

# RP2040 Input Shaper Hardware

In Anlehnung an

<https://klipper.discourse.group/t/raspberry-pi-pico-adxl345-portable-resonance-measurement/1757>

## YouTube Video #92



## Kommerziell

- Fysetc  
<https://github.com/FYSETC/FYSETC-PortableInputShaper>  
<https://de.aliexpress.com/item/1005006079473058.html> direkt an Nozzle
- BTT  
<https://github.com/bigtreetech/ADXL345>  
<https://github.com/bigtreetech/LIS2DW>
- Mellow 3D  
[https://mellow-3d.github.io/fly\\_adxl345\\_usb\\_general.html](https://mellow-3d.github.io/fly_adxl345_usb_general.html)
- Provok3d Nozzle Adxl (STM32 !)  
<https://provok3d.com/product/nozzle-adxl-board/?v=0a10a0b3e53b>
- Fly3d Drucker CNC V6 Düsen Halter  
<https://de.aliexpress.com/item/1005006425869823.html>

## DIY


- RP2040 + ADXL Sensor
- <https://www.printables.com/de/model/586008-nozzle-bracket-for-adxl345-accelerometer>

## Kabel


- mit kurzen Kabeln (~ 20cm) initial testen um Verbindungsprobleme zu vermeiden
- Im nächsten Schritt dann mit längeren Kabeln testen.
  - Die Kabel sollten mindestens gedrillt sein.
  - Einige nutzen auch geschirmtes CAT 6a Kabel

- Grundsätzlich sollte das Kabel so kurz wie eben möglich sein, um Verluste und Störungen gering zu halten.

## Spannungsregler

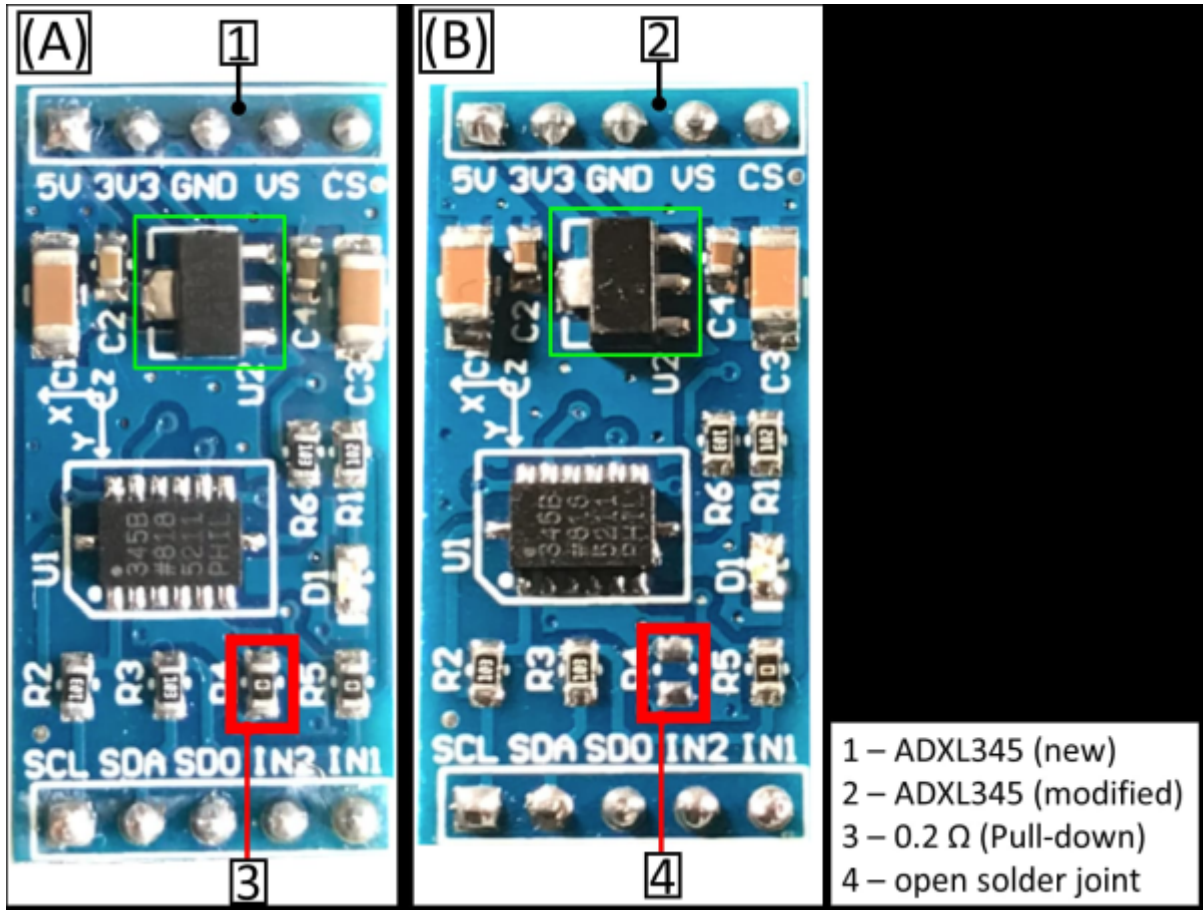
- Der ADXL345 Chip braucht für den Betrieb 3,3V. Er ist nicht 5V kompatibel!
- Durch den Spannungsregler (meist ein IC mit 5 Pins) kann das Board mit 5V versorgt werden. Der ADXL wird dann mit den 3,3V des Spannungsreglers betrieben.
-  Boards ohne Spannungsregler dürfen nicht mit 5V betrieben werden, da sonst der ADXL zerstört wird.
- Boards mit Spannungsregler funktionieren in der Regel auch, wenn sie mit 3,3V an VCC betrieben werden. Am ADXL liegen dann eher 2,8-3,0V an, was aber kein Problem darstellt.
- **Faustregel** Boards immer erst mit 3,3V testen!

