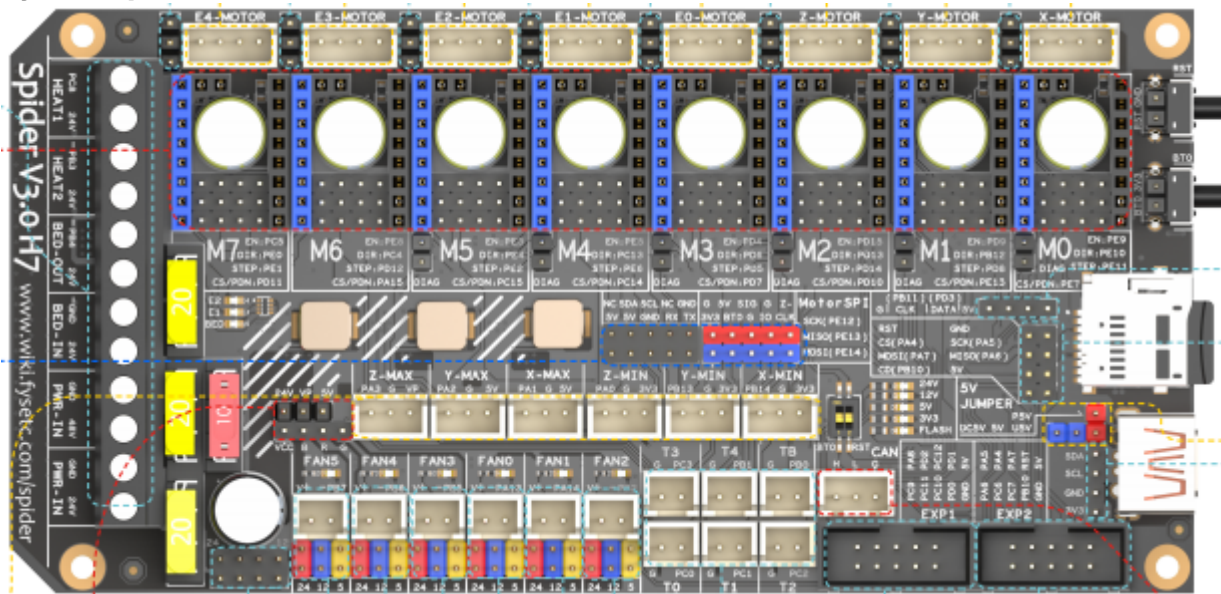


Fysetc Spider H7 (CAN-Bridge)

Schrittweise Anleitung, um das Fysetc **Spider H7** Board über die **USB/CAN Bridge** in Betrieb zu nehmen.

Fysetc Spider Board H7



YouTube Video #???






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

Verkabelung

Stromversorgung

- Der Jumper ist entweder mit DC5V / U5V, DC5V / USB5V oder nur U5V gekennzeichnet.
- Der Jumper DC5V / U5V befindet sich hinter dem USB-C Anschluss.
-  Bei 24V Versorgung muss der Jumper immer auf die Position DC5V 
- **Betrieb**
 - Im Betrieb wird das Board mit 24V versorgt (Anschluss PWR-IN GND / 24V)
 -  Der Jumper auf Position DC5V setzen!
- **Firmware flashen**
 - Das Board wird **nicht mit 24V versorgt**.
Den Jumper auf Position U5V (oder USB5V) setzen.
 - Das Board wird **mit 24V betrieben**.
Den Jumper auf Position DC5V setzen!

