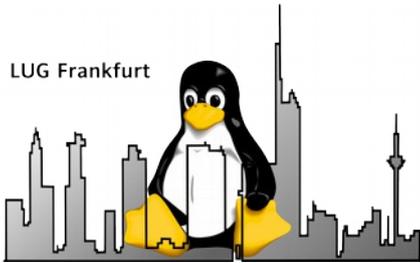


Linux + HTPCs: Teil 2 – Hardware & Verarbeitung von DVB-Streams

von Holger Beetz

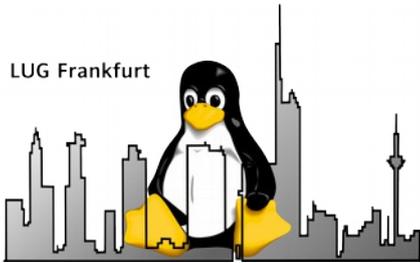
LUG Frankfurt





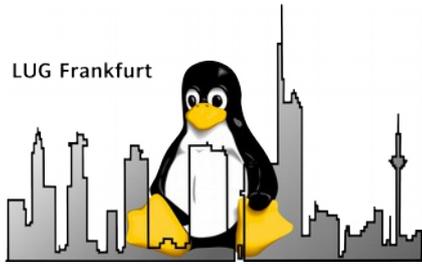
Abgrenzung (Worum geht's nicht !)

- Aufnahme von verschlüsselten Inhalten über DVB-S & DVB-C (Sky, HD+, ...) mittels CI / CAM & Smartcard
- Sat-over-IP / DVB-Network-Devices
- Komplexe DVB-S Anlagen
(Disceq / Unicable / Rotoransteuerung / Sat-Relais / Sat-Weiche)

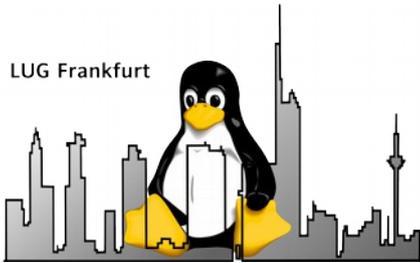


Worum geht's sonst ?

- Grafikkarten
- DVB-T / -C / -S / -S2 Devices →
Worauf sollte man beim Einkauf achten !
- Fernsteuerung(en)
- Alternative Hardware als HTPC
- Verarbeitung von DVB Streams
- Sonstige Quellen für (HD) Videos

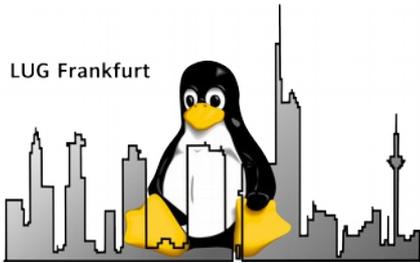


Grafikkarten



Grafikkarten

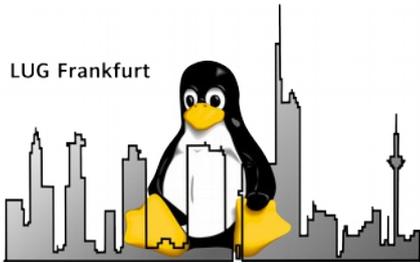
- Sind primär für HDTV / HD-Filme interessant, aber auch SDTV / SD Filme haben Mindestanforderungen (maximale Auflösung, beschleunigtes Video, hardwareunterstütztes Skalieren).
- Besonders bei schwachen Systemen ist GPU Videodecoding gefragt (VDPAU, VA-API für x86).
- SDTV bei VDR über Hollywood+, Full Featured Sat-Karte oder PVR350 (bei Analog TV).
- Nvidia, AMD und Intel haben alle ihre Macken.
- Barebones sind häufig nicht aufrüstbar (Zotac, Shuttle, usw), also hier auf GPU achten.



