

# Klipperscreen überall

## Hinweis

Die Anleitungen hier sind vorwiegend für Debian basierte Systeme (Linux) und unter Windows mittels MSYS2 Mingw64.

Bei anderen Distributionen ist die Installation ggf. different!!

## Pi mit XPT2046

- getestet auf einem **Raspbian OS Bookworm 64Bit mit Desktop** (Standard Image)
- [https://klipperscreen.readthedocs.io/en/latest/Hardware/GPIO\\_35/](https://klipperscreen.readthedocs.io/en/latest/Hardware/GPIO_35/)
- sudo nano /boot/firmware/config.txt
  - siehe auch <https://github.com/raspberrypi/firmware/blob/master/boot/overlays/README>
  - dtoverlay=piscreen,drm
  - dtoverlay=piscreen,drm,rotate=180 für gedreht
  - für Invers vom Touch gibt es noch invx und invy
  - Dieses Overlay auskommentieren : #dtoverlay=vc4-kms-v3d
- Installation von KlipperScreen über kiauh
  - cd ~
  - git clone <https://github.com/th33xitus/kiauh.git>
  - ./kiauh/kiauh.sh
- Touch Screen Kalibrierung
  - [https://klipperscreen.readthedocs.io/en/latest/Troubleshooting/Touch\\_issues/?h=cali](https://klipperscreen.readthedocs.io/en/latest/Troubleshooting/Touch_issues/?h=cali)
  - DISPLAY=:0 xinput
  - Beispiel für X Invert
 

```
DISPLAY=:0 xinput set-prop "ADS7846 Touchscreen" 'Coordinate Transformation Matrix' -1 0 1 0 1 0 0 0 1
```
  - Kalibrierung über xcal tool → muss extra kompiliert werden und ist oft nicht wirklich nötig.
- Touchscreen Test
  - sudo apt install evtest
  - evtest → Input wählen → am Touch rumdrücken :

```
Event: time 1741437484.802100, type 3 (EV_ABS), code 0 (ABS_X), value 973
Event: time 1741437484.802100, type 3 (EV_ABS), code 1 (ABS_Y), value 1156
Event: time 1741437484.802100, type 3 (EV_ABS), code 24 (ABS_PRESSURE), value 162
Event: time 1741437484.802100, ----- SYN_REPORT -----
```

## Linux (kiauh inkl. Pi)

- per kiauh Klipperscreen installieren

- ggf. den Ordner config anlegen  
  mkdir -p ~/printer\_data/config
- nano KlipperScreen.conf

[download](#)

```
[printer Voron]
# Define the moonraker host/port if different from 127.0.0.1 and
7125
moonraker_host: 192.168.30.70
moonraker_port: 7125
```

moonraker\_host mit IP vom Drucker anpassen

- Auf dem Drucker selber sicherstellen das die IP Range erlaubt ist
  - moonraker.conf öffnen
  - [download](#)

```
[authorization]
cors_domains:
...
trusted_clients:
...
192.168.30.0/24
```

## Linux X86

- sudo apt update && sudo apt upgrade -y
- sudo apt install python3 python3-venv libcairo2 libcairo2-dev gir1.2-cairo-1.0 libgirepository1.0-dev gir1.2-glib-2.0 python3-dev mpv libmpv-dev
- git clone <https://github.com/KlipperScreen/KlipperScreen.git>
- cd KlipperScreen
- python3 -m venv env
- source env/bin/activate
- pip install pygobject jinja2 websocket-client requests python-mpv sdbus
- python screen.py