Versorgung Raspberry Pi

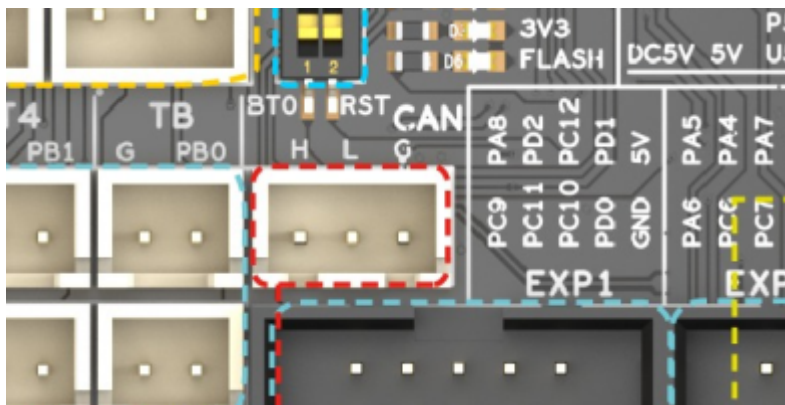
- Ein Raspberry Pi kann direkt über das Board versorgt werden. Ein passendes Kabel liegt bei.
- Über diesen Anschluss wird auch gleich TX / RX verbunden mit dem Pi. Damit wäre ein Betrieb über UART möglich.

48V Anschluss

- 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!

TBD ! 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 Spider H7 3.0 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!

Versorgung Raspberry Pi

- 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.

Bootloader sichern

Das Board wird mit Marlin ausgeliefert (Stand 04.09.2024).

```
pi@TestPi5:~ $ dmesg -HW
[Sep 4 06:54] usb 3-1: USB disconnect, device number 2
[ +0.405181] usb 3-1: new full-speed USB device number 3 using xhci-hcd
[ +0.183840] usb 3-1: New USB device found, idVendor=0483, idProduct=5740,
bcdDevice= 0.00
[ +0.000005] usb 3-1: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[ +0.000002] usb 3-1: Product: MARLIN_STM32H723VG CDC in FS Mode
[ +0.000002] usb 3-1: Manufacturer: STMicroelectronics
[ +0.000002] usb 3-1: SerialNumber: 364334613532
[ +0.028036] cdc_acm 3-1:1.0: ttyACM0: USB ACM device
```

Mit den original Bootloader Settings (128k) lässt sich das Board nicht betreiben. Der Hex Abzug zeigt auch das da scheinbar gar kein Bootloader installiert ist / war! Es gibt also nichts zu sichern ...

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
- An der Seite sind zwei 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
[Sep  4 06:55] usb 3-1: USB disconnect, device number 3
[ +0.448648] usb 3-1: new full-speed USB device number 4 using xhci-hcd
[ +0.154183] usb 3-1: not running at top speed; connect to a high speed hub
[ +0.024000] usb 3-1: New USB device found, idVendor=0483, idProduct=df11, bcdDevice= 2.00
[ +0.000005] usb 3-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ +0.000002] usb 3-1: Product: DFU in FS Mode
[ +0.000002] usb 3-1: Manufacturer: STMicroelectronics
[ +0.000002] usb 3-1: SerialNumber: 364334613532
```

- 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
(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 (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 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@TestPi5:~/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: v0.12.0-296-gcc4ad6670
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/axi/1000120000.pcie/1f00300000.usb/xhci-
hcd.1/usb3/3-1/3-1:1.0
/usr/bin/python3 lib/canboot/flash_can.py -d /dev/serial/by-
path/platform-xhci-hcd.1-usb-0:1:1.0 -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: stm32h723xxv0.0.1-75-g90eb71b
Flashing '/home/pi/klipper/out/klipper.bin'...

[#####]

Write complete: 1 pages
Verifying (block count = 600)...

[#####]

Verification Complete: SHA = ABD0ECEAC094657FE40E51D0EE1F824DFC011383
```

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@TestPi5:~ $ ~/klippy-env/bin/python
~/klipper/scripts/canbus_query.py can0
```

```
Found canbus_uuid=4cd6d5f5fd9a, Application: Klipper
Total 1 uuids found
```

- Die **UUID** (canbus_uuid=**4cd6d5f5fd9a**) 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 4cd6d5f5fd9a
```

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
==== connected =====
001.393: analog_in_state oid=6 next_clock=535019151 value=31275
```

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: 4cd6d5f5fd9a
#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: 4cd6d5f5fd9a
```

- Das Leviathan Board per flashtool.py resetten. Welche UUID das Leviathan hat kann man bei mehreren Busteilnehmern leider nicht ohne weiteres 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 4cd6d5f5fd9a-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 es **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