Grafikkarten (ff)

NVidia

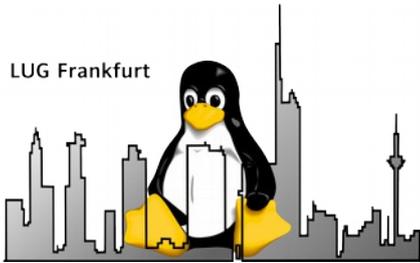
- Gute 3D Performance mit proprietärem Treiber
- Nouveau - 3D "so la la"
- VDPAU für Videobeschleunigung
- Mindestens GeForce 210 für Audio über HDMI.
VDPAU aber schon bei alten Karten ab
GeForce 8800 mit 512MB
- „Primus“ bei der Videobeschleunigung
- Teilweise noch PCI Grafikkarten mit
HD Beschleunigung (GeForce 610)



Grafikkarten (ff)

AMD / ATI

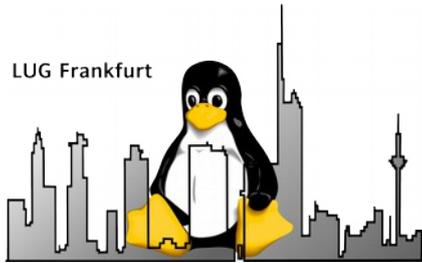
- Gute 3D Performance mit fglrx & radeon
- Vaapi für Videobeschleunigung
- Videobeschleunigung mit den Open Source-Treiber (vaapi)
- Installation von proprietärem Treiber manchmal „wackelig“



Grafikkarten (ff)

Intel

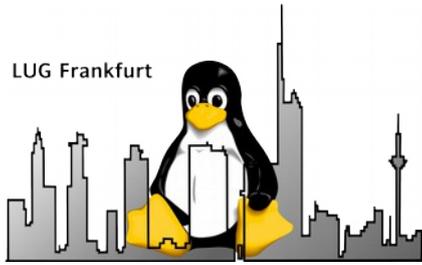
- Nur Intel HD Grafik nutzen (WICHTIG !)
→ kein PowerVR, Intel GMA, usw.
- Mittelmäßige 3D Performance
- Gute Open Source Unterstützung
- vaapi für Videobeschleunigung



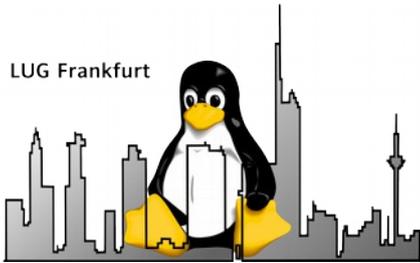
Grafikkarten (ff)

URLs:

- [VDR-Portal empfohlene Systeme für HD](#)
- [VDPAU-Grafikkarten \(VDR Wiki\)](#)

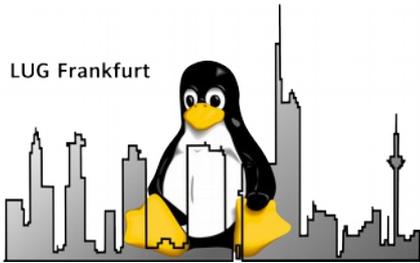


DVB Devices



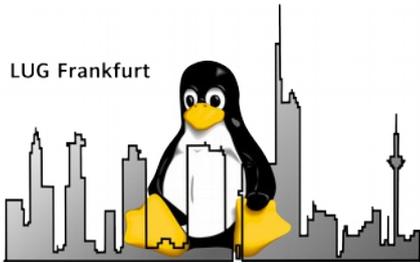
DVB...was ?

- DVB = Digital Video Broadcast
= Digitale Übertragung von Videosignalen
- DVB-T = T wie Terristisch
= „Antennenfernsehen“
- DVB-C = C wie Kabel = „Kabelfernsehen“
- DVB-S = S wie Satellit = „Schüsselfernsehen“
- DVB-S2 = wie DVB-S, aber in HD (h264)
- Analoge Übertragung ist so gut wie tot !



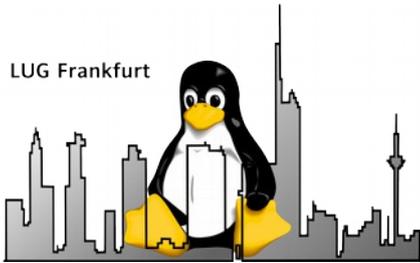
DVB-T Devices

- Das terristische Digital Fernsehen
- Geht fast überall (Theoretisch :-)
- Billige USB-Sticks
- Guter Hardware Support
- Bescheidene Bandbreiten → schlechtere Qualität im Vergleich zu DVB-C und DVB-S
- Häufig Empfangsprobleme
- Gute Antenne mit hoher Position wichtig
- Ablösung durch DVB-T2 zwischen 2016 und 2019 !!!



