komplette Installation

ACHTUNG: Diese Anleitung ist nicht 1:1 so auf alle neuen Boards anwendbar! Sie wurde relativ speziell für das Mellow Fly E3-PRO V3 geschrieben!

In diesem Video zeige ich euch einmal komplett die Installation von Klipper & Co auf Basis des **Mellow Fly E3-PRO V3** Boards und eines Raspberry Pi 3b. Es geht also um Mainsail (OS), Klipper Firmware compilieren und flashen, einen besseren Bootloader, die Drucker Konfiguration, Webcam und KlipperScreen. In diesem Video wird nicht auf alle Teilbereiche im Detail eingegangen. Dafür gibt es eine ganze Reihe an Videos, die ich hier mal verlinken möchte:

#32 - Klipper FAQ - Klipper Grundlagen - https://youtu.be/Bv361latAZo
#33 - Klipper FAQ - Raspberry Pi - Mainsail OS - https://youtu.be/LKBSG2A85sU
#42 - Klipper FAQ - ls /dev/serial/by-id/* Murks - https://youtu.be/Am5ONKeKwvc
#44 - Klipper FAQ - Video mit Crowsnest - https://youtu.be/6PRwHsu7Ipw

Danke an dieser Stelle an Jan Oerter (aka Schmelzerboy) der mich überhaupt zu diesem Video inspiriert hat. Wir haben zusammen eine kleine Reihe für einen Anycube I3 Mega zusammengestellt. Die Playlist dazu verlinke ich hier, wenn sie erstellt worden ist.

Jans Kanal könnt ihr euch hier ansehen und abonnieren 🙂
https://www.youtube.com/@MegaMuelleimer

YouTube Video #46



Einrichtung des Raspberry Pi

- Raspberry Pi Imager installieren & starten (https://www.raspberrypi.com/software/)
- Other specific-purpose OS → 3D Printing → Mainsail OS → 32 Bit Variante nutzen
- SD-Karte in den PC mit mind. 8GB Speicher
- Konfiguration im Raspberry Pi Imager anpassen und SD-Karte schreiben lassen
- Karte in den Raspberry Pi stecken und booten
- per SSH auf den Raspberry Pi einloggen (am besten mit MobaXTerm → https://mobaxterm.mobatek.net/)
- Updates und Tools installieren
  ```
  sudo apt update && sudo apt upgrade -y && sudo apt install -y git
  ```

```
silversearcher-ag wavemon hexedit sudoku tcpdump iptraf mc htop dcfldd
nano usbutils ranger tldr ncdu can-utils multitail fd-find && mkdir -p
~/.local/share && tldr -u
```

- unnütze Pakete entfernen
```
sudo apt autoremove -y modem* cups* pulse* avahi* triggerhappy*
```

CanBoot als neuer Bootloader - Warum?

- Firmware Flashen (Klipper) ohne die Hardware anzufassen
- keine SD-Karten Jonglieren mehr
- das Kopieren der klipper.bin Datei (nebst umbenennen) entfällt
- man kann in der Linux Konsole nachsehen, ob der Bootloader aktiv ist
- keine inkompatiblen oder falsch formatierte SD-Karten mehr
- Das Board kann mittels seriellem Port, USB und CAN-Bus (wenn verfügbar) geflasht werden

> **Hinweis Mellow Fly E3 Pro V3**
> Das Board hat leider ein Problem mit dem DFU Modus (USB **D**evice **F**irmware **U**pgrade). Normalerweise erreicht man den Modus durch drücken der BT0 Taste auf dem Board (oder setzen eines Jumpers) und dann Reset (oder das Board mit Strom versorgen). Bei diesem Board klappt das nicht. Auch mittels Klipper lässt sich das Board nicht in diesen Modus bringen (wenn man `make flash FLASH_DEVICE …` verwendet).
> Hier macht es also umso mehr Sinn einen brauchbaren Weg zu etablieren um das Board mit einer neuen (Klipper) Firmware zu flashen.
>
> **Hinweis Original “SD”-Bootloader**
> Der originle Bootloader geht beim Einspielen von CanBoot verloren. Ein Update über SD-Karte ist damit also nicht mehr möglich. Für die meisten Boards findet man den “alten” Bootloader aber im Internet und könnte ihn wieder herstellen. Wer aber bei Klipper bleibt und sich beim Flashen nicht die Finger brechen will sollte zu CanBoot wechseln 🙂

CanBoot mit anderen Boards ?

