Klipper Installations Anleitung

Für die Klipper Installation empfehle ich euch ein paar Werkzeuge.

|  |  |  |
| --- | --- | --- |
| Name | Beschreibung | Link |
| PuTTY | Terminal-Programm, um auf euren Raspberry zugreifen zu können. | <https://www.putty.org/> |
| WinSCP | WinSCP ist ein grafischer Open Source SFTP und FTP Client für Windows. Den könnt ihr benutzen, um Dateien zwischen eurem PC und dem Raspberry auszutauschen | <https://winscp.net/eng/docs/lang:de> |
|  |  |  |

Ein paar nützliche Links möchte ich euch auch zeigen.

|  |  |
| --- | --- |
| Beschreibung | Link |
| Offizielle Klipper Webseite | <https://www.klipper3d.org/> |
| Klipper auf GitHub | <https://github.com/KevinOConnor/klipper> |
| Eine Klipper Installationsanaleitung | <https://drucktipps3d.de/forum/topic/klipper-anleitung/> |
| KIAUH - Klipper Installation And Update Helper: | <https://github.com/th33xitus/kiauh> |
| Raspberry Pi 4B - usbmount in Raspbian Buster-Lite einrichten | <https://www.dgebhardt.de/raspi-projects/projects/usbmount.html> |
| OctoPrint | <https://octoprint.org/> |
|  |  |

# Vorbereiten des Raspberry Pi.

Diese Anleitung, geht davon aus das ihr schon OctoPrint in Einsatz habt. Und auf dem Drucker Marlin läuft. Bevor ihr Klipper installiert solltet ihr den Raspberry auf den aktuellen Stand bringe. Hierzu verbindet ihr euch mit PuTTY zu eurem Raspberry und führt die folgenden Befehle aus.

|  |
| --- |
| **sudo apt-get update**  **sudo apt full-upgrade** |

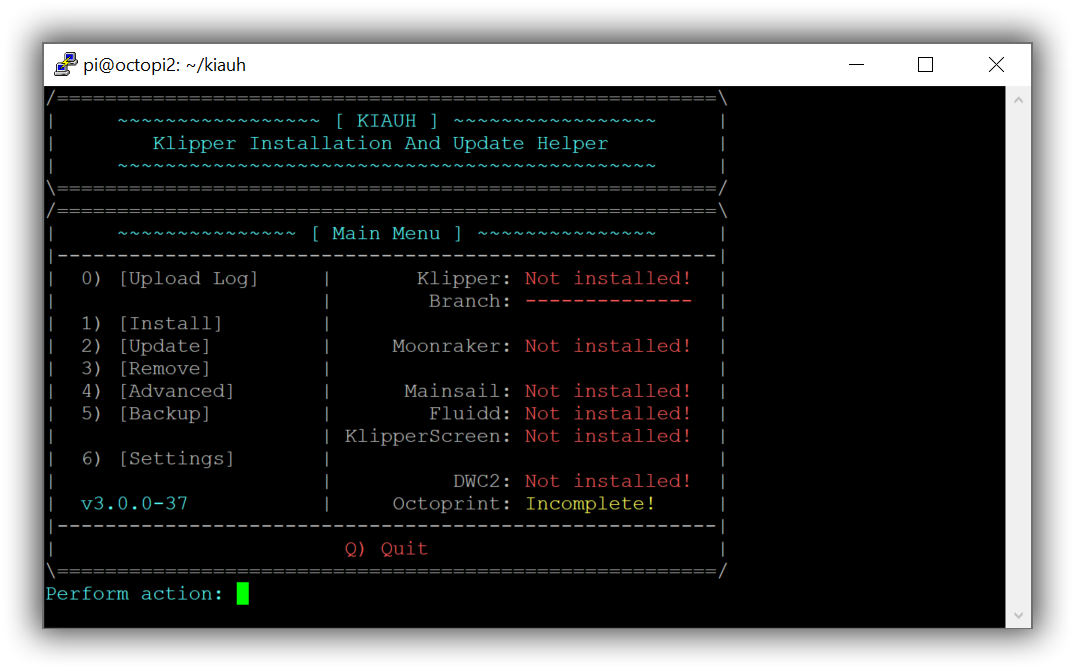
Auch OctoPrint solltet ihr ggf. auf den neusten Stand bringen.

# Installation über KIAUH

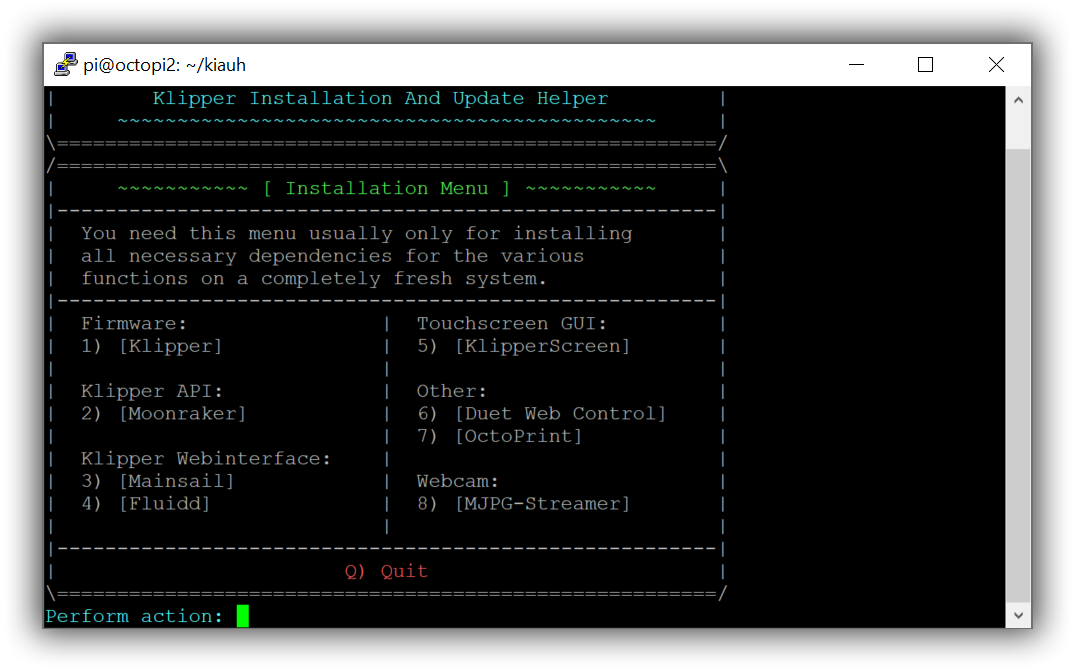
Für die Installation von Klipper nutzen wir eine Scriptsamlung namens KIAUH. Diese wird wie folgt installiert und gestartet.