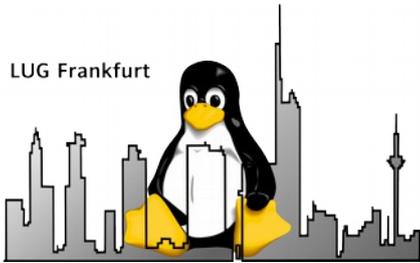
DVB-C Devices

- Aufgrund geringer Verbreitung häufig deutlich teurer als DVB-T oder DVB-S Devices.
- CI / CAM notwendig, wenn man mehr als „Free-to-air“ sehen möchte.
Stichwort: Grundverschlüsselung.
- Immer vom Kabelanschluss abhängig (€€€).
- Geringere Senderauswahl im Vergleich zu DVB-S.
- HDTV Empfang möglich.
- Mehrere Empfänger ohne viel Aufwand möglich.



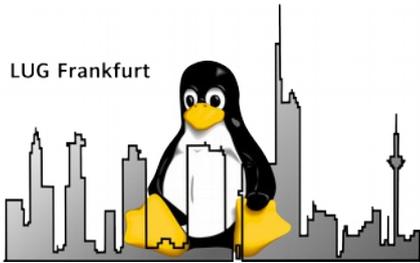
DVB-S / -S2 Devices

- Aufgrund hoher Verbreitung sehr kostengünstig.
- Gute Bildqualität aufgrund hoher Bandbreite.
- HDTV Empfang möglich.
- Große Senderauswahl.
- Viele Mehrfach-Tuner erhältlich.
- Ein LNB pro Empfänger notwendig.
Ansonsten nur Empfang von Sendern auf dem selben Transponder.



DVB Devices – generelle Überlegungen vor dem Kauf

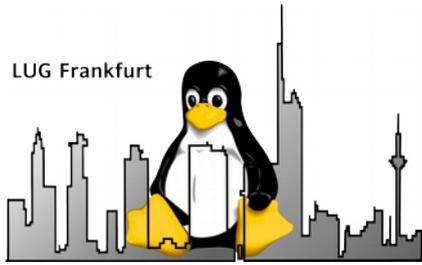
- DVB Device sollte am Besten direkt vom Standardkernel bzw. Video for Linux (V4L) unterstützt werden.
- Alternativ existieren auch Lösungen mit Userspacetreiber (Sundtek), welche aber sehr teuer und oft schwer lieferbar sind.
- Ganz bescheiden ist ein Treiber zum Kompilieren, der vom Hersteller ausgeliefert wird (z.B. DVBSky Devices). Dann muss bei jedem Kernelupdate neu gebaut werden.
- Die exakte Modellbestimmung bzw. Bestimmung der verwendeten Hardware ist häufig schwierig. Manchmal hilft leider nur ein **lspci** oder **lsub** gefolgt von der Suche nach der eindeutigen Geräte ID. Besonders DVB-T Geräte sind solche Wackelkandidaten.



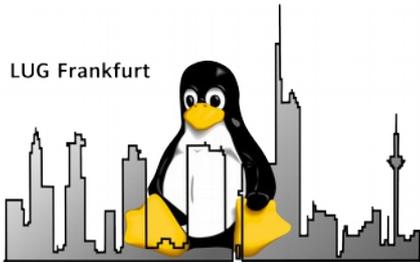
DVB Devices – generelle Überlegungen vor dem Kauf (ff)

- LinuxTV Hardwareübersicht:
http://www.linuxtv.org/wiki/index.php/Hardware_Device_Information
- Ubuntuuser.de Wiki zu DVB Devices:
<http://wiki.ubuntuusers.de/DVB-Karten>
- VDR Wiki:
<http://www.vdr-wiki.de/wiki/index.php/DVB-C>
<http://www.vdr-wiki.de/wiki/index.php/DVB-S>
<http://www.vdr-wiki.de/wiki/index.php/DVB-T>
- VDR Forum allgemein:
<http://www.vdr-portal.de/>

AUGEN AUF VOR(!) DEM KAUF !

