

MK3-Klipper Input_Shaping ([Resonance Compensation](#))

Klipper ist zur Ausführung von „Inputshaping“ mit einem Bewegungssensors des Typ's ADXL345 vorkonfiguriert. Um bei höheren Druckgeschwindigkeiten ausreichend Filamentfluss zur Verfügung stellen zu können, wurde eine E3D V6 Volcano Lösung mit einer 0.4mm Düse umgesetzt bzw. verbaut. Die Hiwin Linearschienen geben mehr Stabilität als die 8mm Rundwellen, weshalb diese an der X- und Y-Achse verbaut sind. Grundaufbau ist ein Prusa MK3.

Beim ADXL345 würde ich empfehlen, möglichst keinen Clon zu verwenden. Der originale Chip kostet einige Euros und funktioniert auf Anhieb, vorausgesetzt man führt folgende 5 Schritte korrekt aus.

Erster Schritt – Vorbereitung und Installation der Hardware, getestet mit einem Pi4 4GB

Anschluss Schema:

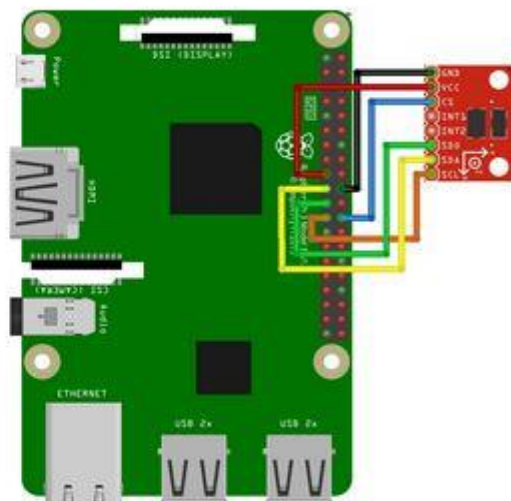
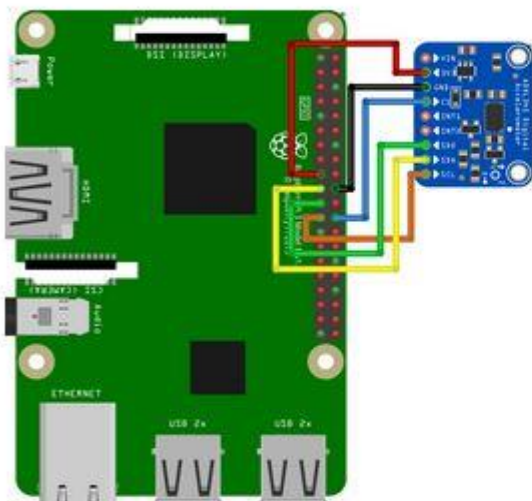
Der ADXL345 muss über SPI an den Raspberry Pi angeschlossen werden. Beachte, dass die I2C-Verbindung, die in der ADXL345-Dokumentation vorgeschlagen wird, einen zu geringen Durchsatz hat und nicht funktionieren wird.

ADXL345 pin	RPi pin	RPi pin name
3V3 (or VCC)	01	3.3v DC power
GND	06	Ground
CS	24	GPIO08 (SPI0_CE0_N)
SDO	21	GPIO09 (SPI0_MISO)
SDA	19	GPIO10 (SPI0_MOSI)
SCL	23	GPIO11 (SPI0_SCLK)

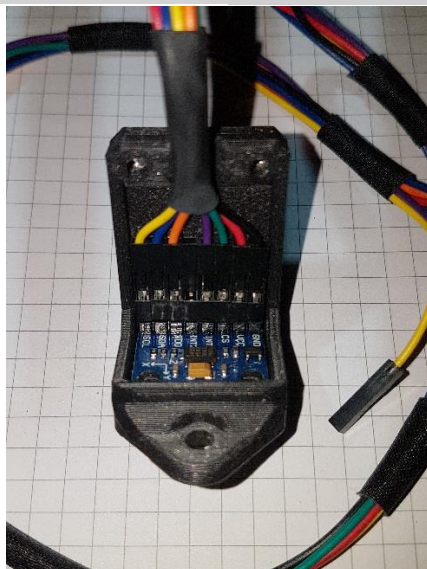
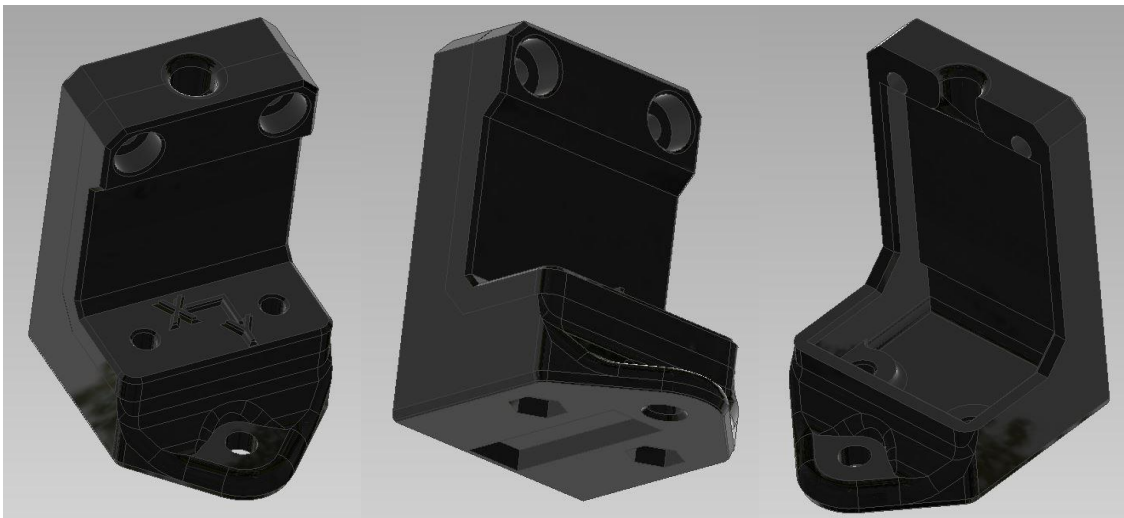
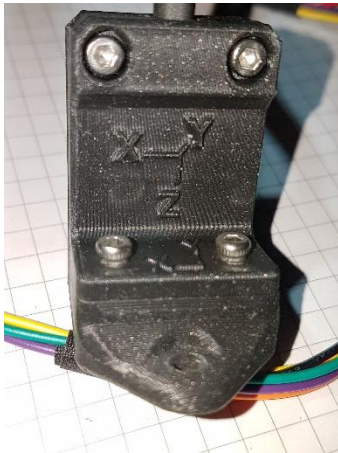
oder Pin17

oder Pin 34

Fritzing wiring diagrams for some of the ADXL345 boards:



Für die Verbindung Pi4 - ADXL345 habe ich 50cm 1-polige Dupont Kabel verwendet und für den Sensor ein passendes Gehäuse mit einem Magneten und einer Fixierbohrung zum Befestigen des Sensors gedruckt.



(INT wird zwar nicht benötigt, habe ich aber dennoch angeschlossen)

Zweiter Schritt – Installation der notwendigen Software und Einrichtung des Pi4:

Nachdem Klipper bereits installiert ist, sind folgende Kommandos in einem Pi4 Terminalfenster abzusetzen:

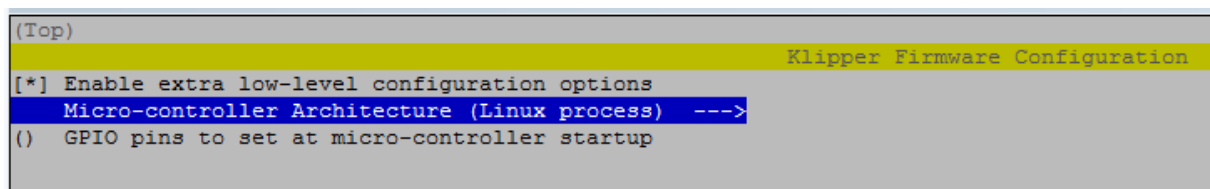
```
cd ~/klipper/  
sudo cp "./scripts/klipper-mcu-start.sh" /etc/init.d/klipper_mcu  
sudo update-rc.d klipper_mcu defaults
```

Das Pi4 wird hiermit als 2. MCU definiert damit die GPIO Pin's angesprochen werden können.

Nun muss die Firmware mit folgenden Kommandos angepasst und neu compiliert werden:

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

Hier beim Menüpunkt „Microcontroller Architecture“ auf „Linux process“ einstellen, speichern und mit folgenden Befehlen neu compilieren:



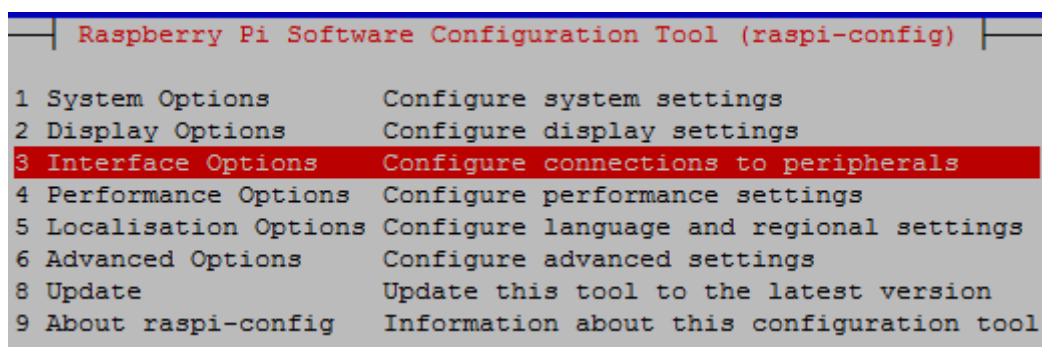
```
sudo service klipper stop  
make flash  
sudo service klipper start
```

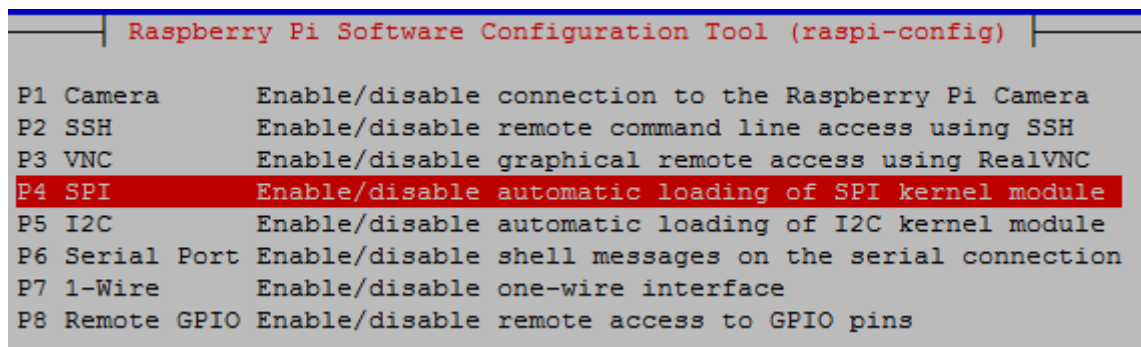
Falls es „permission denied“, also Probleme mit Zugriffsrechten beim Verbinden mit „/tmp/klipper_host_mcu“ geben sollte, kann der User Pi zur Gruppe „tty“ hinzugefügt werden:

```
sudo usermod -a -G tty pi
```

Zusätzlich müssen die SPI Treiber des Pi4 aktiviert werden mit:

```
sudo raspi-config
```





(SPI aktivieren mit yes)

Dritter Schritt: Klipper printer.cfg anpassen bzw. einrichten

Folgende Zeilen in die printer.cfg Konfigurationsdatei eintragen:

```
[mcu rpi]
serial: /tmp/klipper_host_mcu

[adxl345]
cs_pin: rpi:None

[resonance_tester]
accel_chip: adxl345
probe_points:
    110, 110, 50 # als Beispiel - hängt von der Druckergröße ab
```

Nachdem dieser Eintrag erledigt ist und das Pi4 per GPIO mit dem Bewegungssensor verbunden ist, muss Klipper neu gestartet werden bevor wir mit den Tests beginnen können.