## Level Shifter

- Der ADXL ist so designed das er normalerweise mit 3,3V Signalen arbeitet.
- Auch die STM32, RP2040, Raspberry Pi, ... arbeiten im Normalfall mit diesen Spannungen.
- Es gibt aber Controller wie mit 5,0 oder gar geringeren Spannungen wie 1,8V arbeiten. Und hier kommen dann Levelshifter zum Einsatz, die ein Signal in beide Richtungen umformen können.
- Im Bereich des 3D Druck sind ADXL Boards mit Level Shifter in der Regel unnötig. Sollte man doch ein Board mit Levelshifter haben, sollten beide Spannungsversorgungen mit 3,3V belegt werden.
-  Wird ein Levelshifter mit 5V betrieben kann dies einen Controller zerstören, der nur mit 3,3V arbeitet!
- Levelshifter sind in der Regel zu erkennen an zwei kleinen 3 Pin Mosfets nahe den Signalleitungen.

## I2C

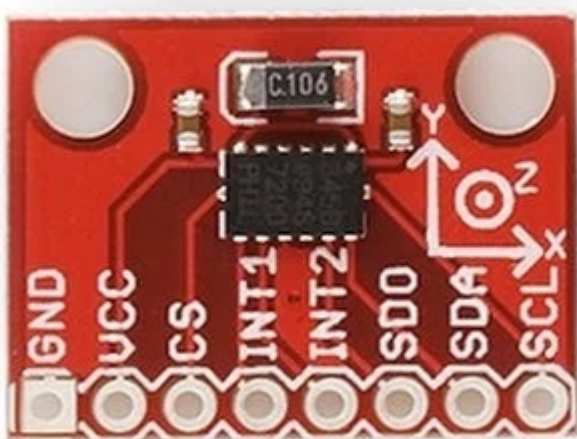
- Einige ADXL Boards werden mit I2C anstatt SPI ausgeliefert. I2C ist aber (ohne 400kbit/s fast mode) zu langsam um die 3200 Samples/s für die Resonanz Messung zu liefern.
- Diese Boards verfügen oft über zusätzliche Pullup Widerstände an den Signalleitungen (SDA, SCL). Es ist möglich die Boards umzulöten. (Siehe Klipper Github Issue #3637)



- Grundsätzlich sollte man aber darauf achten eine SPI Version zu kaufen.

