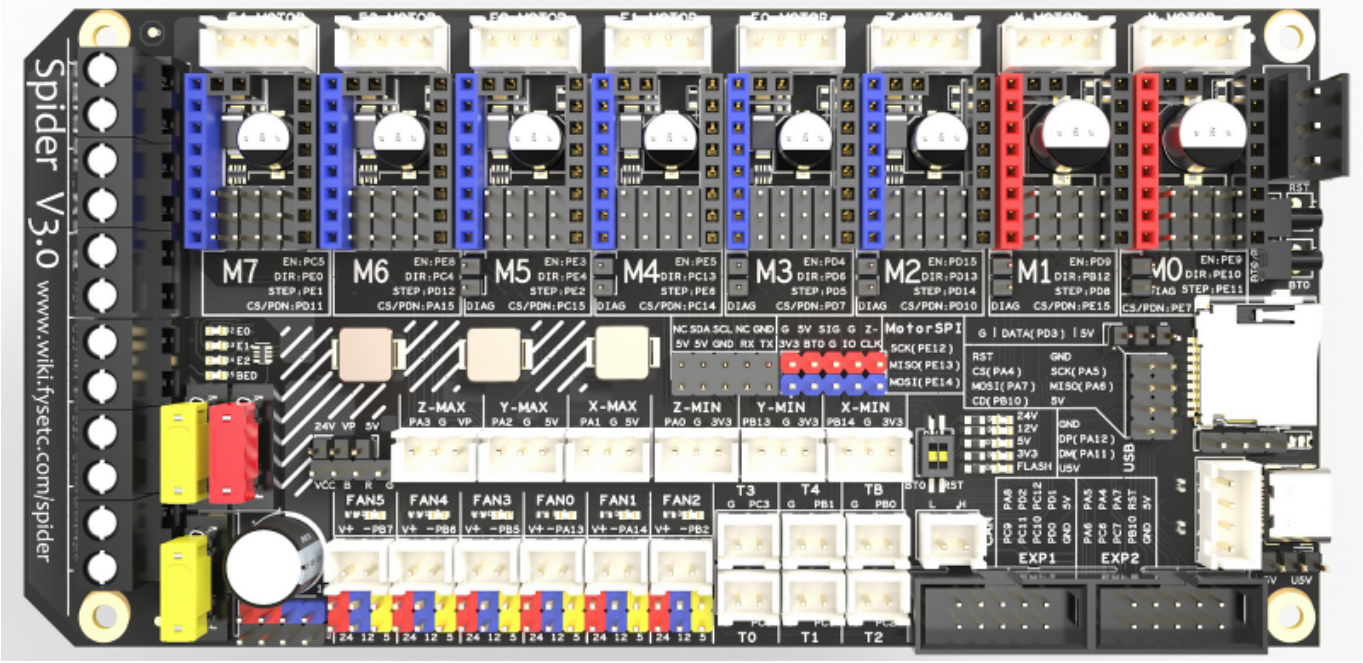
Fysetc Spider (CAN-Bridge)

Schrittweise Anleitung, um das Fysetc **Spider** Board (1.x-3.x) über die **USB/CAN Bridge** in Betrieb zu nehmen.

Fysetc Spider Board 3.0



YouTube Video #xx

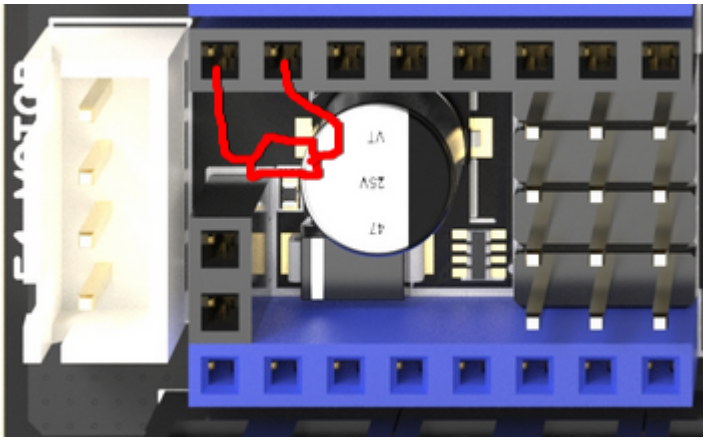
Hinweise

- **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.

