

Display Panel

Install

Wegen zu vielen Problemen mit Wayland wird hier **Trixie mit X11** genutzt.

Das Hauptproblem ist, dass das Display beim Stromsparen einen nervigen Bildschirm anzeigt ... "No Signal" ...

Einstellungen

- `sudo raspi-config`
 - I2C Aktivieren
 - umstellen von Wayland auf X11

100 Mbit Fix

- Man muss dafür per SSH mit WLAN auf dem Pi sein
- `sudo nano /etc/systemd/system/eth0fix.service`

```
[Unit]
Description=100MBit Fix for eth0
After=network-online.target
Requires=network-online.target

[Service]
ExecStart=/sbin/ethtool -s eth0 speed 100 duplex full autoneg on
Type=oneshot

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

- `sudo systemctl enable eth0fix.service`
- `sudo systemctl restart eth0fix.service`

Updates

- Command Line
`sudo apt update && sudo apt upgrade -y && sudo apt install -y git git-lfs silversearcher-ag wavemon hexedit sudoku tcpdump iptraf mc htop dcfldd nano usbutils ranger ncd u can-utils multitail fd-find lsof x11vnc minicom joystick i2c-tools speedtest-cli iotop fio ir-keytable curl inxi lshw nala fzf duf`
- UI Tools

```
sudo apt update && sudo apt upgrade -y && sudo apt install -y flameshot  
terminator cutecom jstest-gtk xclip
```

- Nutzloses Zeug

```
sudo apt autoremove -y modem* cups* avahi* triggerhappy* rpi-connect  
openvpn
```

Display

- sudo raspi-config → Screen Blanking aus !
- Screensaver aus
 - mkdir -p /home/pi/.config/lxsession/LXDE-pi
 - cp /etc/xdg/lxsession/LXDE-pi/autostart /home/pi/.config/lxsession/LXDE-pi/autostart
 - nano /home/pi/.config/lxsession/LXDE-pi/autostart

[download](#)

```
@lxpanel --profile LXDE-pi  
@pcmanfm --desktop --profile LXDE-pi  
#@xscreensaver -no-splash  
@xset s off  
@xset -dpms
```

- sudo apt remove xscreensaver
- Venv einrichten
 - sudo apt -y update && sudo apt-get install -y build-essential cmake make gcc pkg-config python3-dev python3-virtualenv libsystemd-dev libopenblas-dev
 - cd ~ && mkdir -p DisplayOn0ff && cd DisplayOn0ff
 - python -m venv . && source bin/activate
 - pip3 install --upgrade pip && pip3 install RPi.GPIO pynput smbus paho-mqtt psutil
- Compile VL53 Lib ...
 - sudo apt-get install build-essential python-dev
 - cd ~
 - git clone https://github.com/pimoroni/VL53L0X_rasp_python.git
 - cd VL53L0X-python
 - make
 - VL53L0X.py in den Scriptordner kopieren ...
- Tests starten ...

Tests

VL

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 -auth guess -forever -loop -
noxdamage -repeat -rfbauth /etc/x11vnc.pass -rfbport 5900 -shared

[Install]
WantedBy=multi-user.target
```
- `sudo systemctl enable x11vnc.service`