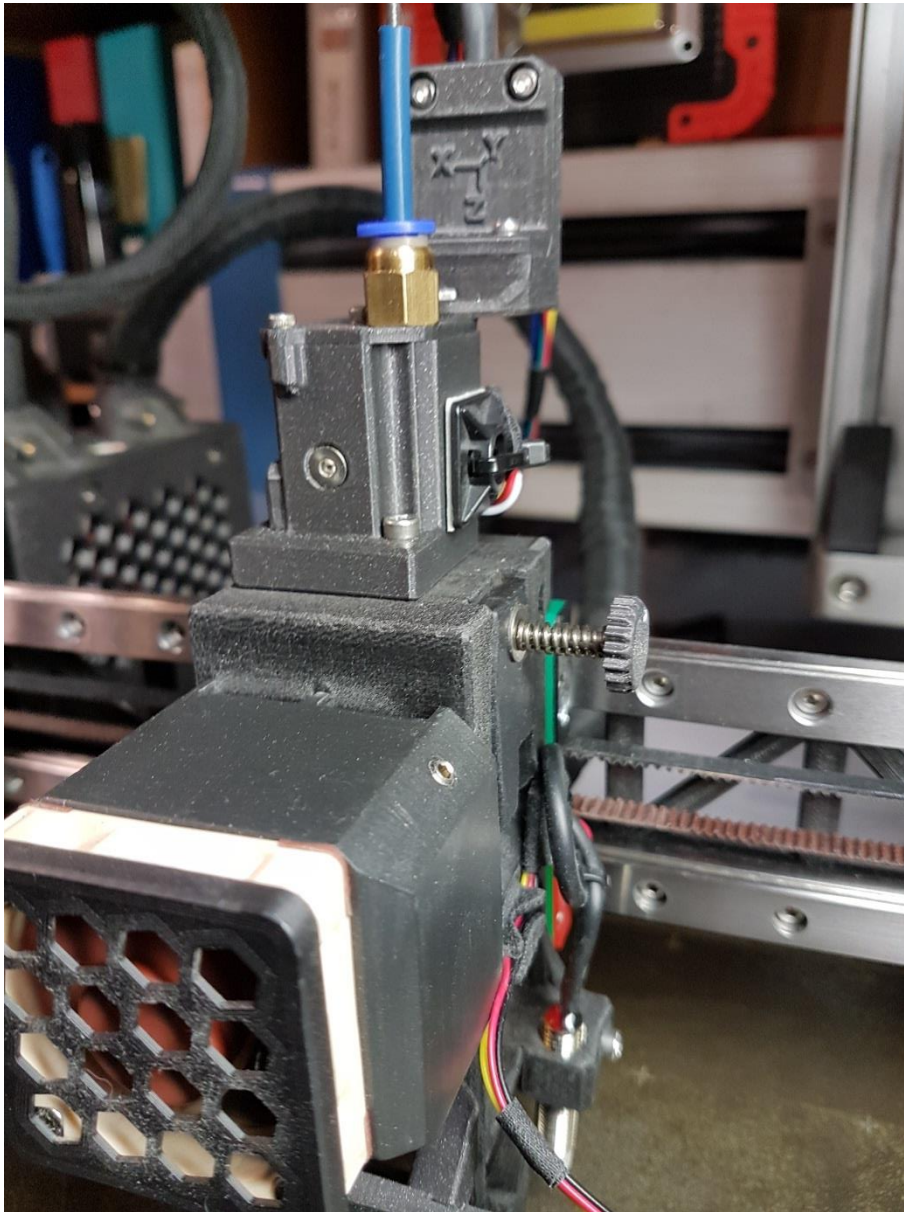
Vor dem ersten Resonanztest sollte das Setup überprüft werden.

Die Eingabe „ACCELEROMETER_QUERY“ ins Octoprint Terminal sollte in etwa solch ein Ergebnis liefern:

```
Send: M105
Recv: ok B:26.4 /0.0 T0:27.3 /0.0
Send: ACCELEROMETER_QUERY
Recv: // adxl345 values (x, y, z): 229.475610, 764.918700, 9867.451230
Recv: ok
Send: M105
```

Tut es das nicht, liegt höchstwahrscheinlich ein Verdrahtungsproblem oder ein nicht kompatibler Bewegungssensor vor.

Nun starten wir die erste Messung in X. Dazu habe ich den Sensor auf den Extruder geschraubt.



Man muss darauf achten, dass das Koordinatensystem des Sensors mit dem Koordinatensystem des Druckers übereinstimmt (X=X, Y=Y, Z=Z).

Nun kann der erste Test der X-Achse mit folgendem Kommando gestartet werden (Vorsicht, der Drucker scheint bei gewissen Frequenzen fast abzuheben! Beim Drucker bleiben und notfalls M112 aktivieren!)

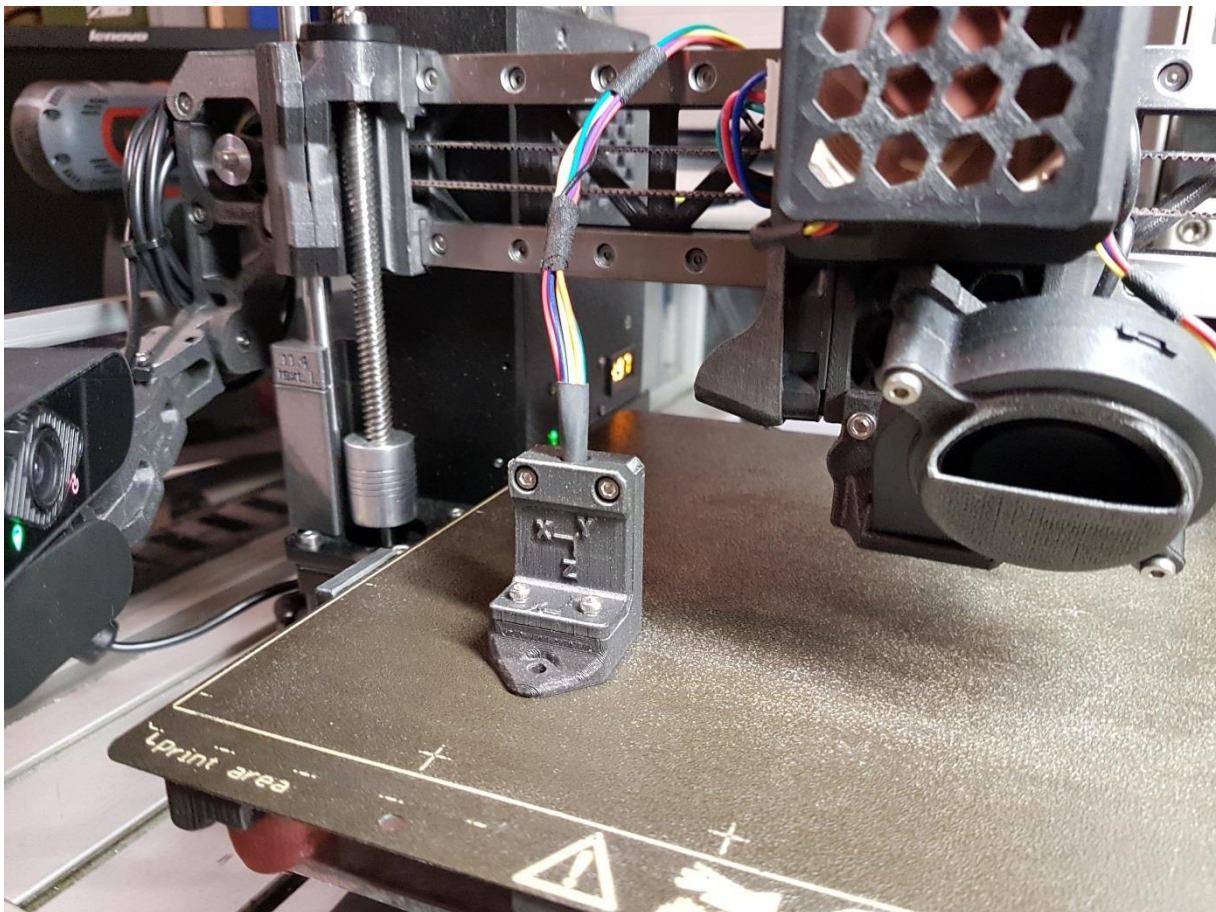
```
TEST_RESONANCES AXIS=X
```

Das dauert einige Minuten. Danach wird eine *.csv Datei für die Ergebnisse (Rohdaten) der X-Achse ins Pi /tmp/ Verzeichnis geschrieben.

