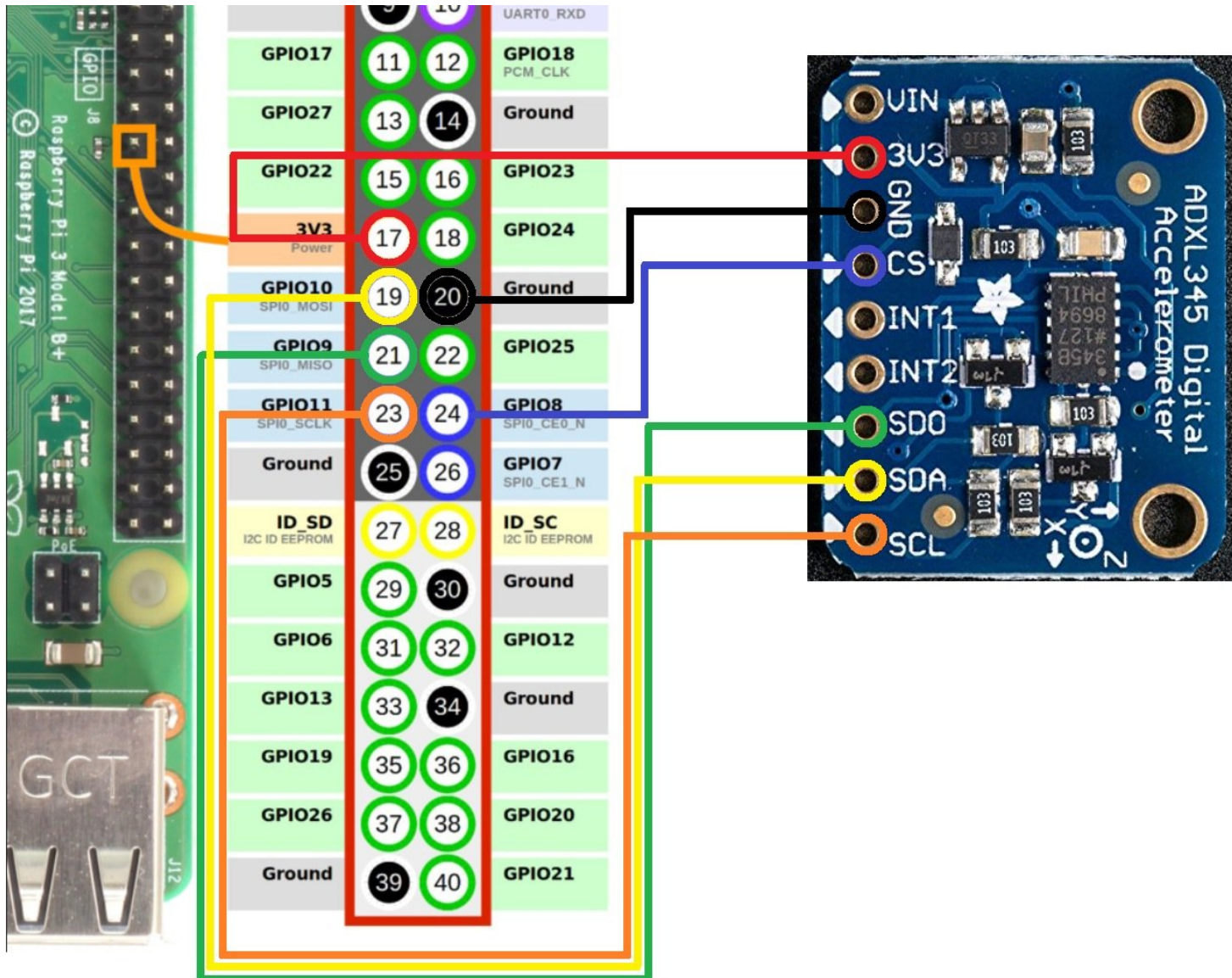




Adding ADXL345 Accelerometer

Written By: David Husolo



INTRODUCTION

You will need a printed adapter to mount the ADXL345 to the rear of the X carriage. You can find one [here](#)

https://www.klipper3d.org/Measuring_Reso...

