

*Klipper Einrichtung

- [*Klipper - Neopixel einrichten und nutzen](#)
- [*Klipper zu spezifischen Druckern](#)
- [*Klipper - Z-Offset richtig einstellen](#)
- [*Klipper - Kiauh installieren](#)
- [*Klipper - Deutsche Übersetzung](#)
- [*Slicer - Start/End Code einrichten / oder Fehler beheben](#)
- [*Klipper - Firmware Retract einrichten](#)

*Klipper - Neopixel

einrichten und nutzen

Habt Ihr Probleme bei der Einrichtung oder Benutzung euer Neopixel. Dann zeigen wir euch wie alles funktioniert.

Konfiguration in Klipper einbinden

Folgende Konfiguration müsst ihr in eure printer.cfg inkludieren oder direkt einfügen.

[Link zur Config](#)

In der Printer.cfg :

```
[include ./neopixel_pin.cfg]
```

Benutzt den Namen der CFG, den ihr hinterlegt habt

Pin (Schnittstelle) definieren:

```
[neopixel sb_leds]
pin:
```

Mit dem Parameter "pin" definiert Ihr die Schnittstelle zu euren LED. Diesen müsst ihr natürlich für euer Board entsprechend rausfinden.

Für den Vyper gilt zum Beispiel: Extended Board am Toolhead -> pin: PA13

Wie viele LEDs sind vorhanden

```
[neopixel sb_leds]
chain_count: 3
```

In diesem Beispiel besteht der LED Strand aus insgesamt 3 LEDs.

Welche LEDs sind verbaut

```
[neopixel sb_leds]
color_order: GRBW
```

Hier müsst ihr schauen, was Ihr euch gekauft hat. Grundlegend gibt es zwei Typen. **WS2812B** = GRB und **SK6812** = GRBW.

Farbe bei Hochfahren

```
initial_RED: 0.0
initial_GREEN: 0.0
initial_BLUE: 0.0
initial_WHITE: 0.0
```

Mit diesen Werten könnt ihr die Farbe nach dem Boot festlegen. Setzt dafür einfach den Wert zwischen 0.0 und 1.0. In der Regel nimmt man 0.0 für aus und 1.0 für ein.

Ansteuerung

Konsolenbefehl zur Ansteuerung

```
SET_LED LED=sb_leds RED=1 GREEN=1 BLUE=0 WHITE=0 INDEX=1 TRANSMIT=1
```

- LED= Name eurer Neopixel Schnittstelle ([neopixel sb_leds]). Hier sb_leds
- RED|GREEN|BLUE|WHITE = Farbe ein oder aus (1 oder 0)
- INDEX= Welche LED im Strand soll angesteuert werden

In dem oben genannten Beispiel wird die LED 1 mit der Farbe Gelb angesteuert (Rot + Grün) Hier sind einige Farbbeispiele in einer Konfig. hinterlegt. ([Neopixel - Farben](#)) `[include ./neopixel_farben.cfg]`

Benutzt den Namen der CFG, den ihr hinterlegt habt

Statusmeldungen im Gcode

Link zur [Statusmeldung-Konfiguration](#)

```
[include ./neopixel_status.cfg]
```

Hier könnt ihr folgende Meldungen einfach in einen Gcode eurer Wahl einbinden.

```
# The following status macros are available:
# STATUS_READY
# STATUS_OFF
# STATUS_BUSY
# STATUS_HEATING
# STATUS_LEVELING
# STATUS_HOMING
# STATUS_CLEANING
# STATUS_MESHING
# STATUS_CALIBRATING_Z
# With additional macros for direct control:
# SET_NOZZLE_LEDS_ON
# SET_LOGO_LEDS_OFF
# SET_NOZZLE_LEDS_OFF
```

*Klipper zu spezifischen Druckern

Anycubic Vyper

[Anycubic Vyper Modifikationen](#)

[Anycubic Vyper allgemein](#)

Anycubic i3 Mega

[Anycubic i3 Mega](#)

Creality - Ender 3 S1/ Pro / Plus

[Creality - Ender 3 S1/ Pro / Plus](#)

Elegoo - Neptune 3

[Elegoo - Neptune 3](#)