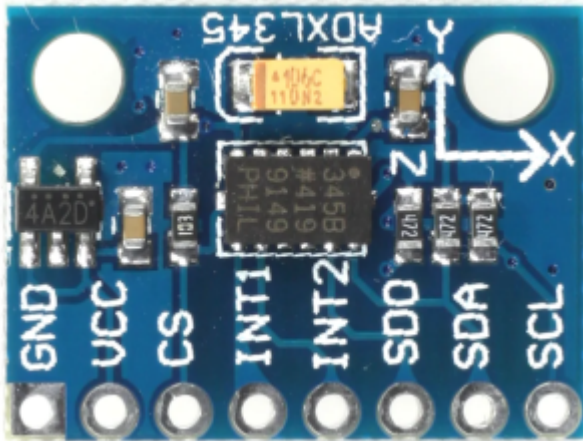
### Sensoren Beispiele

- **Kein** Spannungsregler, **kein** Levelshifter



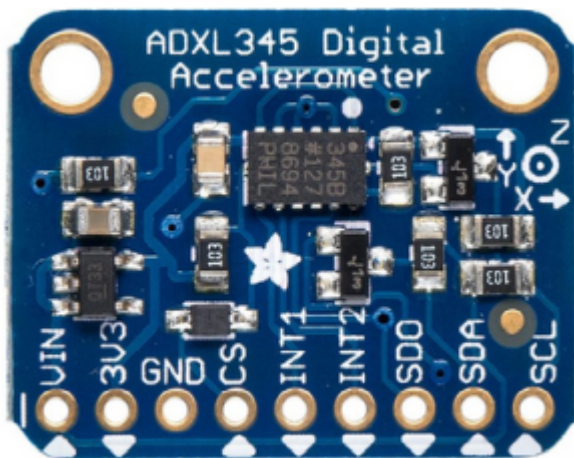
<https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345-BreakoutBoard-v13.pdf>

- Spannungsregler, **kein** Levelshifter



<https://www.roboter-bausatz.de/media/image/c1/f0/e2/GY-291-Schaltplan.jpg>

- Spannungsregler, Levelshifter



<https://learn.adafruit.com/adxl345-digital-accelerometer/downloads>

## Klipper flashen


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

```
[*] Enable extra low-level configuration options
Micro-controller Architecture (Raspberry Pi RP2040) --->
Bootloader offset (No bootloader) --->
Flash chip (W25Q080 with CLKDIV 2) --->
Communication interface (USB) --->
USB ids --->
() GPIO pins to set at micro-controller startup
```

- Mit Q und dann Y beenden
- `make clean && make -j4`
- Die Boot Taste **gedrückt halten (!)** und den USB-C Stecker anschließen.
  - Meistens haben die Boards eh nur eine Taste. Sollten es zwei Tasten sein, muss man schauen was Boot und was Reset ist.
- `dmesg` im Terminal eingeben

```
[126509.556147] usb 1-1.4: new full-speed USB device number 13 using
dwc_otg
[126509.688730] usb 1-1.4: New USB device found, idVendor=2e8a,
idProduct=0003, bcdDevice= 1.00
[126509.688767] usb 1-1.4: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[126509.688783] usb 1-1.4: Product: RP2 Boot
[126509.688795] usb 1-1.4: Manufacturer: Raspberry Pi
[126509.688807] usb 1-1.4: SerialNumber: E0C9125B0D9B
[126509.689785] usb-storage 1-1.4:1.0: USB Mass Storage device detected
[126509.690375] scsi host0: usb-storage 1-1.4:1.0
[126510.727183] scsi 0:0:0:0: Direct-Access      RPI      RP2
3      PQ: 0 ANSI: 2
[126510.727877] sd 0:0:0:0: Attached scsi generic sg0 type 0
[126510.728901] sd 0:0:0:0: [sda] 262144 512-byte logical blocks: (134
MB/128 MiB)
[126510.729746] sd 0:0:0:0: [sda] Write Protect is off
[126510.729766] sd 0:0:0:0: [sda] Mode Sense: 03 00 00 00
[126510.730619] sd 0:0:0:0: [sda] No Caching mode page found
[126510.730636] sd 0:0:0:0: [sda] Assuming drive cache: write through
[126510.744971]  sda: sda1
[126510.746027] sd 0:0:0:0: [sda] Attached SCSI removable disk
```


Es muss sowas am Ende stehen ... **Product: RP2 Boot, Manufacturer: Raspberry Pi**

-  Wird kein Board gefunden, dann kann der Speichertyp (Flash Chip) falsch sein. In dem Fall wechseln auf GENERIC:

```
( ) W25Q080 with CLKDIV 2
(X) GENERIC_03H with CLKDIV 4
```

- Das Board mit Klipper flashen  
make flash FLASH\_DEVICE=2e8a:0003
- Das Board sollte sich dann schon mit Klipper melden ...  
ls -lR /dev/ | grep -v '\->\s../tty' | grep -e 'tty[[:alpha:]]' -e serial

```
/dev/serial:
/dev/serial/by-id:
lrwxrwxrwx 1 root root 13 Jan 22 20:26 usb-
Klipper_rp2040_E66138935F3C6B28-if00 -> ../../ttyACM0
/dev/serial/by-path:
lrwxrwxrwx 1 root root 13 Jan 22 20:26 platform-3f980000.usb-
usb-0:1.4:1.0 -> ../../ttyACM0
```