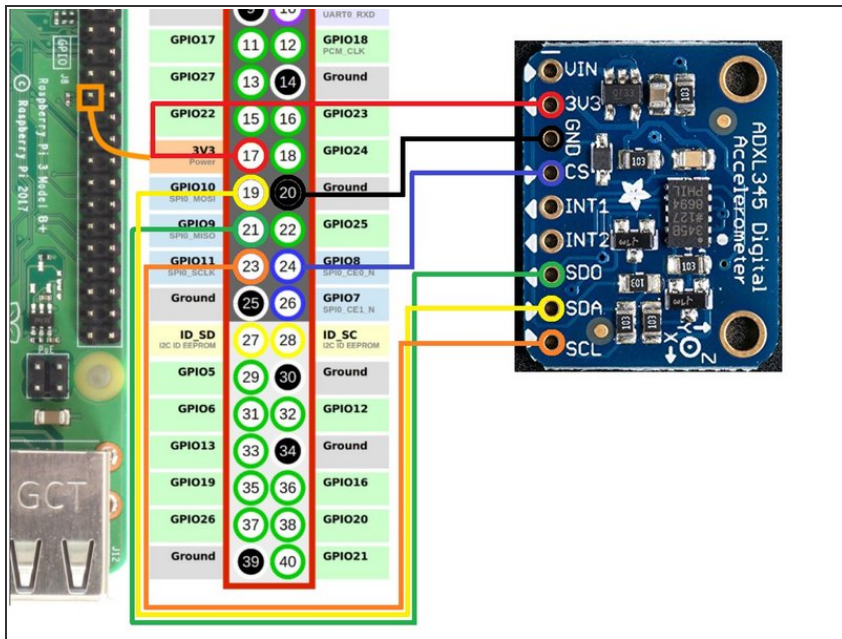


PARTS:

- [ADXL345](#) (1)
- [ADXL345 Printed Mount](#) (1)
- [M3x8 countersunk screw](#) (3)
- [M2.5x6 socket head screw](#) (2)
- [M2.5*3*3.5 Heat Set Inserts](#) (2)

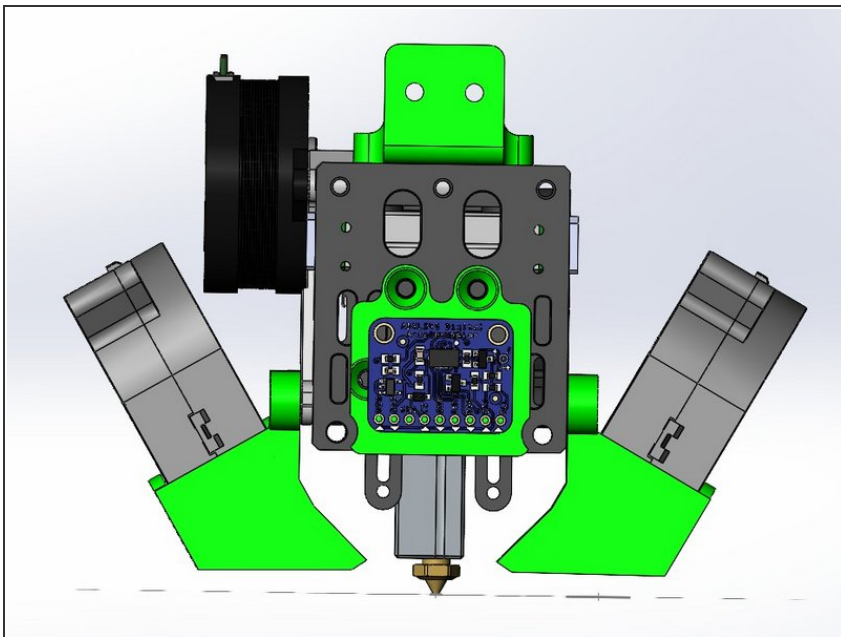
Optional

Step 1 — Wiring



- Connect the Accelerometer to the RPi
 - Accelerometer 3V3 to RPi pin 17 3.3v
 - Accelerometer GND to RPi pin 20 Ground
 - Accelerometer CS to RPi pin 24 GPIO08 (SPI0_CE0_N)
 - Accelerometer SDO 21 to RPi pin GPIO09 (SPI0_MISO)
 - Accelerometer SDA to RPi pin 19 GPIO10 (SPI0_MOSI)
 - Accelerometer SCL 23 to RPi pin GPIO11 (SPI0_SCLK)

Step 2 — Mounting the ADXL345



- There are a number of options you can use to mount the accelerometer. I used this [mount](#)

Step 3 — Installing The Packages



The SPI interface is enabled

- You need to SSH into to your printer and install Numby.
 - `~/klippy-env/bin/pip install -v numpy`
- ❗ Be patient, this package can take 10-20 minutes to install. When finished run:
 - `sudo apt update`
 - `sudo apt install python3-numpy python3-matplotlib -y`
- Verify Linux SPI driver is enabled
 - `sudo raspi-config`
 - #3 Interface Options > P4 SPI > Yes > OK > Finish