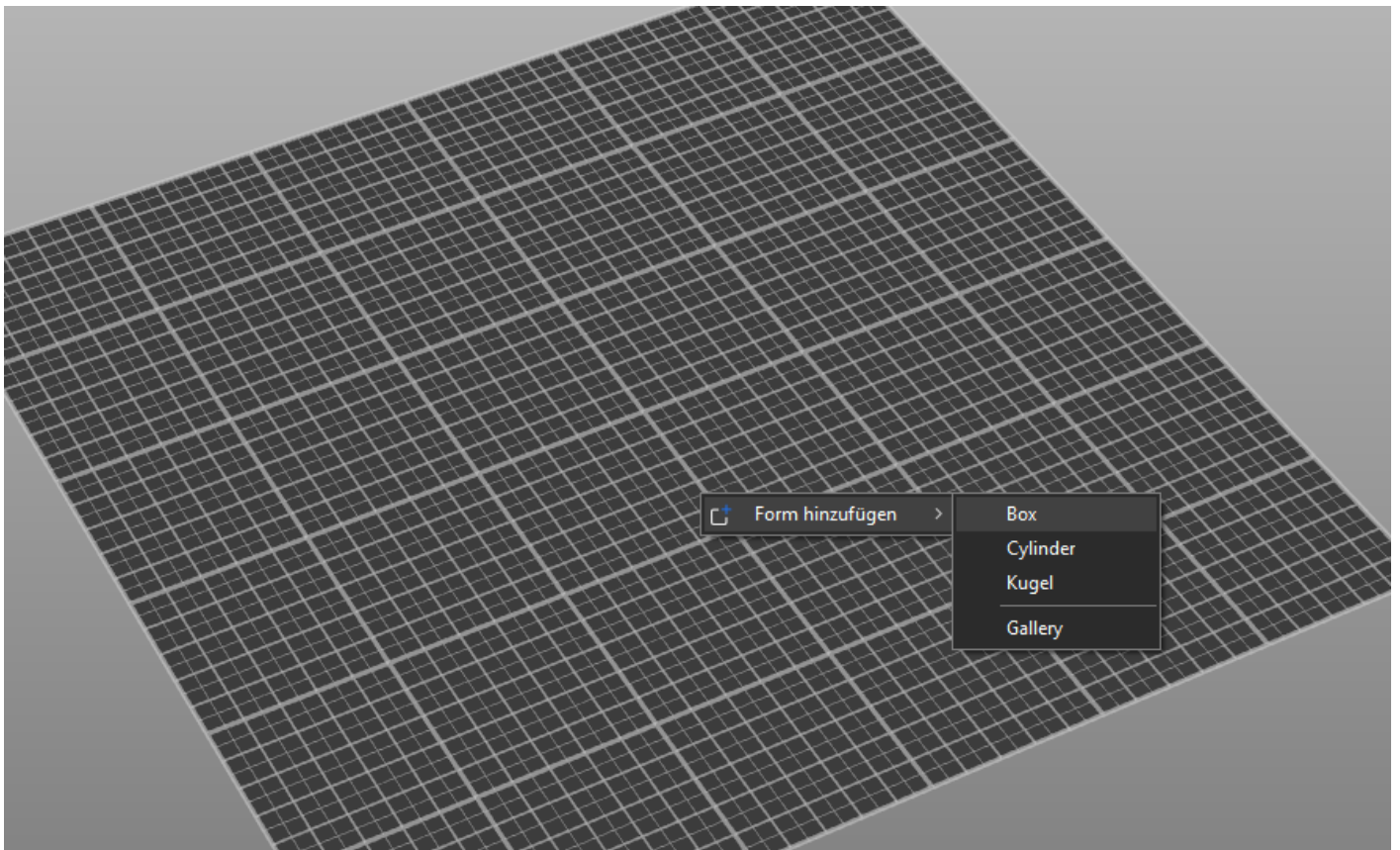
Voron

[Voron](#)

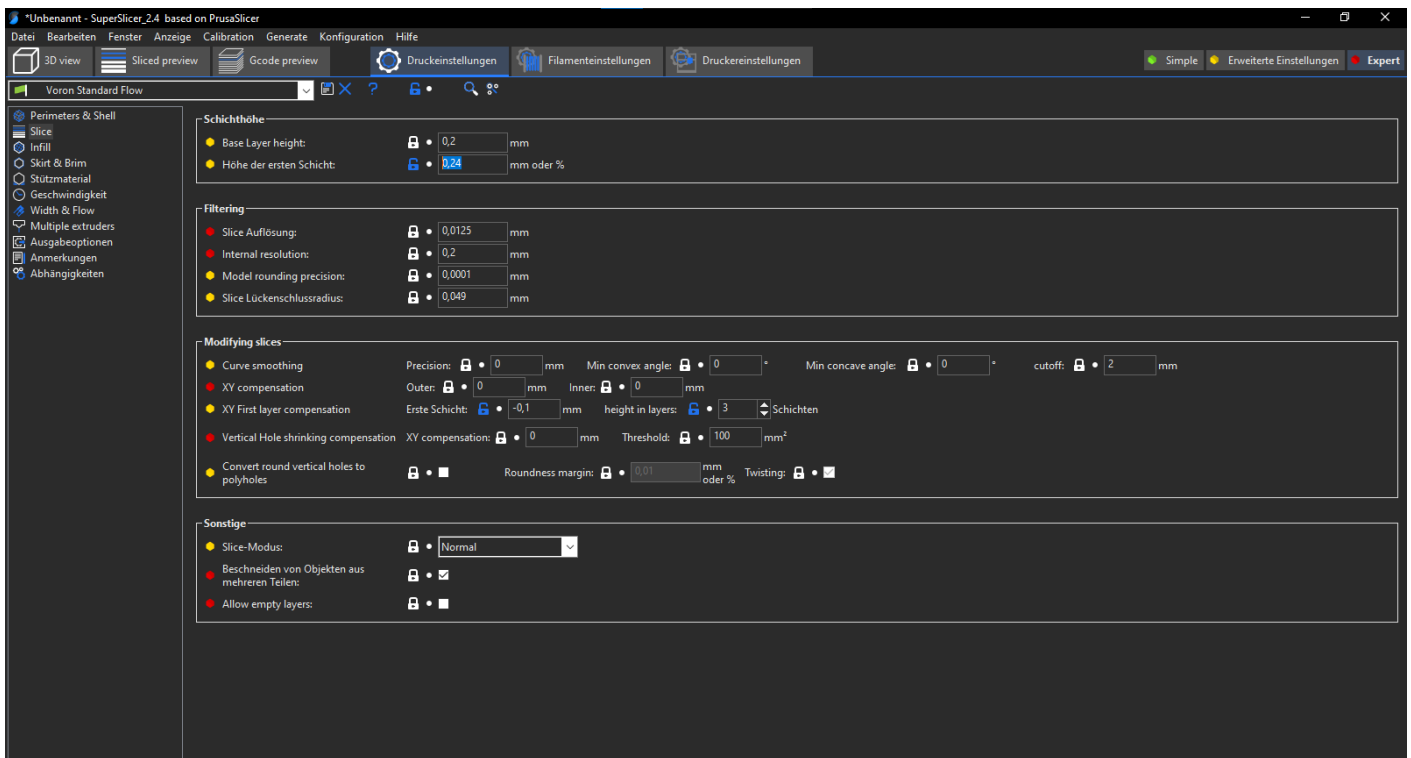
*Klipper - Z-Offset richtig einstellen

Ihr habt den PROBE_CALIBRATE durchgeführt und nach dem ACCEPT mit einem klicken auf SAVE_CONFIG eure Config gespeichert ?
Sehr gut ... doch wie geht es nun weiter ?

Nun ist es daran, euren Z-Offset Live "On the Fly" fein zu tunen, mit den sogenannten Babysteps. Als erstes erstellt Ihr euch im SuperSlicer eine Datei, die die Höhe eurer ersten Schicht hat. Dafür geht Ihr mit Rechtsklick auf die Baufläche im SuperSlicer und fügt eine Form (Box) hinzu.

