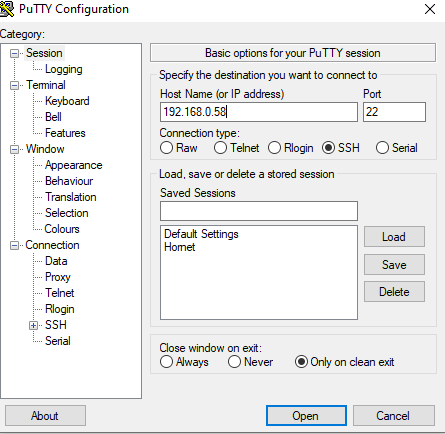
Mit dem Programm Putty und der IP-Adresse des Raspberry Pi auf dem PI einloggen.

Es wird vorausgesetzt, dass das Pi Betriebssystem Raspberry Pi OS Lite in der aktuellen Version bereits installiert wurde und der Pi mittels SSH erreichbar ist. Kurze Anleitung folgt am Ende.

Sollte die IP-Adresse nicht bekannt sein, so muss im Router geschaut werden, welche er zugewiesen bekommen hat, da der Pi mit der Standardeinstellung DHCP, sprich eine automatische IP-Adressvergabe unterstützt. Er sollte im Heimrouter mit dem Namen „raspberrypi“ zu finden sein.

Downloadlink für Putty

[Putty Download](https://www.putty.org/)



Es werden nun der Benutzernamen und ein Passwort abgefragt

User: **pi** / Passwort: **raspberry** (im Grundzustand – sollte später unbedingt geändert werden, mehr dazu gleich)

Bevor wir mit der Installation loslegen, wollen wir erst einmal den Pi auf den aktuellsten Stand heben. Das geht über folgende zwei Befehle, die nacheinander eingegeben werden. Dieser Vorgang kann etwas dauern.

sudo apt-get update

sudo apt-get upgrade

Was machen die beiden Befehle?

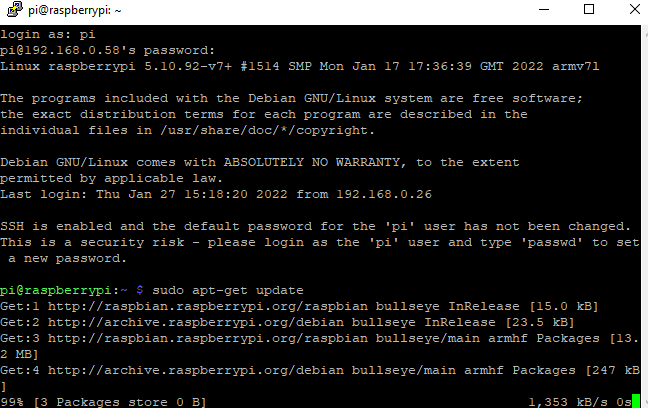
**apt-get update** aktualisiert eure Paketlisten, die Informationen enthalten welche Programmversionen aktuell verfügbar sind und wie die Anhängigkeiten der einzelnen Pakete zueinander sind.

**apt-get upgrade** lädt die neuen Programmversionen herunter und installiert diese. Aber Vorsicht, es ist zwingend vorher der Befehl „**apt-get update**“ auszuführen, damit die aktuellen Paketlisten vorliegen.

Wer noch wissen möchte, was der Befehl „sudo“ davor anstellt – dafür Zitiere ich jetzt mal folgende [Quelle](https://wiki.ubuntuusers.de/sudo/)

*Der Befehl sudo kann Programmaufrufen vorangestellt werden. Er ermöglicht berechtigten Benutzern, das Programm im Namen und mit den Rechten eines anderen Benutzers auszuführen. Beispielsweise um Aufgaben auszuführen, die* [*Administratoren*](https://wiki.ubuntuusers.de/Administrator/) *vorbehalten sind:*

* *Programme installieren (sudo apt-get install ...)*
* *Systemkonfigurationen ändern (sudo nano /etc/fstab)*



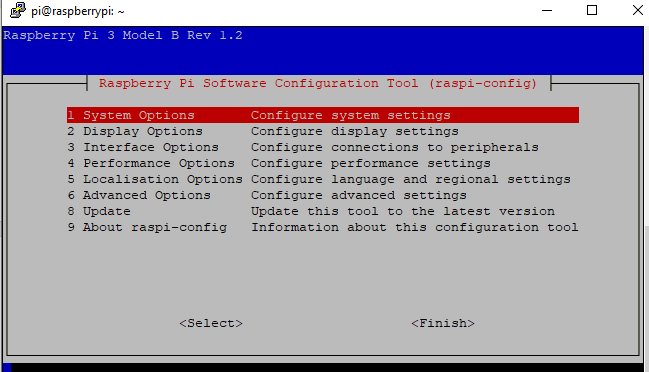
**Als kleiner Tipp: Man kann die Befehle aus diesem PDF mittels der Tastenkombination „STRG+C“ kopieren und in Putty mit der rechten Maustaste einfügen.**

DerRaspberry Pi ist nun aktuell, aber eines fehlt noch – wir können uns ja immer noch mit dem Standardpasswort anmelden. Das werden wir nun ändern 😉 – mit dem Befehl

sudo raspi-config

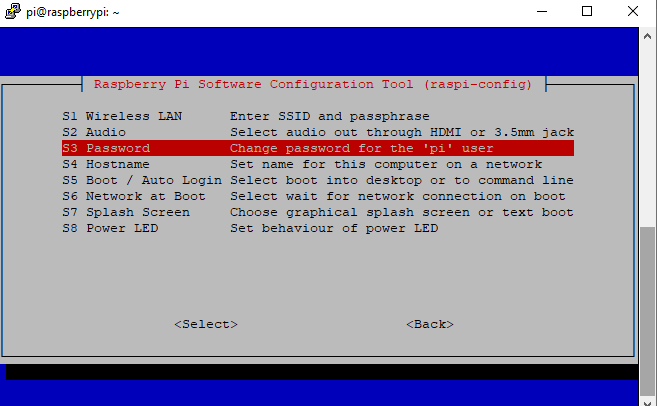
Es erscheint ein Menü, in welchen wir nun die Nummer 1 „System Options“ wählen.