|  |
| --- |
| **cd ~**  **git clone https://github.com/th33xitus/kiauh.git**  **./kiauh/kiauh.sh** |
|  |
|  |
|  |

Nach dem Start von KIAUH erscheint das folgende Menu.



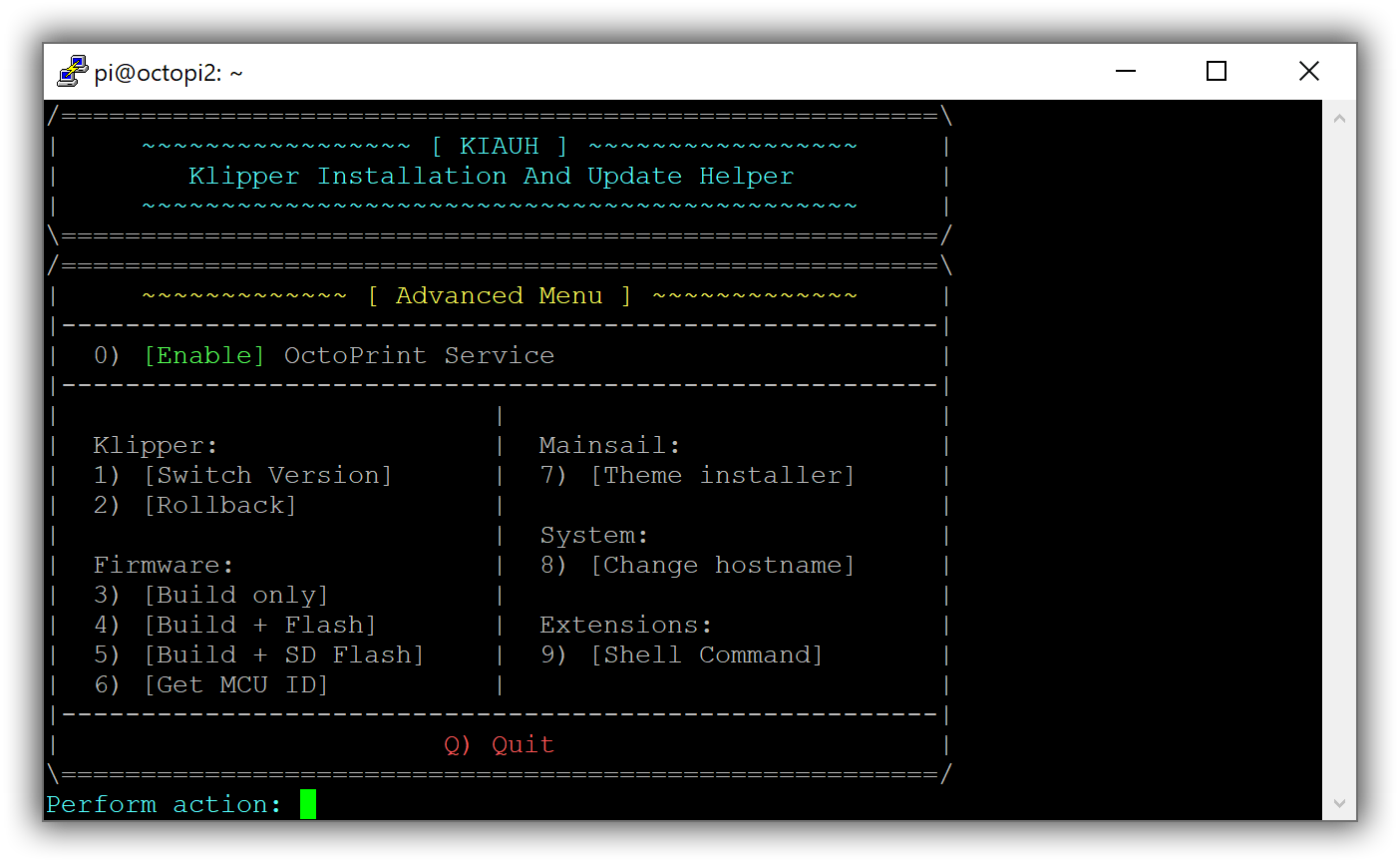
Über die Eingabe von 1 kommt ihr in das Installations Menu.