Falls der Drucker zu arg vibriert, kann man die Frequenz (accel_per_hz) in der printer.cfg von 75 auf 50Hz zurücksetzen und erneut testen.

```
57 min_freq: 5
58 # Minimum frequency to test for resonances. The default is 5 Hz.
59 max_freq: 120
60 # Maximum frequency to test for resonances. The default is 120 Hz.
61 accel_per_hz: 75
62 # This parameter is used to determine which acceleration to use to
63 # test a specific frequency: accel = accel_per_hz * freq. Higher the
64 # value, the higher is the energy of the oscillations. Can be set to
65 # a lower than the default value if the resonances get too strong on
66 # the printer. However, lower values make measurements of
67 # high-frequency resonances less precise. The default value is 75
68 # (mm/sec).
```

Wenn dieser Schritt erfolgreich durchgelaufen ist, kann man zur Y-Achse weiterschreiten. Dazu habe ich den Sensor einfach auf die Druckplatte gestellt. Durch den starken Neodym Magneten steht er dort relativ sicher. Auch hier auf die Ausrichtung des Koordinatensystems achten.



Test starten mit:

```
TEST_RESONANCES AXIS=Y
```

Nach einigen Minuten wird eine weitere *.csv Datei für die Y-Achse nach /tmp/ geschrieben.

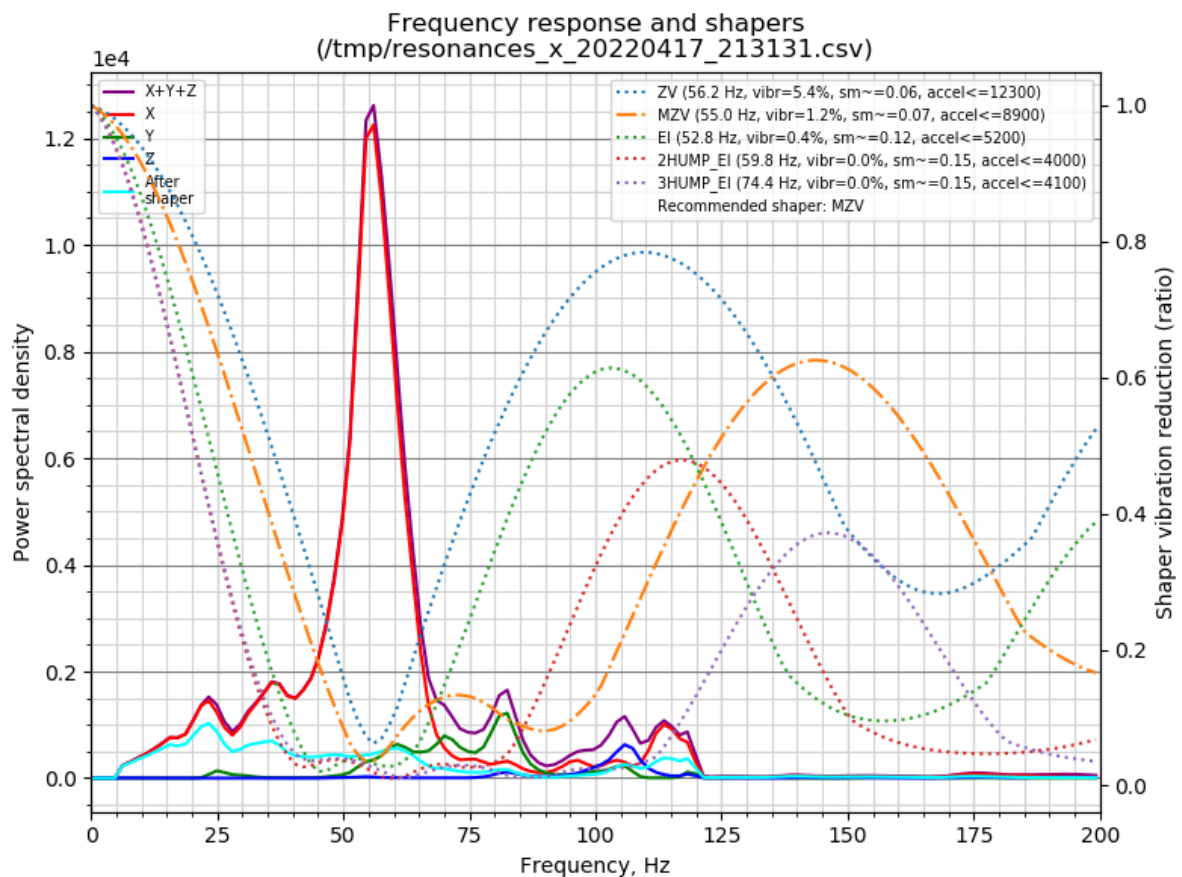
Um das Ergebnis der Tests zu visualisieren, müssen noch 2 Schritte zum Erzeugen der Bilder (*.png) durchgeführt werden.

Am Pi4 Terminal (z.B. Putty) folgende Kommandos absetzen:

```
~/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
```

Es werden 2 Bilder erzeugt die wie folgt aussehen:



(shaper_calibrate_x.png)

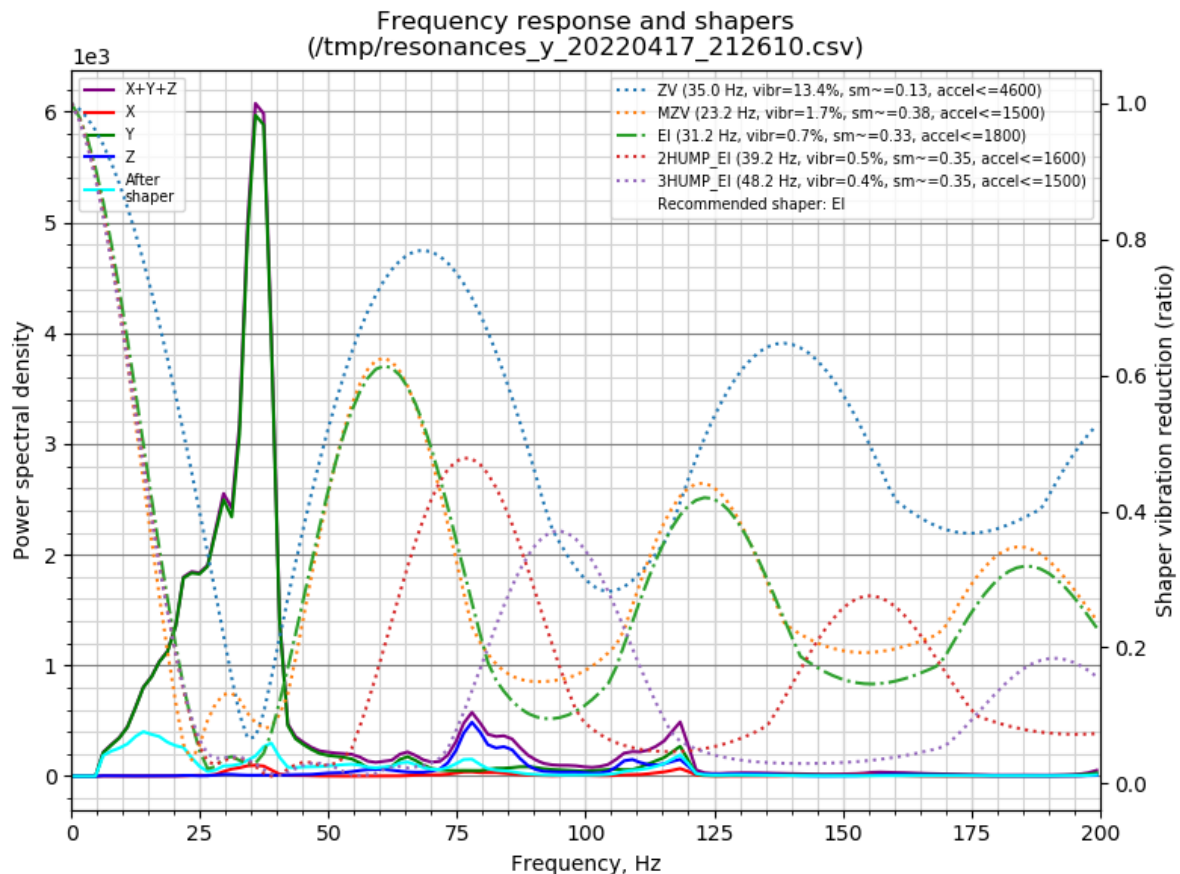
Wie ich dem Ergebnis entnehmen kann, trage ich folgende Werte in die printer.cfg im Sektor [input_shaper] ein:

```
75 #  
76 [input_shaper]  
77 shaper_freq_x: 55  
78 # A frequency (in Hz) of the input shaper for X axis. This is  
79 # usually a resonance frequency of X axis that the input shaper  
80 # should suppress. For more complex shapers, like 2- and 3-hump EI  
81 # input shapers, this parameter can be set from different  
82 # considerations. The default value is 0, which disables input  
83 # shaping for X axis.
```


und

```
95 shaper_type_x: mzv
96 #shaper_type_y:
97 #   If shaper_type is not set, these two parameters can be used to
98 #   configure different input shapers for X and Y axes. The same
99 #   values are supported as for shaper_type parameter.
```

Hier die Y-Achse:



(shaper_calibrate_y.png)

Natürlich werden auch diese Ergebnisse in die printer.cfg eingetragen:

```
84 shaper_freq_y: 31.2
85 #   A frequency (in Hz) of the input shaper for Y axis. This is
86 #   usually a resonance frequency of Y axis that the input shaper
87 #   should suppress. For more complex shapers, like 2- and 3-hump EI
88 #   input shapers, this parameter can be set from different
89 #   considerations. The default value is 0, which disables input
90 #   shaping for Y axis.
```

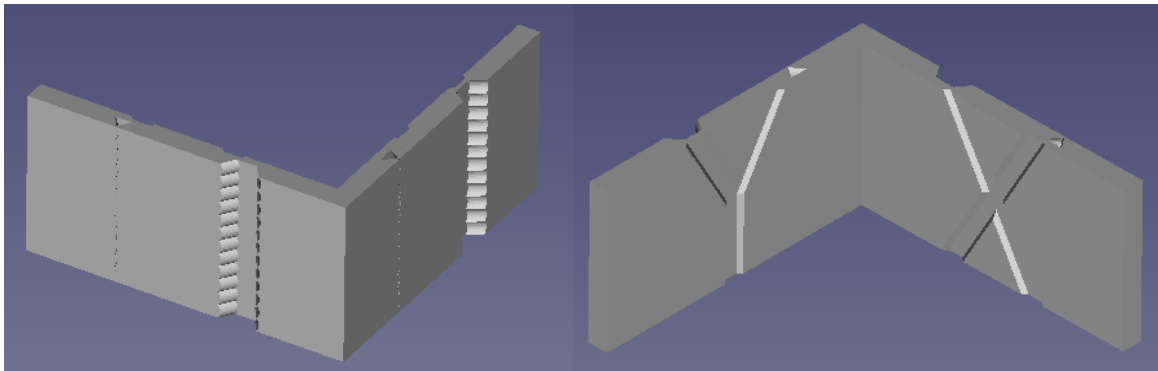
```
96 shaper_type_y: ei
97 #   If shaper_type is not set, these two parameters can be used to
98 #   configure different input shapers for X and Y axes. The same
99 #   values are supported as for shaper_type parameter.
```


Im vierten Schritt geht's ans Testen unter Realbedingungen. Ein Testteil wird gedruckt und die Unterschiede verglichen - mit und ohne Input Shaping.