Die Navigation erfolgt mittels Pfeiltasten



Im nächsten Fenster wählen wir „S3 Passwort“ und folgen der Anleitung.

Zu beachten ist, dass die Passwort-Eingabe verdeckt erfolgt. Bitte nicht wundern, dass ihr nicht schreibt 😉



Wir können hier noch mehrere Einstellungen vornehmen, auf die ich in dieser Anleitung nicht eingehe, aber ihr könnt sie bei Bedarf gerne hier nachlesen. [Link](https://www.elektronik-kompendium.de/sites/raspberry-pi/1906291.htm).

*Änderung vom Hostnamen lasst bitte vorerst sein, da mir aufgefallen ist, dass die nachfolgenden Schritte mit kiauh dann mit Fehlern beendet werden*

Der Pi ist aktuell und das Passwort wurde geändert – Super – jetzt beginnt der Spaß erst

Wir arbeiten mit kiauh weiter

Was ist und macht kiauh – mal kurz zusammengefasst – Es nimmt euch Arbeit ab, in dem es Schritte automatisiert ausführt und darüber hinaus noch ein kleines Menü bietet

[Kiauh](https://github.com/th33xitus/kiauh)

*This script acts as a helping hand for you to get set up in a fast and comfortable way.****This does not mean, it will relieve you of using your brain.exe! 🧠*** *Please also always pay attention to the individual component repositories (all linked below).  
Feel free to give it a try. If you have suggestions or encounter any problems, please report them.*

Merke: Brain.exe darf benutzt werden 😉

Fahren wir fort mit den Befehlen:

sudo apt-get install git -y

cd ~

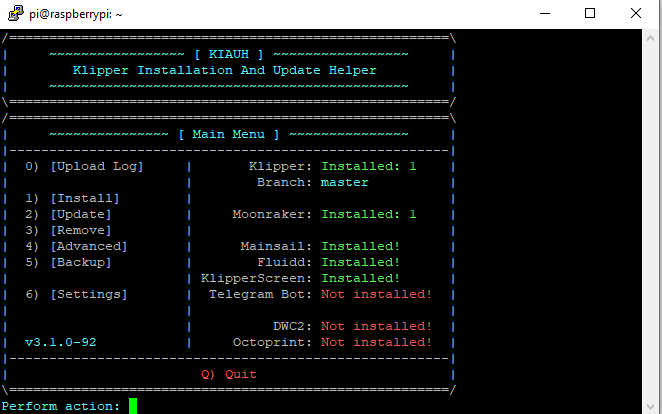
git clone https://github.com/th33xitus/kiauh.git

./kiauh/kiauh.sh

Was passiert dort? Vereinfacht gesagt wir installieren git, wir springen ins „Homeverzeichnis“, laden ein Skript und führen es zum Schluss aus.

Wer noch ein wenig nachlesen möchte was dieses „git“ ist (nein, es ist nicht Github damit gemeint) der kann es unter folgendem Link nachschlagen. [GIT](https://www.raspberry-pi-geek.de/ausgaben/rpg/2019/10/den-raspberry-pi-als-git-server-nutzen/)

Wenn es erfolgreich durchlaufen wurde, solltet ihr jetzt folgendes Menü sehen.



Bei euch steht auf der rechten Seite natürlich überall Not installed!

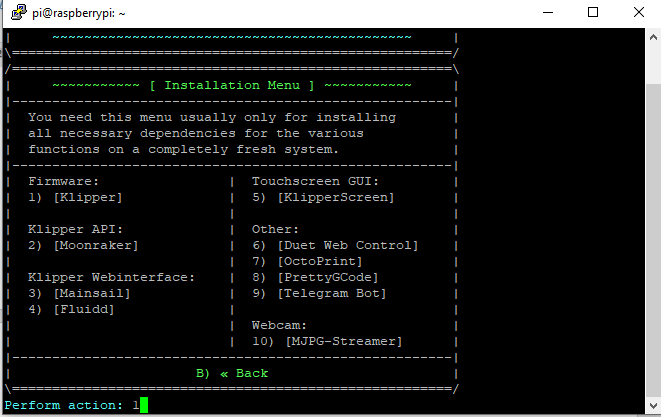
Soweit, so gut. Wir sind aber noch lange nicht fertig

Jetzt liegt es an euch wie ihr vorgehen wollt. Erst die Installation oder die MCU ID auslesen und flashen? Ich fange immer mit den Installationen an

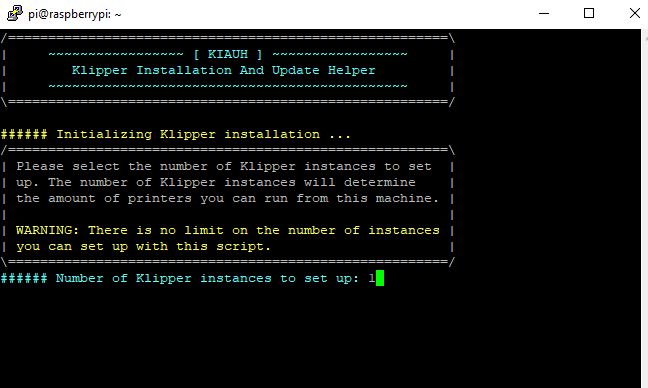
Die Navigation unter kiauh erfolgt mit den Ziffern 0-6 und den Buchstaben „Q“ für Quit und „B“ für Back

Wir wählen unter *Perfom action:* die „1“ , schließlich wollen wir ja etwas installieren

und im nachfolgenden Menü ebenfalls wieder die „1“ um Klipper zu installieren



Wir werden anschließend nach der Menge der Instanzen gefragt, pro Drucker eine. Ich habe an meinem Pi nur einen Drucker hängen, also gebe ich „1“ ein und bestätige bei der nächsten Eingabeaufforderung mit „y“



Glückwunsch! Klipper wurde erfolgreich installiert.