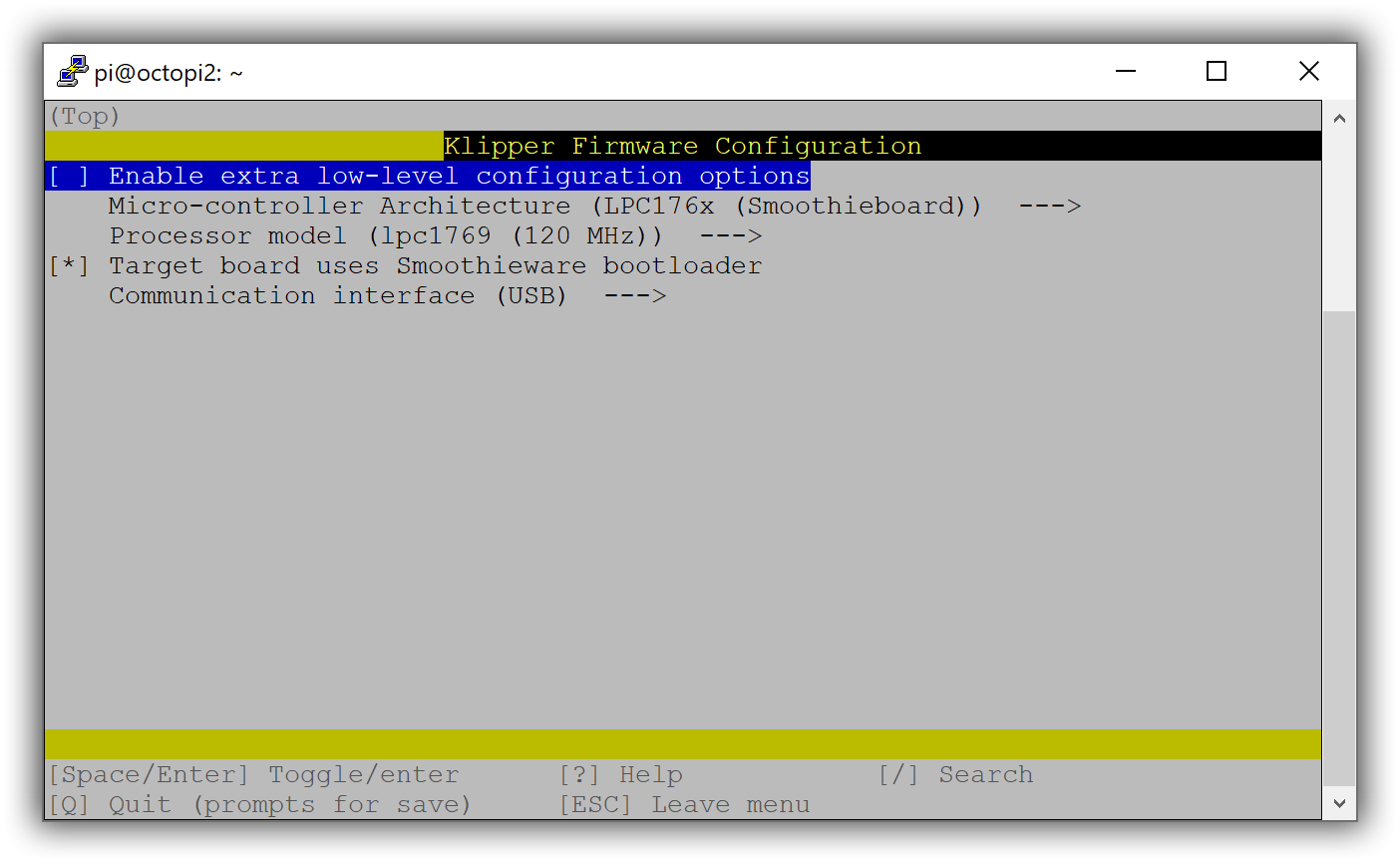
Mit einer erneuten Eingabe von 1, startet ihr die Installation von Klipper. Die nachfolgenden Eingabe Anforderungen, bestätigt ihr einfach mit Enter. Nach der Installation von Klipper geht ihr mit Q wieder auf das Startmenu.

# Erzeugen der Firmware

Vom Hauptmenü aus navigier ihr mit 4 in das Advance Menu.



Mit der Eingabe von 3 starten wir die Erzeugung der Firmware. Nun startet der „Klipper Firmware Configuration“.



Für das SKR 1.4 Turbo wählt ihr als Micro-controller „LPC176x“ aus. Zusätzlich wählt ihr für das Torbo Bord das „Processor model“, „120 MHz“ aus. Mit Q verlassen wir den Konfigurator und die Firmware wird generiert. Diese wir dann im Verzeichnis /home/pi/klipper/out. Mit 2 mahl Q verlasst ihr nun KIAUH. Mit dem Kommando

|  |
| --- |
| **ls ~/klipper/out** |

Solltet ihr auch die Datei „klipper.bin“ sehen. Ihr könntet jetzt mit WinSCP die Datei herunterladen, in „firmware.bin“ umbenennen und auf eine Micro-SD Karte kopieren. Oder geht den folgenden Weg

# RaspberryPi usbmount einrichten

Ein anderer Weg die Firmware auf einen Micro-SD Karte zu bekommen ist mit usbmount. Hierzu müsst ihr folgende schritte durchführen.

Zuerst installiert ihr Usbmount.

|  |
| --- |
| **sudo apt-get install usbmount** |

Damit ihr auch schreibend auf ein USB-Laufwerk zugreifen könnt muss die Konfigurationsdatei usbmount.conf ändern. Hierzu öffnen wir die Datei.

|  |
| --- |
| **sudo nano /etc/usbmount/usbmount.conf** |

Hier sucht ihr nach der Zeile

|  |
| --- |
| **FS\_MOUNTOPTIONS=""** |

Und ändert sie wie folgt um.

|  |
| --- |
| **FS\_MOUNTOPTIONS="-fstype=vfat,gid=users,dmask=0007,fmask=0111"** |

Anschliessend mit STRG + O speichern und mit STRG + X schliessen. Jetzt erfolgt noch eine Änderung im systemd-udevd.service. Hierzu öffnen wir die Datei ebenfalls mit nano.

|  |
| --- |
| **sudo nano /lib/systemd/system/systemd-udevd.service** |

Nun ändern wir die Zeile „PrivateMounts=yes“ wie folgt um.

|  |
| --- |
| **PrivateMounts=no** |

Anschließend wieder mit STRG + O speichern und mit STRG + X schließen. Nun startet ihr den Raspberry mit „sudo reboot“ durch.