Vierter Schritt - Wir drucken ein Testteil nach folgender Vorlage:

https://www.klipper3d.org/Resonance_Compensation.html

Als Testteil wurde der [ringing tower](#) verwendet,



und die Achsen im PrusaSlicer entsprechend den Markierungen ausgerichtet.

Folgende Änderungen wurden vor dem Druckstart noch durchgeführt:

Klipper (printer.cfg):

- Pressure Advance bei den Extrudereinstellungen auf 0 gesetzt
- square_corner_velocity auf 5.0 gesetzt
- max_accel_to_decel auf 7000 gesetzt
- Shaperfrequenzen in X und Y auf 0 gesetzt

PrusaSlicer:

- Layerhöhe auf 0.2 gesetzt
- Infill und Toplayer auf 0 gesetzt
- 1 Perimeter vertikal, 2 Perimeter horizontal
- Speed Außenkontur = 90mm/s

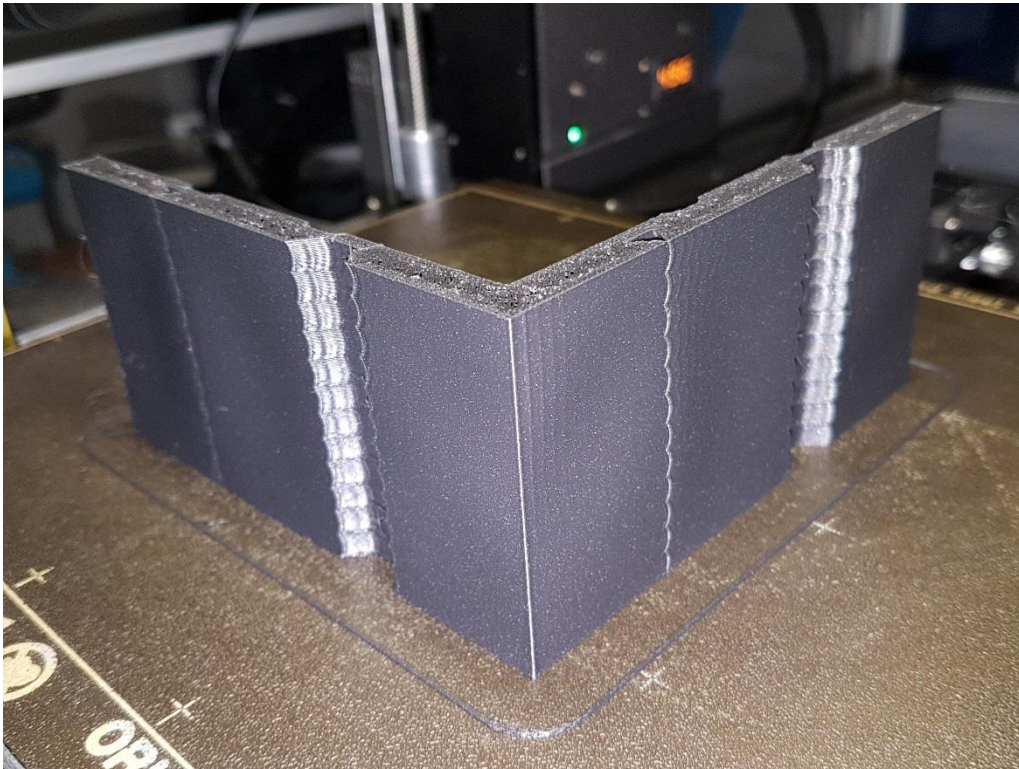
Gedruckt wurde mit Prusa ASA black. Man kann da natürlich andere Materialien auch verwenden, vorzugsweise in grau, da sich da die Struktur gut ablichten lässt.

Bevor der Tower gedruckt wird muss noch folgendes Kommando im Octoprint Terminal an den Drucker gesendet werden:

```
TUNING_TOWER    COMMAND=SET_VELOCITY_LIMIT    PARAMETER=ACCEL    START=1500  
STEP_DELTA=500 STEP_HEIGHT=5
```

Dadurch erhöht Klipper die Beschleunigung um 500mm/s^2 , startend bei 1500mm/s^2 , was bedeutet, dass mit 1500mm/s^2 gestartet und in der letzten Schicht die Beschleunigung auf 7000mm/s^2 hochgesetzt wird.

Danach wird der Druck gestartet (Laufzeit ca. 1 Stunde) und das Ergebnis sieht dann so aus:

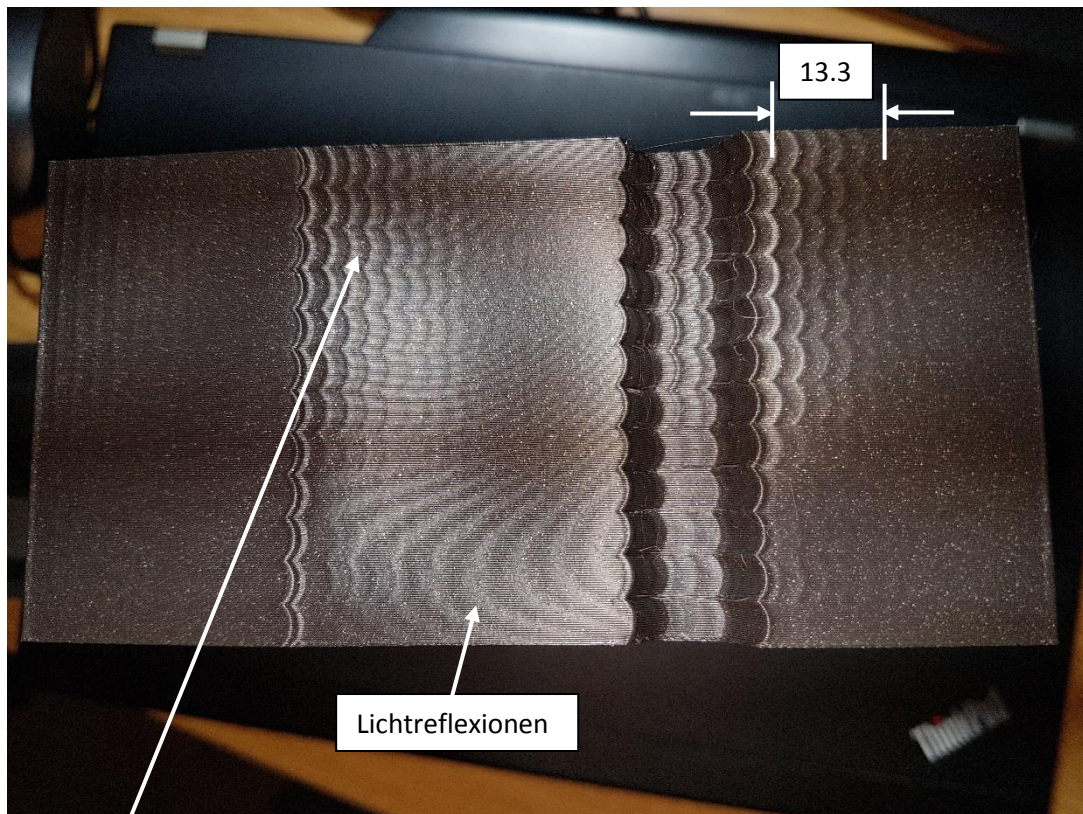


X-Achse:



(sieht noch recht gut aus. Die Y-Achse ist schon bescheidener in Punkto Qualität)

Y-Achse:



Die Wellenbildung ist wesentlich stärker ausgebildet, als bei der X-Achse, was ziemlich sicher daran liegt, dass mehr Masse bewegt werden muss oder die Gurtspannung nicht passt, was aber in meinem Fall ausgeschlossen werden konnte, da ich die Spannung beider Achsen mit meinem Tool überprüft und eingestellt habe.

Die langgezogenen Wellen sind lediglich Lichtreflexionen, welche beim Ablichten mit meiner bescheidenen Belichtung und Kamera entstanden sind.

Nun geht es an die praktische Auswertung:

Gemessen habe ich im oberen (High-Acc.) Bereich den Abstand zwischen 5 Wellentälern, wobei ich auf meinem Digitalmessschieber ein Wert von ca. 13.3mm ablesen konnten.

Nun rechnen wir die Frequenz nach folgender Formel aus:

$$V \cdot N / D \text{ (Hz)} = \text{speed} * \text{Wellentäler} / \text{Abstand} =$$

$$90 * 5 / 13.3 = 33.83\text{Hz}$$

Dieses Ergebnis stimmt in etwa mit den vorher theoretisch ermittelten Ergebnissen mit dem Beschleunigungsmesser von 31.2 Hz überein, weshalb ich einen Wert von 32.5 Hz für die Y-Achse in die printer.cfg eingetragen habe.

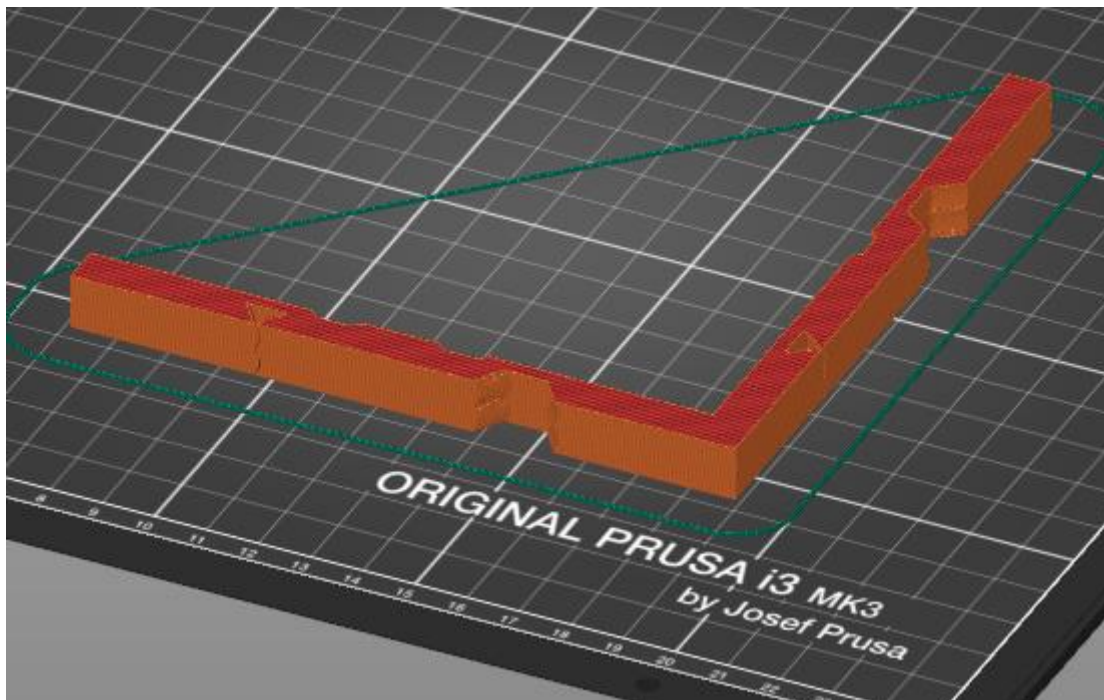
Derselbe Mess- und Rechenvorgang zur X-Achse. Hier habe ich wieder zwischen 5 Wellentälern gemessen und einen Wert von 7.7mm ermittelt.