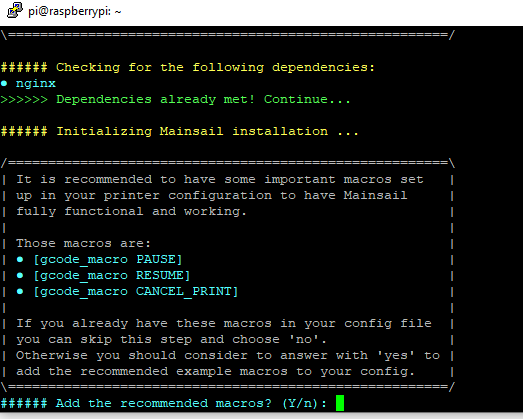
Wir brauchen aber ebenfalls die Weboberfläche bzw. -flächen Mainsail und/oder Fluidd und die Moonraker API, das Bindeglied zwischen Klipper und Mainsail/Fluidd

Wir gehen genau so vor, wie bei der Klipper Installation nur wählen dann jeweils einmal die „2“ und abschließend die „3“ und/oder „4“ für Mainsail oder Fluidd. Ich habe beides installiert

Auch hier werden wir natürlich wieder nach den Instanzen gefragt und müssen diese mit „y“ bestätigen.

**Auf Octoprint gehe ich hier überhaupt nicht ein, da mit Mainsail und Fluidd zwei, auf Klipper zugeschnittene Weboberflächen, oder besser gesagt Human Maschine Interfaces (HMI) existieren.**

Zu beachten ist ebenfalls, dass sowohl bei der Mainsail, als auch Fluidd -Installation gefragt wird, ob ihr die empfohlenen Makros gleich mit installieren wollt. Gleiches für den MJPG-Streamer, der benötigt wird, wenn ihr eine Cam an eurem Raspberry betreiben wollt. Man kann diesen aber auch über das Install-Menü nachträglich installieren.



Ich installiere die Makros nicht, da ich meine makro.cfg z.B selber erstelle. Es schadet aber auch nicht diese zu installieren. **Wichtig**! Nicht einfach blind nutzen, sondern verstehen was die Makros machen, z.B welche Positionen im Pause Makro angefahren werden sollen und ggfs. auf euren Drucker anpassen.

Noch zu erwähnen wäre: **Mainsail und Fluidd parallel betreiben.**

Dies ist natürlich möglich. Da Mainsail und Fluidd gerne den gleichen Port (80) benutzen und dies nicht unbedingt möglich ist, werdet ihr bei der Installation darauf hingewiesen, doch einen anderen validen Port zu benutzen. In meinem Falle wähle ich für Fluidd den Port 85 und kann dann folgendermaßen, als Bsp. mittels IP (Zugriffe sind auch über Namen möglich) und der Portangabe auf beide Oberflächen zugreifen. Dies sieht in meinem Beispiel dann wie folgt aus

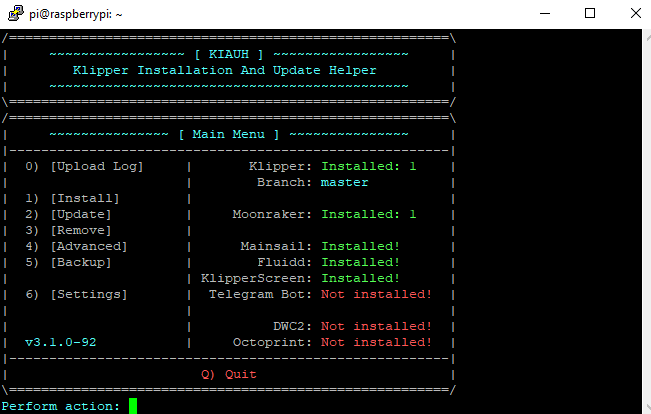
Mainsail: http://192.168.0.58

Fluidd: http://192.168.0.58**:85**

Hat bis hierher alles geklappt, sollte kiauh jetzt folgendes anzeigen. ( Bei mir wurde noch entgegen der Anleitung KlipperScreen installiert, da gehe ich aber hier nicht genau drauf ein)

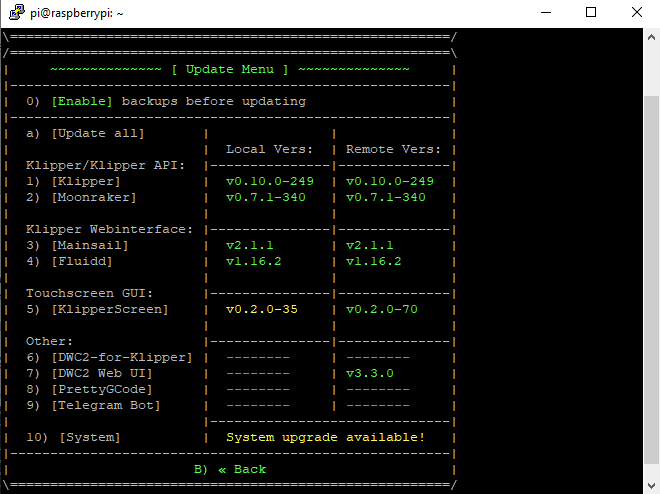
Wer nachlesen möchte wie man KlipperScreen einrichtet, sollte folgenden Link betrachten.

[KlipperScreen](https://klipperscreen.readthedocs.io/en/latest/)



Mittels der Eingabe „2“ -**Update-** könnt ihr euch auch direkt mal die installierten und verfügbaren Versionen anschauen und bei Bedarf aktualisieren. (geht aber übersichtlicher über die Weboberflächen von Mainsail oder Fluidd)

Beachte: Bei mir ist die Installation etwas älter, demzufolge werden bei mir neuere verfügbare Versionen angezeigt. Nach einer Neuinstallation sollten selbstverständlich beide Seiten gleich ausschauen



Ihr könnt jetzt z.B über die Eingabe der IP-Adresse im Browser eurer Wahl, auf die entsprechende Weboberfläche zugreifen, solltet aber noch keine Verbindung zum Drucker erhalten, weil ein wichtiges Element natürlich noch fehlt. Die printer.cfg – kurz die Datei, die Klipper sagt, was, wie, wo angesteuert werden soll.