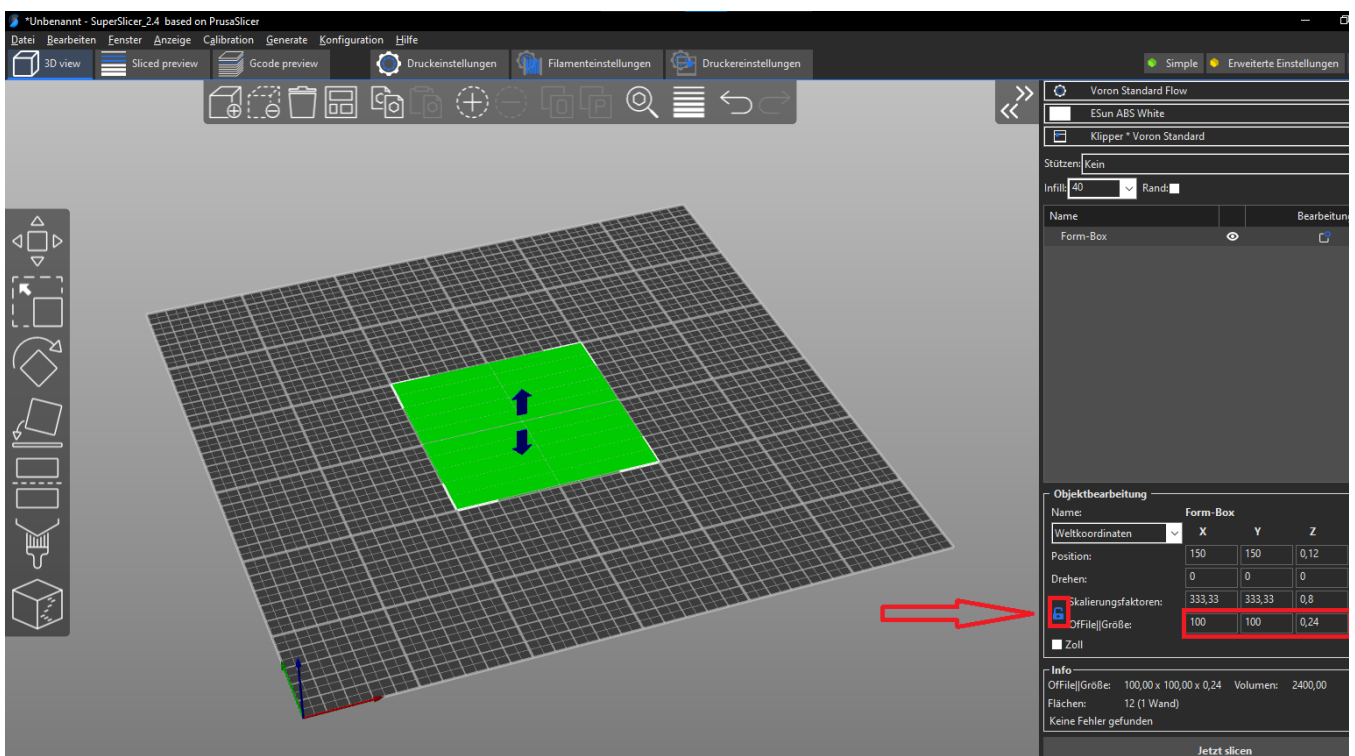


Nun müsst Ihr die Höhe eurer ersten Schicht aus dem Slicer ermitteln.



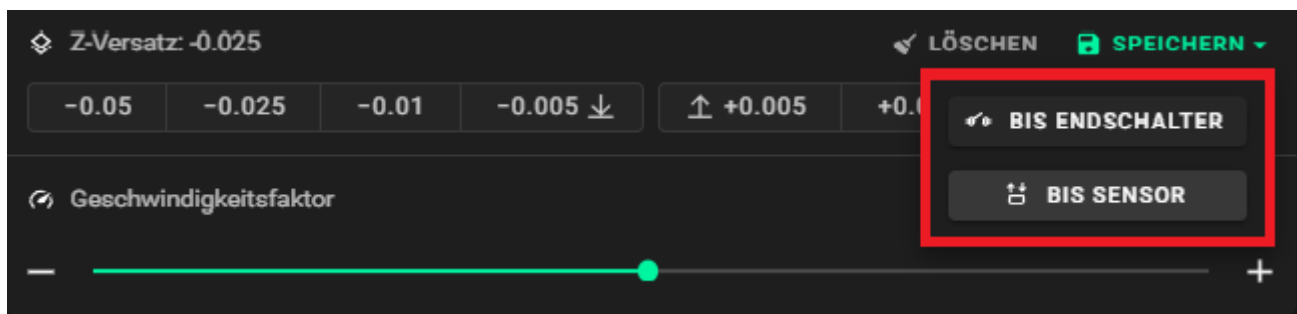
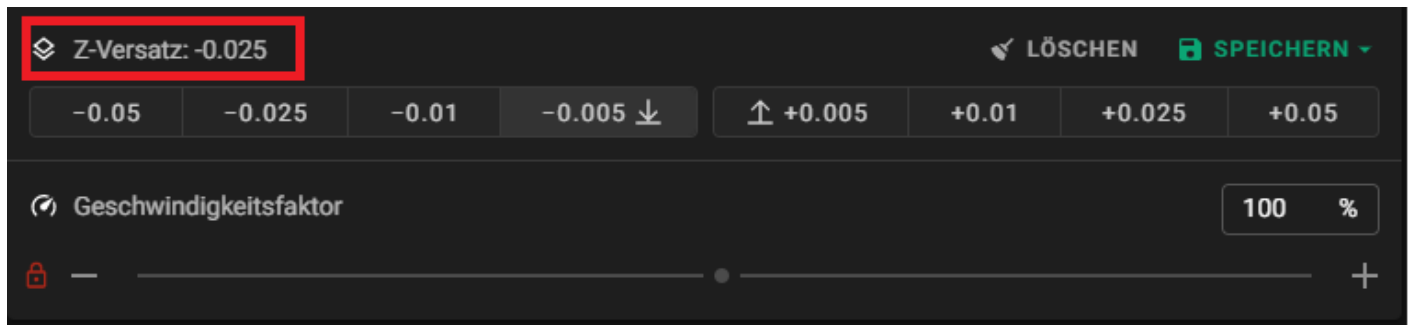
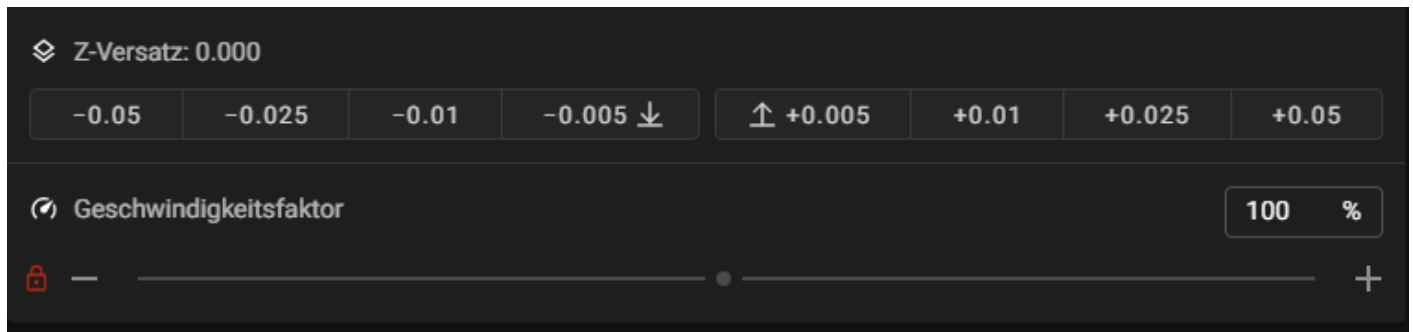
Die folgenden Einstellungen müssen nun noch im Slicer vorgenommen werden, damit Ihr die richtigen Maße für die Form habt.