Nachdem ihr die Micro-SD Karte mittels eines USB-Adapter in den Raspberry gesteckt habt, solltet ihr wie folgt auf die Karte zugreifen können.

|  |
| --- |
| **ls /media/usb/** |

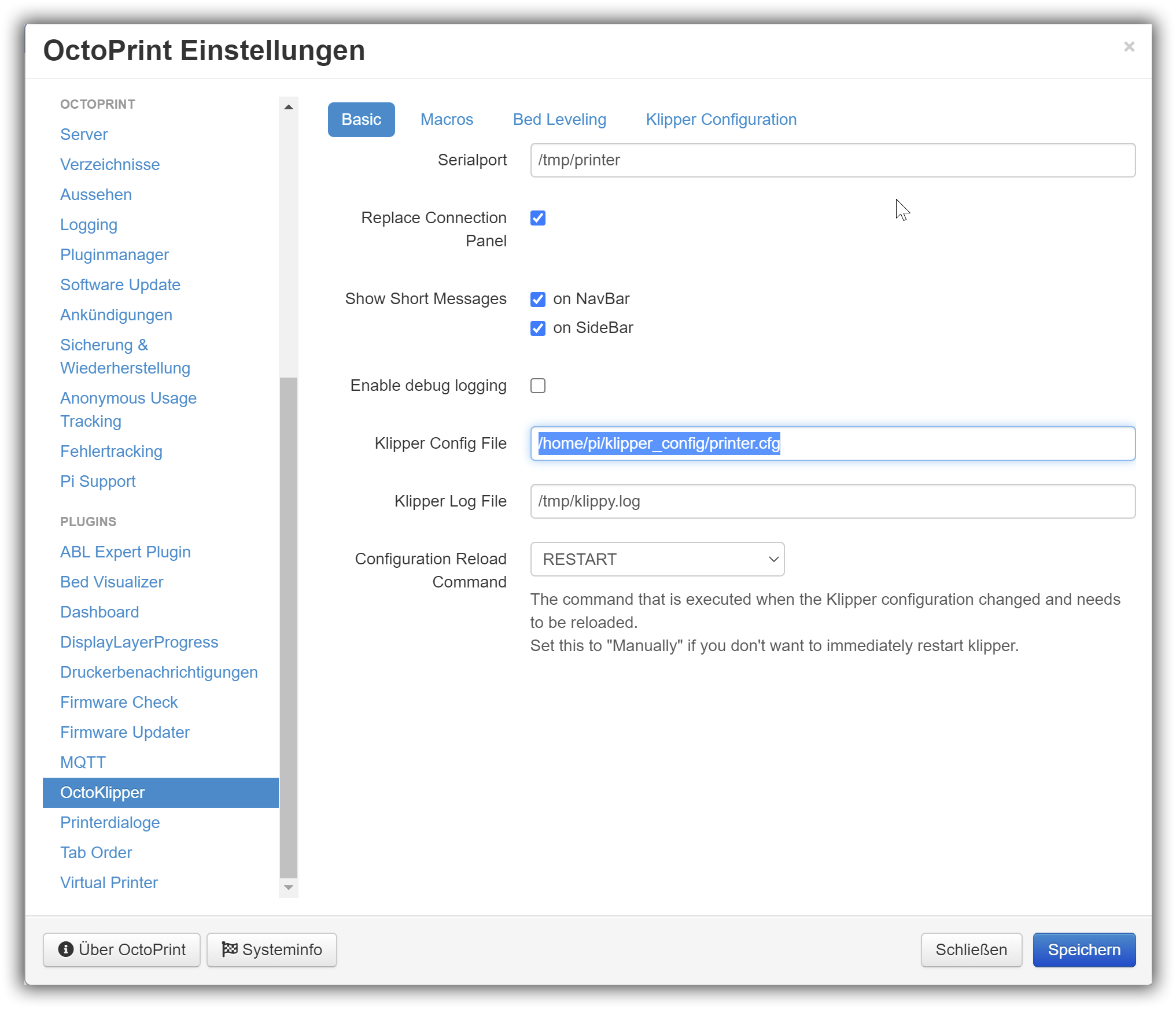
Nun können wir die Firmware auf die Karte kopieren.

