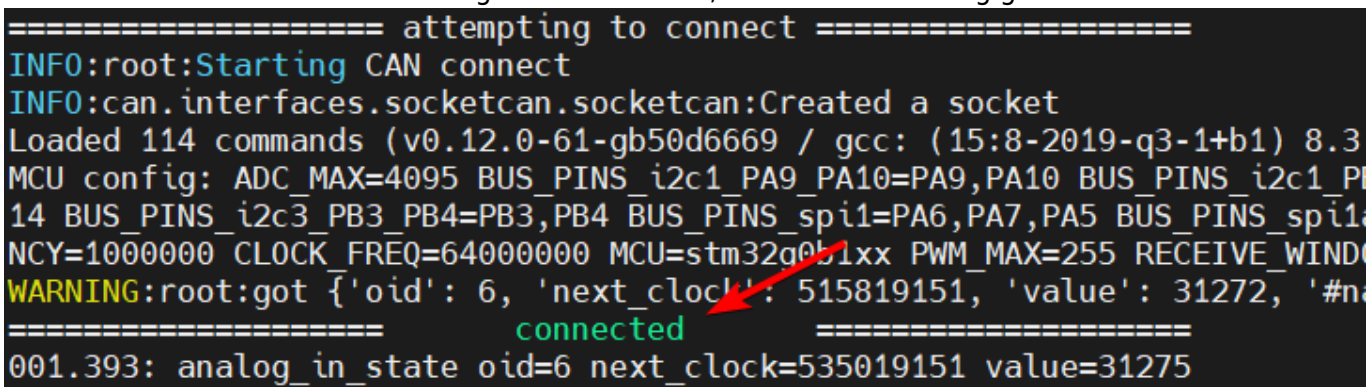
  - Die **UUID** (canbus_uuid=**c39b99373fbc**) notieren !
  - Wird bei diesem Schritt kein Board gefunden, hilft oft ein Reset am Board (entweder über Reset Taster oder 1x Strom weg und wieder dran)

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 c39b99373fbc`

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.



Konfiguration

- `cd ~/printer_data/config`
- **Beispiel Konfiguration**
  https://mellow.klipper.cn/en/docs/ProductDoc/MainBoard/fly-super/fly-super8-pro/cfg/
- `nano ~/printer_data/config/printer.cfg`

```
[mcu]
canbus_uuid: c39b99373fbc
#restart_method: command
```

  - Die Zeile mit `serial` löschen oder auskommentieren
  - Die Zeile mit `restart_method` löschen oder auskommentieren
  - Die Zeile mit `canbus_uuid` entsprechend mit der ermittelten UUID von oben anpassen
- Klipper starten

```
sudo systemctl start klipper.service
```

Klipper Update

> **Hinweis:**
> Das Klipper Update wird über USB eingespielt! Über den CAN-Bus ist ein Update nicht möglich wenn das Board als USB/Can Bridge arbeitet.

- Klipper Dienst stoppen
  `sudo systemctl stop klipper.service`
- Alle CAN UUID's ermitteln
  `grep canbus_uuid ~/printer_data/config/* -n`

```
pi@Pi3Test:~/klipper $ grep canbus_uuid ~/printer_data/config/* -n
/home/pi/printer_data/config/BTT_EBB.cfg:10:canbus_uuid: 44d860c9632b
/home/pi/printer_data/config/printer.cfg:30:canbus_uuid: c39b99373fbc
```

- Das Leviathan Board per flshtool.py resetten. Welche UUID das Leviathan hat kann man bei mehreren Busteilnehmern leider nicht ohne weitere erkennen.
  `~/klippy-env/bin/python ~/katapult/scripts/flashtool.py -i can0 -u <BOARD UUID> -r`

```
pi@Pi3Test:~/klipper $ ~/klippy-env/bin/python
~/katapult/scripts/flashtool.py -i can0 -u c39b99373fbc-r
Sending bootloader jump command...
Bootloader request command sent
Flash Success
```

    - Die Status LED sollte jetzt anfangen zu blinken
- Den Port ermitteln
  `dmesg |tail -n 10`

```
pi@Pi3Test:~/klipper $ dmesg |tail -n 10
[76418.167383] IPv6: ADDRCONF(NETDEV_CHANGE): can0: link becomes ready
[76867.446711] usb 1-1.4: USB disconnect, device number 37
[76867.446933] gs_usb 1-1.4:1.0 can0: Couldnt shutdown device (err=-19)
[76867.787311] usb 1-1.4: new full-speed USB device number 38 using
dwc_otg
[76867.933716] usb 1-1.4: New USB device found, idVendor=1d50,
idProduct=6177, bcdDevice= 1.00
[76867.933741] usb 1-1.4: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[76867.933749] usb 1-1.4: Product: stm32f446xx
[76867.933755] usb 1-1.4: Manufacturer: katapult
[76867.933761] usb 1-1.4: SerialNumber: 350053000851313133353932
[76867.938929] cdc_acm 1-1.4:1.0: ttyACM0: USB ACM device
```

Wie immer brauchen wir die tty... Angabe. In diesem Fall ist is **ttyACM0** wie man in der letzten Zeile sehen kann.