Step 4 — Preparing printer.cfg

```

271
272
273
274
275 #####
276 #   Input shaper & resonance_tester]
277 #####
278
279 [mcu rpi]
280 serial: /tmp/klipper_host_mcu
281
282 [adxl345]
283 cs_pin: rpi:None
284
285 [resonance_tester]
286 accel_chip: adxl345
287 probe_points:
288     179,115,20
289
290
291
292
293
294

```

- Uncomment the input shaper & resonance tester section in the config.
 - `probe_points`: It's recommended to use 1 point in the center of the bed slightly above. I used the same coordinates used when homing Z
- Restart klipper with
 - RESTART

Step 5 — Prechecks



- Check to make sure the RPI can communicate with the ADXL345 by running the following command in console
 - `ACCELEROMETER_QUERY`
- It should return something like:
 - `adxl345 values (x, y, z): -152.983740, 10249.910580, 152.983740`
- ❗ If you get an error like Invalid `adxl345` id (got `xx` vs `e5`), where `xx` is some other ID, it is indicative of the connection problem with ADXL345, or the faulty sensor. Double-check the power, the wiring (that it matches the schematics, no wire is broken or loose, etc.), and soldering quality.

Step 6 — Configuring Resonance



- Home the printer.
- During resonance testing the vibrations can become extremely violent. Make sure you're in proximity of the printer in case you need to cancel the test with M112
- Now you can run the first resonance test for X
 - `TEST_RESONANCES AXIS=X`
- When the test completes run it again for Y
 - `TEST_RESONANCES AXIS=Y`
- When the test is complete it will generate CSV files stored on the pi.

Step 7 — Interpreting Resonance Results

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Feb  5 19:23:42 2022 from 192.168.1.102

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@BLVfsetc:~$ sudo raspi-config
[sudo] password for pi:
Sorry, try again.
[sudo] password for pi:
pi@BLVfsetc:~$ ~/klipper/scripts/calibrate_shaper.py /tmp/resonances_x*.csv -o /tmp/shaper_calibrate_x.png
Fitted shaper 'zv' frequency = 68.2 Hz (vibrations = 4.1%, smoothing == 0.040)
To avoid too much smoothing with 'zv', suggested max_accel <= 18100 mm/sec^2
Fitted shaper 'mzv' frequency = 67.4 Hz (vibrations = 0.3%, smoothing == 0.046)
To avoid too much smoothing with 'mzv', suggested max_accel <= 13400 mm/sec^2
Fitted shaper 'ei' frequency = 80.0 Hz (vibrations = 0.3%, smoothing == 0.050)
To avoid too much smoothing with 'ei', suggested max_accel <= 11900 mm/sec^2
Fitted shaper '2hump_ei' frequency = 58.2 Hz (vibrations = 0.2%, smoothing == 0.159)
To avoid too much smoothing with '2hump_ei', suggested max_accel <= 3800 mm/sec^2
Fitted shaper '3hump_ei' frequency = 50.4 Hz (vibrations = 0.1%, smoothing == 0.323)
To avoid too much smoothing with '3hump_ei', suggested max_accel <= 1700 mm/sec^2
Recommended shaper is mzv @ 67.4 Hz
pi@BLVfsetc:~$ ~/klipper/scripts/calibrate_shaper.py /tmp/resonances_y*.csv -o /tmp/shaper_calibrate_y.png
Fitted shaper 'zv' frequency = 43.2 Hz (vibrations = 1.1%, smoothing == 0.088)
To avoid too much smoothing with 'zv', suggested max_accel <= 7300 mm/sec^2
Fitted shaper 'mzv' frequency = 43.6 Hz (vibrations = 0.0%, smoothing == 0.107)
To avoid too much smoothing with 'mzv', suggested max_accel <= 5600 mm/sec^2
Fitted shaper 'ei' frequency = 52.0 Hz (vibrations = 0.0%, smoothing == 0.119)
To avoid too much smoothing with 'ei', suggested max_accel <= 5000 mm/sec^2
Fitted shaper '2hump_ei' frequency = 64.6 Hz (vibrations = 0.0%, smoothing == 0.129)
To avoid too much smoothing with '2hump_ei', suggested max_accel <= 4600 mm/sec^2
Fitted shaper '3hump_ei' frequency = 77.4 Hz (vibrations = 0.0%, smoothing == 0.137)
To avoid too much smoothing with '3hump_ei', suggested max_accel <= 4400 mm/sec^2
Recommended shaper is zv @ 43.2 Hz
pi@BLVfsetc:~$
```