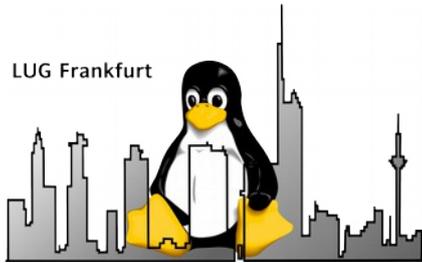


Fernsteuerung(en)

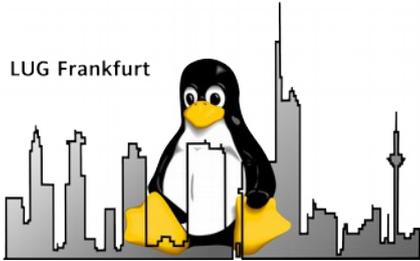


Fernsteuerung(en)

- LIRC mit Standard Fernbedienung
- CIR (Consumer Infrared) mit Standard Fernbedienung
- CEC
 - <http://kodi.wiki/view/CEC>
 - <http://www.pulse-eight.com/store/products/104-usb-hdmi-cec-adapter.aspx>
- X10
 - Microsoft X10 MCE Remotes
- Tablet & Smartphones
 - XBMC:
 - Android: XBMC For Android / Yahtzee / XBMCRemote
 - Apple: XBMCRemote
 - VDR:
 - Android: AndroVDR
 - Apple: vdrRemote (Payware)
- Wireless Tastatur mit Trackball / jeder Remotedesktop

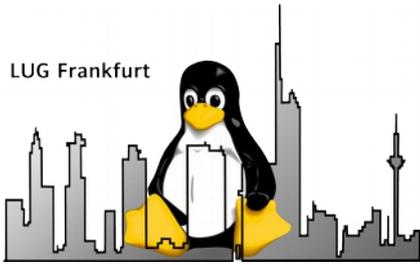


HTPC Alternativen



Alternative Hardware als HTPC

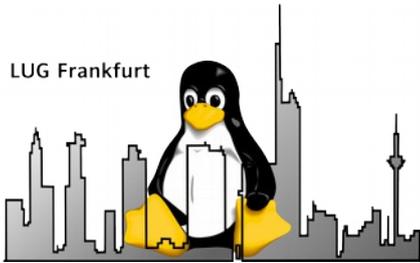
- Raspi:
„King of the Hill“ mit guter XBMC Unterstützung, aber etwas wenig Dampf für DTS- und DTS-HD Ton.
Natives OpenElec Image.
- CuBox-I: Natives OpenElec Image, aber teuer.
- Oroid, BananaPi & Konsorten: XBMC Support und vor Allem GPU-Support häufig sehr wackelig.
- Android Devices generell:
XBMC für Android existiert, aber es gibt häufig Probleme mit HD Filmen. HD Filme lassen sich aber mit anderen Playern oft stressfrei wie auf dem Smartphone abspielen.



Alternative Hardware als HTPC (ff)

Klare Vorteile von ARM:

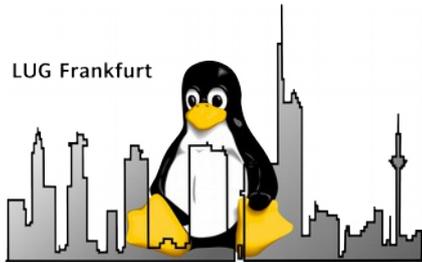
- Sehr stromsparend und mit einfachen Netzteilen betreibbar.
- Vergleichsweise hohe Performance.
- Kompakte Bauform mit günstigen Gehäusen.
- Normale Linux-Distributionen verfügbar
→ Viele Standardfunktionen nutzbar
(NFS, FTP, SSH, Samba, OwnCloud, etc).
- Mit gewissen Einschränkungen als Recording Backend nutzbar.