- Als erstes Klickt Ihr auf der rechten Seite auf das Schloss. Dies sollte nun "offen" dargestellt werden.
- Als zweites tragt Ihr die Werte (je nach Bettgröße oder bedarf) ein. Wichtig ist, das der letzte Punkt eurer ersten Schichthöhe entspricht. (in meinem Fall, 0.24mm)



Nun kann die Datei gesliced und an den Drucker geschickt werden.

Beginnt nun mit dem Drucken und stellt den Offset Live über Mainsail ein. Dies geschieht über diese Toolbar



- Habt ihr eure seitlichen Endstops noch in Benutzung, wählt Ihr nun Speichern bis Endschalter aus. (In der Printer.cfg wie folgt "endstop_pin: PG10 ---> Hardware Endstop")
- Habt Ihr die Variante mit einem BL-Touch und die seitlichen Endstops sind deaktiviert, wählt ihr nun Speichern bis Sensor aus. (In der Printer.cfg wie folgt "endstop_pin: probe:z_virtual_endstop ---> Bltouch")

Wenn Ihr nun soweit seid, dass Ihr die für euch beste Einstellung gefunden habt, ist es sehr wichtig, dass Ihr die Config erneut abspeichert.

Geht dafür einfach über den Save Config Button oben rechts neben dem NOTAUS, oder schreibt SAVE_CONFIG in die Konsole und bestätigt mit Enter.

Die Firmware macht nun einen Neustart.

*Klipper - Kiauh installieren

[Hier ist unsere Anleitung zur Installation von kiauh](#)

*Klipper - Deutsche Übersetzung

[Hier ist der Link PDF zur Deutschen Übersetzung](#)

*Slicer - Start/End Code einrichten / oder Fehler beheben

Probleme und Fehler

Ihr habt das Problem, dass euer Startcode nicht funktioniert?

- Bedmesh wird nicht geladen
- Temperaturen werden nicht übergeben
- Er funktioniert gar nicht

Parameter und Platzhalter

Bei dem Startcode ist es wichtig, dass die sogenannten Parameter und Placeholder sauber definiert sind. Ihr solltet diesen dann im Gcode kontrollieren

Beispiel:

```
START_PRINT BED=[first_layer_bed_temperature] EXTRUDER=[first_layer_temperature]
```

```
START_PRINT Parameter=[Placeholder] .....
```

Parameter

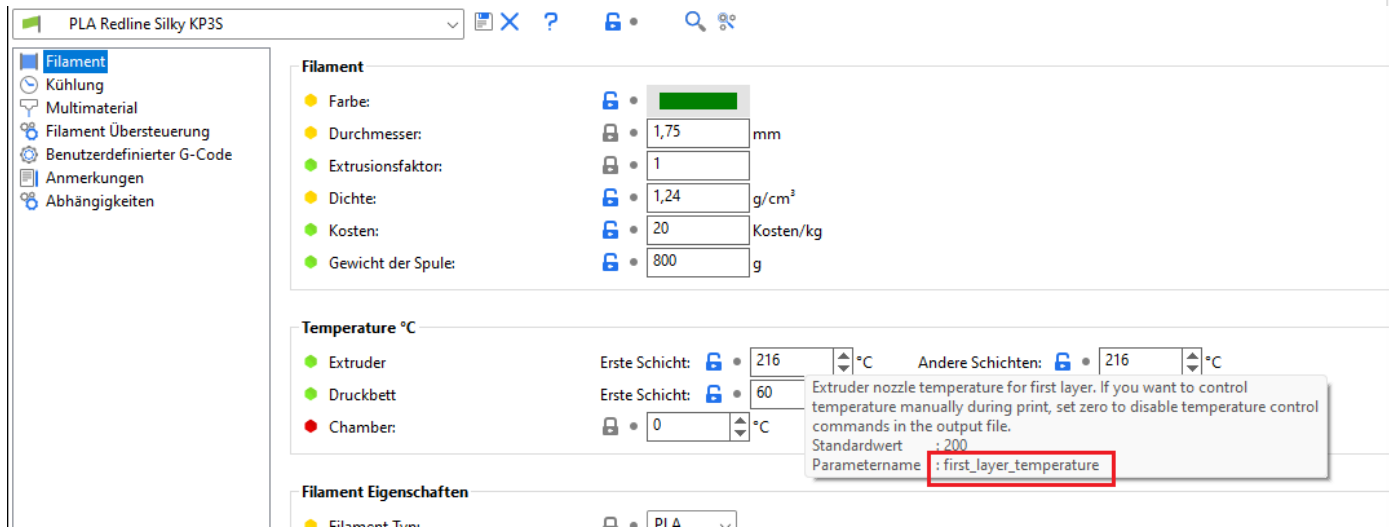
Mit einem Parameter kannst du einen Zahlenwert einem "Wort"(Platzhalter) zuweisen und abfragen.