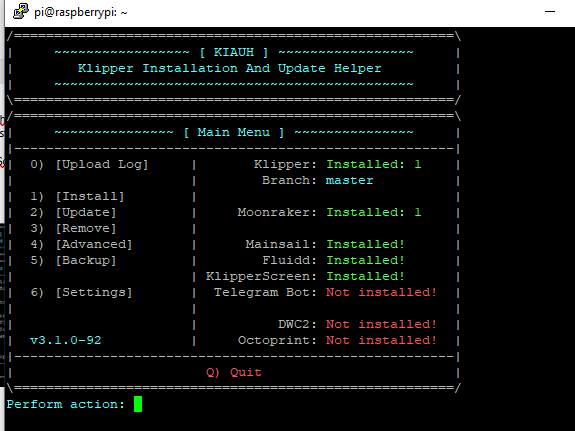
Für die printer.cfg ist es nötig

* das Mainboard eures Druckers entsprechend mit der Klipper Firmware zu flashen um
* in der printer.cfg eine Adresse (mcu id) anzugeben, über welche das Mainboard überhaupt erreichbar ist
* ein paar weitere Parameter (nur ein Paar :D ), auf die ich später kurz zu sprechen komme 😉

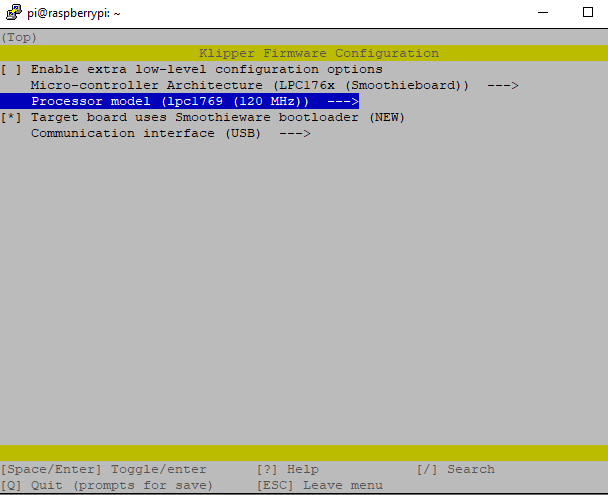
Fangen wir mal an das Mainboard mit der Klipper FW zu flashen. Es gibt hier natürlich unzählige Mainboards und ich kann nicht auf alle eingehen. Als Beispiel nutze ich ein BTT SKR 1.4 Turbo. (Fast alle BTT Board SKR mini, SKR1.2 pro, SKR 1.3, SKR 1.4, Octopus 1.1, Octopus Pro etc.) werden auf diese Weise mit der Firmware versorgt

Hier ist die Besonderheit, dass es nicht direkt über USB geflasht werden kann, sondern die Firmware auf eine SD-Karte verschoben und diese beim Starten vom Mainboard dort eingesetzt sein muss.

Alles beginnt wieder im Hauptmenü von kiauh.



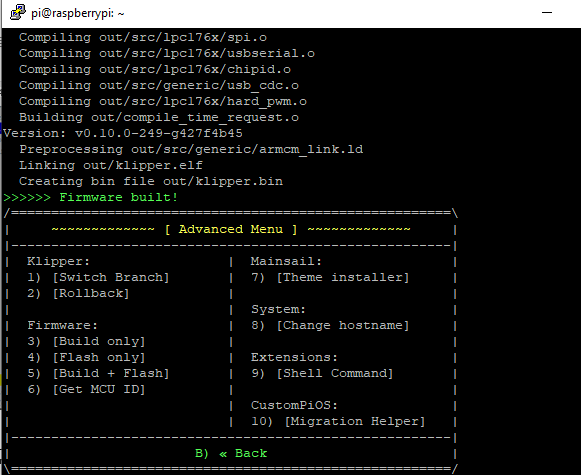
Wir wählen die „4“ Advanced und im folgenden Menü die „3“ **Build only** und schwups sind wir im untenstehenden Menü angelangt



Da ich weiß, dass mein Turbo Board einen LPC1769 Chipsatz mit 120Mhz besitzt, stelle ich diesen und den Smoothie Bootloader ein.

Wenn ich alle Einstellung getätigt habe, verlasse ich das Menü über die taste „Q“ und bestätige das Speichern mit „y“.

Nun wird automatisch die Firmware erstellt. Wenn alles geklappt hat, sollte es so ausschauen

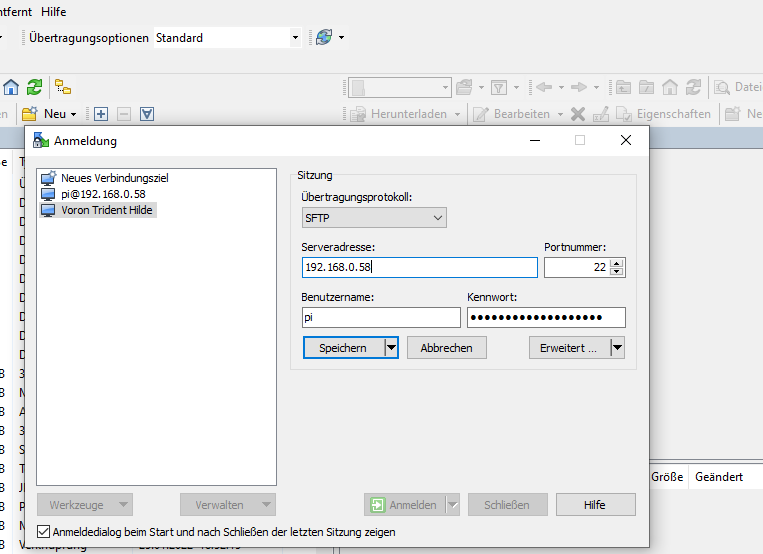


Jetzt müssen wir ja irgendwie an die gerade erstellte Firmware kommen. Dafür nutze ich das Programm WinSCP