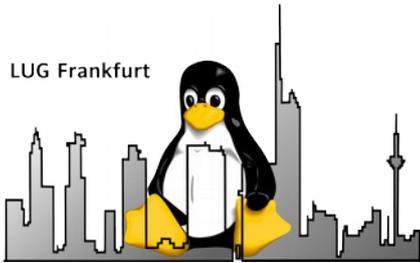
Alternative Hardware als HTPC (ff)

Klare Nachteile von ARM:

- Unterstützung externer Hardware häufig schwieriger als bei x86 Hardware. Gerade bei DVB-Geräten.
- Oft unverständlich hoher Preis in der Summe (> 80€)
Gerade mit Zubehör (Gehäuse, eMMC Card, usw.)
- Performance etwas geringer als „gleichwertige“
x86 Hardware → Lieber Mehrkernprozessor kaufen
- Häufig unnötige Macken (wie Raspi mit instabiler USB
Implementierung und NIC Anbindung über USB)

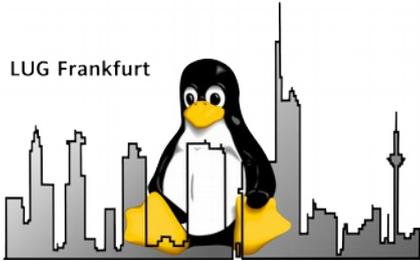


Verarbeitung von DVB Streams



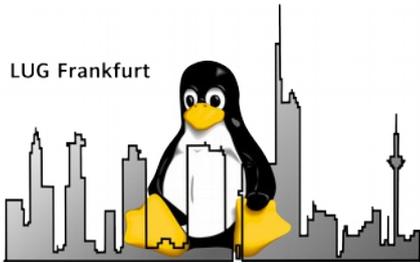
Verarbeitung von DVB Streams

- SD Aufnahmen in MPEG2 lassen sich am Besten mit ProjectX oder AVIDemux in Version 2.5.x bearbeiten. Häufigste Quellen sind DVB-T, DVB-S und DVB-C in Standard Definition.
- HD Aufnahmen in h264 lassen sich am Besten mit AVIDemux 2.6.x verarbeiten. ProjectX unterstützt kein h264 und AVIDemux 2.5.x ist gelegentlich sehr instabil.
Problem bei MKV: AC3 5.1 goes 2.0
Lösung: Als Zielcontainer MPEG TS nehmen
- Geschnitten wird hierbei immer an den I-Frames. Gespeichert wird im Streamcopy-Modus ohne erneutes Encoden des Videos.



Verarbeitung von DVB Streams (ff)

- DVB Aufnahmen nach Möglichkeit **nicht demuxen.** Hierbei wird häufig der Ton asynchron.
- Bricht die Speicherung der geschnittenen Aufnahme im Streamcopy-Modus mit AVIDemux ab, so sind häufig Fehler im Stream. Vorheriges Streamcopy des ungeschnittenen Videos mit TSMuxer, Project X, avconv oder ffmpeg helfen manchmal. Ansonsten kann man das geschnittene Video auch neu in einem Codec nach Wahl (re)encodieren.

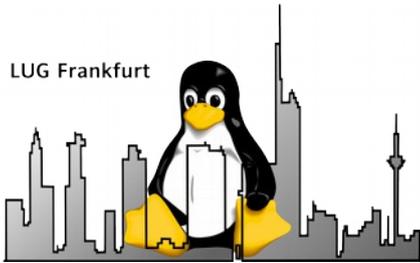


Verarbeitung von DVB Streams (ff)

- Mehrere TS-Segmente lassen sich binär aneinanderhängen:
*cat *.ts >> output.ts*

Die so gewonnene Datei lässt sich problemlos weiterverarbeiten. Das klappt häufig auch mit Aufnahmen von Satreceivern. SMPlayer zickt bei der Wiedergabe solcher Dateien häufig rum. VLC hingegen geht.

- Handbrake kann komfortabel SDTV und HDTV Sendungen konvertieren. Auch solche, die beim Öffnen mit AVIDemux einen Absturz verursachen. Nachfolgend kann man das encodete Video schneiden.



Verarbeitung von DVB Streams (ff)

- Konvertieren von TS-Dateien in MKV mittels avconv.

