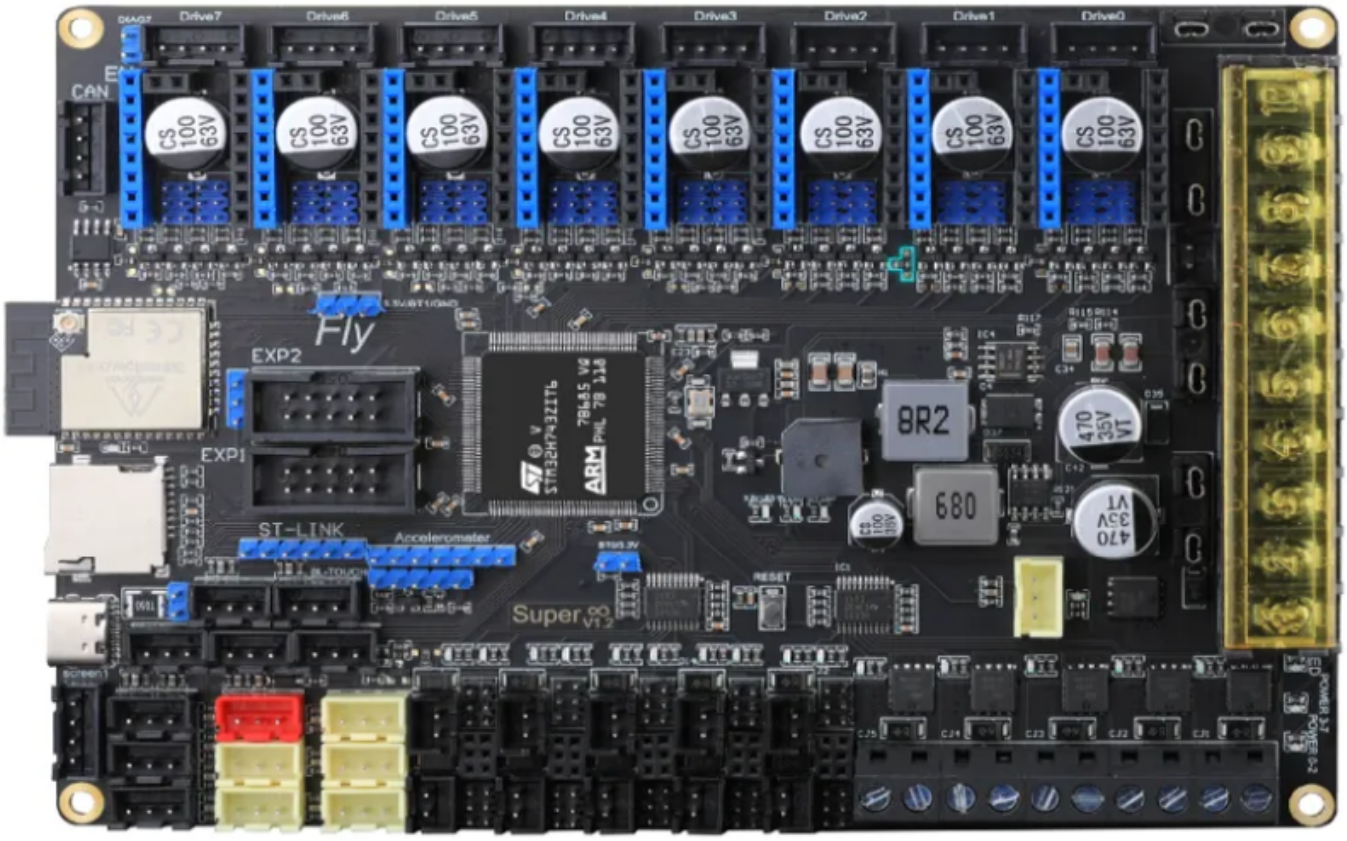
Mellow Fly-Super8Pro (USB)

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

Mellow Fly-Super8Pro



YouTube Video #1xx

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!