3,3V Problem

Bei den Spider 1.x Boards gibt es evtl. ein Problem mit der Spannungsversorgung. Es kann unter Umständen vorkommen, dass die Elkos im Eingang und an den Schrittmotoren noch Spannung führen. Beim Einsetzen von Stepper Treibern kann es dann zu einem schlagartigen Entladen kommen. Dabei kann es die Stepper Treiber zuschießen. Deshalb bei den älteren Spider Boards folgendes machen:

- Das Board komplett stromlos machen. Am besten alle (!) Kabel ab - ALLE.
- Jetzt einen 100k Widerstand zwischen PWR-IN (24V GND) anschließen
- Einen weiteren 100k Widerstand steckt man auf einen Treiberslot an folgende Stelle (das ist immer Pin 15 & 16):



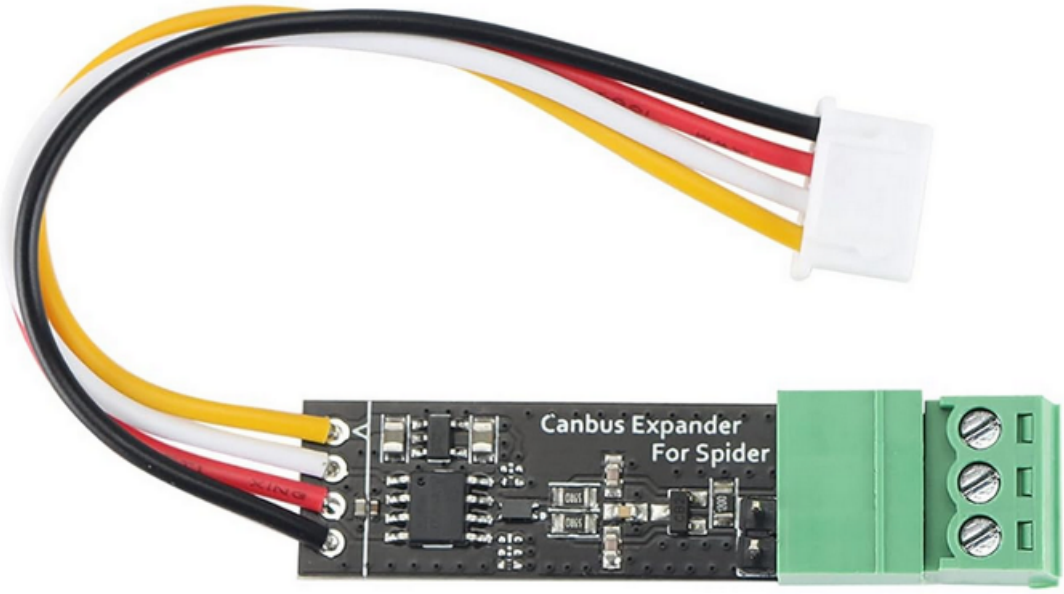
- Das lässt man dann mal 10-15 Minuten so liegen.
- Widerstände abbauen und das Bord verkabeln ...