|  |
| --- |
| **cp ~/klipper/out/klipper.bin /media/usb/firmware.bin** |

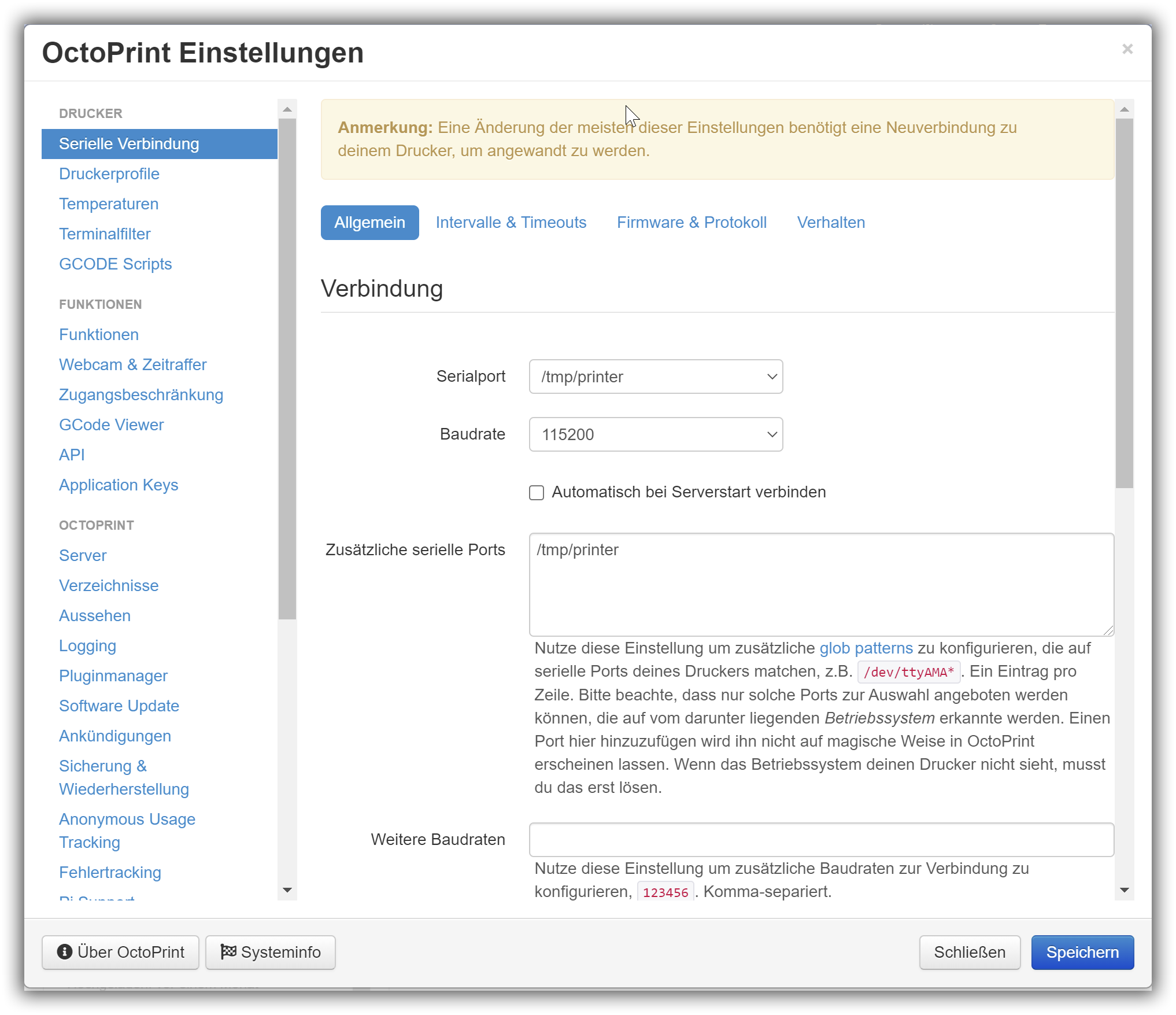
Die so präpariert Karte, könnt ihr nun in euren Drucker einfügen.

# Präparieren von Octoprint

Als nächstes muss Octoprint noch präpariert werden. Hierzu installiert ihr als erstes das Plugin OctoKlipper. Nach der Installation passen wir als erstes den Verweis auf die Konfigurationsdatei im OctoKlipper an. Dies muss nun auf die Datei „/home/pi/klipper\_config/printer.cfg“ verweisen.



Zusätzlich stellen wir das Serialport in Octoprint auf „/tmp/printer“ um.



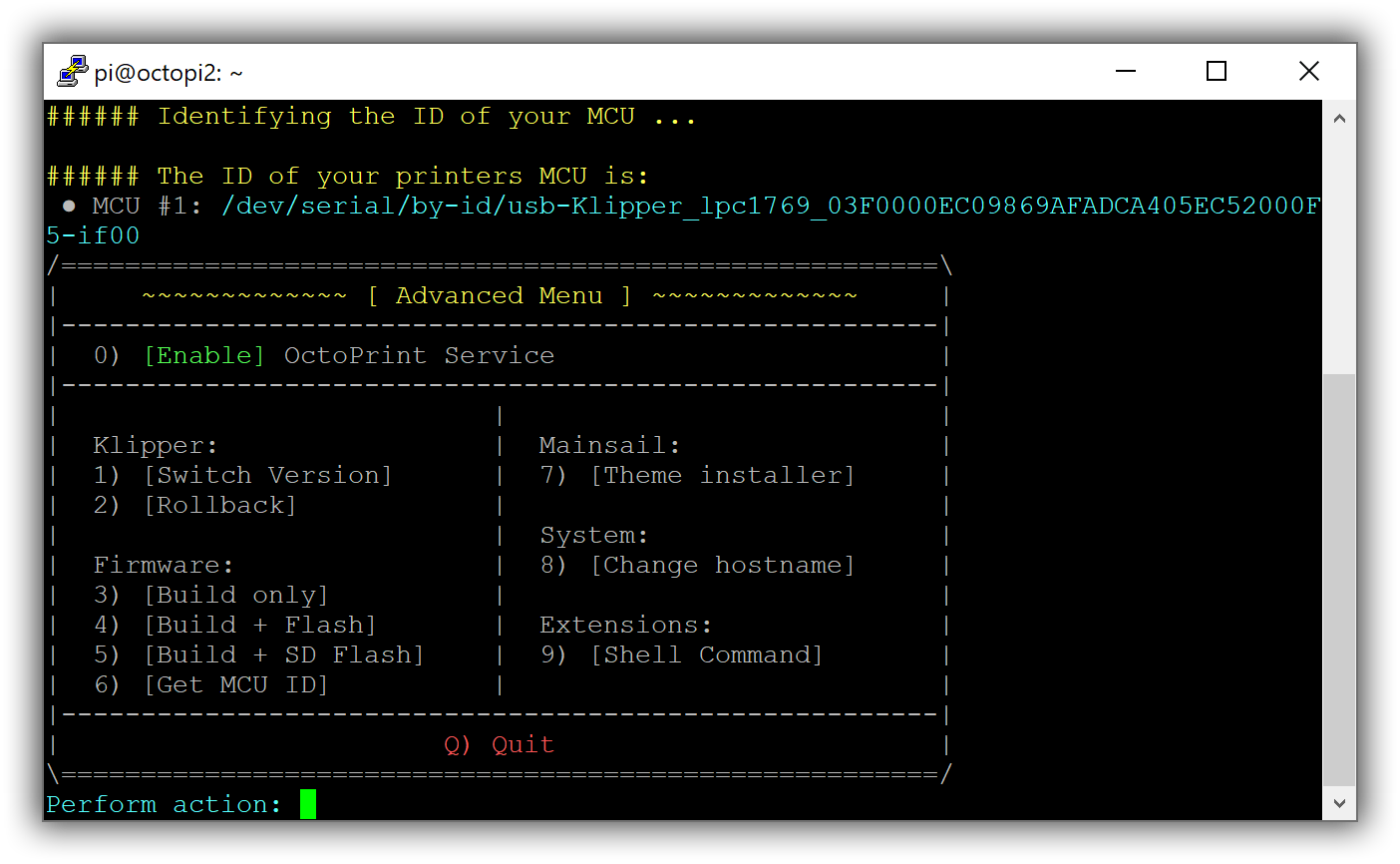
Nach dem Speichern muss auch der Raspberry erneut durchgestartet werden. Anschließend kann die Klipper Konfiguration angepasst werden.

