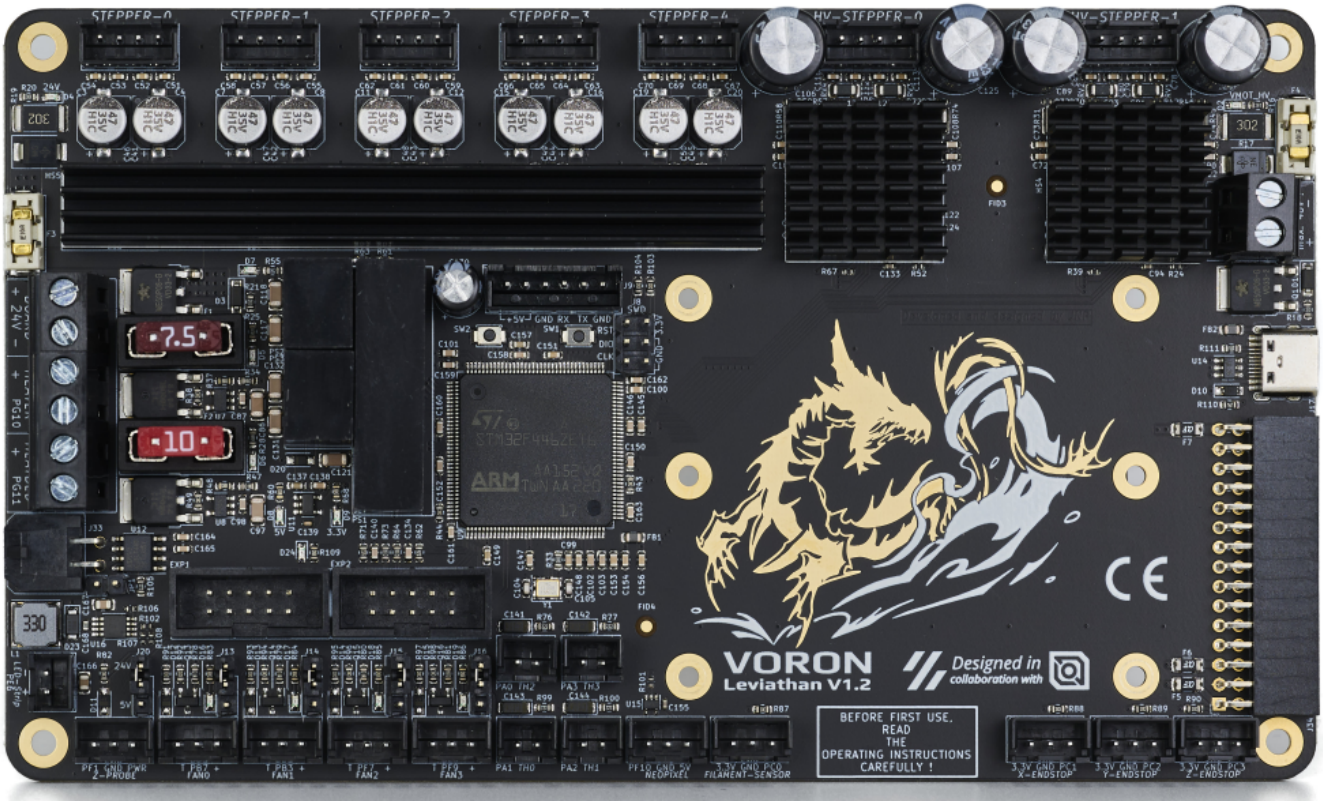
Leviathan V1.2 (USB)

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

Leviathan V1.2



YouTube Video #85

Video

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!
- In der Originalanleitung wird darauf verwiesen, dass die Firmware mit einem 32KiB Bootloader