[Download WinSCP](https://winscp.net/eng/download.php)

WinSCP ist ein FTP Client für Windows, um es mal ganz einfach auszudrücken. Damit ist es möglich die Dateien und Strukturen meines Raspberry Pi wie im Windows Explorer anzeigen zu lassen

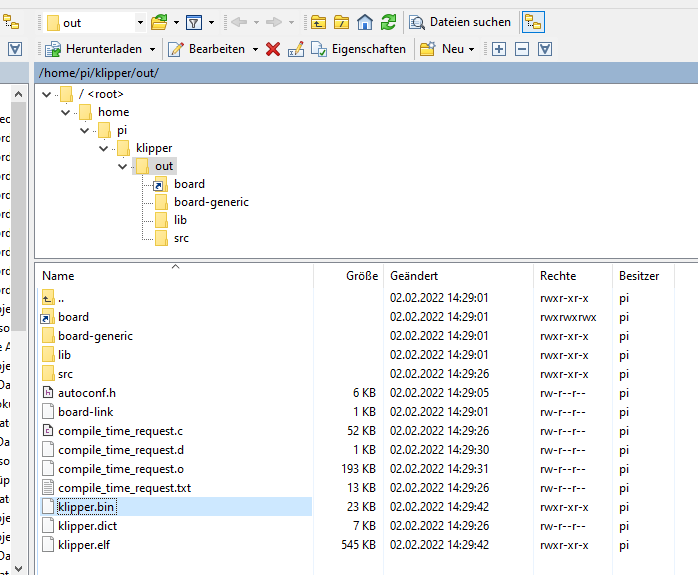
Ich wähle mich auch wieder mit der IP-Adresse von Raspberry Pi ein und gebe meinen Nutzernamen: pi, sowie mein (hoffentlich geändertes) Passwort ein. Übertragungsprotokoll steht in meinem Falle auf SFTP



Wir navigieren zum Pfad /home/pi/Klipper/out/ und sehen dort unsere Datei klipper.bin.

Diese, so mache ich es immer, ziehe ich mir bei etwas verkleinertem Fenster von WinSCP mittels Drag&Drop auf meinen Desktop und benenne diese in **firmware.bin** um.

Da WinSCP einen Datei Explorer enthält, kann ich ebenfalls auf der linken Seite meinen entsprechenden lokalen Windows Ordner auswählen und die klipper.bin Datei direkt dorthin kopieren/transferieren.



Die umbenannte Datei mit dem Namen **firmware.bin** wird nun auf eine leere, in FAT32 formatierte Micro SD-Karte kopiert.

Jetzt heißt es Drucker ausschalten, wenn der Pi schon mit eurem Board verbunden ist, bitte die USB-Verbindung ebenfalls trennen, da manche Board ggfs. auch über diese mit Spannung versorgt werden.

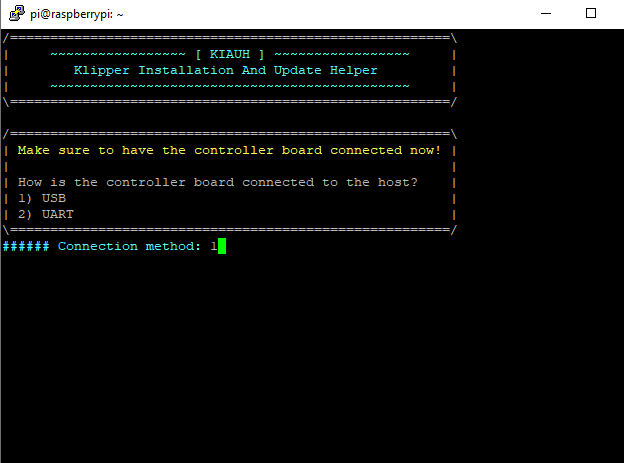
Alles aus? Ok, dann Micro SD-Karte ins Board stecken, Spannung wieder einschalten und um sicher zu gehen 1min warten. (geht wesentlich schneller)

Ihr wollt kontrollieren, ob die Firmware nun tatsächlich auf das Board gespielt wurde?

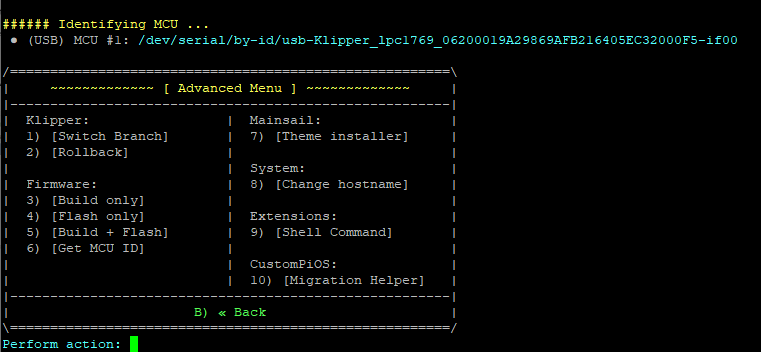
SD-Karte wieder in den Rechner stecken und kontrollieren ab die Datei **firmware.bin** in **firmware.CUR** umbenannt wurde. Ist dies der Fall – Glückwunsch die FW wurde erfolgreich geflasht.

Doch dies ist nur ein Teil – noch könnt ihr euch nicht über Mainsail auf das Mainboard schalten. Dafür fehlt noch ein kleiner Schritt. Die ID vom Mainboard

Fix zurück zu kiauh – „4“ Advanced Menü und mit „6“ Get MCU ID asuwählen



Da unser Board mittels USB angeschlossen ist wählen wir die „1“ und bekommen unsere ID angezeigt



Bitte kopieren und bspw. Im Editor speichern!

Was uns jetzt noch fehlt ist eine printer.cfg . Auch hier gibt es wieder unzählige und ich werde nur auf eine eingehen, das Vorgehen ist eigentlich immer gleich.

