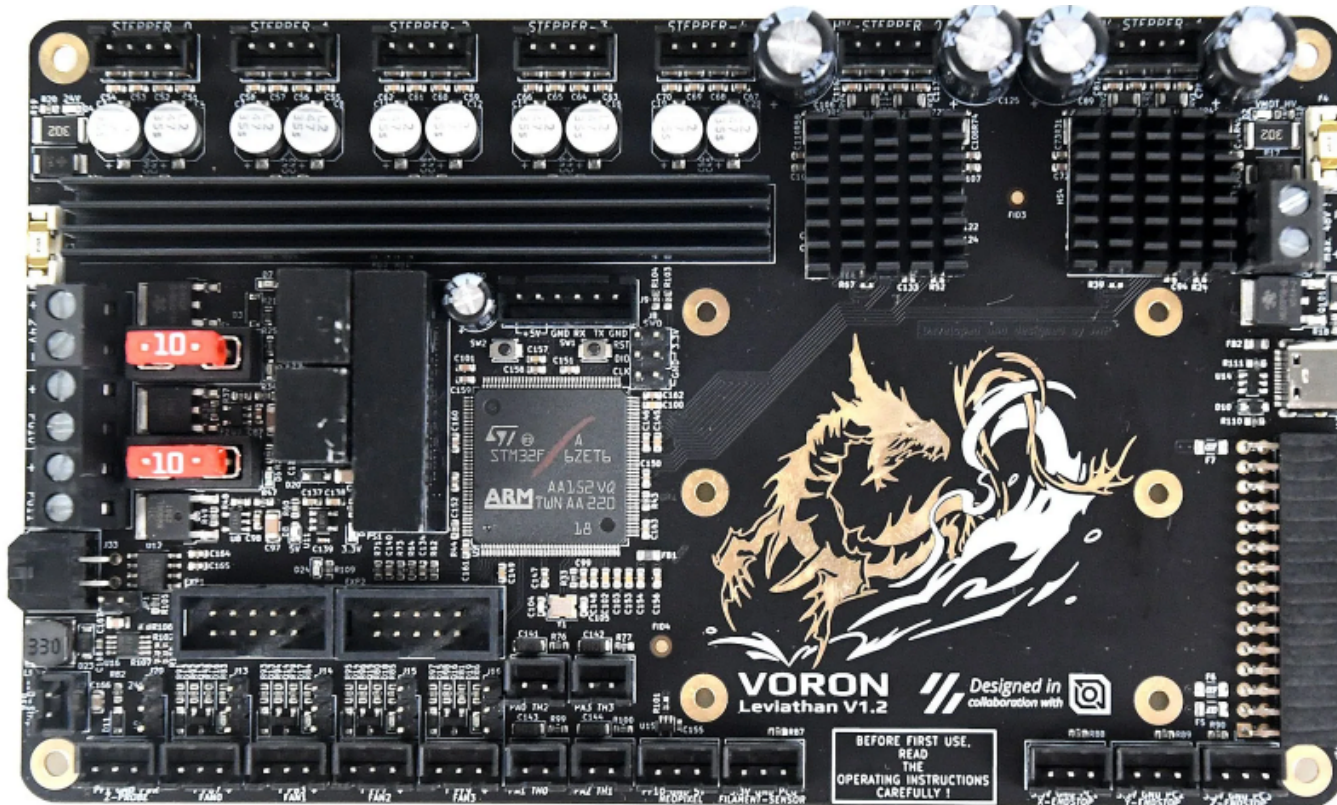


YTVI Leviathan V1.2 (CAN-Bridge)

Schrittweise Anleitung, um das LDO **Leviathan V1.2** Board über die **USB/CAN Bridge** in Betrieb zu nehmen.

Leviathan V1.2



YouTube Video #86



Hinweise

- **SBC** bedeutet in der Anleitung **Single Board Computer**. 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!