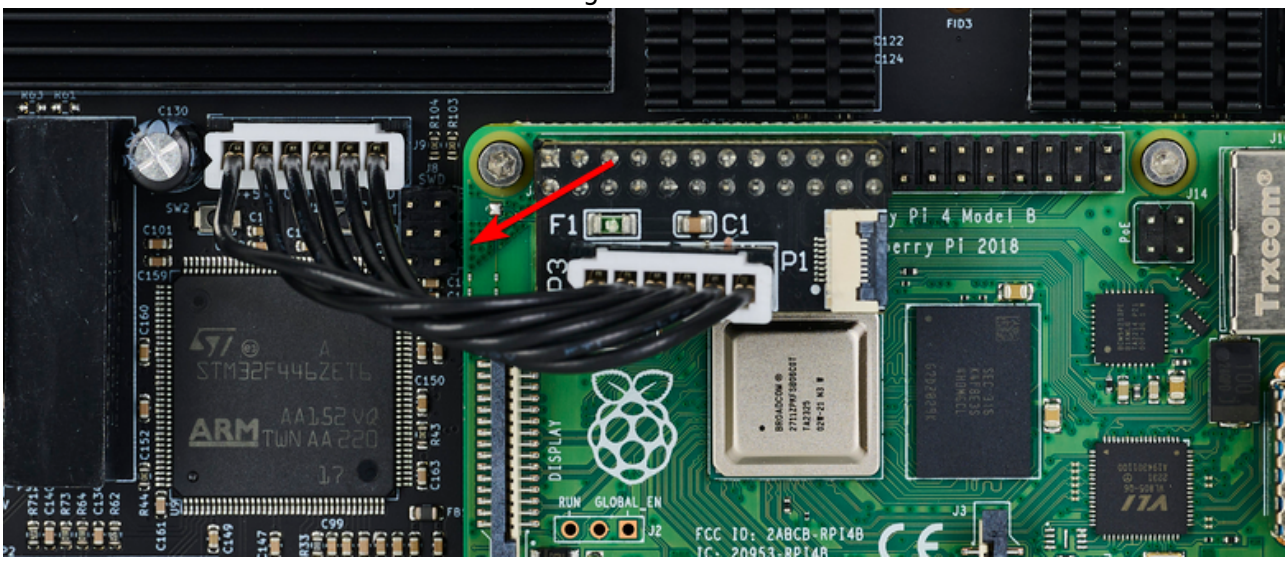
Offset kompiliert werden soll. Allerdings konnte ich auf meinem Testboard keinen Bootloader entdecken (der macht auch gar keinen Sinn, weil es keinen SD-Karten Slot gibt). Das würde dazu führen, dass das Update nicht funktioniert!
- Wenn das Board per USB betrieben wird, ist kein Bootloader nötig! Klipper kann das Board selber updaten → siehe Klipper Update …
- Endlich mal ein Board **ohne** diesen **unnützen SD-Kartenslot** 😎
- Kein Molex Stecker für CAN, keine Ersatzsicherungen

Stromversorgung

- 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.
- Im Betrieb wird das Board mit 24V versorgt (Anschluss BOARD + 24V -)
- 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

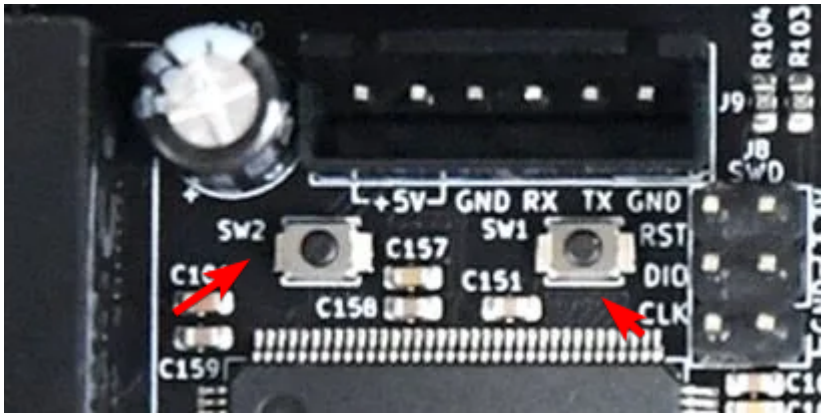
- Ganz normale USB-C Verkabelung
- zusätzlich die Stromversorgung
- Ein Raspberry Pi kann direkt auf das Board geschraubt werden. Es liegt eine Adapterplatine bei, die über das 6 polige Kabel mit dem Board verbunden wird. Über diesen Anschluss wird der Raspberry Pi mit 5,06V versorgt. Über diesen Anschluss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



- Das Board kann auch anstatt USB über UART betrieben werden
  PA9/PA10 sind die entsprechenden Pins dafür.
  Dafür muss die Firmware neu kompiliert werden!

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

Klipper flashen

> **Hinweis**
> Wenn das Board mit USB betrieben wird, braucht es **keinen extra Bootloader**!