Erneute Berechnung ergibt: $90 * 5 / 7.7 = 58.44 \text{ Hz}$, was ca. ungefähr dem theoretisch ermittelten Wert von 55Hz nahekommt. Hier habe ich in die printer.cfg einen Wert von 56.7 für die X-Achse eingetragen.

Im letzten Schritt erfolgt ein Test mit den eingetragenen Werten und eine Feinjustierung.

Schritt 5 – Optimierung, Feinjustierung

Bei der Feinoptimierung habe ich den Tower nur 10mm hoch gedruckt. Alle ermittelten Werte wurden in die printer.cfg eingetragen und der erneute Druck erfolgte mit denselben Slicereinstellungen bei Beschleunigungswerten von 7000mm/s^2 bzw. 3500mm/s^2 .



```
[printer]
kinematics: cartesian
max_velocity: 320 # 500 is max. limit for orig. MK3
max_accel: 7000
max_accel_to_decel: 3500 #max_accel/2
max_z_velocity: 12
max_z_accel: 200
square_corner_velocity: 5.0
```



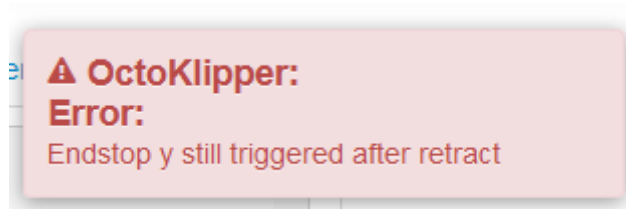
```

[input_shaper]
shaper_freq_x: 56.7
# A frequency (in Hz) of the input shaper for X axis. This is
# usually a resonance frequency of X axis that the input shaper
# should suppress. For more complex shapers, like 2- and 3-hump EI
# input shapers, this parameter can be set from different
# considerations. The default value is 0, which disables input
# shaping for X axis.
shaper_freq_y: 32.5
# A frequency (in Hz) of the input shaper for Y axis. This is
# usually a resonance frequency of Y axis that the input shaper
# should suppress. For more complex shapers, like 2- and 3-hump EI
# input shapers, this parameter can be set from different
# considerations. The default value is 0, which disables input
# shaping for Y axis.
#shaper_type:
# A type of the input shaper to use for both X and Y axes. Supported
# shapers are zv, mzv, zvd, ei, 2hump_ei, and 3hump_ei. The default
# is mzv input shaper.
shaper_type_x: mzv
shaper_type_y: ei
# If shaper_type is not set, these two parameters can be used to
# configure different input shapers for X and Y axes. The same
# values are supported as for shaper_type parameter.
damping_ratio_x: 0.1
damping_ratio_y: 0.1
# Damping ratios of vibrations of X and Y axes used by input shapers
# to improve vibration suppression. Default value is 0.1 which is a
# good all-round value for most printers. In most circumstances this
# parameter requires no tuning and should not be changed.
#

```

Wie ich feststellen musste, hat die Y-Achse nach dem Eintrag der 32.5 im input_shaper Register Probleme beim Referenzieren der Y-Achse gemacht.

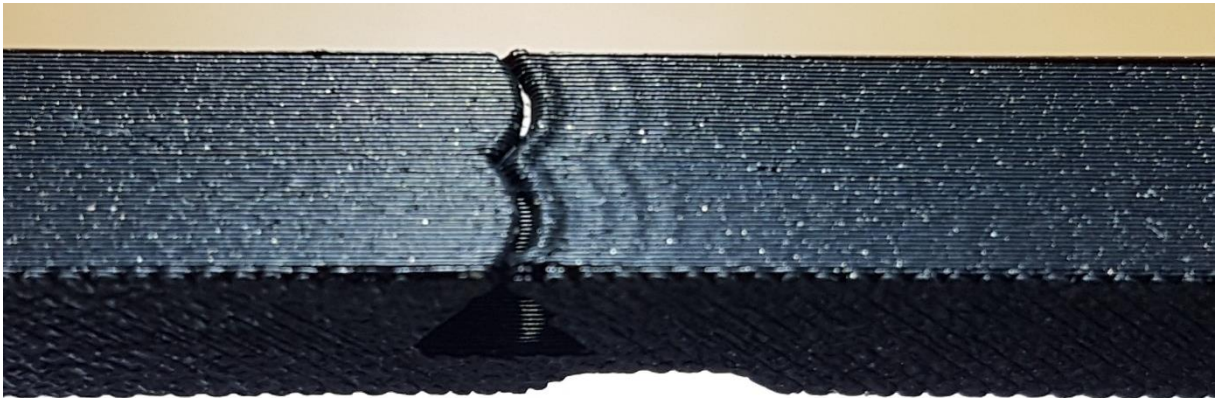
G28Y



Erhöhung der Homing Geschwindigkeit von 50 auf 100mm/s haben den Fehler temporär eliminiert. Alternativ kann man auch am Parameter „sense_resistor“ etwas nachjustieren. Ich habe in meinem Fall den Wert etwas reduziert.

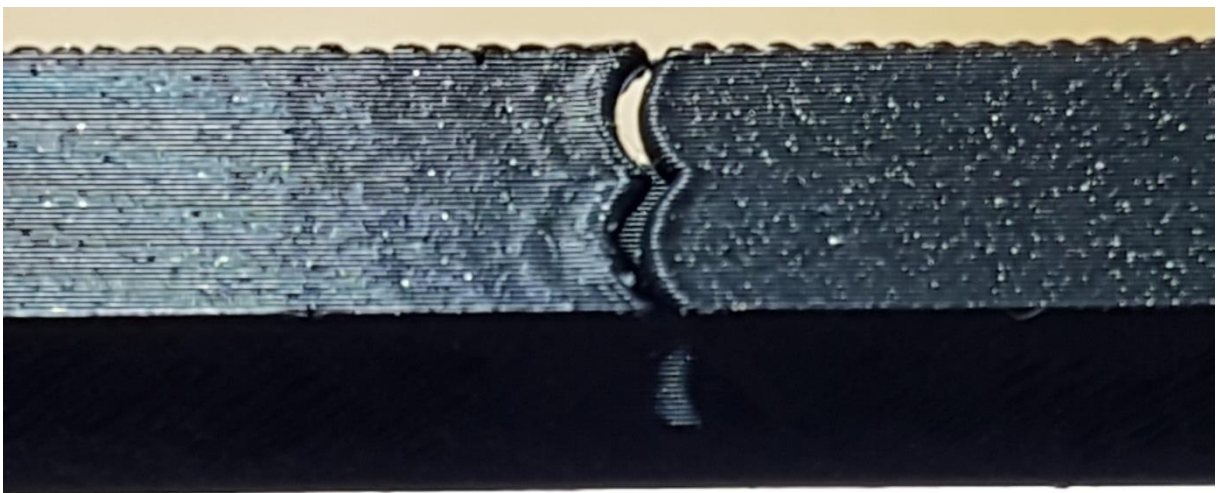
Nachdem die kleine Hürde beseitigt war – ein erneuter Druck mit den ermittelten und eingetragenen I-shaping Werten.

Das Ergebnis zur X-Achse mit denselben Slicerparametern, mit Beschleunigung (7000mm/s²), Speed 90mm/s und einer Bauteilhöhe von 10mm. Diese Werte entsprechen den Werten der obersten Schicht mit max. Werten des vorher gedruckten Towers.



Wie man sieht, Besserung aber noch nicht optimal

...und die Y-Achse:



Auch hier ist noch Optimierungspotential vorhanden.

Allerdings muss ich dazu erwähnen, dass ich mit 7000mm/s² die empfohlenen max. Einstellungen von 1800mm/s² für die Y-Achse weit überschritten habe und dort noch anpassen muss.

— EI (31.2 Hz, vibr=0.7%, sm~=0.33, accel<=1800)

Die Beschleunigung der X-Achse würde soweit passen, werde ich aber auch reduzieren, d.h., ein erneuter Testdruck mit reduzierten Beschleunigungswerten und erhöhtem Druckspeed bis 300mm/s.