Viele sind bereits als Beispiel vorhanden und können frei angepasst werden.

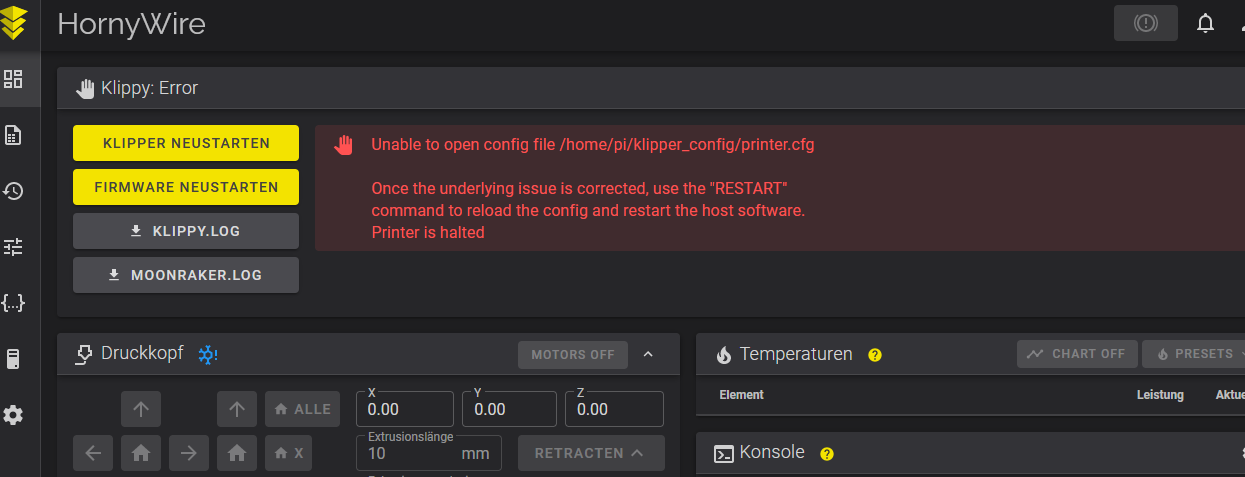
Erste Anlaufstelle ist Klipper Github unter folgenden Link

[Klipper Config](https://github.com/Klipper3d/klipper/tree/master/config)

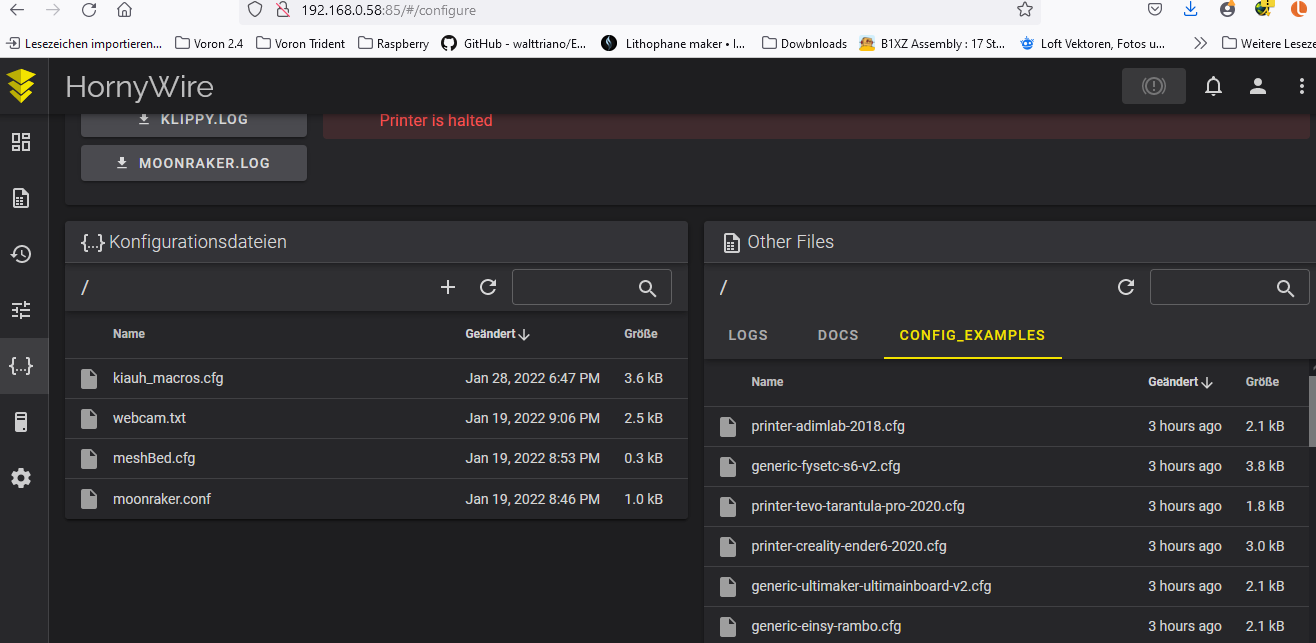
oder in der Google-Suche „klipper+euer board“ eingeben. Dort wird man eigentlich immer fündig