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

```
[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F446)  --->
    Bootloader offset (No bootloader)  --->
    Clock Reference (12 MHz crystal)  --->
    Communication interface (USB (on PA11/PA12))  --->
```

```
    USB ids  --->
()  GPIO pins to set at micro-controller startup
```

  - beenden mit Q und Y
- Klipper kompilieren
  `make -j4`
- Klipper flashen
  `dfu-util -R -a 0 -s 0x08000000:mass-erase:force -D ~/klipper/out/klipper.bin`
  Das Ergebnis sollte sein `File downloaded successfully`

Port ermitteln

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

```
pi@Pi3Test:~/klipper $ dmesg -HW
[Jan23 19:11] usb 1-1.4: new full-speed USB device number 29 using dwc_otg
[  +0.147013] 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
[  +0.000017] usb 1-1.4: Product: stm32f446xx
[  +0.000012] usb 1-1.4: Manufacturer: Klipper
[  +0.000011] usb 1-1.4: SerialNumber: 350053000851313133353932
[  +0.008035] cdc_acm 1-1.4:1.0: ttyACM0: USB ACM device
```

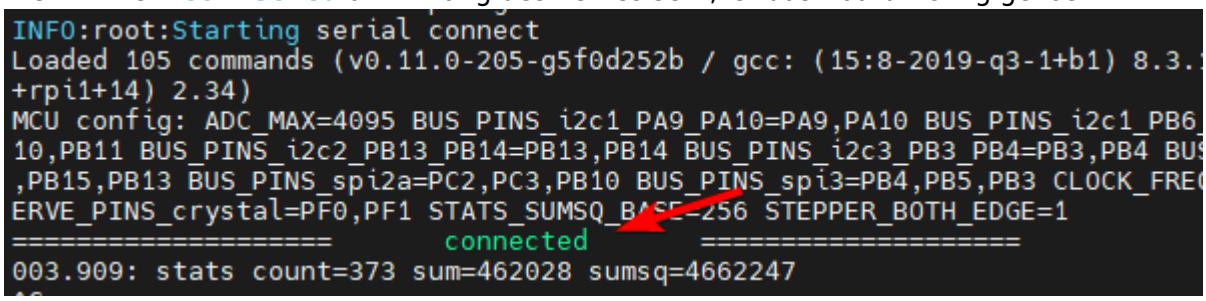
  - Wir brauchen die Information mit **tty...** also in diesem Fall **ttyACM0**
  - STRG+C drücken, um die Meldungen zu beenden
- Den zugehörigen Link ermitteln
  `ls -lR /dev/ | grep -v '\->\s../tty' | grep -e 'tty[[:alpha:]]' -e serial`
  - Wir brauchen die Info unter `/dev/serial/by-id:` :
    `lrwxrwxrwx 1 root root 13 Nov 11 15:15 usb-Klipper_stm32f446xx_350053000851313133353932-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_stm32f446xx_350053000851313133353932-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_stm32f446xx_350053000851313133353932-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.