— MZV (55.0 Hz, vibr=1.2%, sm~=0.07, accel<=8900)

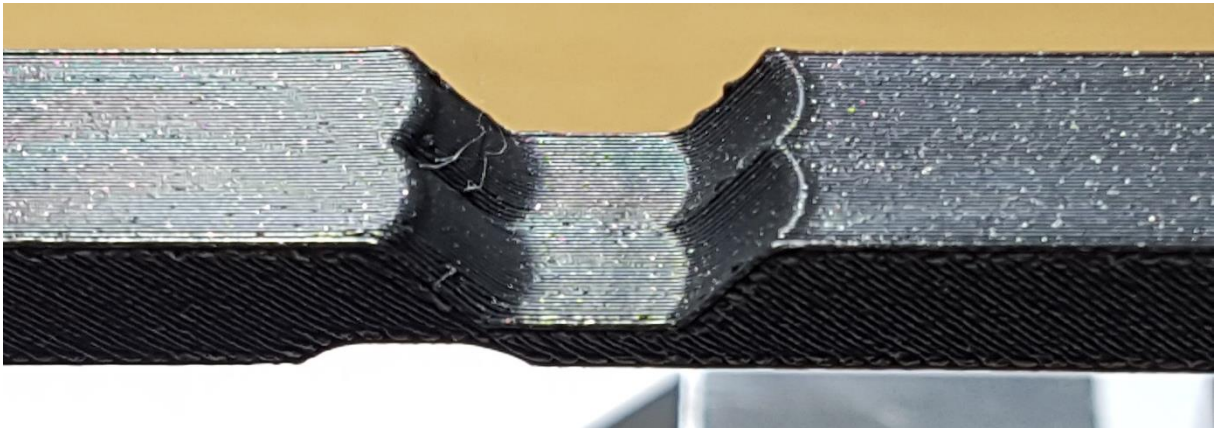
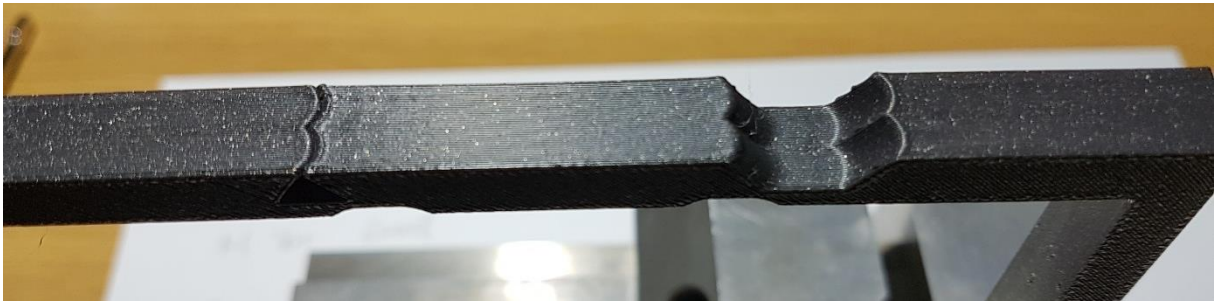
Nachfolgend das Ergebnis des 2. Versuches nach weiteren Änderungen der Beschleunigung und Geschwindigkeitswerte. Pressure Advance steht immer noch auf 0.

```
[printer]
kinematics: cartesian
max_velocity: 320 # 400-500 is max. limit for orig. MK3
max_accel: 3600
max_accel_to_decel: 1800 #max_accel/2
max_z_velocity: 12
max_z_accel: 200
square_corner_velocity: 5.0
```

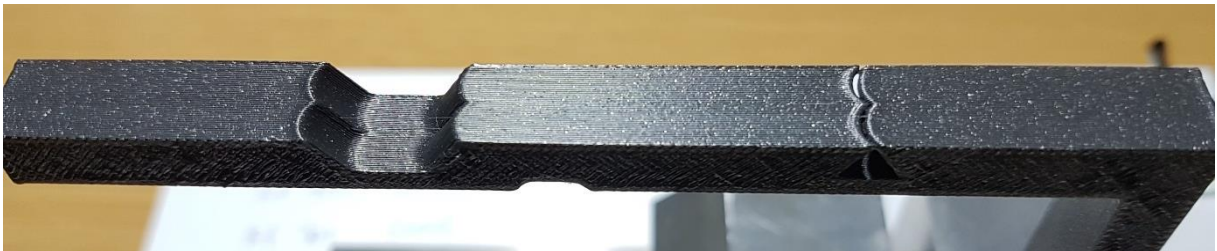
Geschwindigkeiten:

Geschwindigkeit für Druckbewegungen		Geschwindigkeit für Bewegungen zwischen den Druckvorgängen	
● Konturen:	<input type="text" value="300"/> mm/s	● Eilgang:	<input type="text" value="320"/> mm/s
● Dünne Außenkonturen:	<input type="text" value="100"/> mm/s oder %		
● Außenkonturen:	<input type="text" value="225"/> mm/s oder %		
● Infill:	<input type="text" value="300"/> mm/s		
● Massives Infill:	<input type="text" value="150"/> mm/s oder %		
● Oberes massives Infill:	<input type="text" value="100"/> mm/s oder %		
● Stützmaterial:	<input type="text" value="300"/> mm/s		
● Schnittstellenmaterial zum Stützmaterial:	<input type="text" value="100%"/> mm/s oder %		
● Überbrückungen:	<input type="text" value="300"/> mm/s		
● Lückenfüllung:	<input type="text" value="150"/> mm/s		
● Bügeln:	<input type="text" value="15"/> mm/s		
		Veränderer	
		● Druckgeschwindigkeit der ersten Schicht:	<input type="text" value="100"/> mm/s oder %
		Beschleunigungskontrolle (fortgeschritten)	
● Konturen:	<input type="text" value="0"/> mm/s ²		
● Infill:	<input type="text" value="0"/> mm/s ²		
● Überbrückung:	<input type="text" value="0"/> mm/s ²		
● Erste Schicht:	<input type="text" value="0"/> mm/s ²		
● Standard:	<input type="text" value="0"/> mm/s ²		
		Automatische Geschwindigkeit (fortgeschritten)	
● Maximale Druckgeschwindigkeit:	<input type="text" value="300"/> mm/s		
● Maximale Volumengeschwindigkeit:	<input type="text" value="0"/> mm ³ /s		

Ergebnis X-Achse:



Ergebnis Y-Achse:



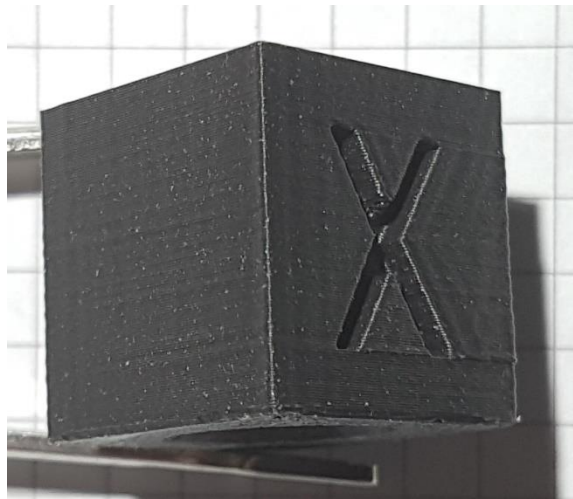
Auf den Bildern ist im Bereich der Ecken zu erkennen, dass der Parameter „max_accel_to_decel: 1800“ evtl. noch zu optimieren ist, da sich die Ecken leicht aufwölben. Im nächsten Versuch habe ich PA hinzugefügt, aber vorher den Testwürfel gedruckt, um nach der Änderung der Beschleunigungs- und Geschwindigkeitswerte inkl. aktivem InputShaping einen optimalen Wert zu ermitteln.

Vor dem Start des Druckes wurde folgendes Kommando an das Octoprint Terminal übergeben:

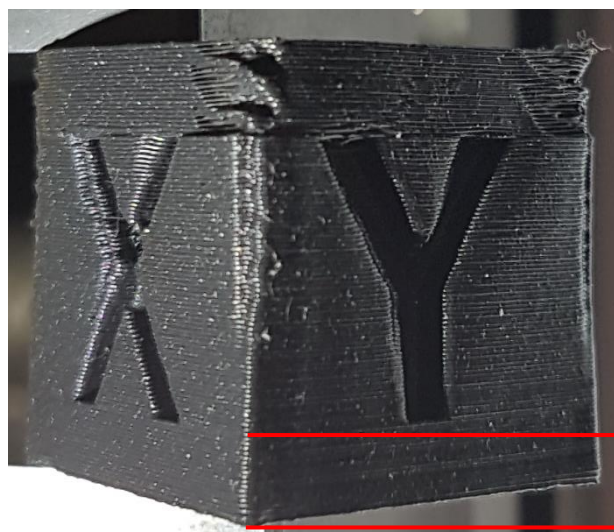
```
TUNING_TOWER COMMAND=SET_PRESSURE_ADVANCE PARAMETER=ADVANCE START=0  
FACTOR=.005
```

Damit wird beim Druck der PA Factor (bei 0 beginnend) per Layer um 0.005 erhöht.

Vor der Aktivierung von InputShaping bei einem optimal ermittelten PA Wert von 0.04 sah das Ergebnis so aus:



Das Ergebnis bei aktiviertem InputShaping und PA 0.040 jetzt:



4.5mm -> PA = 0.0225

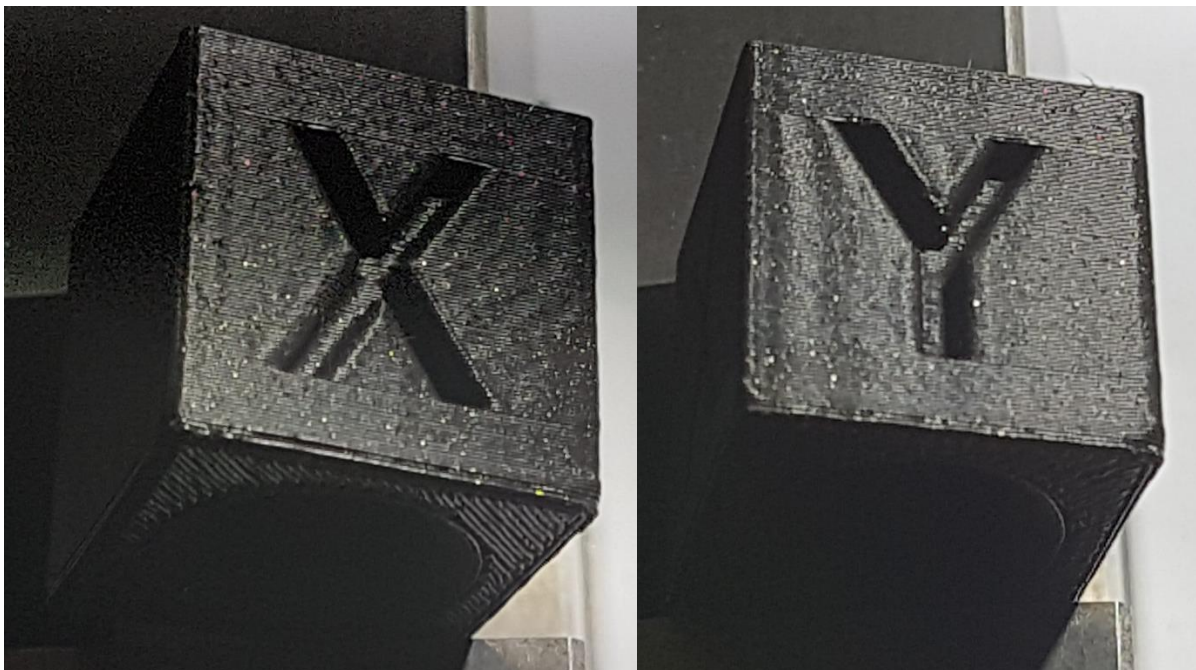
Das bedeutet also, dass der PA Wert reduziert und nach Aktivierung von InputShaping neu kalibriert werden muss.