Zunächst den Inhalt der Datei feststellen:

```
avconv -i input.ts
```

...

```
Stream #0.0[0x13ed]: Video: h264 (High), yuv420p, 1280x720 [PAR 1:1 DAR 16:9], 50 fps, 50 tbr, 90k tbn, 100 tbc
```

```
Stream #0.1[0x13ee](deu): Audio: mp2, 48000 Hz, stereo, s16p, 256 kb/s
```

```
Stream #0.2[0x13ef](mis): Audio: mp2, 48000 Hz, stereo, s16p, 256 kb/s
```

```
Stream #0.3[0x13f2](deu): Audio: ac3, 48000 Hz, stereo, fltp, 448 kb/s
```

```
Stream #0.4[0x13f1](deu): Subtitle: dvbsub (hearing impaired)
```

....

Nun kopieren wir die AC3-Tonspur und das Video in eine MKV-Datei:

```
avconv -i input.ts -c:v copy -map 0:0 -c:a copy -map 0:3 output.mkv
```

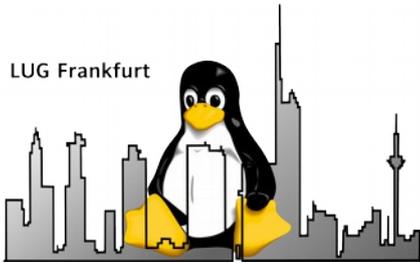
-c:v copy = Videostream kopieren

-c:a copy = Audiostream kopieren

-map 0:0 = Nur Videostream 0 mitnehmen

-map 0:3 = Nur AC3 Audiostream 3 mitnehmen

Defekte Segmente des DVB-Streams werden für Audio und Video gleichermaßen gedropped und der Stream bleibt synchron.



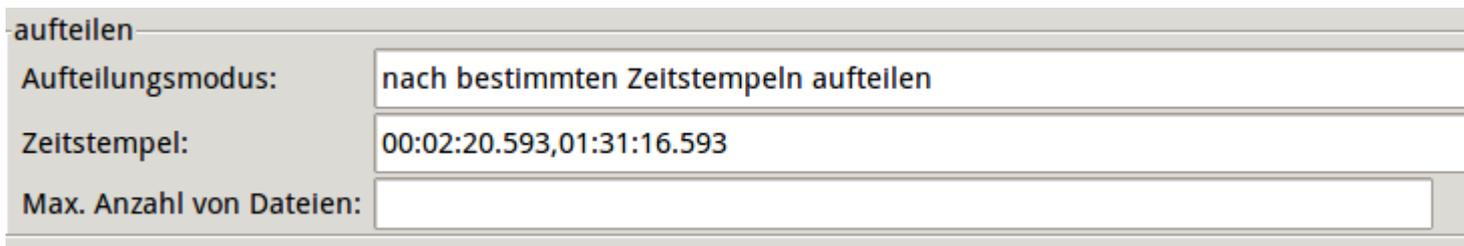
Verarbeitung von DVB Streams (ff)

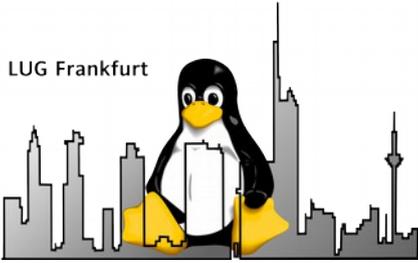
Schneiden von MKV Dateien mit der Matroska Merge GUI

Zeitcode für Schnittpunkte aus AVIDemux kopieren:



Zeitcodes in Matroska Merge GUI einfügen:

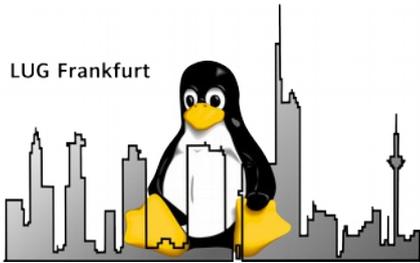




Schneiden mit VDR



Vdr-sxfe in Schnittdarstellung



Schneiden mit VDR (ff)