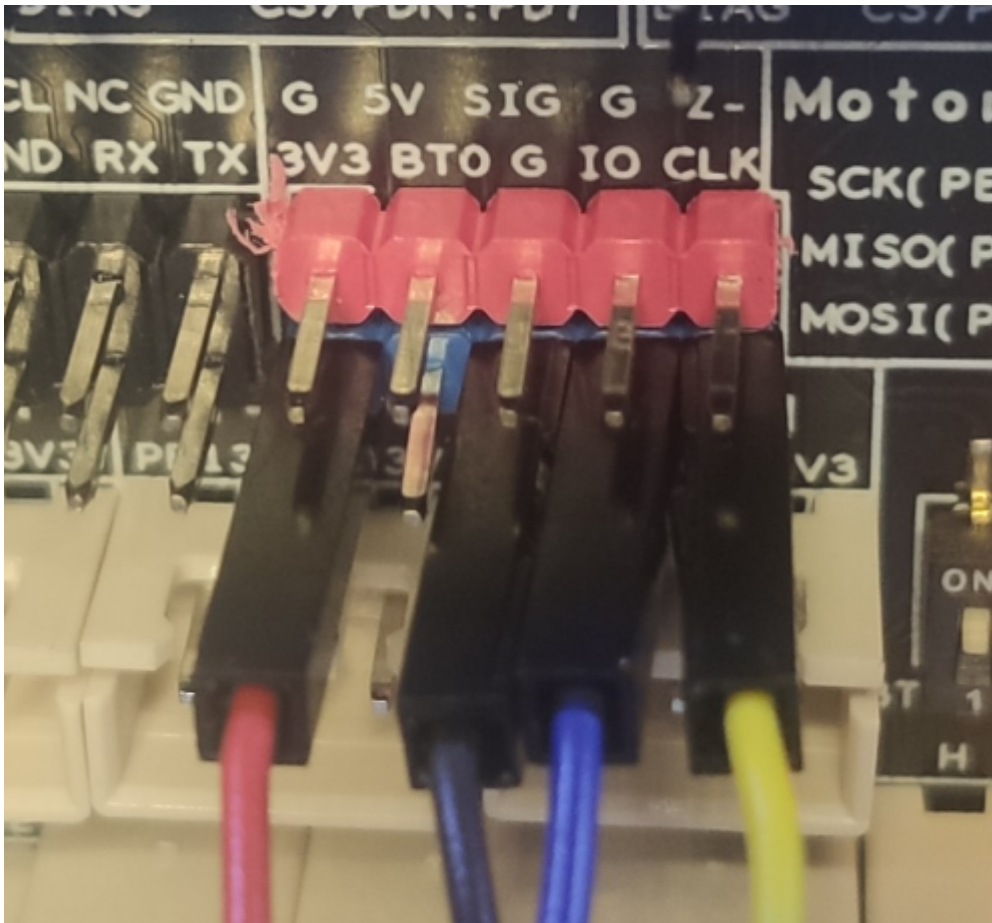
Sonstiges

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



ST-Link (SWD)

Das Board verfügt über einen 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

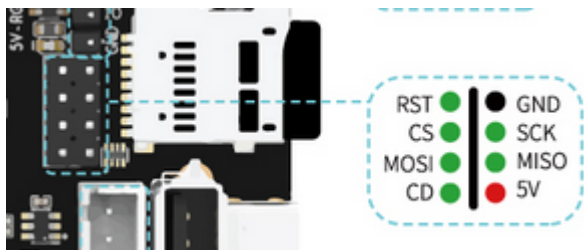
ADXL345 (Input Shaper)

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

- Github Repo
<https://github.com/FYSETC/FYSETC-SPIDER-H7>
- Schaltplan
Spider H7
https://github.com/FYSETC/FYSETC-SPIDER-H7/blob/0a01a5dd6fb8491fde522b36390bc144476f1634/Schematic/SPIDER_H7_SCH.pdf
- Klipper Konfig
TBD !

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

Permanent link:
https://www.drklipper.de/doku.php?id=klipper_faq:flash_guide:stm32h723:fysetc_spider_h7_can_bridge&rev=1727407871

Last update: **2024/09/27 05:31**