- Wenn nicht einfach mal USB raus und wieder rein und fertig 

## kurzer Test

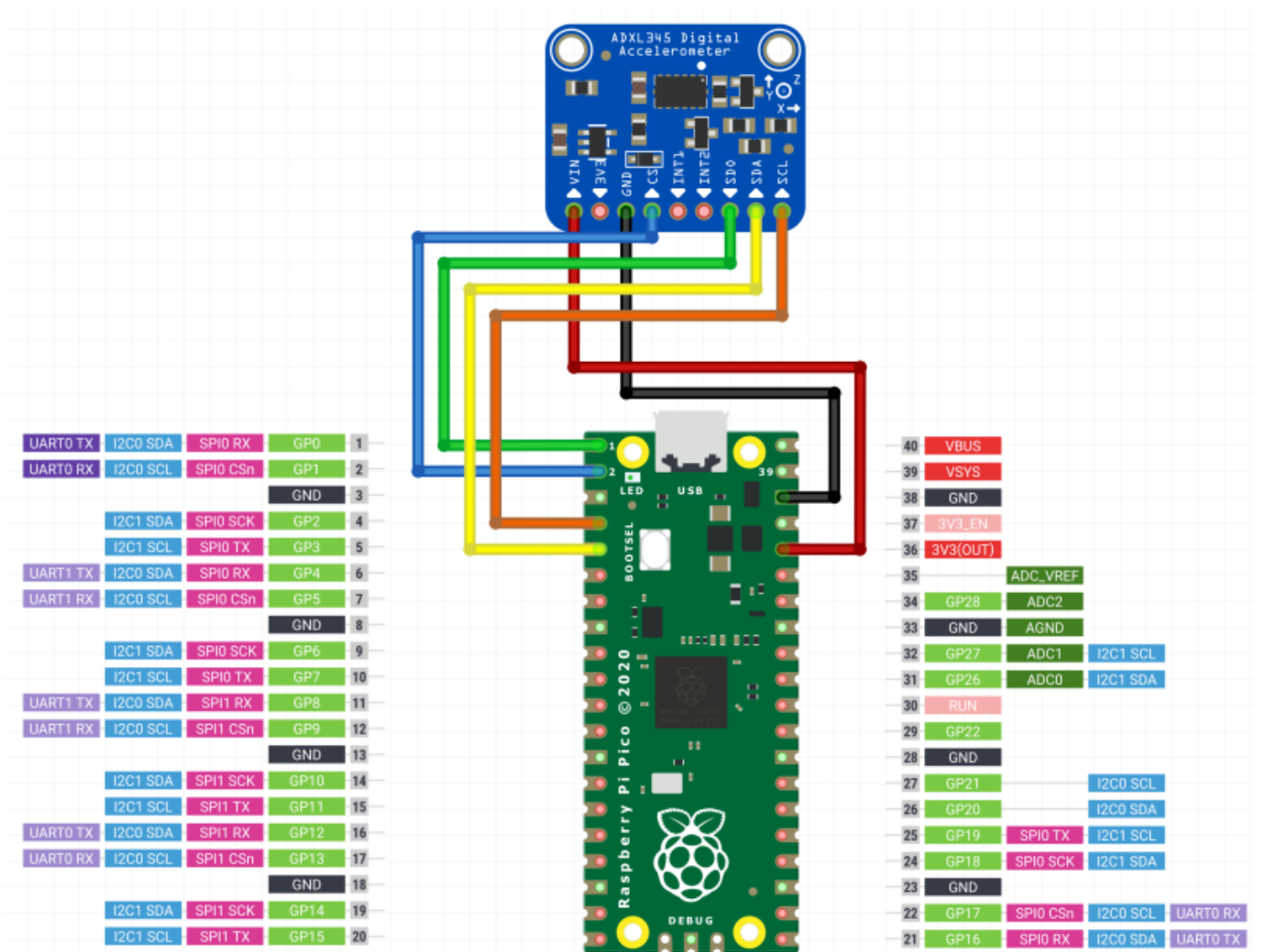
Ob das Board korrekt mit Klipper läuft, lässt sich mit folgendem Befehl schnell testen:  
~/klippy-env/bin/python ~/klipper/klippy/console.py /dev/serial/by-id/usb-Klipper\_rp2040\_E6625C05E7A17228-if00