# Aufbereiten der Klipper Konfiguration

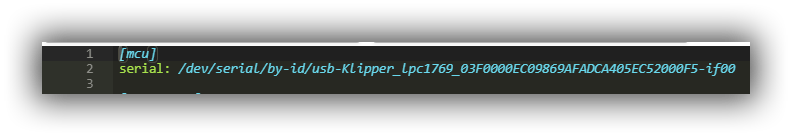
Je nach dem welches Controller Bord ihr habt könnt ihr eine Beispiel Konfigurationsdatei nutzen. Zum einen gibt es im Verzeichnis /home/pi/klipper/config viel Konfigurationsdateien die ihr als Vorlage nutzen könnt. Mit dem befehl

|  |
| --- |
| **sudo cp ~/klipper/config/generic-bigtreetech-skr-v1.4.cfg ~/klipper\_config/printer.cfg** |

Kopiert ihr z.B. eine Vorlage für ein SKR 1.4 Bord. Nun brauchen wir die ID des Serial-Port. Hierzu starten wir wieder KIAUH. Hier navigieren wir mit 4 in das Advance Menu. Dort geben wir 6 ein, um die MCU ID zu ermitteln. Wir bestätigen die nun erscheinende Frage. Danach sollte die MCU-ID auf dem Bildschirm zu sehen sein.



Diese müsst ihr Kopieren und in die printer.cfg eintragen.



Ihr findet eine printer.cfg im gleichen Verzeichnis wie dieses Dokument. Alle Stellen, die ich anpassen musste, sind mit dem String „#ThRo Sovol SV01“ gekennzeichnet. Spätesten wenn ihr nun den Raspberry erneut durchgestartet, solltet ihr auf dem Display die folgenden Informationen sehen.



# Prüfen der Schrittmotoren

Das wir jetzt mit dem Drucker kommunizieren können, prüfen wir als erstes ob wir die wichtigen Schrittmotoren ansprechen können. Mit dem Befehl „STEPPER\_BUZZ STEPPER=stepper\_x“ müssen wir sehen das sich der X-Motor bewegt. Die Namen der Schrittmotoren können wir aus der printer.cfg entnehmen. In meinem Fall sind das stepper\_x, stepper\_y, stepper\_z, stepper\_z1 und extruder. Nun können wir die Richtung prüfen. Vorrausetzung ist das ihr den folgenden Abschnitt in der printer.cfg eingetragen habt.

[force\_move]  
enable\_force\_move: True

Dann könnt ihr den z.B. den X-Motor bewegen. Hierzu gebt ihr das Kommando

FORCE\_MOVE STEPPER=stepper\_x DISTANCE=10 VELOCITY=5

In eurem Octoprint Terminal ein. Beobachtet wie sich euer Motor bewegt und prüft, ob es die richtige Richtung ist. Gegeben falls müst ihr die Zeile „dir\_pin: !P2.6“ der printer.cfg ändern. Ein „!“ bedeutet das der Pin negiert wird. Bei meinem Drucker musste ich dies für X und Y Negieren für Z und den Extruder nicht.

# Leveln des Bettes

Um das Bett erst einmal, möglichst gerade zu bekommen, benutze ich den Z-Endschalter. Hierzu ändere ich die folgenden Zeilen in der printer.cfg, wie dargestellt um.

[stepper\_z]  
endstop\_pin: P1.27  
position\_endstop: 0.0  
#endstop\_pin: probe:z\_virtual\_endstop

Hierdurch ist der BLTouch deaktivier und der Endschalter wird genutzt. Nun könnt ihr mit „G28“ euern Drucker homen. Vorausgesetzt ihr haben den folgenden Block in eurer printer.cfg

[bed\_screws]  
screw1: 47,52  
screw2: 258,52  
screw3: 258,228  
screw4: 47,228

Könnt ihr nun mit dem Kommando „BED\_SCREWS\_ADJUST“ eure Schrauben Positionen anfahren und mit der Papiermethode justieren. Nach jeder Justierung gebt ihr dann das Kommando „ADJUSTED“ ein. Dadurch wird die nächste Position angefahren. Wenn alles passt, gebt ihr für jede Position noch ein „ACCEPT“ ein. Um diese Prozedur auch direkt am Drucker durchführen zu können habe ich das Menu erweitert. Hierzu navigiert ihr im Display zu „Setup -> Calibration -> Bed Leveling“.

