

# Videodisplay

## Doku

Doku zur Implementierung der DRK Wawa Video-Bildschirme

## Installation des Betriebssystems

- Verwendete Version: Raspberry Pi OS with desktop (ca. 1.2GB)
- Verwendeter Installer: [Raspberry Pi Imager](#)

## Konfiguration des Raspberry Pis

Quelle: [Raspberry Pi Video Surveillance Monitor](#)

Boot up the Raspberry Pi. We need to make a few changes to the configuration to improve performance.

There is a tool preinstalled called raspi-config which offers a graphical way to tweak the operating system.

Launch the config tool by running `sudo raspi-config` over an SSH connection or using the local keyboard/monitor.

A graphical menu should pop up that you can now use.

We want to change the following:

Boot Options → Wait for Network at Boot

Interfacing Options → SSH → Yes

Advanced Options → Expand File System

Advanced Options → Memory Split → 256

## Installieren der Camera-SW:

```
wget https://github.com/Anonymousdog/displaycameras/archive/0.8.3.3.zip
unzip 0.8.3.3.zip
cd displaycameras-0.8.3.3
```

Now, make the installer executable and run the install script. When asked about the custom GPU memory split just press "Enter" to accept the default.

After it is done installing, it will prompt you to see the README.md.

```
chmod +x install.sh
sudo ./install.sh
```

## Additional Information

## Configuration

Now the next part of the setup will greatly depend on the monitor and number of cameras you want to display. displaycameras supports screen rotation so you can have it cycle through multiple screens of cameras. In the example\_layouts directory, there are several different examples of camera layouts as well as an in-depth document discussing how the layouts work. I recommend reading through my example and the document to get a general idea on how the system works before creating your config file.

I'm going to walk through an 8 camera setup. We want to show the cameras in a 2x2 grid, so 4 cameras at once. Therefore, we need 4 windows that will be on-screen showing 4 different cameras. 4 windows will be off-screen showing 4 more cameras that can't actively be seen. Look at the diagram below for a visual demonstration.

Every 5 seconds the camera in window 5 will move to window 4 and the camera from window 4 will move to window 3 and so on. This will create an effect of rotating cameras through the visible display. For an 8 camera solution, you could also have a 3x3 grid with a blank space and no rotation. displaycameras is very customizable for whatever makes sense in your environment.

## Configuration Files

We need to configure two files for our setup. The first is in /etc/displaycameras/layout.conf.default so you can edit it launching nano, vim or your favorite text editor while running an SSH session. There are 5 variables we need to set in this file, windows, window\_positions, camera\_names, camera\_feeds, and rotate.

Windows

Änderungen in Configfile können wegen fehlenden Root-Berechtigungen vom User Pi nicht vorgenommen werden

Also über Kommandozeile die Datei überschreiben:

```
cd /home/pi/Desktop
sudo cp -i layout.conf.default /etc/displaycameras
```

## Anpassen des Pi- und Root-User Passwords

PI-User: Über Menü > Einstellungen > Raspberry PI Config > System Root-User: Über die CLI:

```
sudo -i
passwd
```

## Remote Desktop für einfachere Bedienung einrichten

Quelle: [RASPBERRY PI – REMOTE DESKTOP \(XRDP\) INSTALLIEREN](#)

### Schritt 1: Boot to Desktop aktivieren

Zunächst muss der Raspberry Pi so konfiguriert werden, dass der (grafische) Desktop nach dem booten gestartet wird. Dies kann mit wenigen Klicks über raspi-config konfiguriert werden:

```
sudo raspi-config
```

Dort den Eintrag »3 Enable Boot to Desktop/Scratch« auswählen und anschließend »Desktop Log in as user 'pi' at the graphical desktop« bestätigen.

### Schritt 2: XRDP Server installieren

HINWEIS – Raspberry Pi mit PIXEL-Desktop

Leider funktioniert XRDP mit dem neuen Raspbian Image nicht mehr auf Anhieb. Der integrierte RealVNC Server und XRDP stehen in Konflikt zueinander und können nicht gemeinsam verwendet werden. Um diesen Konflikt zu beheben, muss RealVNC vor der Installation von XRDP zuerst deinstalliert werden:

```
sudo apt-get purge realvnc-vnc-server
```

Nachdem boot to Desktop aktiviert wurde, folgt nun die eigentliche Installation des Remote Desktop Servers. Unter Linux bietet XRDP eine freie Implementierung des Remote Desktop Protocols. Diese können wir direkt mittel des Paketmanagers APT installieren. Nach Abschluss der Installation wird der XRDP-Daemon automatisch gestartet.

```
sudo apt-get install xrdp
```

Die grundlegende Installation ist damit bereits erledigt. Der XRDP-Server wird zukünftig beim Systemstart automatisch im Hintergrund gestartet. Bei Bedarf können die XRDP-Einstellungen in den Dateien /etc/xrdp/xrdp.ini und /etc/xrdp/sesman.ini noch angepasst werden. Dies ist jedoch nicht zwingen notwendig und dürfte nur in besonderen Anwendungsfällen benötigt werden. Deshalb gehe ich darauf nicht näher ein.