## Motion Detect

- Sensor : VL53L0X
- Altes Script

```
import sys
import time
import VL53L0X
import os
import subprocess
import RPi.GPIO as GPIO
from pynput import mouse

def on_move(x, y):
 global Display_Last_Action
 print('Pointer moved to {0}'.format((x, y)))
 sys.stdout.flush()
 Display_Last_Action = time.time()
 print("Time %d" % (Display_Last_Action))

def on_click(x, y, button, pressed):
 global Display_Last_Action
 print('{0} at {1}'.format('Pressed' if pressed else 'Released', (x,
y)))
 sys.stdout.flush()
```

```
Display_Last_Action = time.time()
print("Time %d" % (Display_Last_Action))

def on_scroll(x, y, dx, dy):
 global Display_Last_Action
 print('Scrolled {0} at {1}'.format('down' if dy < 0 else 'up', (x,
y)))
 sys.stdout.flush()
 Display_Last_Action = time.time()
 print("Time %d" % (Display_Last_Action))

def toggleOnOff():
 GPIO.output(36, GPIO.HIGH)
 time.sleep(0.4)
 GPIO.output(36, GPIO.LOW)
 print("Toggle Switch")
 sys.stdout.flush()

listener = mouse.Listener(
 on_move=on_move,
 on_click=on_click,
 on_scroll=on_scroll)
listener.start()

Last Time Event happend on Display
Display_Last_Action = time.time()
Display_State = True
Display_Timeout_Sec = 180

RPi.GPIO Layout verwenden (wie Pin-Nummern)
GPIO.setmode(GPIO.BOARD)
RuntimeWarning: This channel is already in use, continuing anyway.
Use GPIO.setwarnings(False) to disable warnings.
GPIO.setwarnings(False)

Pin 36 (GPIO 16) auf Output setzen -> Display ON / OFF
GPIO.setup(36, GPIO.OUT)
Pin 38 (GPIO 20) auf Output setzen -> Brightness +
GPIO.setup(38, GPIO.OUT)
Pin 40 (GPIO 21) auf Output setzen -> Brightness -
GPIO.setup(40, GPIO.OUT)

Create a VL53L0X object
tof = VL53L0X.VL53L0X(i2c_bus=1,i2c_address=0x29)
tof.open()
tof.start_ranging(VL53L0X.VL53L0xAccuracyMode.BETTER)

while 1:
 # Distanz vom VL53L0X Sensor messen
 distance = tof.get_distance()
```

```

 #print("%d mm, %d cm" % (distance, (distance/10)))

 # Display ggf. abschalten
 if time.time()-Display_Last_Action >= Display_Timeout_Sec:
 if Display_State == True:
 #subprocess.call("DISPLAY=:0 xset dpms force off",
shell=True)
 toggleOnOff()
 Display_State = False
 print("Display wird abgeschaltet ...")
 sys.stdout.flush()

 # Abstand vom Sensor pruefen
 if distance < 900:
 print("DETECTED, %d cm , %d sec" % (distance/10, time.time()-
Display_Last_Action))
 sys.stdout.flush()
 Display_Last_Action = time.time()
 print("Time %d" % (Display_Last_Action))

 # Display ggf. wieder einschalten
 if time.time()-Display_Last_Action < Display_Timeout_Sec:
 if Display_State == False:
 #subprocess.call("DISPLAY=:0 xset dpms force on",
shell=True)
 toggleOnOff()
 Display_State = True
 print("Display wird eingeschaltet ...")
 sys.stdout.flush()

 # Display Einschalten
 #os.system('DISPLAY=:0 xdotool key "shift"')
 ###subprocess.call("DISPLAY=:0 xset s reset", shell=True)
 #subprocess.call("DISPLAY=:0 xset dpms force on", shell=True)
 #subprocess.call("DISPLAY=:0 xset dpms 120 120 120",
shell=True)

 time.sleep(0.1)

tof.stop_ranging()
tof.close()

```

- Service

```

pi@SHome-Display-VPN-DNS:/ $ cat lib/systemd/system/display.service
[Unit]
Description=Display Controller

[Service]
Environment=DISPLAY=:0
Environment=XAUTHORITY=/home/pi/.Xauthority

```

```
ExecStart=/usr/bin/python /home/pi/DisplayOn/MotionDetect.py
Restart=always
RestartSec=30s
KillMode=process
TimeoutSec=infinity

[Install]
WantedBy=graphical.target
```

## Chromium Start

```
pi@SHome-Display-VPN-DNS:~ $ cat chrome.sh
#!/bin/sh -e

pfad="http://192.168.30.22:8082/vis/index.html?main#Home"

Chromium ggf. beenden
chromerunning=`ps aux | grep chromium |wc -l`
echo $chromerunning

if test $chromerunning -gt 1; then
 echo "Beende alle Chromium Instanzen ..."
 sudo killall /usr/lib/chromium-browser/chromium-browser-v7
 sleep 4
fi

Chromium zurücksetzen
sed -i 's/"exited_cleanly":false/"exited_cleanly":true/'
~/.config/chromium/'Local State'
sed -i 's/"exited_cleanly":false/"exited_cleanly":true/;
s/"exit_type":"[^"]\+"/"exit_type":"Normal"/'
~/.config/chromium/Default/Preferences

Chromium starten
DISPLAY=:0 /usr/lib/chromium-browser/chromium-browser-v7 --touch-
events=enabled --noerrdialogs --check-for-update-interval=31536000 &

Chromium in Fullscreen
sleep 15
#WID=$(DISPLAY=:0 xdotool search --onlyvisible --class chromium|head -1)
#DISPLAY=:0 xdotool windowactivate ${WID}
#DISPLAY=:0 xdotool key F11

An Display Position navigieren mit der Maus und Vollbild auslösen
DISPLAY=:0 /usr/bin/xdotool mousemove --sync 954 200 click 1

exit 0
```

From:

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

Permanent link:

[https://drklipper.de/doku.php?id=haussteuerung:display\\_panel&rev=1766240542](https://drklipper.de/doku.php?id=haussteuerung:display_panel&rev=1766240542)

Last update: **2025/12/20 15:22**