# Z-Offset ermitteln

Nun wollen wir den Z-Offset des BLTouch ermitteln. Hier ändere ich die folgenden Zeilen in der printer.cfg, wie dargestellt um.

[stepper\_z]  
#endstop\_pin: P1.27  
#position\_endstop: 0.0  
endstop\_pin: probe:z\_virtual\_endstop

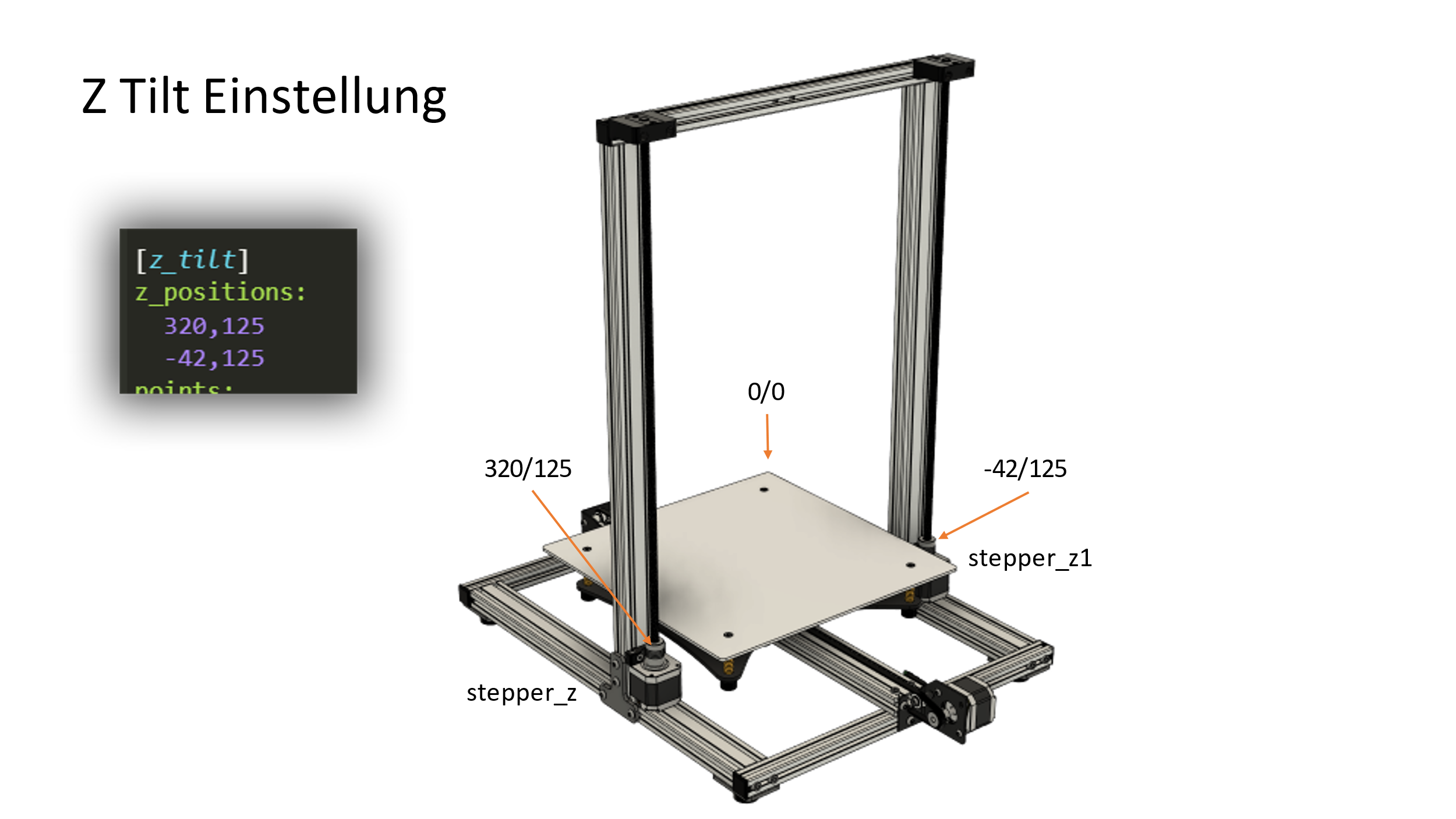
Hierdurch ist der BLTouch wieder aktiv. Wir starten nun die Kalibrierung in dem wir erst ein „G28“ und dann ein „PROBE\_CALIBRATE“ Kommando eingeben. Mit „TESTZ Z=-0.1“ fahren wir die Nozzel näher ans Bett. Diesen Befehl wieder holen wir so oft, bis das untergelegte Papier sich noch gerade so bewegen lässt. Wenn es passt, geben wir „ACCEPT“ ein und die Position wird angezeigt. Um diese in die printer.cfg zu übernehmen geben wir „SAVE\_CONFIG“ ein. Um diese Prozedur auch direkt am Drucker durchführen zu können habe ich das Menu erweitert. Hierzu navigiert ihr im Display zu „Setup -> Calibration -> Z offset cal.“.

# Z-Tilt justieren

Was unter Marlin als „Z Steppers Auto-Alignment“ bekannt ist gibt es natürlich auch unter Klipper. Hierzu müsst ihr den folgenden Block geplegt haben.

[z\_tilt]  
z\_positions:  
 320,125 # stepper\_z  
 -42,125 # stepper\_z1  
points:  
 10,125  
 240,125  
speed: 100  
horizontal\_move\_z: 10  
retries: 5  
retry\_tolerance: 0.05

Um die Werte der “z\_positions” besser zu verstehen habe ich das folgenden Bild angelegt.