Das PA Ergebnis errechnet sich wie folgt:

$$\text{pressure_advance} = \text{<start>} + \text{<measured_height>} * \text{<factor>}$$
$$\text{pressure_advance} = 0 + 4.50 * 0.005 = \mathbf{0.0225} \quad (\text{siehe vorheriges Bild})$$

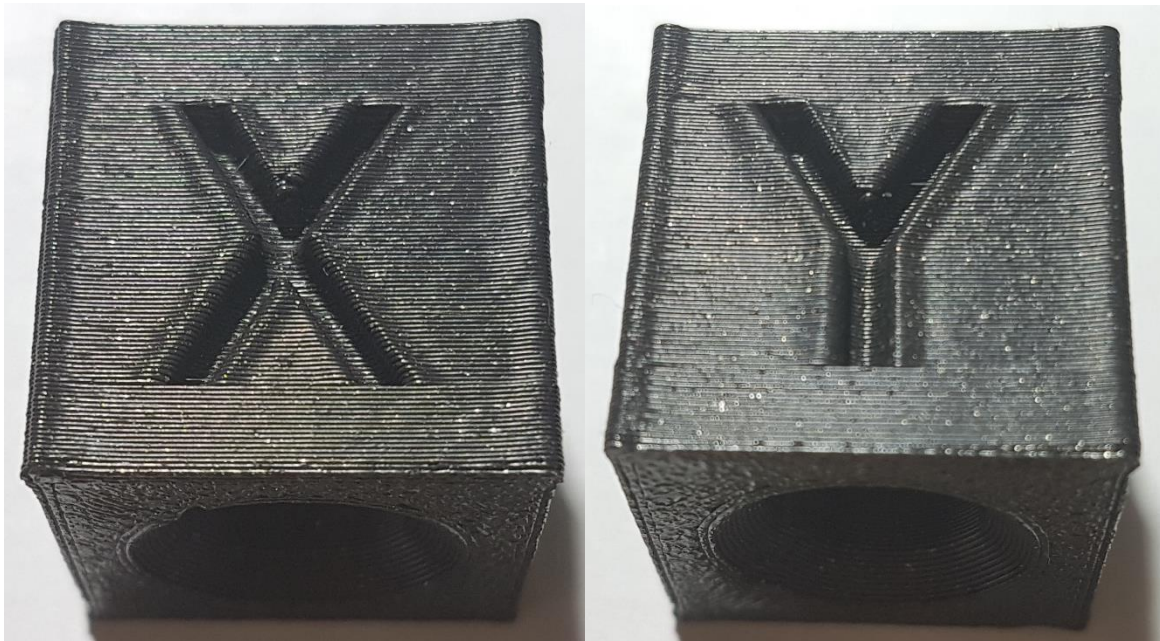
Das wird nun so in die printer.cfg eingetragen und wiederholt mit aktiviertem InputShaping und Pressure Advance gedruckt.

Das Ergebnis (Cal. Cube):



Sowohl an der X- als auch an der Y-Achse sind die Wülste an den Ecken zurückgegangen, dafür hat sich aber wieder ein leichtes Ghosting bzw. Ringing bemerkbar gemacht. Ob dafür PA verantwortlich ist, teste ich erneut, indem ich PA wieder auf 0 setze und mit demselben Gcode den Cube erneut drucke.

Hier nun das Ergebnis mit InputShaping ohne PA:

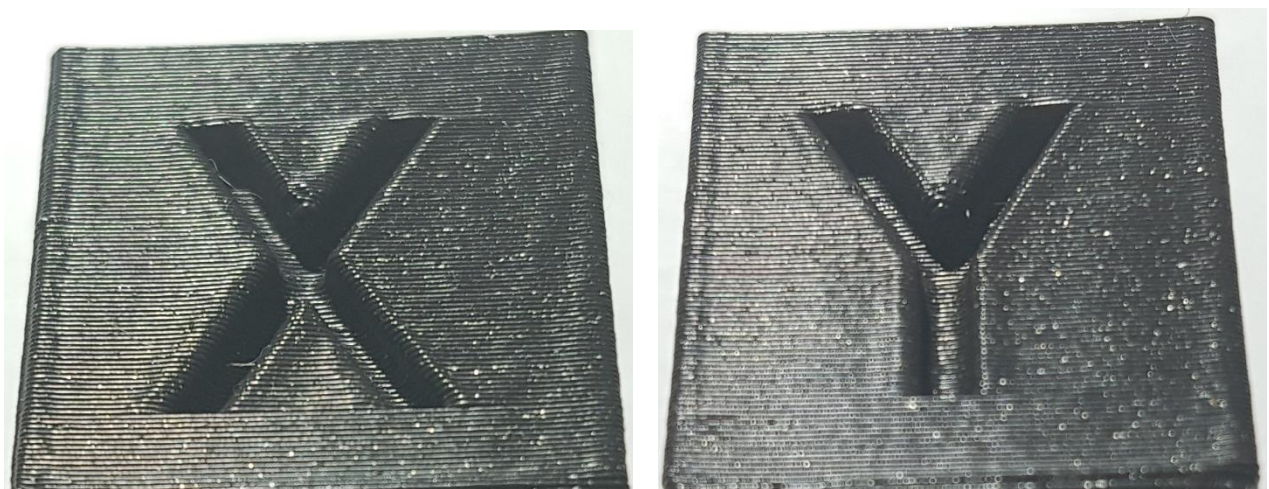


Das Ringing hat sich nun weitgehend zurückgezogen, dafür sind die Wülste an den Ecken wieder zum Vorschein getreten. Gut – dann versuchen wir es mit einem anderen Parameter (ähnlich dem Jerk) -> square_corner_velocity

Ich habe den Defaultwert von 5 auf verschiedene Werte reduziert und erhöht. Dasselbe Ergebnis – die Wülste bleiben. Auch eine Reduktion von PA war nicht wirklich eine Lösung, da sich das Ringing bei geringsten Änderungen wieder abbildete.

Im nächsten Versuch habe ich **max_accel_to_decel** von 1800mm/s² auf 3600mm/s² **erhöht** (Reduktionen auf 1200mm/s² und weniger haben auch nichts verändert bzw. verbessert) und **PA** auf 0.013 **reduziert**.

Ergebnis:



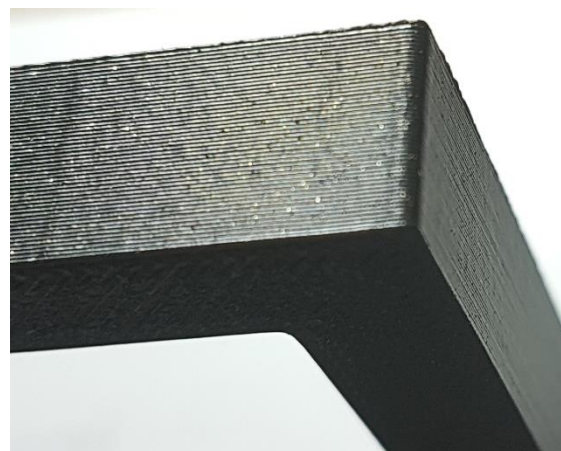
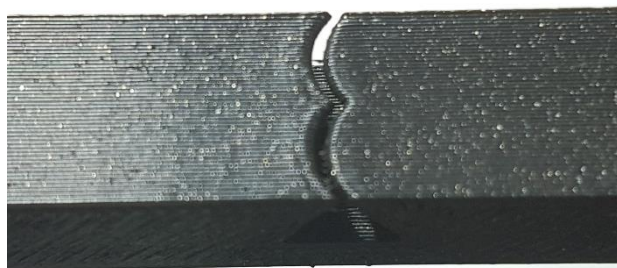
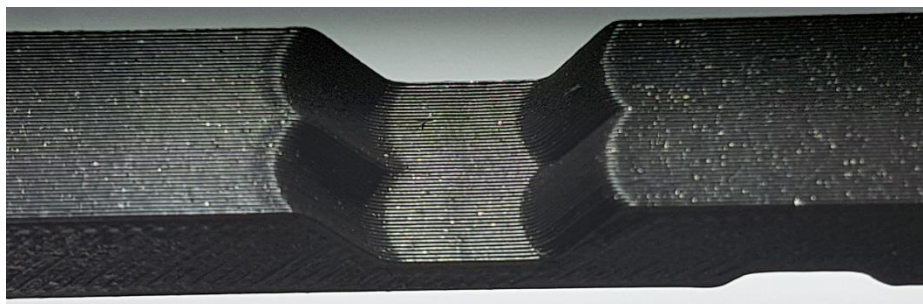


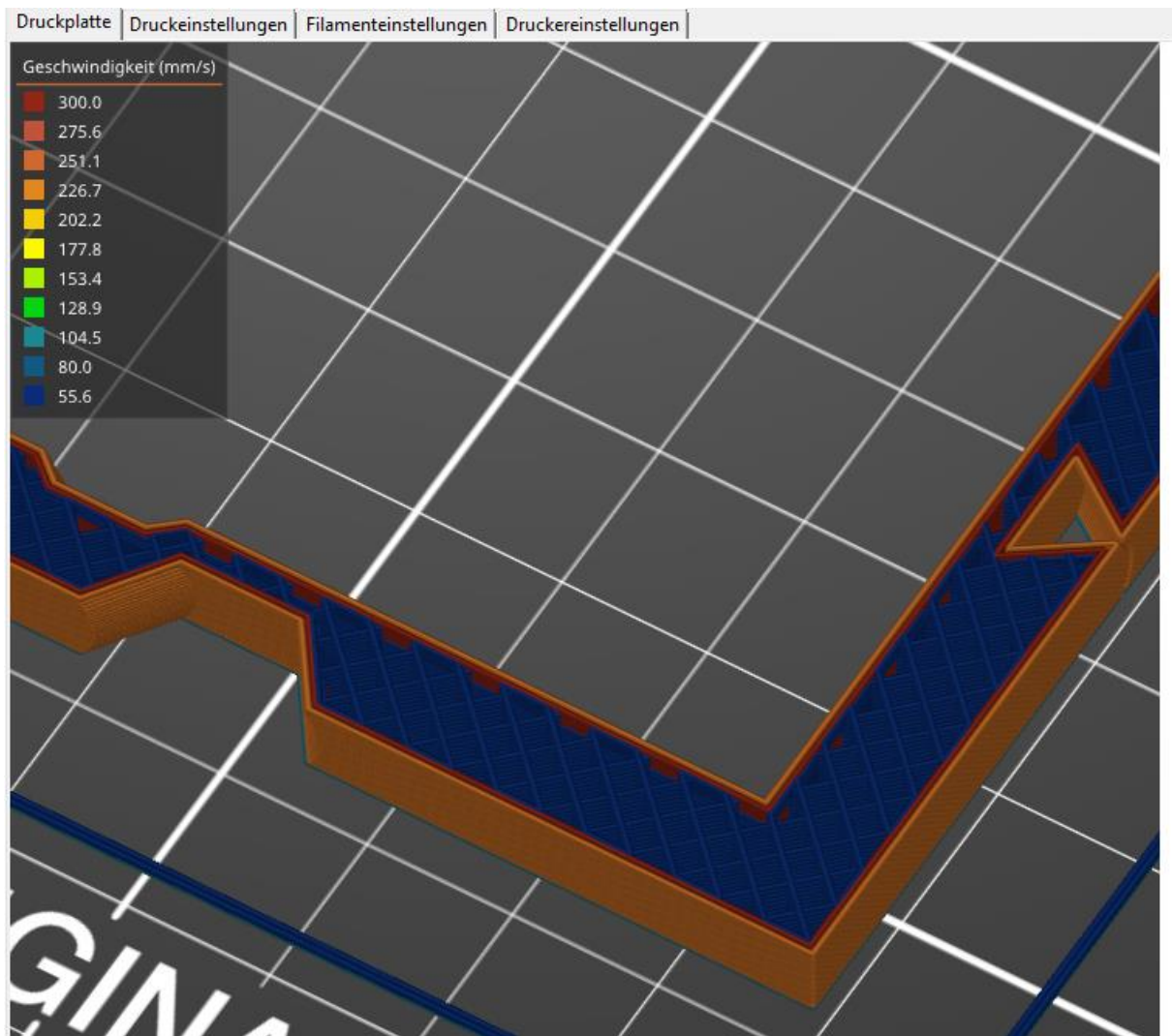
An der rot markierten Stelle findet der Layerwechsel statt, was naturgemäß eine Ausbuchtung erzeugt. Ich verlege diese meist in die Ecken, da man diesen Bereich gut manuell nacharbeiten und optisch aufbessern kann.