Im Falle des Startcodes sind es meist Variablen wie:

BED / EXTRUDER / BED_TEMP / EXTRUDER_TEMP / CHAMBER_TEMP / SHEET / AREA_START / AREA_END

Placeholder (Platzhalter) im Slicer

Mit den sogenannten Placeholder kannst du Werte aus dem Slicer abfragen und einer Variable zuweisen.



Prusa / Super Slicer Placeholder:

https://help.prusa3d.com/article/list-of-placeholders_205643

Beispiel:

Extruder Temperatur des First Layers: "first_layer_temperature"

Cura Platzhalter:

http://files.fieldofview.com/cura/Replacement_Patterns.html

Startcode Beispiele:

Hier findest du Beispiele unserer Startcodes, passend zu den Slicern.

Bitte gleiche diese mit deinen **Variablen** im Makro ab.

Ebenfalls sind die **Klammern** wichtig {} []

Cura Startcode:

```
START_PRINT BED={material_bed_temperature_layer_0} EXTRUDER={material_initial_print_temperature}
```

Prusa Startcode:

```
START_PRINT BED=[first_layer_bed_temperature] EXTRUDER=[first_layer_temperature]
```

Super Slicer Startcode:

```
START_PRINT BED={first_layer_bed_temperature} EXTRUDER={first_layer_temperature}
```

Orca Slicer Startcode:

```
START_PRINT EXTRUDER=[nozzle_temperature_initial_layer] BED=[hot_plate_temp_initial_layer]
```

Wie wird eine Variable mit Parametern beschrieben

Annahmen:

- Startcode im Slicer : **START_PRINT BED={first_layer_bed_temperature} EXTRUDER={first_layer_temperature}**
- Schreibfehler im Start_Print:
{% set extruder = params.EXTRUDER|default(0) %}
{% set bed = params.BED|default(0) %}

Vom Slicer werden die **Parameter** BED und EXTRUDER dem Start_Print Makro zur Verfügung gestellt.

Beispiel aus Gcode:

```
172 ;TYPE:Custom
173 M115.U3.1.0.; tell.printer.latest.fw.version
174 M117.Start.Print.Script.loading
175 START PRINT.BED=60.EXTRUDER=216.
176 M117.Printing.started
177 M107
```

Das Makro macht folgendes:

set xyz = setzte oder beschreibe in die variable xyz

extruder/bed = Variablenname

params.EXTRUDER|default(0) = Die Werte aus dem Parameter (params) Extruder

Komplettes Beispiel

Quelle:

[Crydteam - Vyper Extended Github - marcos.cfg](#)

[gcode_macro START_PRINT]

description: All what needs to be done at print start

gcode:

```
#### set defaults ####
{% set extruder = params.EXTRUDER|default(0) %}
{% set bed = params.BED|default(0) %}
#### end off definition ###

G28                      ; Home
M83                      ; Extruder relative mode
M104 S140                ; Extruder heat up standby temp 140
M190 S{bed}              ; Bed heat up
G1 X0 Y0 F2200[ ] [ ] [ ] [ ] [ ] ; Go to front
M109 S{extruder}         ; Extruder heat up to target temp
G92 E0.0                 ; Reset extruder length
G90                      ; Absolute positioning
BED_MESH_PROFILE LOAD="default"
PRIME_LINE                ; First move
```

Überprüfung ob der Slicer Werte in die Parameter schreibt:

1. Ein Objekt slicen und Gcode Exportieren
2. Gcode mit einem Editor öffnen
3. Mit der Sucher (STRG+F) nach eurem Start_Print Startcode suchen
4. Prüfen ob für Extruder und Bed auch eine Temperatur eingetragen wurde

Beispiel Codeausschnitt

```
172 ;TYPE:Custom
173 M115.U3.1.0.;.tell.printer.latest.fw.version
174 M117.Start.Print.Script.loading
175 START PRINT.BED=60.EXTRUDER=216.
176 M117.Printing.started
177 M107
```

*Klipper - Firmware Retract einrichten

Vorstellung Firmware Retract in Klipper

Hallo,

Jeder von euch kennt den Retract und dessen Funktion. Die meisten von euch definieren diesen im Slicer. Aber was hat denn jetzt der Firmware retract für Vorteile.

- **Konsistenz über verschiedene Slicer:** Da die Retract-Einstellungen in der Firmware verwaltet werden, ist es einfacher, konsistente Ergebnisse über verschiedene Slicer-Programme hinweg zu erzielen. Dies bedeutet, dass die Benutzer nicht für jeden Slicer individuelle Retract-Einstellungen anpassen müssen.
- **Anpassungsfähigkeit:** Firmware Retract ermöglicht es, Retract-Einstellungen zentral in der Firmware zu ändern, ohne den G-Code jedes Mal anpassen zu müssen. Dies erleichtert das Experimentieren mit verschiedenen Retract-Einstellungen, um die bestmögliche Druckqualität zu erzielen.
- **Live Anpassung möglich:** Ihr könnt den Retract während des Druckes anpassen.
- **Vereinfachung des G-Codes:** Da der Retract direkt von der Firmware gesteuert wird, muss der G-Code nicht so viele Befehle enthalten, was zu einer Vereinfachung des G-Codes führt. Dies kann die Dateigröße der G-Code-Dateien verringern.
(in der Praxis vermutlich nicht relevant)
- **Implementierbarkeit:** Der Retract kann je nach Filamenttyp über den Startcode eingestellt werden.

Einrichtung FW Retract in Klipper

Offizielle Klipper Doku zu FW Retract:

https://www.klipper3d.org/Config_Reference.html?h=retract#firmware_retraction

Offizielle Klipper Doku zu G10 / G11:

https://www.klipper3d.org/G-Codes.html#firmware_retraction

Möglichkeit 1 : Direkt in printer.cfg

```
[firmware_retraction]
```

```
# G10 für Retract and G11 Deretract
```

```
retract_length: 3
```

```
# Die Laenge des Rueckzuges in mm wenn ein G10 Befehl angesteuert wird.
```

```
# Ebenfalls wenn nicht weiter definiert gilt er für den Deretract G11.
```

```
# Die Standardeinstellung ist 0mm.
```

```
retract_speed: 40
```

```
# Rueckzugsgeschwindigkeit in mm/s. Der Standardwert ist 20 mm/s.
```

```
#unretract_extra_length: 0
```

```
# Hier wird die zusaetzliche Laenge (in mm) definiert, die bei einem Deretract
```

```
# gemacht werden soll.
```

```
unretract_speed: 40
```

```
# Die Geschwindigkeit für einen Deretract in mm/s. Der Standardwert ist 10 mm/s.
```