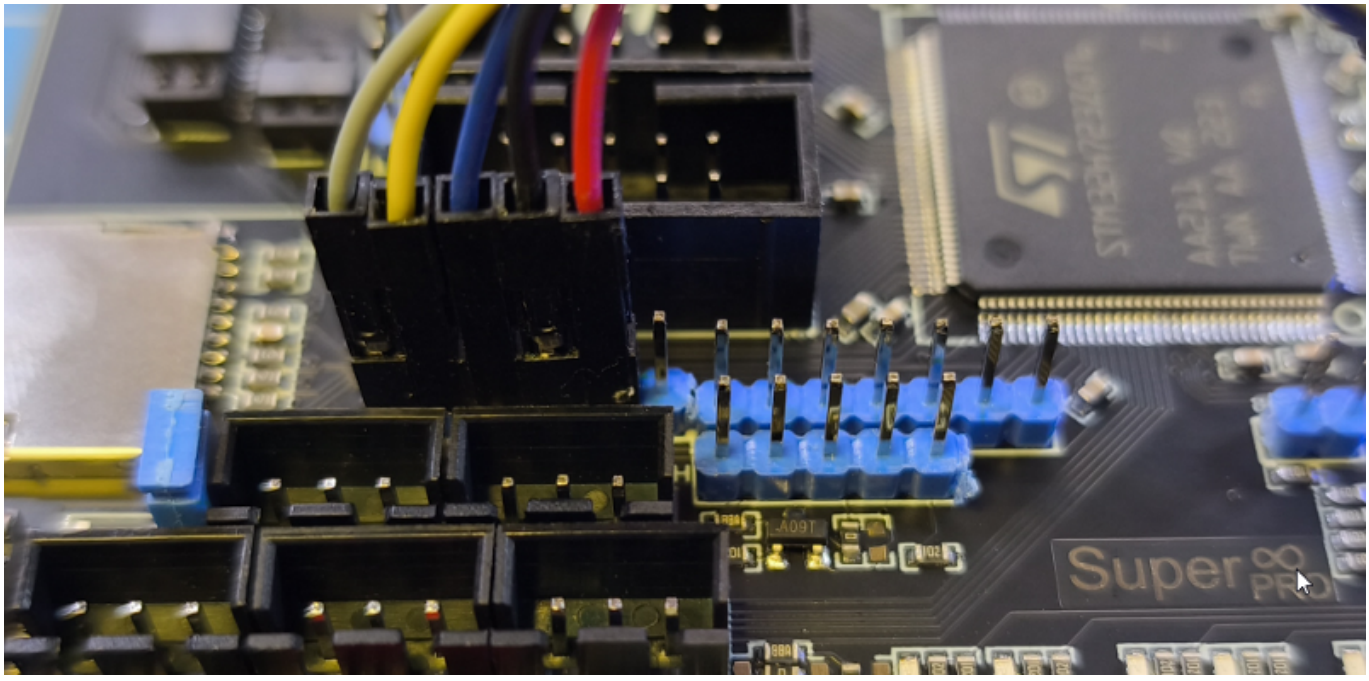
- `cd ~/klipper`
- `make menuconfig`
  → Die Einstellungen sind dieselben wie oben unter Klipper flashen angegeben.
- Klipper flashen
  `make -j4 flash FLASH_DEVICE=/dev/ttyACM0`
  Den ermittelten Port halt am Ende ggf. anpassen.
- Klipper starten
  `sudo systemctl start klipper.service`

Sonstiges

Diese Punkte sind nicht immer Bestandteil vom YouTube Video, aber nützlich 😎

ST-Link (SWD)

Das Board verfügt über einen ST-Link Port. Mit einem entsprechenden ST-Link Adapter kann das Board auch direkt geflasht werden.



Von links nach rechts

- grau → Reset
- gelb → CLK → SWCLK vom ST-Link
- blau → IO → SWDIO vom ST-Link
- schwarz → GND → Masse Anschluss
- ❗ rot → 3V3 → 3,3V Anschluss
  Achtung, der 3,3V Anschluss wird nur beim ST-Link V3 benötigt. Bei den billigen China **V2 Adaptern** darf dieser Pin **nicht angeschlossen** werden!

STM32 Temperatur

Der interne Temperatur Sensor des STM32 kann mit folgendem Konfig Schnibsel ausgelesen werden:

```
[temperature_sensor Levi]
sensor_type              : temperature_mcu
sensor_mcu               : mcu
```

Links

- https://mellow-3d.github.io/fly_super8_pro_h723_general.html
- https://mellow.klipper.cn/en/docs/ProductDoc/MainBoard/fly-super/fly-super8-pro/
- Github Repo
  https://github.com/Mellow-3D/Fly-Super8Pro
- Schaltplan
  https://github.com/Mellow-3D/Fly-Super8Pro/blob/0b982743ea8ddf187300ba3878263ac45f9bf40b/Hardware/Super8Pro_Schematic.pdf
- Klipper Konfig
  https://mellow.klipper.cn/en/docs/ProductDoc/MainBoard/fly-super/fly-super8-pro/cfg/