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]
##--------------------------------------------------------------------
serial: /dev/serial/by-id/usb-
Klipper_stm32f446xx_3500530008513131333539 32-if00
```

    - Die Zeile mit `serial` entsprechend mit dem ermittelten Pfad von oben anpassen

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
    - Am Ende kann es zu einem Fehler kommen. Davon nicht irritieren lassen. Wichtig ist diese Zeile:
      `File downloaded successfully`
- Klipper Dienst starten
  `sudo systemctl start klipper.service`
- Sollte sich das Board nicht melden, am besten den Drucker einmal stromlos machen und neu starten.

Sonstiges

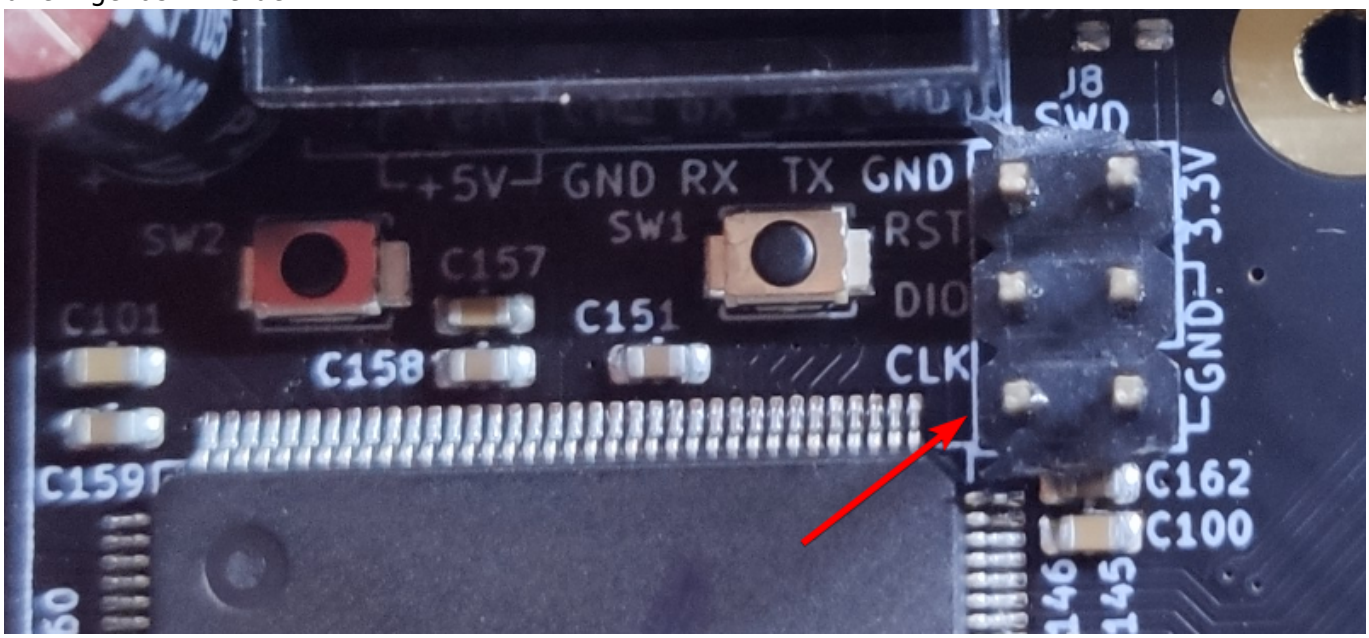
Diese Punkte sind nicht immer Bestandteil vom YouTube Video, aber nützlich 😎

Auslieferungszustand

- Klipper v0.11.0-148-g52f4e20c
- kein Bootloader

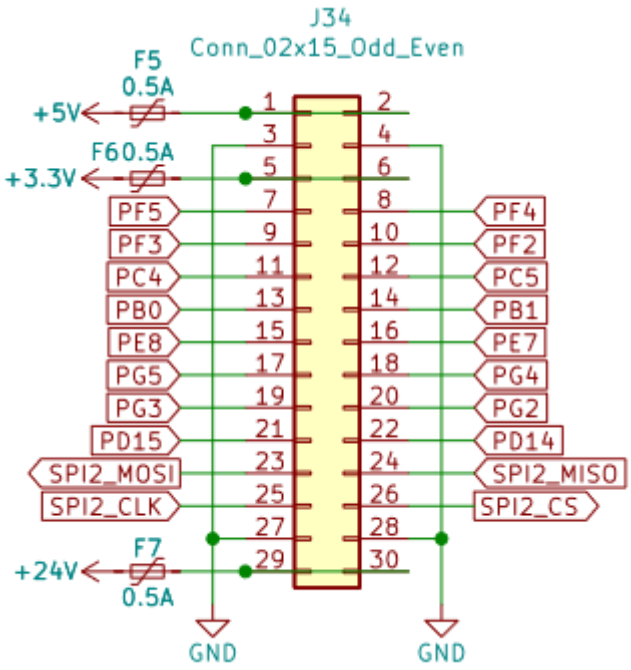
ST-Link (SWD)

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



Extra Port

Das Board hat einen Expansion Port unter dem USB-C Port. Der Port kann im Grunde für alle möglichen Erweiterungen genutzt werden.





ADXL345 (Input Shaper)

Ein ADXL345 Sensor für Input Shaper kann direkt an das Board angeschlossen werden.

- Anschluss am Expansion Port

ADXL345 Pin	Leviathan V1.2 Expansion Port Pin	STM32 Pin
GND	GND (3 / 4 / 27 / 28)	
VCC	+5V (1 / 2)	
CS	SPI2_CS (26)	PB12
INT1	N/A	
INT2	N/A	
SDO	SPI2_MISO (24)	PB14
SDA	SPI2_MOSI (23)	PB15
SCL	SPI2_CLK (25)	PB13

- Konfig Anpassung

```
[adxl345]
```

```
axes_map     : x,y,z
cs_pin       : PB12
spi_bus      : spi2

[resonance_tester]
accel_chip   : adxl345
probe_points : 150, 150, 20 # Center of your bed, raised up a little
```

- **Test** in der MainSail Konsole mittels
  ACCELEROMETER_QUERY
  Als Ergebnis sollte in etwa sowas kommen:
  `accelerometer values (x, y, z): -1110.308913, 1184.329507, 11414.822920`
- Sollte der Test folgenden Fehler bringen ist die Verkabelung falsch!
  `Invalid adxl345 id (got 0 vs e5)`

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

- 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

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

Permanent link:
https://drklipper.de/doku.php?id=klipper_faq:flash_guide:stm32f446:leviathan_v1.2_usb&rev=1707729874

Last update: **2024/02/12 10:24**