CanBoot ist derzeit kompatibel zu den meisten STM32 Boards. Aber auch der Raspberry Pi RP2040 sowie zwei LPC Controller werden unterstützt. Damit deckt CanBoot ein sehr breites Spektrum an Druckerboards ab.

CanBoot flashen

> **Hinweis andere Boards**
> Diese Anleitung zu CanBoot bezieht sich auf das Mellow Fly Board. Wenn ihr ein anderes Board nutzt müsst ihr ggf. die Parameter bei `make menuconfig` anpassen. Die findet ihr im Normalfall in der Beschreibung wie man Klipper für euer Board kompiliert.

- `cd ~`
- `git clone https://github.com/Arksine/CanBoot`
- `cd CanBoot`
- `make menuconfig` (Beenden mittels Q → Y)

```
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F407)  --->
    Build CanBoot deployment application (32KiB bootloader)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (USB (on PA11/PA12))  --->
    Application start offset (32KiB offset)  --->
    USB ids  --->
()  GPIO pins to set on bootloader entry
[*] Support bootloader entry on rapid double click of reset button
[ ] Enable bootloader entry on button (or gpio) state
[ ] Enable Status LED
```

- CanBoot kompilieren
  `make -j4`
- `deployer.bin` auf SD-Karte (FAT32) kopieren und in `firmware.bin` umbenennen
- SD-Karte in das Fly Board und mit Strom versorgen
  - 20 Sekunden warten (Blaue LED blinkt am Anfang 1x kurz auf)
  - Strom wegnehmen und SD-Karte raus nehmen
  - wieder einschalten
- Auf der SD-Karte muss die Datei jetzt `FLY.CUR` oder `firmware.cur` heißen !
- `dmesg`

```
[ 2361.296349] usb 1-1.1: USB disconnect, device number 20
[ 2362.202635] usb 1-1.1: new full-speed USB device number 21 using
dwc_otg
[ 2362.450874] usb 1-1.1: New USB device found, idVendor=1d50,
idProduct=6177, bcdDevice= 1.00
[ 2362.450948] usb 1-1.1: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[ 2362.450975] usb 1-1.1: Product: stm32f407xx
[ 2362.450997] usb 1-1.1: Manufacturer: CanBoot
[ 2362.451016] usb 1-1.1: SerialNumber: 2B003500114739343730333 7
[ 2362.467908] cdc_acm 1-1.1:1.0: ttyACM0: USB ACM device
```

- `ttyACM0` merken für das Flashen der Klipper Firmware

Klipper flashen

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

```
[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F407)  --->
    Bootloader offset (32KiB bootloader)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (USB (on PA11/PA12))  --->
    USB ids  --->
()  GPIO pins to set at micro-controller startup
```

- `make flash FLASH_DEVICE=/dev/ttyACM0`

```
pi@TestPi:~/klipper $ make flash FLASH_DEVICE=/dev/ttyACM0
  Flashing out/klipper.bin to /dev/ttyACM0
Entering bootloader on /dev/ttyACM0
Device reconnect on
/sys/devices/platform/soc/20980000.usb/usb1/1-1/1-1.1/1-1.1:1.0
/usr/bin/python3 lib/canboot/flash_can.py -d /dev/serial/by-
path/platform-20980000.usb-usb-0:1.1:1.0 -f out/klipper.bin

Attempting to connect to bootloader
CanBoot Connected
Protocol Version: 1.0.0
Block Size: 64 bytes
Application Start: 0x8008000
MCU type: stm32f407xx
Flashing '/home/pi/klipper/out/klipper.bin'...

[##################################################]

Write complete: 2 pages
Verifying (block count = 441)...

[##################################################]

Verification Complete: SHA = 913C5425E676A11DDF249DAEE669CF4362044AAE
CAN Flash Success
```

> **Hinweis Flashen im Betrieb**
> Wenn das Board in der printer.cfg eingebunden ist, müsst ihr 2 Schritte extra machen für das Flashen:
> `sudo systemctl stop klipper.service`
> dann das `make flash ….`
> `sudo systemctl start klipper.service`
>
> **Hinweis flash_can.py**
> Man kann auch direkt das Tool flash_can.py von CanBoot verwenden :
> `pi@TestPi:~/klipper $ make flash FLASH_DEVICE=/dev/ttyACM0`