Möglichkeiten 2: Extra CFG-Datei anlegen und einbinden (empfohlen)

- Lege eine cfg namens fw_retract.cfg im Stamm Konfigverzeichnis an. (Einfach über die Weboberfläche mit Datei erstellen)
- Kopiere folgenden Inhalt in diese CFG

```
[firmware_retraction]
```

```
# G10 für Retract and G11 Deretract
```

```
retract_length: 3
```

```
# Die Laenge des Rueckzuges in mm wenn ein G10 Befehl angesteuert wird.
```

```
# Ebenfalls wenn nicht weiter definiert gilt er für den Deretract G11.
```

```
# Die Standardeinstellung ist 0mm.
```

```
retract_speed: 40
```

```
# Rueckzugsgeschwindigkeit in mm/s. Der Standardwert ist 20 mm/s.
```

```
#unretract_extra_length: 0
```

```
# Hier wird die zusaetzliche Laenge (in mm) definiert, die bei einem Deretract
```

```
# gemacht werden soll.
```

```
unretract_speed: 40
```

```
# Die Geschwindigkeit für einen Deretract in mm/s. Der Standardwert ist 10 mm/s.
```

- Inkludiere die CGF in deiner printer.cfg mit folgendem Code

```
[include fw_retract.cfg]
```

Einrichtung FW Retract im Slicer:

SuperSlicer

SuperSlicer_2.5.59 based on PrusaSlicer

Datei Bearbeiten Fenster Anzeige Calibration Generate Konfiguration Hilfe

3D view Sliced preview Gcode preview Druckereinstellungen Filamenteinstellungen

SB * VyperSB

Allgemein benutzerdefinierter G-Code Maschinengrenzen Extruder 1 Anmerkungen Abhängigkeiten

Fähigkeiten

- Extruder: 1
- Einzelextruder mit Multi-Material: ☐
- Milling cutters: 0

Hochladen zum Druckhost

Note: All parameters from this group are moved to the Physical Printer settings (see changelog).

A new Physical Printer profile is created by clicking on the "cog" icon right of the Printer profiles combo box, by selecting the "Add physical printer" item in the Printer combo box. The Physical Printer profile editor opens also when clicking on the "cog" icon in the Printer settings tab. The Physical Printer profiles are being stored into SuperSlicer/physical_printer directory.

Firmware

- G-Code Typ: Klipper
- Unterstützt Stealth Modus: ☐
- Print remaining times: ☐ Method: M73
- Gcode precision: xyz decimals: 3 Extruder decimals: 5
- Processing limit: Maximum G1 per second: 1500 Minimum extrusion length: 0,035
- Illegal characters:

Cooling fan

- Speedup time: 0 s Only for overhangs: ☒
- Kickstart duration: 0 s
- Fan PWM from 0-100: ☐

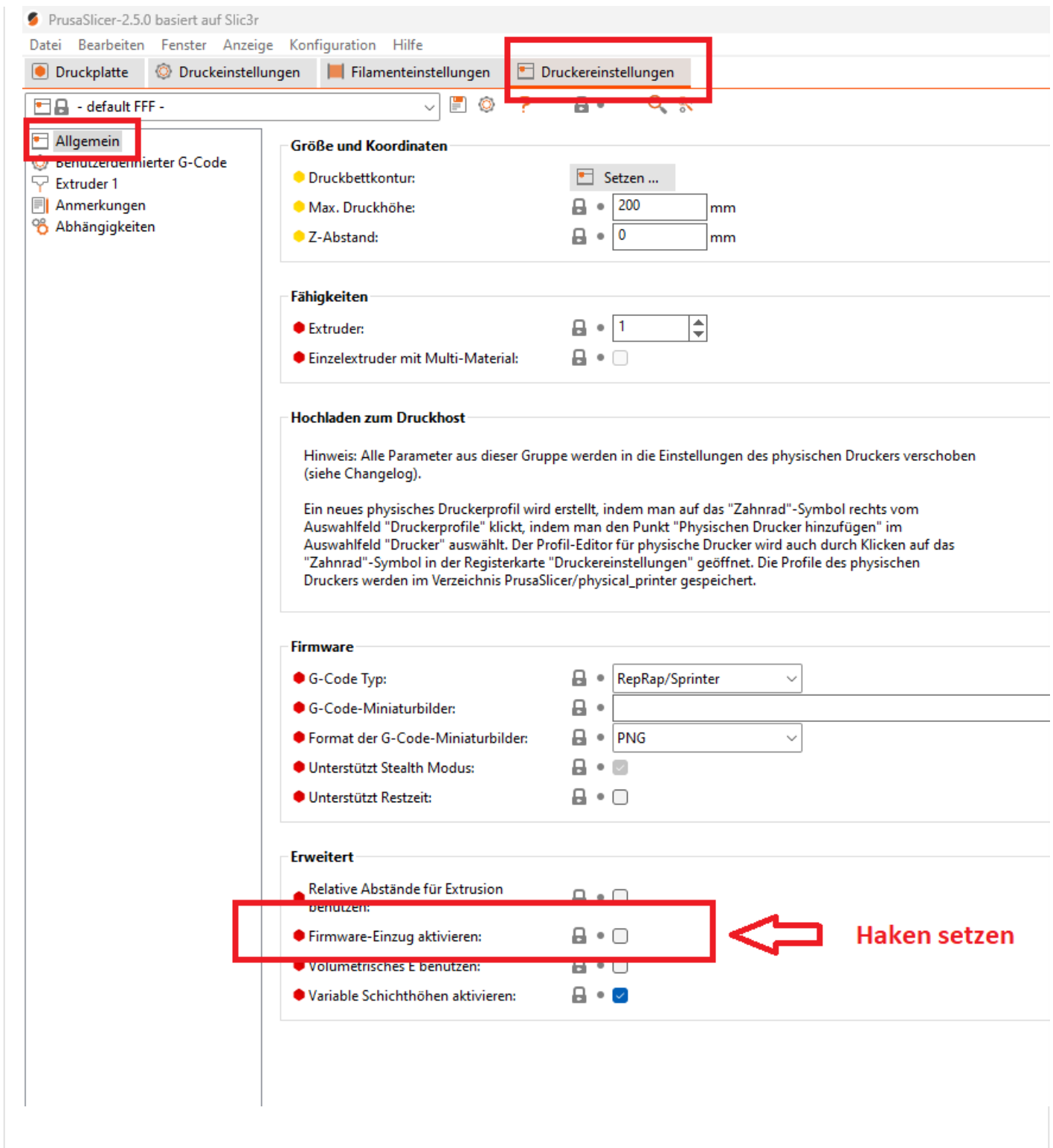
Thumbnails

- Size for Gcode: Small: x: 32 y: 32 Big: x: 400 y: 300
- Color override: ☒
- Thumbnail options: Format of G-code thumbnails: PNG Bed on thumbnail: ☐