CAN Bus Anschluss

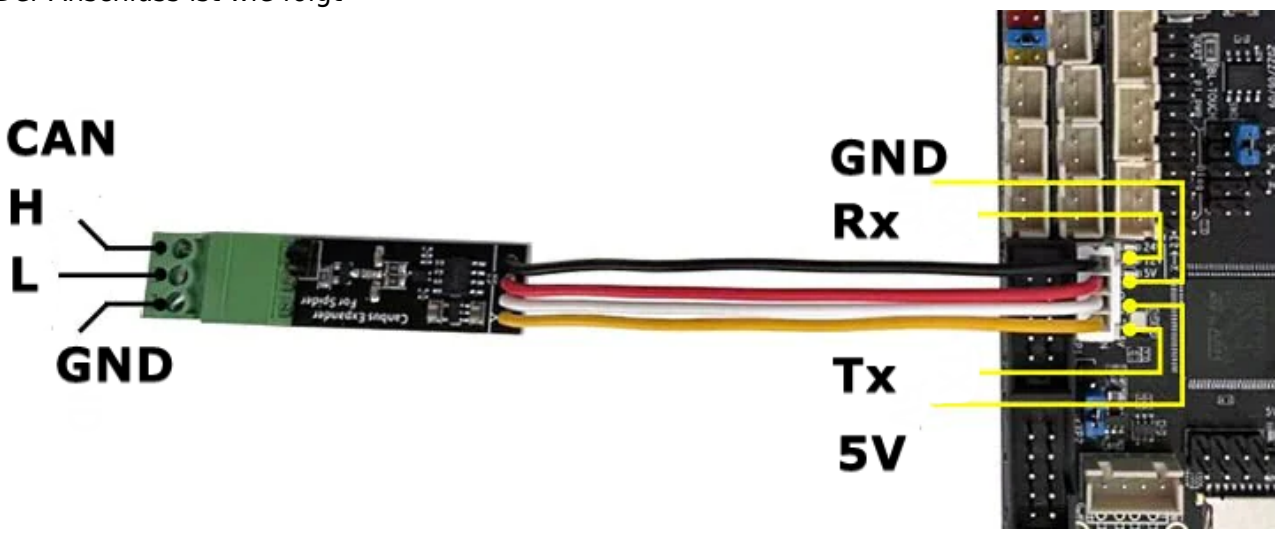
- Wer den CAN Bus überprüfen will, kann im **ausgeschalteten Zustand** den Buswiderstand mit einem Ohmmeter messen. Es müsste zwischen CAN H und CAN L ca. 60Ω ergeben. Vorausgesetzt, es ist ein zweiter Busteilnehmer verkabelt und passend terminiert.

Spider bis 2.3

- Alle Boards bis Version 2.3 haben keinen fest verbauten CAN Transceiver. Sie können also nicht direkt an einem CAN Bus betrieben werden.
- Es gibt ein Adapterboard (CANBUS Expander Module) von Fysetc mit einem entsprechenden Transceiver:



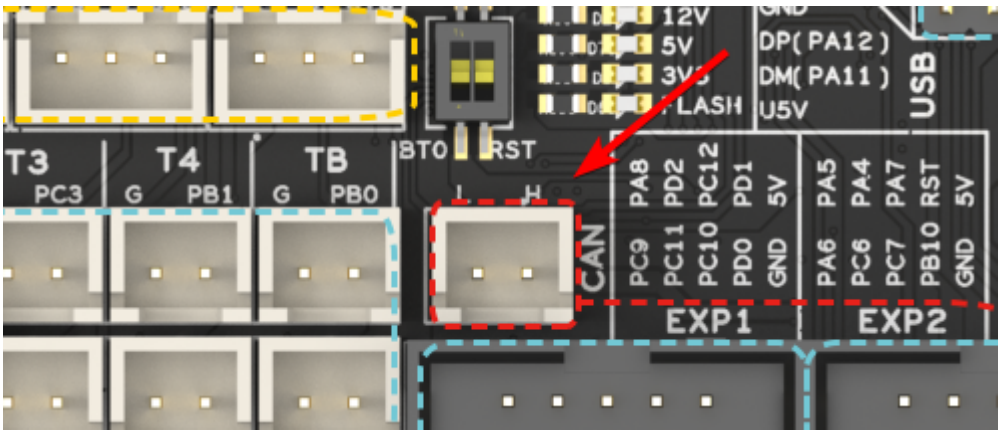
- Der Anschluss ist wie folgt



- Auf dem Spider Board befindet sich eine JST Buchse, die mit CAN beschriftet ist.
- Die 120Ohm Busterminierung kann mit dem Jumper ein- / ausgeschaltet werden - je nach Bedarf.
- Wenn das Board der erste oder letzte Busteilnehmer ist, dann muss der 120Ω Jumper gesetzt werden.