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

TBD

- Alle Treiber können per Jumper selektierbar mit 24V oder 48V betrieben werden.
- Die 48V am Eingang werden nur für die Treiber genutzt!
- 24V müssen trotzdem anliegen!

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: 2F0015000951313430323835
[  +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!

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

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 (STM32H723)  --->
    Bootloader offset (No bootloader)  --->
    Clock Reference (25 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 und USB wieder anstecken

```
pi@TestPi5:~/klipper $ dmesg -HW
[Sep 4 07:56] usb 3-1: USB disconnect, device number 10
[  +0.437458] usb 3-1: new full-speed USB device number 11 using xhci-hcd
[  +0.190009] usb 3-1: New USB device found, idVendor=1d50, idProduct=614e, bcdDevice= 1.00
[  +0.000005] usb 3-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[  +0.000002] usb 3-1: Product: stm32h723xx
[  +0.000002] usb 3-1: Manufacturer: Klipper
[  +0.000001] usb 3-1: SerialNumber: 31000E000951323530343536
[  +0.241050] cdc_acm 3-1:1.0: ttyACM0: USB ACM device
```

  - Wir brauchen die Information mit **tty...** also in diesem Fall **ttyACM0**
  - STRG+C drücken, um die Meldungen zu beenden
- 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 Mar 2 06:59` **`usb-Klipper_stm32h723xx_31000E000951323530343536-if00`** `→ ../../ttyACM0`
  - Achte darauf das am Ende die gleiche tty Bezeichnung steht wie sie im vorherigen Schritt ermittelt wurde (hier also ttyACM0)
  - Was wir für die Konfig brauchen ist dann am Ende:
    `/dev/serial/by-id/usb-Klipper_stm32h723xx_31000E000951323530343536-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_stm32h723xx_31000E000951323530343536-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.

```
INFO:root:Starting serial connect
Loaded 105 commands (v0.11.0-205-g5f0d252b / gcc: (15:8-2019-q3-1+b1) 8.3.
+rpi1+14) 2.34)
MCU config: ADC_MAX=4095 BUS_PINS_i2c1_PA9_PA10=PA9,PA10 BUS_PINS_i2c1_PB6_
10,PB11 BUS_PINS_i2c2_PB13_PB14=PB13,PB14 BUS_PINS_i2c3_PB3_PB4=PB3,PB4 BU
,PB15,PB13 BUS_PINS_spi2a=PC2,PC3,PB10 BUS_PINS_spi3=PB4,PB5,PB3 CLOCK_FRE
ERVE_PINS_crystal=PF0,PF1 STATS_SUMSQ_BASE=256 STEPPER_BOTH_EDGE=1
====================       connected       ====================
003.909: stats count=373 sum=462028 sumsq=4662247
```

Abbrechen kann man die Abfrage mittels `STRG + C`.

Konfiguration

TBD

- `cd ~/printer_data/config`
- ACHTUNG NOCH KEINE AKTUELLE KONFIG
  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]
serial: /dev/serial/by-id/usb-Klipper_stm32f446xx_390028000950315239323320-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`