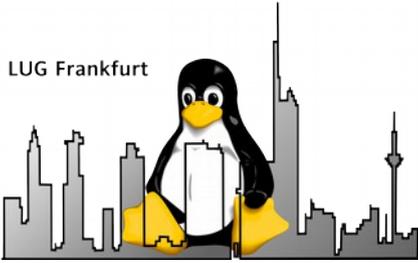
- Aufzeichnung wiedergeben
- Einblenden der "Timeline" mit Ok
- Schnellvorlauf bzw. Rücklauf zu den Schnittpunkten
- Schnittmarke setzen mit 0
- Wiederholen bis alle Werbeblöcke weg sind
- Anwählen von gesetzten Schnittmarken mit 7 oder 9
- Verschieben gewählten Schnittmarken mit 4 oder 6
- Starten der Schnittoperation mit 2

Wichtig: Originalaufnahme und Kopie bleiben erhalten -> Doppelter Festplattenbedarf

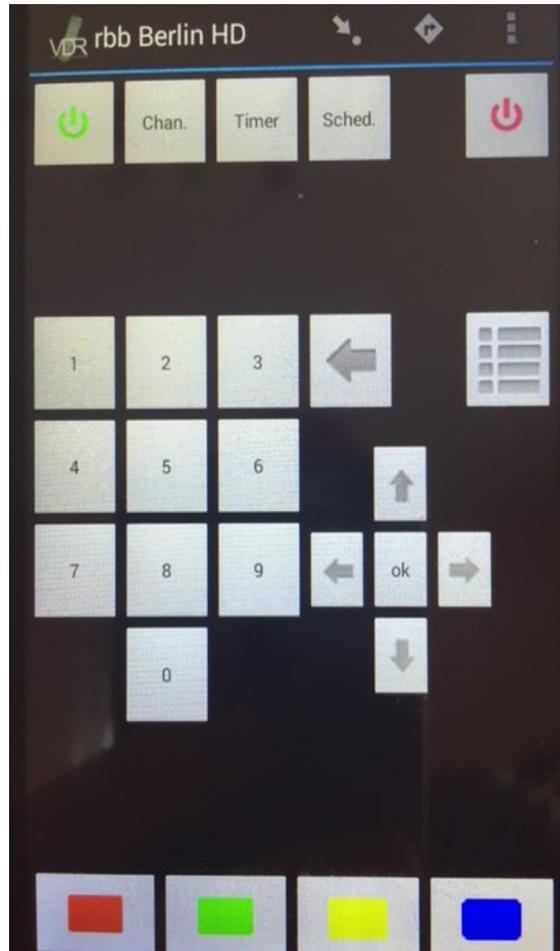
Nachteil: Spulen ist leider relativ träge über die Fernbedienung.

Siehe auch:

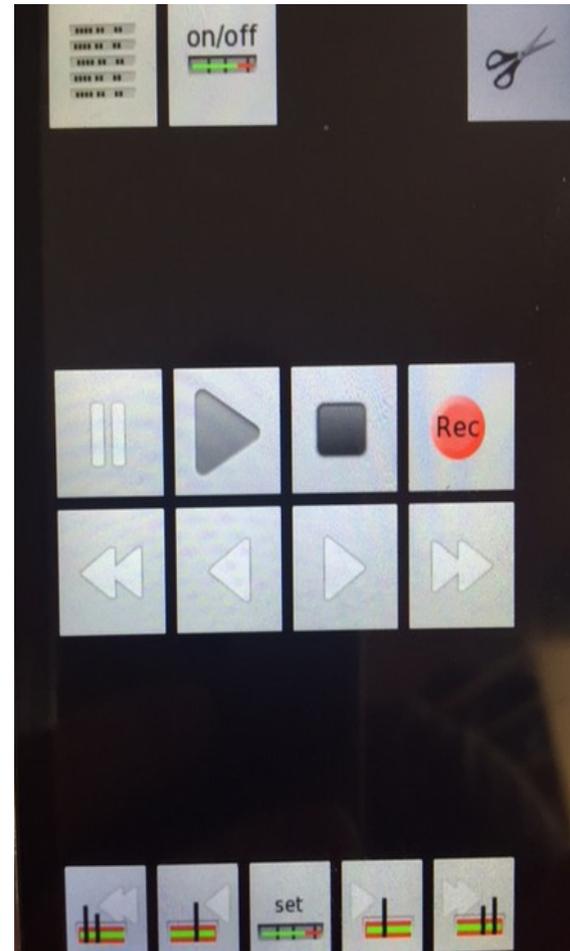
http://www.vdr-wiki.de/wiki/index.php/Benutzerhandbuch#Schneiden_einer_Aufzeichnung