Erweiterte Einstellungen

- Relative Abstände für Extrusion benutzen: ☒
- Firmware-Einzug aktivieren: ☒
- Volumetrisches E benutzen: ☐
- Variable Schichthöhen aktivieren: ☐

PrusaSlicer



Cura


Wenn es jemand am Laufen hat gerne als Info an uns.

Über das Addon "Printer Settings" bekommt man zwar die Option, aber es würde nicht sauber in den Gcode geschrieben.



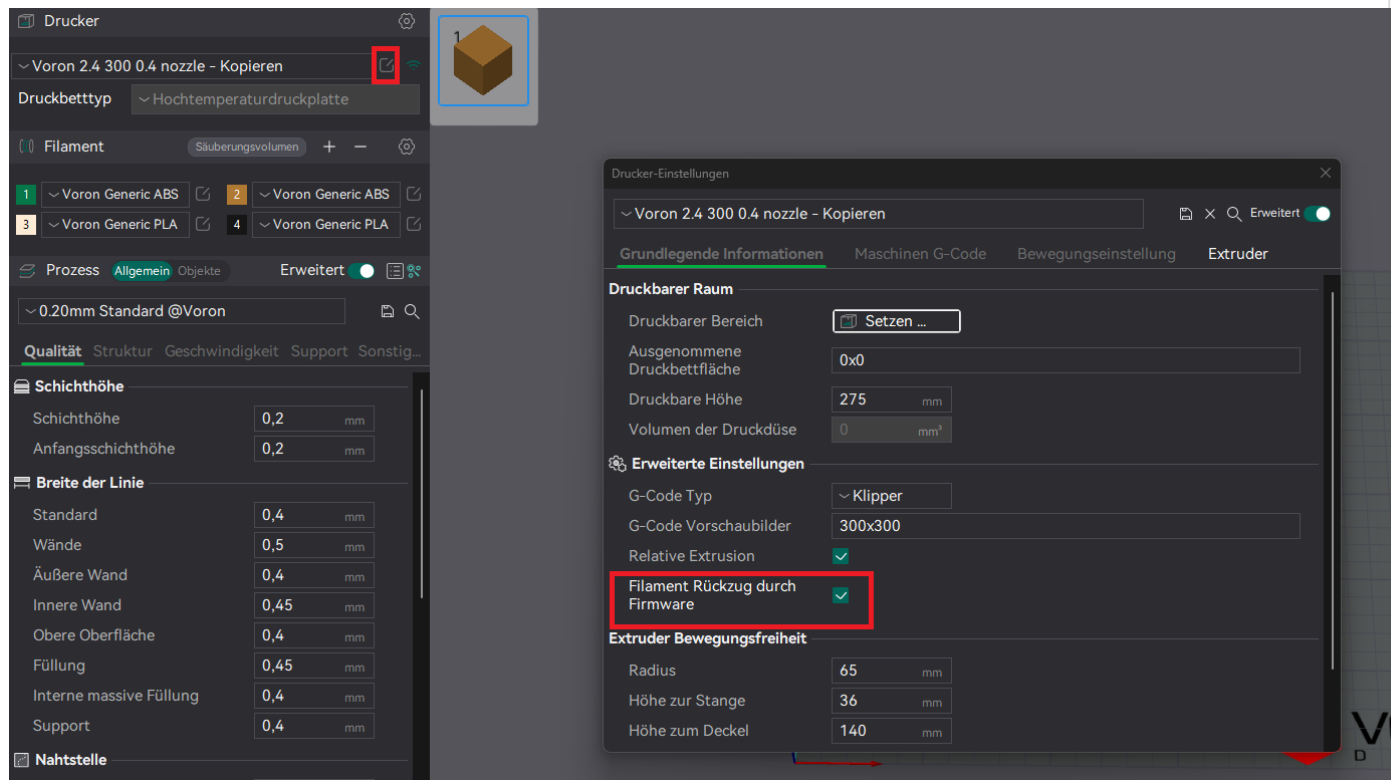
Printer Settings 3.6.0

Adds printer settings to the sidebar

Von **fieldOfView** 

Installieren

Orca Slicer



GCODE Kontrolle ob FW retract geschrieben wurde

GCODE slicen und exportieren. Danach mit einem Editor öffnen und nach G10 oder G10 suchen

```
11069 ;WIPE_START
11070 G1 X122.3 Y134.204 F27000
11071 ;WIPE_END
11072 G10.; retract
11073 ;WIPE_START
11074 G1 F21600
11075 G1 X110.65 Y134.35 E-1.22982
```

```

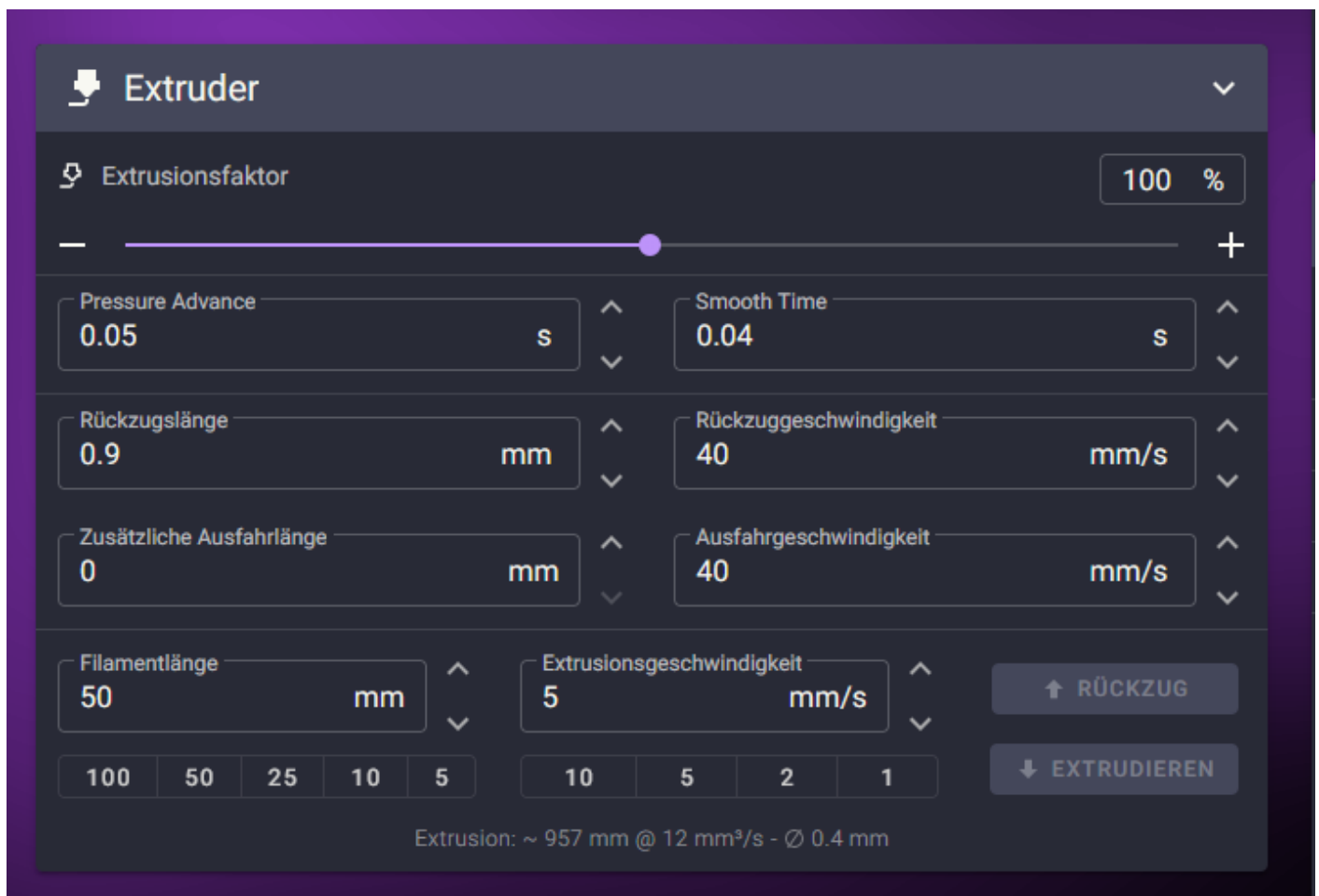
11058 G1·X122.5·Y134.55
11059 G1·Z24.44
11060 G11·;·unretract
11061 ;TYPE:External·perimeter
11062 ;WIDTH:0.4
11063 G1·F3000
11064 G1·X110.45·Y134.55·E0.33989
11065 G1·X110.45·Y110.45·E0.67978
11066 G1·X134.55·Y110.45·E0.67978

```

Retract über die Mainsail Oberfläche einstellen:

Ihr könnt jetzt den Retract im "Extruder" Menü einstellen.

Hier ein Beispiel:



Filamentabhängiger Retract

[Link zum Video "Filamentabhängigen Gcode"](#)

<https://www.youtube.com/embed/voOaRlwSczl?t=585s>

1. Slicer für die Weitergabe der Variable einrichten

Richtiger Filamenttyp im Slicer setzten

The screenshot shows the PrusaSlicer interface with the 'Filamenteinstellungen' tab active. The left sidebar has 'Filament' selected. The main panel displays the following settings:

- Filament:**
 - Farbe: [Black]
 - Durchmesser: 1,75 mm
 - Extrusionsfaktor: 0,9