```
#####
# Input shaper & resonance_tester]
#####

[mcu rpi]
serial: /tmp/klipper_host_mcu

[adxl345]
cs_pin: rpi:None

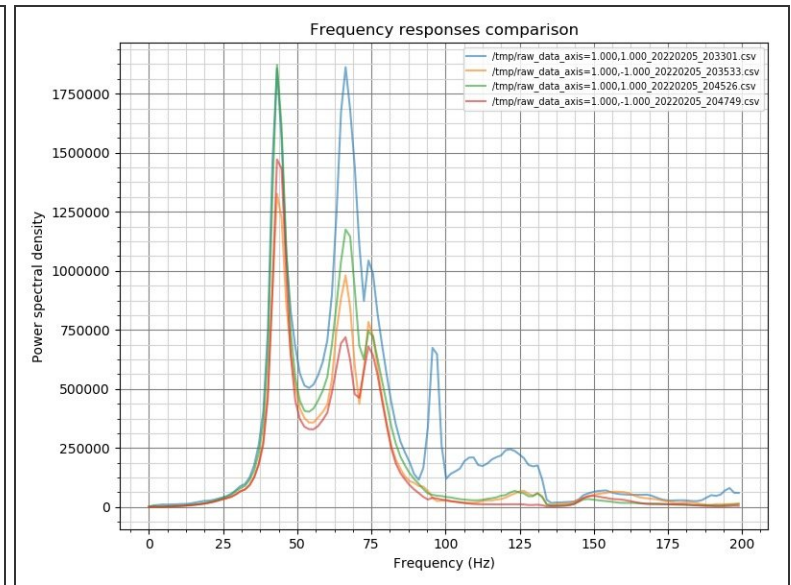
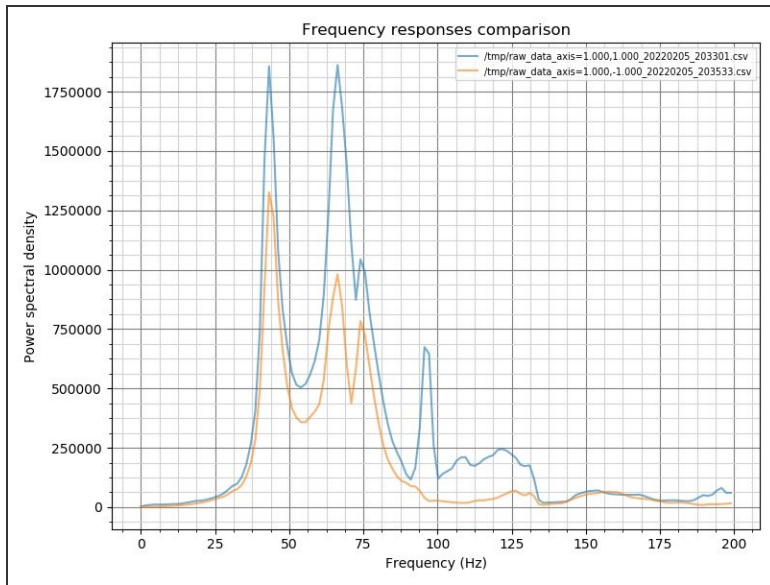
[resonance_tester]
accel_chip: adxl345
probe_points:
    179,115,20

[input_shaper]
shaper_type_x = mzv
shaper_freq_x = 67.4
shaper_type_y = zv
shaper_freq_y = 43.2

#####
# Fan Control
```

- SSH back in to the printer and run the following 2 commands
 - `~/klipper/scripts/calibrate_shaper.py /tmp/resonances_x*.csv -o /tmp/shaper_calibrate_x.png`
 - `~/klipper/scripts/calibrate_shaper.py /tmp/resonances_y*.csv -o /tmp/shaper_calibrate_y.png`
- This script will generate the charts `/tmp/shaper_calibrate_x.png` and `/tmp/shaper_calibrate_y.png` with frequency responses
- ① You can use WINSOCP to copy them to your PC if you like but it's not necessary.
- You will also get the suggested frequencies for each input shaper, as well as which input shaper is recommended for your setup.
 - You can see my values for X
 - You can see my values for Y

Step 8 — Comparing Belt Tension



- With a CoreXY you can also use Klipper to compare belt tension between X and Y. This doesn't tell you if the belts are too tight or too loose. It tells you have the belts are tensioned equally.
- Run the following commands in the console
 - TEST_RESONANCES AXIS=1,1 OUTPUT=raw_data
 - TEST_RESONANCES AXIS=1,-1 OUTPUT=raw_data
- SSH into the pi and run
 - `~/klipper/scripts/graph_accelerometer.py -c /tmp/raw_data_axis*.csv -o /tmp/resonances.png`
 - Based off the generated graph, X had much higher frequencies than Y. I had to tighten the left belt tensioner to to bring the frequencies closer to each other.
- As you can see from the 2nd test the frequencies for X and Y are closer together, but could still use a little more adjustment. I think a lot of the frequency variation is due to my cable chain. At lower frequencies it rattles considerably more only when calibrating X.