Spider ab 3.0

- Das Spider 3.0 Board hat einen Transceiver direkt verbaut.
- CAN H / CAN L kann direkt am Board angeschlossen werden:



CAN L ist hier links, CAN H rechts.

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

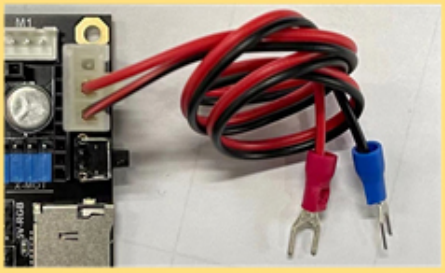
Verkabelung

- Stromversorgung über 24V / GND
- CAN Bus bei Boards < 3.0 über den Adapter, bei Boards >= 3.0 direkt am Board.
- Evtl. USB-C Verkabelung zum initialen Flashen von Klipper. Bei 24V Versorgung den 5V Jumper auf DC5V
- Ein Raspberry Pi kann direkt über das Board versorgt werden. Ein passendes Kabel liegt bei.
  - Über diesen Anschluss wird auch gleich TX / RX verbunden mit dem Pi. Damit wäre ein Betrieb über UART möglich.
- 24V / 48V für die beiden X/Y Treiber

48V Driver Support（M1&M2）:

1.If you use 24V, you need to connect the 24V-PIN and 48V-PIN, and use the jumper plug that we provide.

2.If you use 48V, you need to connect an external power supply to the 48-PIN and GND-PIN using the connecting wires provided by us.



Bootloader sichern

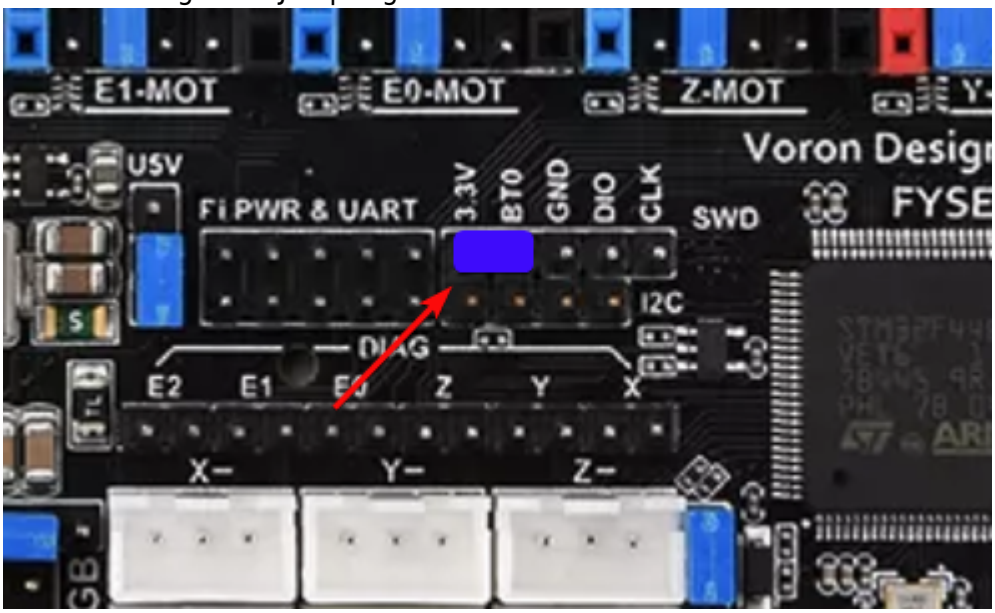
Eine Sicherung lohnt nicht. Das Board wird mit Klipper ausgeliefert und kann schnell komplett neu geflasht werden.