- Endlich mal ein Board **ohne** diesen **unnützen SD-Kartenslot** 😎
- Kein Molex Stecker für CAN, keine Ersatzsicherungen

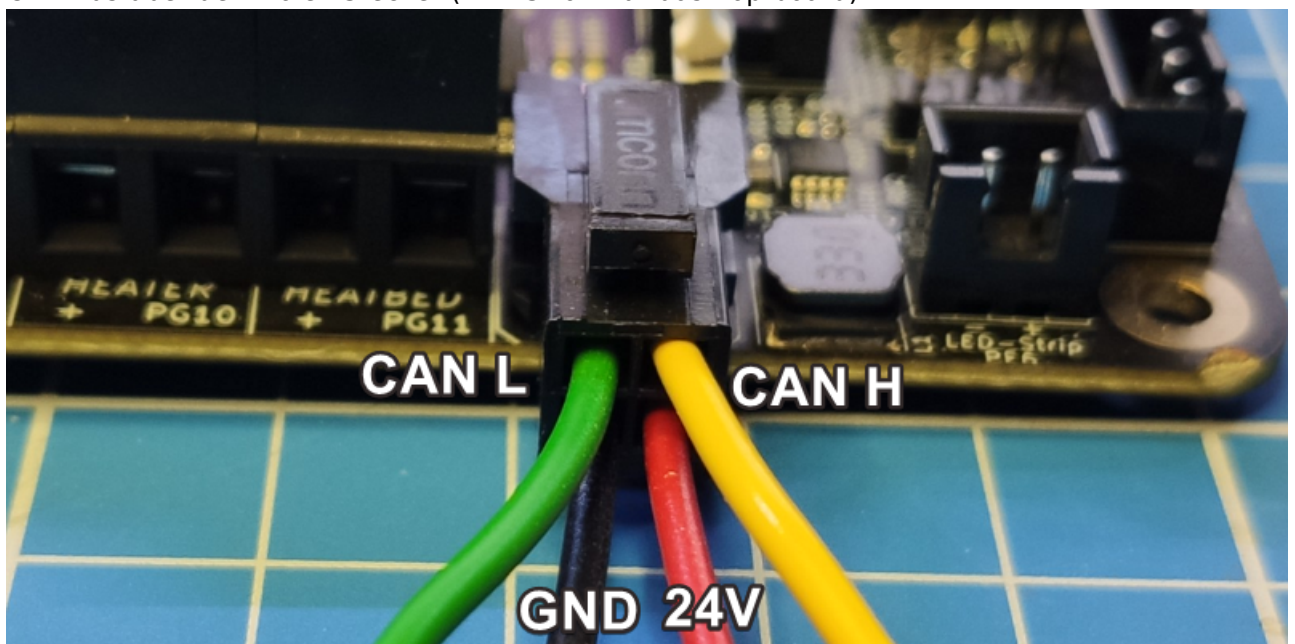
Stromversorgung

- Im Betrieb wird das Board mit 24V versorgt (Anschluss BOARD + 24V -)
- Das Board kann zum initialen Flashen alleine über den USB-C Port versorgt werden. Das Flashen funktioniert natürlich auch, wenn das Board über 24V betrieben wird.
- Es gibt keinen dedizierten Jumper für die 5V Versorgung.
- Für die beiden TMC5160 muss über dem USB-Port extra eine Stromversorgung installiert werden (24-28V).
- Es gibt einen extra Port für die Versorgung des Raspberry Pi. Er liefert 5,06V bei max. 3A - somit sollte auch kein "Undervoltage detected" auftreten. Ein Pi 5 wird sich dennoch über die

Stromversorgung beschweren 😊

Verkabelung

- [Stromversorgung](#) über 24V / GND
- CAN Bus über den Molex Stecker (inkl. Strom für das Kopfboard)



- additiv USB-C Verkabelung zum initialen Flashen von Klipper
- Ein Raspberry Pi kann direkt auf das Board geschraubt werden. Es liegt eine Adapterplatine bei, die über das 6 polige Kabel mit dem Board verbunden wird. Über diesen Anschluss wird der Raspberry Pi mit 5,06V versorgt. Über diesen Anschluss wird auch gleich TX / RX verbunden mit dem Pi. Damit ist ein Betrieb über UART möglich.



CAN Bus Terminierung

- Wenn das Board der erste oder letzte Busteilnehmer ist, dann muss der 120Ω Jumper gesetzt werden.



- 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.

Bootloader sichern

Das Board hat bei Auslieferung keinen Bootloader. Es muss nichts gesichert werden. Sollte Katapult als Bootloader vorhanden sein, ist es eh sinnvoll den neu zu kompilieren bei Bedarf.

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
 - `dmesg -HW`
 - Die **Boot Taste** (SW2) am Board gedrückt halten und dabei das Board über USB mit dem SBC verbinden



Wenn das Board schon über 24V angeschlossen ist geht auch folgendes **Boot Taste** (SW2) gedrückt halten, **Reset Taste** (SW1) einmal drücken und loslassen, dann die **Boot Taste** (SW2) loslassen.