Board Check

- `dmesg`
- `ls -lR /dev/ | grep -v '\->\s../tty' | grep -e 'tty[[:alpha:]]' -e serial`
- `~/klipper $ ../klippy-env/bin/python ./klippy/console.py /dev/ttyACM0`

kiauh für Update nutzen

- `cd ~`
- `git clone https://github.com/th33xitus/kiauh.git`
- `./kiauh/kiauh.sh`
- `4) [Advanced]` auswählen → `4) [Build + Flash]` auswählen
- Den Anweisungen im Terminal folgen 🙂

Raspberry PI MCU

Um den ADXL345 am Raspberry Pi zu betreiben brauchen wir die extra MCU die mit auf dem Raspberry Pi läuft. Diese MCU sorgt dafür, dass die GPIO Pins des Raspberry Pi aus Klipper Sicht verwendbar werden. Eingeschlossen sind dabei auch I2C und SPI Bus (über den der ADXL Sensor angeschlossen wird).

- `cd ~/klipper/`
- `sudo cp ./scripts/klipper-mcu.service /etc/systemd/system/`
- `sudo systemctl enable klipper-mcu.service`
- `make menuconfig`

```
[*] Enable extra low-level configuration options
    Micro-controller Architecture (Linux process)  --->
()  GPIO pins to set at micro-controller startup
```

- `sudo usermod -a -G tty pi`
- `sudo service klipper stop`
- `make flash -j4`
- `sudo service klipper start`

printer.cfg einspielen

Eine vorgefertigte Konfiguration findet ihr für das **Mellow Fly E3 Pro V3** in der Github Repo von Jan. Diese wird einfach geladen und an die richtige Stelle kopiert.

- `cd ~`
- `git clone https://github.com/Schmelzerboy/I3-Mega-Klipper`
- `cp ./I3-Mega-Klipper/Konfiguration\ Drucker/*.cfg ~/printer_data/config/`
- Den seriellen Port ermitteln
  `ls -lR /dev/ | grep -v '\->\s../tty' | grep -e 'tty[[:alpha:]]' -e serial`
  
  `/dev/serial/by-id/usb-Klipper_stm32f407xx_2B0035001147393437303337-if00`
- Den Port in der printer.cfg eintragen. Das geht mittels nano in der Konsole.
  `nano ~/printer_data/config/printer.cfg`
  Port eintragen und dann schließen/speichern mittels mit STRG + X → Y für Yes → Enter
  
- `sudo systemctl restart klipper.service`

Hinweis zur printer.cfg

Zu jedem Teil in der printer.cfg gibt es sehr ausführliche Informationen in der Config Reference → https://www.klipper3d.org/Config_Reference.html . Wer dem Englischen nicht zu zugetragen ist sollte sich mit folgenden Links behelfen:

- https://translate.google.com/?hl=de
- https://www-klipper3d-org.translate.goog/Config_Reference.html?_x_tr_sl=auto&_x_tr_tl=de&_x_tr_hl=de&_x_tr_pto=wapp
- https://www.deepl.com/translator

Notes Video Wozu Konfig ? 1 Erweiterung 2 Pins 3 Einstellungen

KlipperScreen mit kiauh

> **Hinweis KlipperScreen**
> Ihr müsst KlipperScreen nur dann installieren wenn am Raspberry Pi auch ein Touch Bildschirm angeschlossen ist. Ist das nicht der Fall könnt ihr diesen Schritt einfach überspringen!

- `cd ~`
- `sudo raspi-config`
  - → `1 System Options`
  - → `S5 Boot / Auto Login`
  - → `B1 Console`
  - raspi-config beenden und den Pi neu booten lassen
- Wieder per SSH einloggen ...
- kiauh starten
  `./kiauh/kiauh.sh`
  - → `1) [Install]`
  - → `5) [KlipperScreen]`
  - warten .... 🙂 (da wird einiges nachinstalliert ....)
  - Wenn alles installiert ist mit B → Q kiauh verlassen
- `sudo reboot`

Fehlerbehebung

- Logfile ansehen
  `cat /home/pi/.local/share/xorg/Xorg.0.log`
- Wenn am Ende was steht mit `xf86OpenConsole: Cannot open virtual console 2 (Permission denied)` dann folgendes ausführen
  `sudo bash -c "echo needs_root_rights=yes>>/etc/X11/Xwrapper.config"`
- `sudo service KlipperScreen restart`

Mainsail öffnen

- Mit der IP-Adresse vom Raspberry Pi die Webseite im Browser öffnen

    - die gleiche IP-Adresse, die ihr auch für SSH verwendet habt ...
- Es sollte sich Mainsail öffnen ...