- :!: Ab hier läuft KS schon im Fullscreen

Eine Konfig macht Sinn



# Windows

- <https://www.msys2.org/> installieren (Stand jetzt msys2-x86\_64-20250221.exe)
- **MSYS2 MINGW64** starten
- Updates einspielen  
pacman -Syu  
Danach die Konsole neu starten und ggf. auch den Befehl nochmal wiederholen!
- Nötige Pakete installieren

```
pacman -S \
python3 cmake git make base-devel \
mingw-w64-x86_64-ag \
mingw-w64-x86_64-gcc \
mingw-w64-x86_64-toolchain \
mingw-w64-x86_64-pkg-config \
mingw-w64-x86_64-python-pip \
mingw-w64-x86_64-python-setuptools \
mingw-w64-x86_64-python-wheel \
mingw-w64-x86_64-python-cairo \
mingw-w64-x86_64-python-gobject \
mingw-w64-x86_64-gobject-introspection \
mingw-w64-x86_64-gtk3 \
mingw-w64-x86_64-ninja \
mingw-w64-x86_64-mpv \
mingw-w64-x86_64-ffmpeg \
mingw-w64-x86_64-libass
```

- git clone <https://github.com/KlipperScreen/KlipperScreen.git>
- cd KlipperScreen
- python -m venv env
- source env/bin/activate
- python.exe -m pip install --upgrade pip
- pip install pygobject requests ninja2 python-mpv websocket-client  
sdbus sdbus\_networkmanager führen noch zu Problemen ...
- Anpassung für DPMS  
nano screen.py  
Zwei Zeilen auskommentieren. (Suchen geht mit STRG + W)

```
self.use_dpms =
self._config.get_main_config().getboolean("use_dpms", fallback=True)
# self.use_dpms &= functions.dpms_loaded
# self.set_dpms(self.use_dpms)
```

- Konfig im KlipperScreen Ordner anlegen  
nano KlipperScreen.conf



Ansonsten startet KS im Fullscreen !!

- Start mit python screen.py

## Windows Button

Wie starte ich das mit einem Button auf dem Desktop ?

- Es muss Python auf Windows installiert sein!
- neue Datei anlegen `startks.py`

[startks.py](#)

```
import subprocess
import os

# MSYS2-Pfad und Venv-Pfad anpassen
MSYS2_PATH = r"C:\msys64"

# Bash-Befehl, der ausgeführt werden soll
bash_command = f"cd ~/KlipperScreen && source env/bin/activate &&
python screen.py"

# Vollständiger Pfad zur bash.exe
bash_exe = os.path.join(MSYS2_PATH, "usr", "bin", "bash.exe")

# Führe den Befehl unsichtbar aus
subprocess.run([bash_exe, "--login", "-c", bash_command],
               creationflags=subprocess.CREATE_NO_WINDOW)
```

- Jetzt eine Verknüpfung erstellen auf `pythonw C:\Program Files\Python310\pythonw.exe`
- Einen Namen angeben "KlipperScreen" z.B. und Dialog beenden
- Rechtsklick auf das Icon und Eigenschaften aufrufen
  - Ziel : "C:\Program Files\Python310\pythonw.exe" "C:\Users\Dominik Schmidt\Desktop\test.py"
  - Icon ggf. anpassen

## X11VNC

- `sudo apt install x11vnc`
- für einen ersten Test kann man das verwenden  
`x11vnc -usepw -forever -display :0`
- **Einrichtung als Dienst**
- `sudo x11vnc -storepasswd /etc/x11vnc.pass`
- `sudo nano /lib/systemd/system/x11vnc.service`

```
[Unit]
Description=Start X11VNC
After=multi-user.target
```

```
[Service]
Type=simple
ExecStart=/usr/bin/x11vnc -display :0 -usepw -forever -loop -noxdamage
-repeating -rfbauth /etc/x11vnc.pass -rfbport 5900 -shared

[Install]
WantedBy=multi-user.target
```