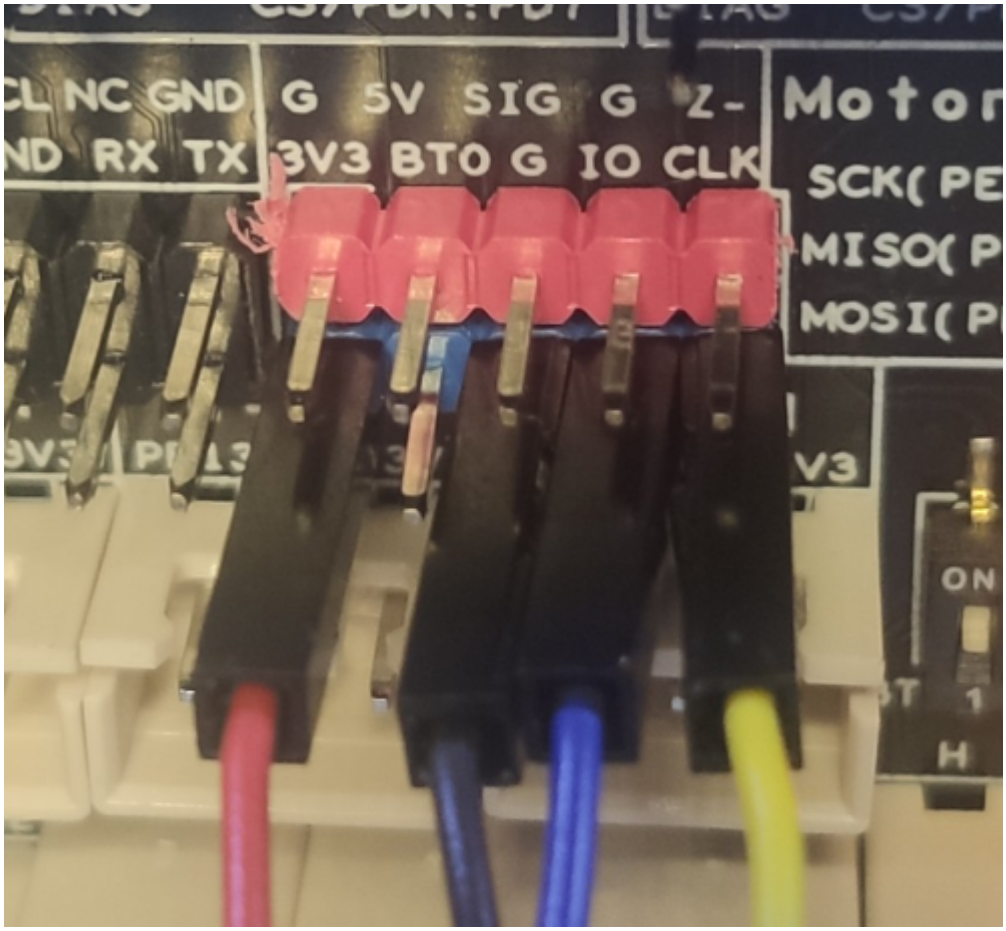
- Klipper Dienst starten
  `sudo systemctl start klipper.service`
- Sollte sich das Board nicht melden, am besten den Drucker einmal stromlos machen und neu starten.

Sonstiges

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

ST-Link (SWD)

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



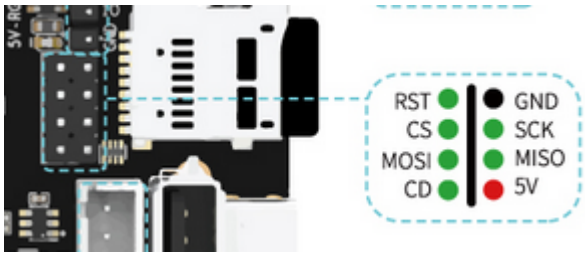
- 3V3 → 3,3V Anschluss
- G → Masse Anschluss
- IO → SWDIO vom ST-Link
- CLK → SWCLK vom ST-Link

ADXL345 (Input Shaper)

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

Bis Board 2.2

Ab Board 2.3



ADXL345 Pin	Spider Board (SPI1)	STM32 Pin
GND	GND	
VCC	+5V	
CS	CS	PA4
INT1	N/A	
INT2	N/A	
SDO	MISO	PA6
SDA	MOSI	PA7
SCL	CLK	PA5

- Konfig Anpassung

```
[adxl345]
axes_map     : x,y,z
cs_pin       : PA4
spi_bus      : spi1

[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

- 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
  TBD !

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

Permanent link:
https://drklipper.de/doku.php?id=klipper_faq:flash_guide:stm32h723:mellow_fly-super8pro&rev=1732900667

Last update: **2024/11/29 18:17**