 - Dichte: 1,24 g/cm³
 - Kosten: 20 Kosten/kg
 - Gewicht der Spule: 800 g
- Temperature °C:**
 - Extruder: Erste Schicht: 220 °C
 - Druckbett: Erste Schicht: 60 °C
 - Chamber: 0 °C
- Filament Eigenschaften:**
 - Filament Typ: PLA (highlighted with a red box and arrows)
 - Lösliches Material: []
 - Shrinkage: 100 %
 - Max line overlap: 100 %

Wie sieht jetzt die Variable aus?

Filament Eigenschaften

The close-up shows the 'Filament Typ' dropdown set to 'PLA'. A tooltip is displayed with the following text:

- Die Materialart des Filaments zur Verwendung in benutzerdefinierten G-Codes.
- Standardwert : PLA
- Parametername : filament_type

2. Variable im Startmakro des Slicers hinterlegen

Startcode im Superslicer

```
START_PRINT BED={first_layer_bed_temperature} EXTRUDER={first_layer_temperature}  
MATERIAL={filament_type}
```

3. Prüfen ob Variable übergeben wurde

So soll es aussehen

```
M117 Start Print Script loading  
START_PRINT BED=60 EXTRUDER=220 MATERIAL=PLA  
; custom gcode end: start_gcode  
.....
```

- Objekt slicen
- Gcode mit editor öffnen
- Nach "Material" durchsuchen -> Es sollte euer Filamenttyp hinterlegt sein

Gcode für die Materialabfrage in einem Makro

```
# Materialabhaengige Parameter wie PA, FLOW, Retract, Mesh usw.  
RESPOND MSG="Material: {material}"  
{% if material == "PLA" %}  
#SET_RETRACTION RETRACT_LENGTH=1.2  
#SET_RETRACTION RETRACT_SPEED=30  
#SET_RETRACTION UNRETRACT_SPEED=30  
{% elif material == "PET" %}  
#SET_RETRACTION RETRACT_LENGTH=1.5  
#SET_RETRACTION RETRACT_SPEED=20  
#SET_RETRACTION UNRETRACT_SPEED=20  
{% elif material == "ABS" %}  
#SET_RETRACTION RETRACT_LENGTH=0.9  
#SET_RETRACTION RETRACT_SPEED=40  
#SET_RETRACTION UNRETRACT_SPEED=40
```

{% endif %}