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) 8.3.0
+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 BUS
,PB15,PB13 BUS_PINS_spi2a=PC2,PC3,PB10 BUS_PINS_spi3=PB4,PB5,PB3 CLOCK_FREQ
ERVE_PINS_crystal=PF0,PF1 STATS_SUMSQ_BASE=256 STEPPER_BOTH_EDGE=1
=====
                          connected
=====
003.909: stats count=373 sum=462028 sumsq=4662247
```

## RP2040 SPI Ports



## Hardware SPI

Der RP2040 verfügt theoretisch über 7 nutzbare SPI Ports. Dafür sind aber meistens nicht alle Pins rausgeführt, oder Pins haben eine unglückliche Doppelbelegung.

spi0d und spi1c sind nur eingeschränkt verfügbar / nutzbar!

SPI Bus	CS	SDO (MISO)	SDA (MOSI)	SCL (CLK)	getestet	Konfiguration
<b>RPi Pico pin Name</b>	CSn	SPI RX	SPI TX	SPI SCK		
<b>spi0a</b>	gpio1	gpio0	gpio3	gpio2	OK	cs_pin : RP2040:gpio1 spi_bus : spi0a
<b>spi0b</b>	gpio5	gpio4	gpio7	gpio6	OK	cs_pin : RP2040:gpio5 spi_bus : spi0b
<b>spi0c</b>	gpio17	gpio16	gpio19	gpio18	OK	cs_pin : RP2040:gpio17 spi_bus : spi0c
<b>spi0d</b>	gpio ?	gpio20	gpio23	gpio22		
<b>spi1a</b>	gpio9	gpio8	gpio11	gpio10	OK	cs_pin : RP2040:gpio9 spi_bus : sp1a
<b>spi1b</b>	gpio13	gpio12	gpio15	gpio14	OK	cs_pin : RP2040:gpio12 spi_bus : spi1b
<b>spi1c</b>	gpio ?	gpio24	gpio27	gpio26		

## Software SPI

SPI Bus	CS	SDO (MISO)	SDA (MOSI)	SCL (CLK)	Konfiguration	Notes
<b>Fysetc Portable Input Shaper</b>	gpio13	gpio12	gpio11	gpio10	cs_pin : RP2040:gpio13 spi_software_sclk_pin: RP2040:gpio10 spi_software_mosi_pin: RP2040:gpio11 spi_software_miso_pin: RP2040:gpio12 axes_map: x, -z, y	
<b>BTT ADXL345 V2.0</b>	gpio9	gpio8	gpio11	gpio10	cs_pin : RP2040:gpio9 spi_software_sclk_pin: RP2040:gpio10 spi_software_mosi_pin: RP2040:gpio11 spi_software_miso_pin: RP2040:gpio8 axes_map: -x, -y, -z	Alternativ spi_bus: sp1a
<b>BTT LIS2DW V1.0</b>	gpio9	gpio8	gpio11	gpio10	cs_pin : RP2040:gpio9 spi_software_sclk_pin: RP2040:gpio10 spi_software_mosi_pin: RP2040:gpio11 spi_software_miso_pin: RP2040:gpio8 axes_map: -y, x, -z	Alternativ spi_bus: sp1a

SPI Bus	CS	SDO (MISO)	SDA (MOSI)	SCL (CLK)	Konfiguration	Notes
<b>BTT EBB SB2209</b>	gpio1	gpio3	gpio0	gpio2	cs_pin: EBBCan:gpio1 spi_software_sclk_pin: EBBCan:gpio2 spi_software_mosi_pin: EBBCan:gpio0 spi_software_miso_pin: EBBCan:gpio3 axes_map: z, -y, x	
<b>Mellow Fly-ADXL345-USB</b>	gpio9	gpio12	gpio11	gpio10	cs_pin : RP2040:gpio9 spi_software_sclk_pin: RP2040:gpio10 spi_software_mosi_pin: RP2040:gpio11 spi_software_miso_pin: RP2040:gpio12	

### Hinweis

Wenn Software SPI verwendet wird, können beliebige gpio Pins verwendet werden!  
Bei den Kaufversionen sind die Pins allerdings fest vorgegeben!

## Konfiguration

- Eigene Konfigurations-Datei für den ADXL anlegen

[adxl345.cfg](#)

```
[mcu RP2040]
serial      : /dev/serial/by-id/usb-Klipper_rp2040_E66138935F717729-if00
baud       : 250000
restart_method : command