Die Werte geben die theoretische Position an, die die Nozzel anfahren müsste, um über der jeweiligen Z-Spindel zu stehen. Um die Justierung vergleichbar mit Marlin durchführen zu können habe ich ein Makro für den G-Code G34 angelegt.

[gcode\_macro G34]  
gcode:  
 {% if printer.toolhead.homed\_axes != "xyz" %}  
 G28  
 {% endif %}  
 Z\_TILT\_ADJUST

Hierdurch wird vor dem Kommando „Z\_TILT\_ADJUST“ ggf. ein „G28“ ausgeführt.

# E-Steps justieren

Das Kalibrieren des Extruders ist in Klipper etwas anders als in Marlin. Ich beschreibe hier das Vorgehen in Klipper 0.9.0, welches „rotation\_distance“ zur Kalibrierung nutzt.

Als Basis für den block [extruder], mischen die muster Konfiguration des SV01 und der BIGTREETECH SKR V1.4

|  |  |  |
| --- | --- | --- |
| **Sovol Sv01** | **BIGTREETECH SKR V1.4** | **Eigene** |
| step\_pin: PA4  dir\_pin: !PA6  enable\_pin: !PA2  microsteps: 16  rotation\_distance: 7.680  nozzle\_diameter: 0.400  filament\_diameter: 1.750  heater\_pin: PB4  sensor\_type: EPCOS 100K B57560G104F  sensor\_pin: PK5  control: pid  pid\_Kp: 31.147  pid\_Ki: 2.076  pid\_Kd: 116.803  min\_temp: 0  max\_temp: 265 | step\_pin: P2.13  dir\_pin: !P0.11  enable\_pin: !P2.12  microsteps: 16  rotation\_distance: 33.500  nozzle\_diameter: 0.400  filament\_diameter: 1.750  heater\_pin: P2.7  sensor\_type: EPCOS 100K B57560G104F  sensor\_pin: P0.24  control: pid  pid\_Kp: 22.2  pid\_Ki: 1.08  pid\_Kd: 114  min\_temp: 0  max\_temp: 260 | step\_pin: P2.13  dir\_pin: P0.11  enable\_pin: !P2.12  microsteps: 16  rotation\_distance: 7.680  nozzle\_diameter: 0.400  filament\_diameter: 1.750  full\_steps\_per\_rotation: 400  heater\_pin: P2.7  sensor\_type: ATC Semitec 104GT-2  sensor\_pin: P0.24  control: pid  min\_temp: 0  max\_temp: 300  min\_extrude\_temp: 140  max\_extrude\_only\_distance: 150 |

Einige Anpassungen an die Eigene Konfiguration können wir direkt einfügen. Beim „dir\_pin“ muss in meinem Fall die Negierung der Richtung entfallen. Da der „Titan Aero“ einen 0.9 Grad Schrittmotor hat, füge ich die Zeile „full\_steps\_per\_rotation: 400“ ein. Die Default Einstellung für einen 1.8 Grad Schrittmotor währe „full\_steps\_per\_rotation: 200“. Da der „Titan Aero“ hat außerdem einen anderen Thermistor, weshalb ich hier „ATC Semitec 104GT-2“ eingetragen habe. Die PID werte können wir weglassen, weil wir sie später ermitteln. Ich setze die „max\_temp“ auf 300, weil der „Titan Aero“ das schaffen sollte. Ich habe auch die „min\_extrude\_temp“, von Default 170 auf 140 heruntergesetzt. Der Wert für „max\_extrude\_only\_distance“ erlauben mir mehr als 50mm mit einem Befehl zu extrudieren. Der Default erlaubt nur 50mm, was für diese Kalibrierung störend währe.

Wie bei Marlin auch, wir einen definierte länge Filament Extrudieren und gemessen. Hierzu gibt es zwei Vorgehensweisen. Die erste ist wir erhitzende Nozzel auf einen passenden Wert für unser Filament und beginnen mit der Prozedur. Oder wir entfernen die Nozzel um das Filament kalt zu extrudieren. Dann müssen wir den Block [extruder] um die Zeile „min\_extrude\_temp: 0“ erweitern oder anpassen. G1 Z100

1. Alten Wert von „rotation\_distance“ in Formel eintragen
2. Relative Extrudieren einstellen mit M83
3. Als nächstes markieren wir uns, oberhalb des Extruders 120mm entfernt, das Filament.
4. Nun extrudieren wir 100mm mit G1 E100 F100
5. Restlänge Messen. Und in Formel eintragen

Die Formel sieht vereinfacht so aus.

Neuer Wert = Alter Wert \* (120 – Restlänge) / 100

# Pressure Advance

„Pressure Advance“ das in Marlin “Linear Advance” heißt, ermöglicht uns die Kanten scharfer zu drucken. Klipper hat hierfür eine sehr einfache Methode. Hierzu laden wir zuerst die folgende STL Datei herunter. <https://github.com/KevinOConnor/klipper/blob/master/docs/prints/square_tower.stl>

Diese Datei laden wir in unseren Slicer z.B. Cura. Folgende Einstellungen sollten wir anpassen.

|  |  |
| --- | --- |
| Parameter |  |
| Schichtdicke: 0.3 | ~75% des Nozzel Durchmessers |
| Fülldichte: 0 |  |
| Druckgeschwindigkeit: 100 |  |
| Füllgeschwindigkeit: 100 |  |
| Wandgeschwindigkeit: 100 |  |
| Geschwindigkeit Außenwand:100 |  |
| Geschwindigkeit Innenwand:100 |  |
| Bewegungsgeschwindigkeit:300 |  |

Mit dieser Einstellung Slicen wir das und sicher den GCode. Dieser kann nun in Octoprint hochgeladen werden.