Vorgehen Flashen

- Wer sein Board das erste mal mit Klipper einrichtet muss die folgenden Schritte durchgehen:
  - DFU-Modus aktivieren
  - Katapult
  - Port ermitteln
  - Klipper
  - SBC
- 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`
- **bis Spider 2.3**
  Hier muss folgender Jumper gesetzt werden :
  

  Danach die Reset-Taste (die ist oberhalb vom USB-C Anschluss) einmal drücken.
  Der Jumper kann auch im Betrieb gesetzt werden.
- **ab Spider 3.0**
  Bei diesem Board gibt es an der Seite 2 Taster. Der Taster der zum USB-C Port zeigt ist BT0. Der da drüber ist Reset.
  Den BT0 Taster gedrückt halten, einmal auf Reset drücken und dann BT0 wieder loslassen.
- Das Board meldet sich mit **`Product: STM32 BOOTLOADER`** oder **`Product: DFU in FS Mode`**

```
pi@Pi4Test:~ $ dmesg -HW
[Feb27 15:48] usb 1-1.1: USB disconnect, device number 11
[  +0.370339] usb 1-1.1: new full-speed USB device number 12 using
xhci_hcd
```

```
[  +0.106951] usb 1-1.1: New USB device found, idVendor=0483,
idProduct=df11, bcdDevice=22.00
[  +0.000036] usb 1-1.1: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[  +0.000018] usb 1-1.1: Product: STM32  BOOTLOADER
[  +0.000014] usb 1-1.1: Manufacturer: STMicroelectronics
[  +0.000014] usb 1-1.1: SerialNumber: STM32FxSTM32
```

  - 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 (STM32F446)  --->
    Build Katapult deployment application (Do not build)  --->
    Clock Reference (12 MHz crystal)  --->
    Communication interface (USB (on PA11/PA12))  --->
    Application start offset (32KiB 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
(PD3)   Status LED GPIO Pin
```

  - **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
- Die Status LED vom Board sollte jetzt blinken (Die LED kann ja noch Board an unterschiedlichen Stellen sein!)

Port ermitteln

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

```
pi@Pi3Test:~/katapult $ dmesg -HW
[  +0.261499] usb 1-1.1: new full-speed USB device number 13 using
xhci_hcd
[  +0.111977] usb 1-1.1: New USB device found, idVendor=1d50,
idProduct=6177, bcdDevice= 1.00
[  +0.000036] usb 1-1.1: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[  +0.000017] usb 1-1.1: Product: stm32f446xx
[  +0.000014] usb 1-1.1: Manufacturer: katapult
[  +0.000014] usb 1-1.1: SerialNumber: 390028000950315239323320
[  +0.016088] cdc_acm 1-1.1:1.0: ttyACM0: USB ACM device
```

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

Klipper flashen

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

```
[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F446)  --->
    Bootloader offset (32KiB bootloader)  --->
    Clock Reference (12 MHz crystal)  --->
    Communication interface (USB to CAN bus bridge (USB on PA11/PA12))
--->
    CAN bus interface (CAN bus (on PD0/PD1))  --->
    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@Pi3Test:~/klipper $ make -j4 flash FLASH_DEVICE=/dev/ttyACM0
  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: v0.12.0-102-g9f41f53c
```

```
  Linking out/klipper.elf
  Creating hex file out/klipper.bin
  Flashing out/klipper.bin to /dev/ttyACM0
Entering bootloader on /dev/ttyACM0
Device reconnect on
/sys/devices/platform/soc/3f980000.usb/usb1/1-1/1-1.4/1-1.4:1.0
/usr/bin/python3 lib/canboot/flash_can.py -d /dev/serial/by-
path/platform-3f980000.usb-usb-0:1.4:1.0 -f out/klipper.bin

Attempting to connect to bootloader
CanBoot Connected
Protocol Version: 1.0.0
Block Size: 64 bytes
Application Start: 0x8008000
MCU type: stm32f446xx
Flashing '/home/pi/klipper/out/klipper.bin'...