[resonance_tester]
accel_chip : adxl345          # adxl345 / lis2dw
probe_points : 150, 150, 20 # an example of the center of the bed, check yours

[adxl345]
# [lis2dw]
axes_map      : x,y,z          # Default
# axes_map    : x,z,y          # Installed with vertical ADXL
# axes_map    : x,-z,y         # Fysetc Portable Input Shaper
# axes_map    : -y,x,-z        # BTT LIS2DW V1.0

cs_pin        : RP2040:gpio1
# Hardware SPI Bus
spi_bus       : spi0a
```

```
# Software SPI Bus
# spi_software_sclk_pin: RP2040:gpio10
# spi_software_mosi_pin: RP2040:gpio11
# spi_software_miso_pin: RP2040:gpio12
```

- folgende Anpassungen sind nötig:
  - seriellen Port eintragen
  - Den Chip Typ `accel_chip` anpassen
  - Die Bettmitte bei `probe_points` ggf. ändern
  - Evtl. die `axes_map` anpassen
  - Den Bus korrekt konfigurieren mit `cs_pin` und `spi_bus` (bzw. `spi_software_...`)

## Dual ADXL

Es ist möglich, mehrere ADXL an einem Controller zu betreiben, z.B. für Drucker mit beweglichem Bett. In dem Fall werden die Achsen über unterschiedliche ADXL vermessen.

[https://www.klipper3d.org/Measuring\\_Resonances.html#bed-slinger-printers](https://www.klipper3d.org/Measuring_Resonances.html#bed-slinger-printers)

Eine Konfiguration dafür könnte so aussehen

```
[mcu RP2040]
serial      : /dev/serial/by-id/usb-Klipper_rp2040_E66138935F717729-if00
baud       : 115200      # 250000
restart_method : command

[resonance_tester]
accel_chip_x  : adxl345 hotend
accel_chip_y  : adxl345 bed
probe_points  : 150, 150, 20

[adxl345 hotend]
cs_pin       : RP2040:gpio1
spi_bus      : spi0a

[adxl345 bed]
cs_pin       : RP2040:gpio5
spi_bus      : spi0b
```

## 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 oder die Konfiguration falsch!  
`Invalid adxl345 id (got 0 vs e5)`

## Auswertung

[https://www.klipper3d.org/Measuring\\_Resonances.html#installation-instructions](https://www.klipper3d.org/Measuring_Resonances.html#installation-instructions)

## Einbauposition

Grundsätzlich sollte der ADXL345 so nah wie möglich an der Nozzle platziert werden. Am besten direkt an der Nozzle montiert ...

## Unterschiede

- **SB2209**
- **Fysetc**
- **DIY Nozzel Sensor**

## Links

- Klipper Dokumentation  
[https://www.klipper3d.org/Measuring\\_Resonances.html](https://www.klipper3d.org/Measuring_Resonances.html)
- Klipper with ADXL345 + Raspberry Pi clone (RP2040 zero)  
[https://travis90x.altervista.org/klipper-adxl345-raspberry-pi-rp2040-zero/?doing\\_wp\\_cron=1708409012.1427578926086425781250](https://travis90x.altervista.org/klipper-adxl345-raspberry-pi-rp2040-zero/?doing_wp_cron=1708409012.1427578926086425781250)
- Connect your ADXL345 to your host Raspberry Pi via a RP2040-Zero  
<https://github.com/bassamanator/rp2040-zero-adxl345-klipper>
- ADXL Nozzle Mount  
<https://www.teamfdm.com/files/file/720-adxl345-nozzle-mount/>
- Fysetc Input Shaper  
<https://github.com/FYSETC/FYSETC-PortableInputShaper/tree/main>
- Discourse  
<https://klipper.discourse.group/t/raspberry-pi-pico-adxl345-portable-resonance-measurement/1757>
- Beschleunigungssensor ADXL345 hinzufügen  
<https://de.ifixit.com/Anleitung/Beschleunigungssensor+ADXL345+hinzuf%C3%BCgen/147745>
- Vergleich der Position  
<https://www.youtube.com/watch?v=ufmUmeDnr0Y>

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

Permanent link:  
[https://www.drklipper.de/doku.php?id=klipper\\_faq:rp2040\\_input\\_shaper\\_hardware&rev=1708626743](https://www.drklipper.de/doku.php?id=klipper_faq:rp2040_input_shaper_hardware&rev=1708626743)

Last update: **2024/02/22 19:32**