Das Ergebnis (Speed 300mm/s, InputShaping und PA 0.013 aktiv) sieht für mich für den Anfang mal zufriedenstellend aus und werde ich vorerst so belassen, bis ich die o.g. Gewichtsreduktionen durchgeführt habe.

Maßlich liegt alles Bereich von $\pm 0.05\text{mm}$. Motoren und Gurte zeigen sich bzgl. Temperaturentwicklung und Verschleiß bislang wenig beeinflusst von den höheren Geschwindigkeiten. Das muss aber über einen längeren Zeitraum beobachtet werden.

Hier nochmal die ersten 10mm des Towers mit denselben Einstellungen:





(PrusaSlicer 2.3.1 - überall 2 Perimeter, 20% Infill, Druckzeit ohne Heizphase ca. 15 Minuten)

Hier noch 2 kurze Videos zum Klipperprojekt in Action:

<https://youtu.be/Ptz9xDf4OTc>

https://youtu.be/yqthw_KTa2g



Zum Abschluss:

Um mit einem guten Ergebnis ins Ziel zu laufen, würde ich sagen, dass Folgendes wichtig ist:

- Massen (Drucktisch, Extruder) so gering wie möglich halten
- die Mechanik stabil und möglichst spielfrei aufbauen (Führungswellen, Rahmensteifigkeit, keine losen Schrauben...
- korrekte Gurtspannung
- etwas oder etwas mehr Geduld beim Justieren der Tuningparameter, Extrusionswerte vorher kalibrieren
- zum Kalibrieren von InputShaping möglichst keinen axdl345 Clone verwenden
- PA experimentell einstellen oder gegebenenfalls weglassen, da es dem InputShaping entgegenwirkt. Hier muss man ein harmonisches Zusammenspiel durch Ermittlung von optimalen Werten finden. Es geht, ist aber noch ausbaufähig.

Ich werde sicherlich nicht die letzte Anpassung oder Optimierung durchgeführt haben. Im laufenden Betrieb wird sich zeigen, wo noch „geschraubt“ werden muss. Auf jeden Fall werde ich noch Gewichte an Extruder und hauptsächlich am Drucktisch reduzieren und das Prozedere erneut durchführen. Ganz optimal bzgl. Ringing wird's wahrscheinlich nicht werden, aber ein guter Kompromiss zwischen Speed und Oberflächenqualität bzw. Maßhaltigkeit, was für viele meiner Langzeit Druckteile ein Gewinn an Zeit sein wird, da sich die Druckzeit bis zu 70% verkürzt. Nebenbei erwähnt: Die Stepermotoren sind in den vielen Testversuchen nicht mal warm geworden.

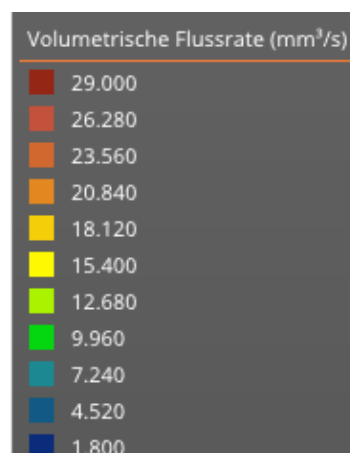
Die kleinen, filigranen Teile darf dann der gute alte MK3 (Prusa FW 3.8.1, ebenfalls mit Octoprint) abarbeiten. Der hat schon 4 1/2 Jahre und viele KM Filamentfluss auf dem Buckel, läuft aber Dank regelmäßiger Wartung und Reinigung nach wie vor wie am ersten Tag.

Das genauere Prozedere der oben genannten Schritte ist [hier](#) beschrieben, allerdings ist der Artikel nicht in Deutsch verfügbar bzw. habe ich noch nichts Vergleichbares dazu gefunden.

Außer viel Zeit und Geduld für das Softwaretechnische (Klipper und printer.cfg) habe ich kein Geld für's Klipperprojekt (außer den Drucker selbst) investiert, da mein System ja ohnehin mit einem Pi und Octoprint ausgerüstet war. Den ADXL345 habe ich hier gekauft:

<https://electronics.semaf.at/ADXL345-Acceleration-Sensor>

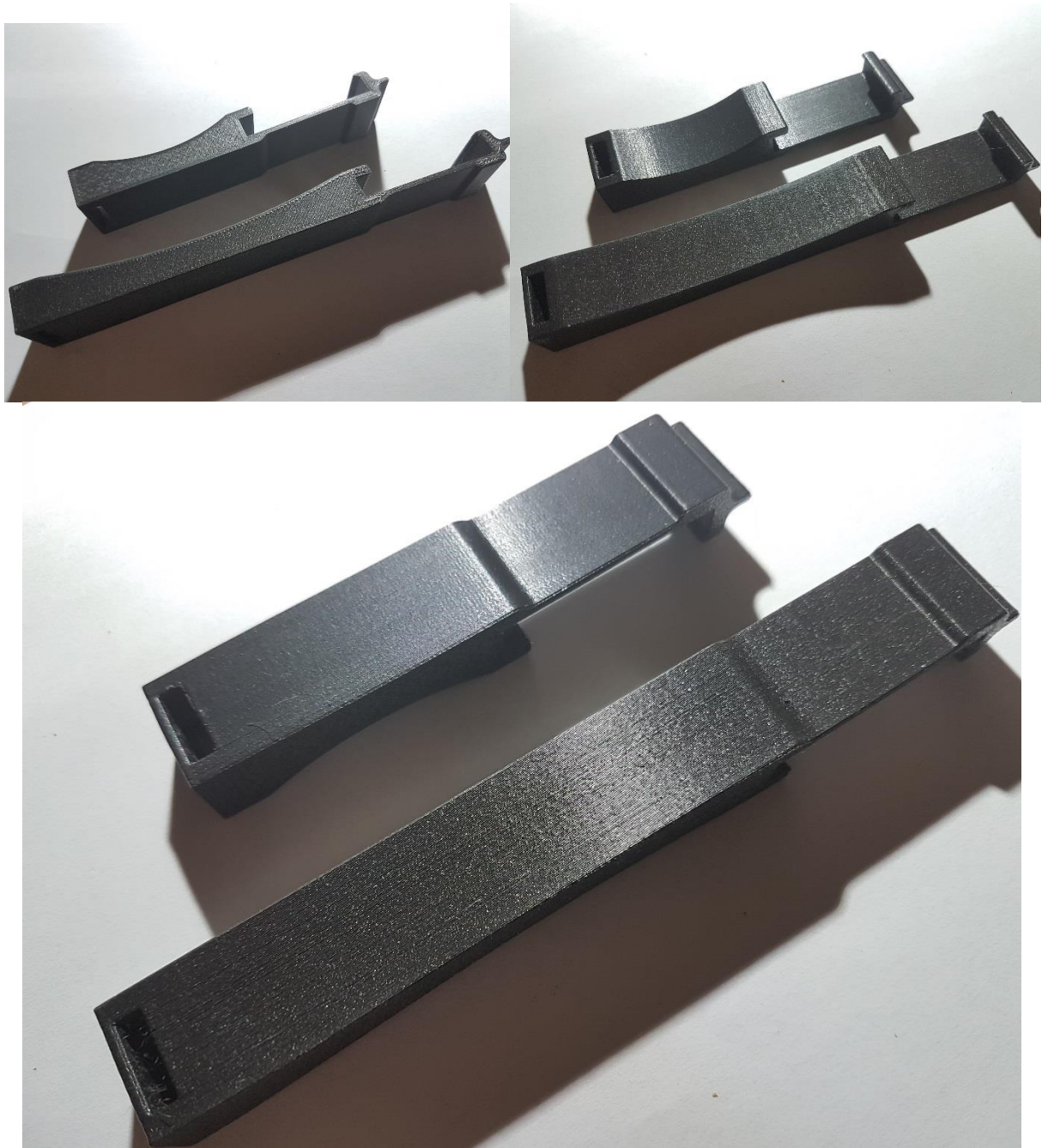
Noch etwas zum Volcano Hotend (Kupfer/Nickel/Bimetall, Düse Messing 0.4mm). In meinen Tests hat dieses ohne Probleme volumetrische Flussraten bis 29mm³/s abgearbeitet.



Ich zeige noch einen Vergleich zweier gedruckter Bauteile, einmal Standard MK3 Speed, und einmal Klipper MK3 Speed.

Das kürzere Bauteil wurde auf dem originalen MK3 mit 50mm/s Druckgeschwindigkeit in 93 Minuten gedruckt. Das längere wäre theoretisch in 136 Minuten fertig gewesen.

Das längere Bauteil hat der Klipper-MK3 in 50 Minuten (minus 63% Druckzeit) mit einer Druckgeschwindigkeit bis 300mm/s abgearbeitet.



(ASA black und ASA Galaxy black, Layerhöhe 0.2mm, 50% Infill, Temp. 260/105)

Karl, Wien am 22.04.2022