### Schritt 3: Neustart

Zum Abschluss der Einrichtung empfiehlt sich ein Neustart des Raspberry Pi:

```
sudo reboot
```

### Schritt 4: Remote Desktop Client

Um nun Remote auf den Raspberry Pi zugreifen zu können, benötigen wir auf dem Client eine entsprechende Remote Desktop Client Software.

## Standby des Monitors verhindern (sonst verwirren die Kameras immer wieder)

Quelle: [Raspberry Pi: Monitor-Standby am HDMI-Ausgang ausschalten](#)

### Raspberry Pi: Monitor-Standby am HDMI-Ausgang ausschalten

Normalerweise wird bei Inaktivität des Nutzers irgendwann der Monitor abgeschaltet bzw. in Standby geschaltet. Die meisten Betriebssysteme sind standardmäßig so eingestellt, dass sie nach einer definierten Zeit den Bildschirm abschalten. Wenn man dann die Maus bewegt oder die Tastatur bedient geht die Anzeige wieder an. Aber es gibt auch Situationen, da möchte man nicht, dass der Bildschirm ausgeht.

### Auf dem Desktop den Standby ausschalten

Im Desktop-Betrieb wird der Bildschirm nach 10 Minuten Inaktivität in den Standby geschickt. Das heißt, er wird Schwarz. Die einfachste Methode den Standby dauerhaft abzuschalten ist, einen Bildschirmschoner zu installieren, und ihn danach zu deaktivieren. Damit wird auch der Standby bei Inaktivität ausgeschaltet.

Zur Installation gibt man auf der Kommandozeile folgenden Befehl ein:

```
sudo apt install xscreensaver
```

Mit „Return“ fortfahren und die Installation abwarten.

Den „Bildschirmschoner“ findet man dann im Menü unter „Einstellungen“. Beim ersten Aufruf muss der XScreenSaver-Dienst auf Nachfrage mit „OK“ gestartet.

Danach stellt man im Reiter „Anzeige-Modi“ den Modus auf „Bildschirmschoner deaktivieren“. Danach bleibt der Bildschirm dauerhaft an. Und das Programm kann man dann beenden.

### Auf der Kommandozeile den Standby ausschalten

Hinweis: Es kann sein, dass sich der Monitor auf der Kommandozeile nicht in den Standby schaltet. Es ist also nicht mehr notwendig den Standby auszuschalten.

Mit dem Kommando „xset“ kann man verschiedene Parameter der Bildschirmausgabe, sowie Maus und Tastatur einstellen.

Mit dem Kommando „xset q“ zeigen Sie alle einstellbaren Parameter und ihre aktuellen Werte an.

## Umsetzungsidee

- Hardware: RPi4
- Software:
  - Debian base-install
  - [displaycameras](#), auf Basis von
  - [OMXPlayer](#)
  - Alternative: [motioneyeos](#) noch nicht genau angeschaut, taugt das dafür?

Grundumsetzung:

- Test RTSP Stream in OMXPlayer
- Statisches Displaygrid anlegen für die Kameras (auf 2 Monitoren)
- Kurzes Beispiel für [displaycameras](#):

<https://selfhostedhome.com/raspberry-pi-video-surveillance-monitor/>

Weitere Ideen:

- Bildunterschrift für jede Kamera (mit Standort info) -> Kann das OMXPlayer Subtitle-Feature hierfür genutzt werden? -> dann Repo von displaycameras forken und Änderungen dort machen
- Alle Streams als Kleinbild auf 1 Monitor, auf 2. Monitor wird ein selektierter Stream groß angezeigt
  - Steuerung über Network-Socket?
  - Steuerung von LARDIS aus möglich?
    - Selektieren einer gewünschten Kamera
    - Automatisches Umschalten der Türkamera bei Klingel
- 2 unterschiedliche Configs? 1x ELZ (mit schaltbarem Großbild), 1x Lageraum (statisch)

From:

<https://10.4.110.13:8082/> - **DokuWiki**

Permanent link:

<https://10.4.110.13:8082/doku.php?id=iuk:intern:wawa:videoueberwachung:videodisplay&rev=1684483150>

Last update: **2023/05/19 07:59**