Webcam mit crowsnest

- USB Kamera an den Pi anschließen
- `sudo systemctl restart crowsnest.service`
- Das Logfile anzeigen lassen
  `cat ~/printer_data/logs/crowsnest.log`
- In der Ausgabe nach einem solchen Eintrag suchen:
- Den Device Link kopieren (hier `/dev/v4l/by-id/usb-Novatek_WebCam_130_HD2-video-index0`)
- `nano ~/printer_data/config/crowsnest.conf`
    - Den ersten Eintrag anpassen mit dem Kamera Device Link
    - Evtl. auch die Auflösung anpassen → die findet man ebenfalls im `crowsnest.log`
    - nano beenden mit STRG + X → Y → Enter
- `sudo systemctl restart crowsnest.service`
- In der Mainsail Webseite eine neue Kamera hinzufügen
- Auf Save Webcam und das sollte es gewesen sein
- Im Dashboard sollte nun die Webcam angezeigt werden

Fly Board zurücksetzen

> **Hinweis 1**
> Die folgende Anleitung funktioniert so NUR mit dem Mellow Fly E3 Pro V3 Board. Auch die verlinkte Firmware ist nur mit diesem Board kompatibel!
>
> **Hinweis 2**
> Dieser Schritt ist auch nur dann nötig, wenn ihr wieder über SD-Karte flashen wollt. Wenn CanBoot einmal richtig geflsht ist, braucht ihr diesen Part nicht! Die folgende Anleitung dient eher als Notanker !!

Um das Mellow Fly Board auf den alten Bootloader (Flashen mit SD-Karte) zurückzusetzen, sind folgende Schritte nötig:

- STM32CubeProgrammer installieren
- Das Fly Board an einen USB/Seriell Adapter anschließen
    - Tx Pin vom Board muss an Rx Pin vom USB Adapter
    - Rx Pin vom Board muss an Tx Pin vom USB Adapter
    - GND muss auch verbunden werden zum USB Adapter
- STM32CubeProgrammer starten, UART auswählen und den seriellen Port von dem Adapter einstellen
- Das Board sollte jetzt mit Strom versorgt werden ...
- Dann am Board BT0 **gedrückt halten**, und Reset einmal kurz drücken.

- Jetzt in der STM Software auf das grüne Connect oben rechts klicken
- Und dann BT0 Taste am Board loslassen
- Wenn folgender Fehler kommt
  Error: Activating device: KO. Please, verify the boot mode configuration and check the serial port configuration. Reset your device then try again...
  dann ist vermutlich TX / RX auf einer Seite vertauscht.
- Wenn alles geklappt hat, kriegst du einen Teil vom Speicher angezeigt und unten rechts den Controllertyp:
- Jetzt erstmal Speicher löschen
  (1) richtigen Reiter öffnen
  (2) Full Chip erease
- Den Original Bootloader von hier laden
  https://github.com/Mellow-3D/f407bootloader/releases/download/v1.01/STM32.bin
- Wieder auf die erste Seite wechseln (1) und die geladene BIN Datei laden (2)
- Auf Download klicken und flaschen
- Fertig

Links

- github Fly-E3-Pro-v3
  https://github.com/Mellow-3D/Fly-E3-Pro-v3
- Fly-407ZG Bootloader
  https://github.com/Mellow-3D/f407bootloader
  https://github.com/Mellow-3D/f407bootloader/releases
- MobaXTerm
  https://mobaxterm.mobatek.net/
- CanBoot
  https://github.com/Arksine/CanBoot
- kiauh
  https://github.com/th33xitus/kiauh

From:
https://www.drklipper.de/ - **Dr. Klipper Wiki**

Permanent link:
https://www.drklipper.de/doku.php?id=klipper_faq:46_-_komplette_installation

Last update: **2024/10/29 05:57**