Letzter Schritt

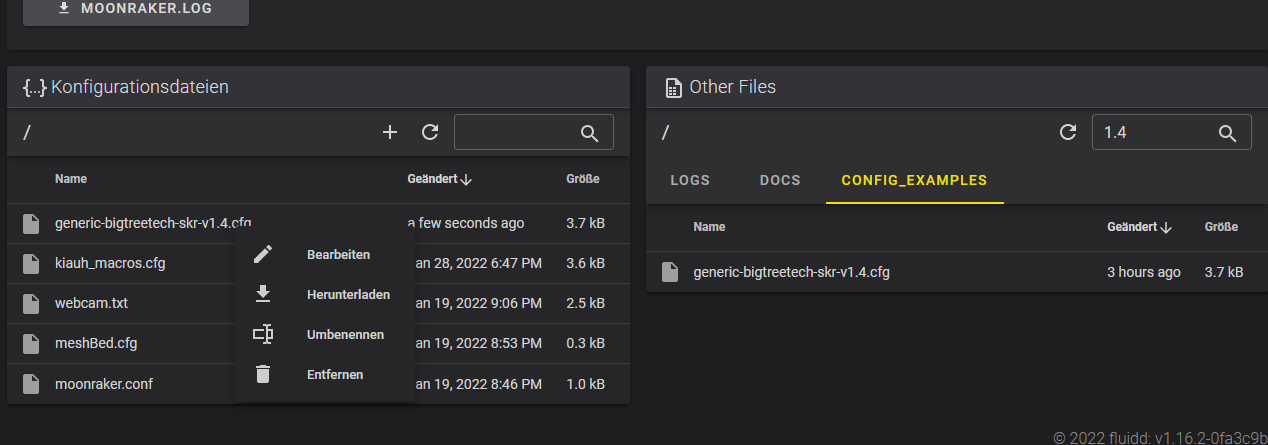
Wir loggen uns auf unsere Weboberfläche Mainsail/Fluidd ein



Es wird direkt mal gemeckert, dass keine printer.cfg vorhanden ist. Dies ist nur im meinem Falle so, da ich diese für die Anleitung gelöscht habe. Ihr findet bereits eine printer.cfg vor, die allerdings noch nicht viel enthält. Also fix auf den oben genannten Klipper-Config Link geklickt und die passende Config für unser SKR 1.4 Turbo rausgesucht und hochgeladen…. Stopp alles möglich, aber das sollte doch einfacher gehen…. Tuts ja auch. Die Config Examples wurden bereits installiert und können über Mainsail oder Fluidd eingesehen werden. Hier das Beispiel wie man über Fluidd an die Configs gelangt.



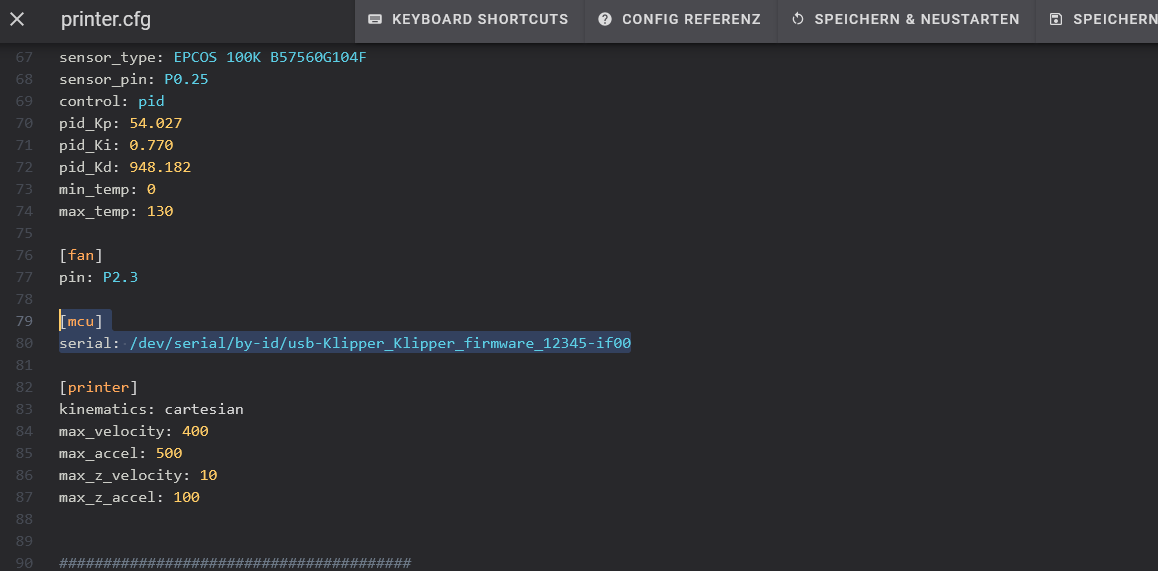
In die Suchmaske tippen wir nun 1.4 ein und sehen die generic Config vom 1.4 Board. Drag&Drop wäre zu schön hat bei mir aber nicht geklappt. Also einmal auf der rechten Seite die Datei mit der linken Maustaste anklicken und herunterladen auswählen. Auf der linken Seite das + anwählen und die gerade herunter geladene Datei hochladen. Abschließend diese mit der linken Maustaste auswählen und in printer.cfg umbenennen. Dieses Vorgehen ist nur eine Möglichkeit von vielen.



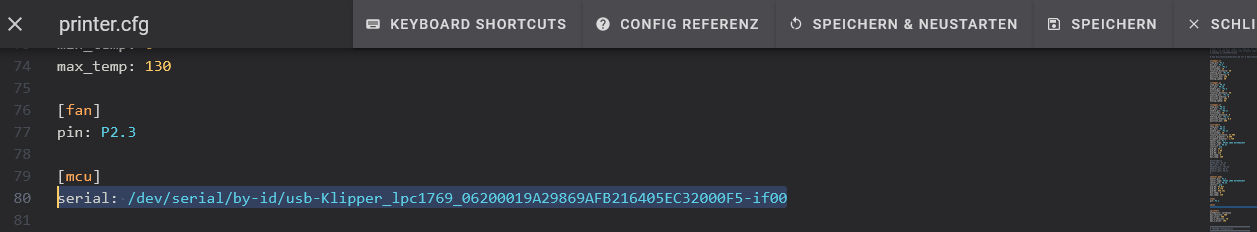
Die printer.cfg Datei kann direkt unter Fluidd bearbeitet werden. Linke Maustaste auf die Datei und bearbeiten auswählen.