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

Write complete: 2 pages
Verifying (block count = 488)...

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

Verification Complete: SHA = 14F85E3BBD86FCBA99F270F03AD3FC60DDCE8F71
CAN Flash Success
```

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

```
pi@Pi4Test:~/klipper $ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 016: ID 1d50:606f OpenMoko, Inc. Geschwister Schneider
CAN adapter
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. 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@Pi3Test:~/klipper $ ~/klippy-env/bin/python
~/klipper/scripts/canbus_query.py can0
Found canbus_uuid=c57dc3b70ff0, Application: Klipper
Total 1 uuids found
```

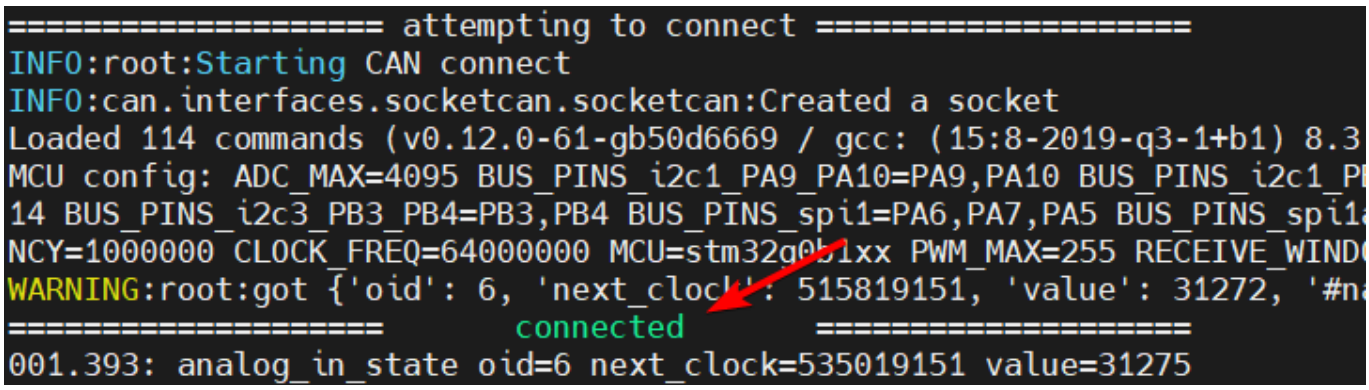
  - Die **UUID** (canbus_uuid=**c57dc3b70ff0**) notieren !
  - Wird bei diesem Schritt kein Board gefunden, hilft oft ein Reset am Board (entweder über Reset Taster oder 1x Strom weg und wieder dran)

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 c57dc3b70ff0`

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**
  `wget https://raw.githubusercontent.com/FYSETC/FYSETC-SPIDER/main/firmware/Klipper/printer.cfg -O printer.cfg`
- `nano ~/printer_data/config/printer.cfg`

```
[mcu]
canbus_uuid: c57dc3b70ff0
#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: c57dc3b70ff0
```

- 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 c57dc3b70ff0 -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: 3500530008513131333539 32
[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`

Links

- Github Repo
  https://github.com/FYSETC/FYSETC-SPIDER
- Schaltplan
  Spider 1.x
  https://github.com/FYSETC/FYSETC-SPIDER/blob/main/hardware/V1.x/Spider%20V1.0C%20SCH.pdf
  Spider 2.2
  https://github.com/FYSETC/FYSETC-SPIDER/blob/main/hardware/V2.2/Spider%20V2.2%20SCH.pdf

Spider 2.3
https://github.com/FYSETC/FYSETC-SPIDER/blob/main/hardware/V2.3/Spider_V2.3_SCH.pdf
Spider 3.0
https://github.com/FYSETC/FYSETC-SPIDER/blob/main/hardware/V3.0/spider_V3.0_sch.pdf

- Klipper Konfig
https://github.com/FYSETC/FYSETC-SPIDER/blob/main/firmware/Klipper/printer.cfg