- sudo systemctl enable x11vnc.service

## Konfig

- <https://klipperscreen.readthedocs.io/en/latest/Configuration/>
- Suchpfade:
  - ~/printer\_data/config/KlipperScreen.conf
  - ~/.config/KlipperScreen/KlipperScreen.conf
  - \${KlipperScreen\_Directory}/KlipperScreen.conf
- Vorlage

```
[main]
#language = None
#theme =
width = 800
height = 600
show_cursor = True
use_dpms = False
screen_blanking_printing = 14400
#screen_blanking = off
#default_printer =
autoclose_popups = True
auto_open_extrude = True

[printer Voron]
# Define the moonraker host/port if different from 127.0.0.1 and 7125
moonraker_host: 192.168.30.70
moonraker_port: 7125

[printer Franzmann]
# Define the moonraker host/port if different from 127.0.0.1 and 7125
moonraker_host: 192.168.30.75
moonraker_port: 7125

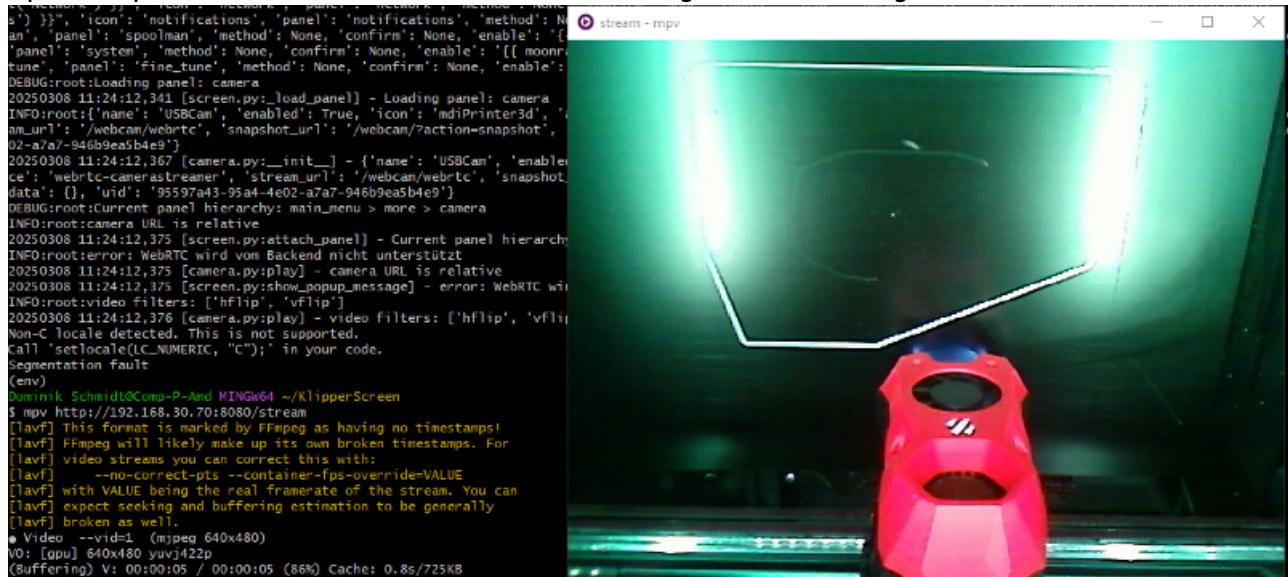
#~# --- Do not edit below this line. This section is auto generated ---
#~#
#~#
#~# [main]
#~# screen_blanking_printing = 14400
#~# use_dpms = False
#~# show_cursor = True
#~#
```

Last update:

2025/03/08 klipper\_faq:klipperscreen\_ueberall https://www.drklipper.de/doku.php?id=klipper\_faq:klipperscreen\_ueberall&rev=1741459637  
19:47

## Video MPV

- Camera Streamer hat einen kleinen Webserver zum Testen  
<http://<IP>:8080/>  
Und dort kann man auch die /stream URL sehen und testen
- mpv http://<IP>:8080/stream startet eine eigene Anwendung mit dem Livestream :



From:

<https://www.drklipper.de/> - Dr. Klipper Wiki

Permanent link:

[https://www.drklipper.de/doku.php?id=klipper\\_faq:klipperscreen\\_ueberall&rev=1741459637](https://www.drklipper.de/doku.php?id=klipper_faq:klipperscreen_ueberall&rev=1741459637)

Last update: **2025/03/08 19:47**