- Das Board meldet sich mit **Product: DFU in FS Mode**

```
pi@Pi3Test:/boot $ dmesg -HW
[Jan23 18:39] usb 1-1.4: USB disconnect, device number 20
[ +0.805709] usb 1-1.4: new full-speed USB device number 21 using
dwc_otg
[ +0.141330] usb 1-1.4: New USB device found, idVendor=0483,
idProduct=df11, bcdDevice=22.00
[ +0.000038] usb 1-1.4: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[ +0.000017] usb 1-1.4: Product: STM32 BOOTLOADER
[ +0.000013] usb 1-1.4: Manufacturer: STMicroelectronics
[ +0.000012] usb 1-1.4: 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
(PE1) 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 (SW1) drücken
 - oder das Board einmal stromlos machen

Port ermitteln

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

```

pi@Pi3Test:~/katapult $ dmesg -HW
[Feb 1 11:01] usb 1-1.4: new full-speed USB device number 16 using dwc_otg
[ +0.147164] usb 1-1.4: New USB device found, idVendor=1d50, idProduct=6177, bcdDevice= 1.00
[ +0.000039] usb 1-1.4: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ +0.000016] usb 1-1.4: Product: stm32f446xx
[ +0.000013] usb 1-1.4: Manufacturer: katapult

```

```
[ +0.000011] usb 1-1.4: SerialNumber: 350053000851313133353932  
[ +0.005586] 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

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 PB5/PB6)) --->  
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/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: v0.12.0-102-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

```

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=3c98a1ec0529, Application: Klipper
Total 1 uuids found
```

- Die **UUID** (canbus_uuid=**3c98a1ec0529**) 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 3c98a1ec0529
```

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_WIND
WARNING:root:got {'oid': 6, 'next_clock': 515819151, 'value': 31272, '#n
=====
001.393: analog_in_state oid=6 next_clock=535019151 value=31275
```

Konfiguration

- cd ~/printer_data/config
- **Beispiel Konfiguration Voron 2**
wget https://raw.githubusercontent.com/MotorDynamicsLab/Leviathan/master/Klipper_config/voron2_leviathan_v1.2.cfg -O printer.cfg
- **Beispiel Konfiguration Trident**
wget https://raw.githubusercontent.com/MotorDynamicsLab/Leviathan/master/Klipper_config/trident_leviathan_v1.2.cfg -O printer.cfg
- nano ~/printer_data/config/printer.cfg

```
[mcu]
canbus_uuid: 3c98a1ec0529
#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 Update

- 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: 3c98a1ec0529
```

- Das Leviathan Board per flashtool.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 <LEVIATHAN UUID> -r`

```
pi@Pi3Test:~/klipper $ ~/klippy-env/bin/python
~/katapult/scripts/flashtool.py -i can0 -u 3c98a1ec0529 -r
Sending bootloader jump command...
Bootloader request command sent
Flash Success
```

- 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.

- `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`

- `cansniffer can0`
- `~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0`

Links

- Github Repo
<https://github.com/MotorDynamicsLab/Leviathan/tree/master>
- LDO Infoseite
<https://www.ldomotion.com/p/guide/VORON-Leviathan-V12>
- Manual
https://github.com/MotorDynamicsLab/Leviathan/blob/e4fb6d27322140ee2509b9061d3bbc16aa7cf56c/Manual/Leviathan_V1.2_Manual.pdf
- Schaltplan
https://github.com/MotorDynamicsLab/Leviathan/blob/e4fb6d27322140ee2509b9061d3bbc16aa7cf56c/Schematic/Leviathan_V1.2.pdf
- Klipper Konfig
Voron :
https://github.com/MotorDynamicsLab/Leviathan/blob/e4fb6d27322140ee2509b9061d3bbc16aa7cf56c/Klipper_config/voron2_leviathan_v1.2.cfg
Trident :
https://github.com/MotorDynamicsLab/Leviathan/blob/e4fb6d27322140ee2509b9061d3bbc16aa7cf56c/Klipper_config/trident_leviathan_v1.2.cfg
- Bootlader Entry bei Bridge Mode
https://github.com/Klipper3d/klipper/blob/master/docs/Bootloader_Entry.md

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

Permanent link:
https://drklipper.de/doku.php?id=klipper_faq:flash_guide:stm32f446:leviathan_v1.2_can_bridge&rev=1706797324

Last update: **2024/02/01 15:22**

