

Tutorial

<https://tutorials-raspberrypi.de/>

<https://www.raspberrypi.org/documentation/installation/installing-images/linux.md>

Image auf SD installieren:

```
unzip 2020-05-27-raspbian-buster-full-armhf.zip |  
dd bs=4M of=/dev/mmcblk0p1 iflag=fullblock oflag=direct status=progress;  
sync
```

Image kopieren

```
sudo dd if=/dev/[SD CARD DEVICE NAME] of=[IMAGE FILE NAME].iso bs=4M  
status=progress
```

Bei if= den Devicenamen ohne Partitions-ID nehmen, also "/dev/mmcblk0 ", nicht "/dev/mmcblk0p1".
Dann landen zwei Partitionen in einem Image.

Pishrink: <https://github.com/Drewsif/PiShrink> verkleinert Image, sodass es sich beim ersten boot automatisch auf die Grösse der neuen SDcard vergrößert:

```
odysseus3:/run/media/thommie/ab51503b-6005-4c24-960f-415e289a2eb1 #  
pishrink.sh -v raspi05_backup2.img raspi05_backup2_shrink.img  
pishrink.sh v0.1.3  
pishrink.sh: Copying raspi05_backup2.img to raspi05_backup2_shrink.img...  
...  
pishrink.sh: Gathering data ...  
Creating new /etc/rc.local  
pishrink.sh: Checking filesystem ...  
rootfs: 116220/3852672 files (0.3% non-contiguous), 1578868/15525376 blocks  
resize2fs 1.46.5 (30-Dec-2021)  
pishrink.sh: Shrinking filesystem ...  
resize2fs 1.46.5 (30-Dec-2021)  
Resizing the filesystem on /dev/loop0 to 1425936 (4k) blocks.  
Begin pass 2 (max = 473157)  
Relocating blocks XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
Begin pass 3 (max = 474)  
Scanning inode table XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
Begin pass 4 (max = 9713)  
Updating inode references XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
The filesystem on /dev/loop0 is now 1425936 (4k) blocks long.  
  
pishrink.sh: Shrinking image ...
```

```
pishrink.sh: Shrunk raspi05_backup2_shrink.img from 60G to 5.7G ...
```

Kopie raus schreiben

```
sudo dd if=~/.raspbian_backup.img of=/dev/sdf
```

Image mounten

Partitionsstruktur des Images analysieren

```
odysseus3:/mnt # sfdisk -d /srv/downloads/raspi05_backup_310521.img
label: dos
label-id: 0x91906680
device: /srv/downloads/raspi05_backup_310521.iso
unit: sectors
sector-size: 512

/srv/downloads/raspi05_backup_310521.img1 : start=          8192, size=
524288, type=c
/srv/downloads/raspi05_backup_310521.img2 : start=        532480, size=
60047360, type=83
```

Loopback devices erzeugen

```
losetup -P -f /srv/downloads/raspi05_backup_310521.iso
```

oder mit kpartx

```
odysseus3:~ # kpartx -a /srv/downloads/raspi05_backup_310521.iso
```

Dann werden die loop devices unter /dev/mapper erzeugt und sind mount-bar.

```
odysseus3:/mnt # mount -o loop /dev/mapper/loop0p1 /mnt/loop0p0/
```

Danach loop devices mit `losetup -D` entfernen.

Watchdog

<https://diode.io/raspberry%20pi/running-forever-with-the-raspberry-pi-hardware-watchdog-2020/>

OBS compile

```
sudo cmake -DUNIX_STRUCTURE=1 -DENABLE_PIPEWIRE=OFF -D-DENABLE_WAYLAND=OFF
-DBUILD_BROWSER=OFF -DCMAKE_INSTALL_PREFIX=/usr ..
```

```
sudo make -j4
```

OBS mit Software Rendering

OBS bzw. OPENGGL versucht beim Starten standardmässig das Rendering über eine GPU, das geht im RASPI schief. Daher muss man über eine Umgebungsvariable das Software-Rendering hart definieren. Für OBS wird dafür der Desktop Shortcut angepasst:

```
/usr/share/applications/com.obsproject.Studio.desktop
```

und darin

```
Exec=env LIBGL_ALWAYS_SOFTWARE=1 obs
```

Networking

Der RASPI nutzt dhcpcd als dhcp Client. Bridge settings nicht über /etc/interfaces setzen, sondern über systemd-networkd.

https://wiki.archlinux.org/title/systemd-networkd#Network_bridge_with_DHCP

rtmp Server für Streaming

<https://blog.unixweb.de/video-streaming-server-mit-restreamer-2-0/>

<https://blog.unixweb.de/video-streaming-server-auf-einem-raspberry-pi/>

Hier wird als Streaming Lösung der datarhei Restreamer von <https://datarhei.com/> der FOSS GmbH Schweiz in einem Docker Container benutzt. Siehe <https://hub.docker.com/r/datarhei/restreamer>

From:

<https://wiki.netzwissen.de/> - **netzwissen.de Wiki**

Permanent link:

<https://wiki.netzwissen.de/doku.php?id=raspi&rev=1683823044>

Last update: **05/03/2024 - 10:52**