Abschließend suchen wir uns die Sektion [mcu]

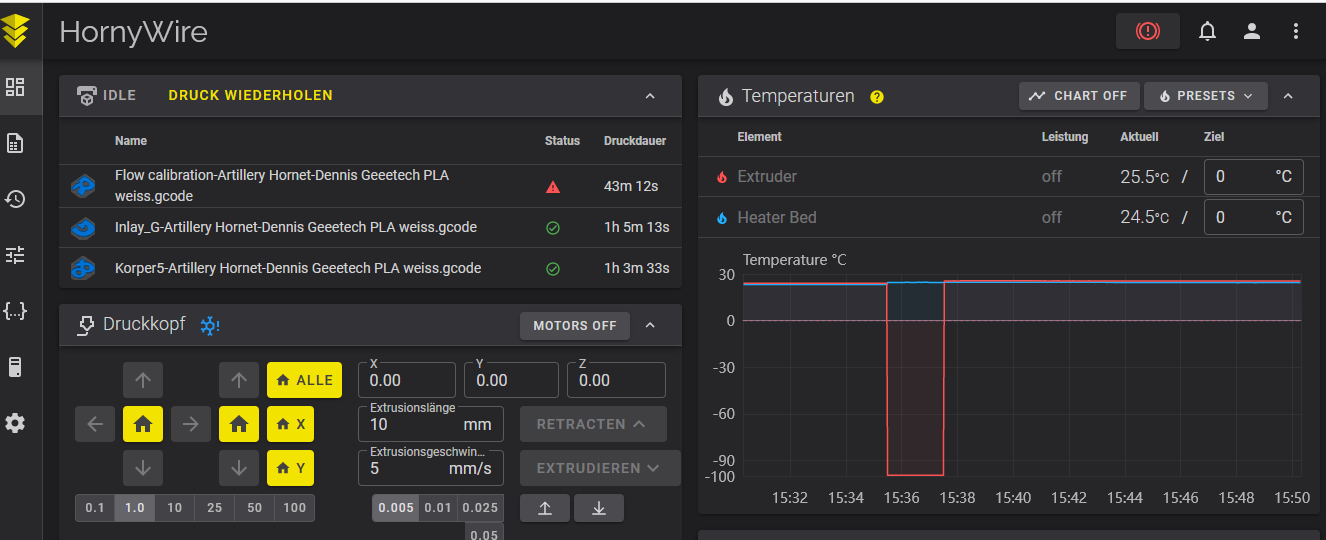
Dort ist bereits eine ID eingetragen, aber natürlich nicht die vorhin ermittelte von unserem Mainboard.



Hoffe ihr habt die ID im Editor gespeichert und könnt diese kopieren und in der printer.cfg eintragen. Wenn die kiauh macros vorhin ebenfalls installiert wurden, vergesst bitte nicht diese mit der Befehlszeile [include kiauh\_macros.cfg] in die printer.cfg zu integrieren. Wenn das erledigt ist klickt auf „Speichern & Neustarten“



Herzlichen Glückwunsch. Euer Drucker ist mit Klipper ausgestattet



**ABER!!! Dies ist nur ein kleiner Teil der printer.cfg. Da es nur ein Config Beispiel war, heißt es nicht, dass eure Belegung passt, dass eure Sensoren passen, dass eure Endschalter auch wirklich auslösen, etc.**

In dieser Anleitung sollte es primär um die Installation von Klipper / Moonraker / Mainsail / Fluidd auf einem blanken Raspberry Pi gehen. Alles Weitere, gerade auch das Initial Setup könnt ihr unter folgenden Links nachlesen. Wichtig ist auch immer ein Pinout eures Boards, damit ihr euch überhaupt einen Überblick verschaffen könnt, an welchem Port/Pin welcher Aktor/Sensor angeschlossen ist. Das entsprechende Pinout ist sehr leicht über Google zu finden. Euer Boardname + Pinout in die Suche eingeben.

[Klipper Configuration Check](https://www.klipper3d.org/Config_checks.html)

[Klipper Doku](https://www.klipper3d.org/Overview.html)

Mal noch fix, wie ganz am Anfang erwähnt zum Raspberry PI OS Lite Betriebssystem für den Pi und dessen Installation.

Was wird benötigt:

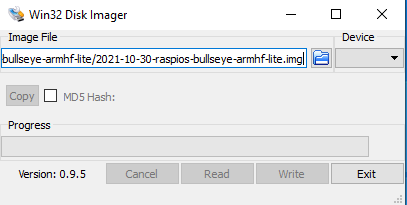
Leere, formatierte SD-Karte ab 8GB

[Win32Diskimager](https://win32diskimager.download/download-win32-disk-imager/)

[Raspberry PI OS Lite Image Datei](https://www.raspberrypi.com/software/operating-systems/)

Raspberry PI OS Lite Image herunterladen und entpacken

Mittels Win32Diskimager, das Image auswählen und auf die SD-Karte schreiben



Bitte anschließend bei einer Formatierungsaufforderung dieser nicht Folge leisten. Windows erkennt nur die Partition nicht, muss es auch nicht.

Im Windows Explorer wird eine Partition Namen „Boot“ angezeigt.

In dieser Partition erstellen wir uns nun eine Datei mit dem Namen ssh , allerdings ohne Dateiendung.

Der einfache Weg ist – rechte Maustaste in den Boot Ordner– Im Kontextmenü NEU auswählen und ein Textdokument erstellen. Dieses in **ssh** umbenennen und die Dateiendung .**txt** löschen. Windows wird wegen dem Dateityp meckern, dies stört uns aber nicht weiter.

Wofür machen wir das Ganze? Damit wir über Putty.exe mittels SSH-Verbindung auf den Pi zugreifen können.

Solltest ihr keine Möglichkeit haben via Netzwerkkabel auf den Pi zuzugreifen, sondern nur über Wlan, dann ist es wichtig eine Wlan Konfig Datei vorab zu erstellen und ebenfalls in die Boot Partition der SD-Karte zu legen.

Eine Anleitung findet ihr [hier](https://www.raspberry-pi-geek.de/ausgaben/rpg/2017/10/ssh-und-wlan-schon-bei-der-installation-konfigurieren/)

Wichtig: Achtet darauf, dass eure Wlan SSID und das Passwort zwischen den “ “ – Zeichen sitzen.

Micro SD in den Pi und ab geht die Post