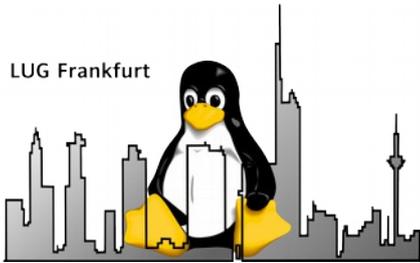
Schneiden mit VDR (ff)



AndroVDR – Standard Layout



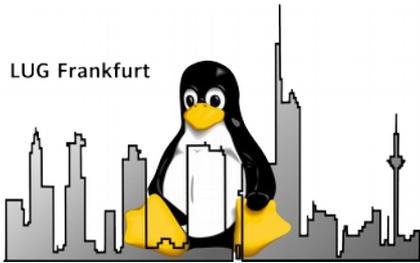
AndroVDR – Schnitt Layout



Schneiden mit VDR (ff)

Stolpersteine bei der Anbindung von AndroVDR und Konsorten:

- IP Adresse oder IP Range des Smartphones muss svdrphosts.conf freigegeben sein.
- Port 6419 für SVDRP (Simple VDR Protocol) muss in der Firewall freigegeben sein (*Hüstel*).
- Optimalerweise sollte nur ein Gerät / Programm mit dem VDR über SVDRP sprechen, weil es sonst zu Funktionsstörungen kommen kann. Also besser z.B. nicht vdradmin-am und AndroVDR gleichzeitig nutzen.
- Aufgrund des längeren Weges (WLAN, LAN, SVDRP Protocol) reagiert AndroVDR manchmal beim Schneiden / Spulen etwas träger.

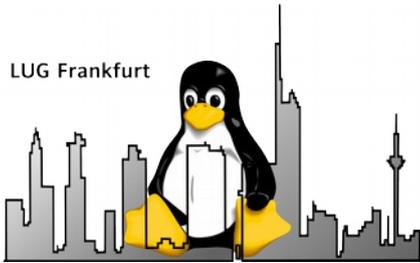


DVB-Verarbeitung

Eine einheitliche Gesamtlösung wie TSDoctor oder DVR-Studio fehlt leider mit Features wie:

- Automatischer Werbeerkenennung
 - Beseitigung von DVB Streamfehlern
 - Entfernen von NALU Filler Packages
 - Export als BluRay, DVD und in andere Formate (mit Profilen für Smartphone, Tablet, usw)
 - Schnelles Scrollen & verlustfreies Schneiden
- Naheliegende Lösung:
Linux VDR plus Scripte & Tools (nächste Folie)

Siehe hierzu: </etc/vdr/recording-hooks/R90.custom>

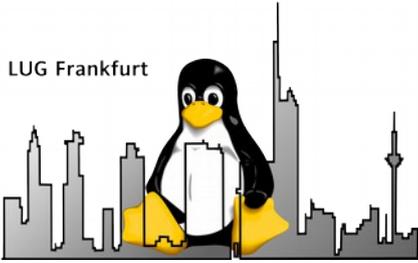


DVB-Verarbeitung (ff)

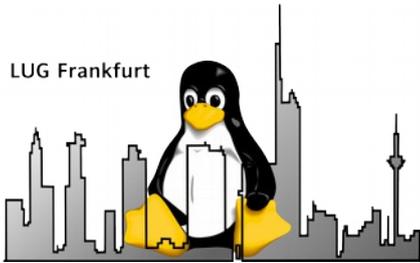
Automation ist Trumpf:

- Automatische Werbeerkenkung:
noad oder *markad* (als VDR Plugins).
- Erzeugung & Konvertierung per *avconv*, *ffmpeg*, *mencoder*, *mkvmerge*.
- Erzeugung BD Struktur per *TSMuxer*.
- Automatisches Zusammenfügen von TS-Dateien:
processTS (als VDR Postprocess Kommando).
- Automatische AC3 Erkennung und Remuxen nach MKV:
findAC3 (in der Shell oder als Befehl im Filemanager).

Die Python Skripte *processTS* und *findAC3* gibt es später als Anhang zum Vortrag.



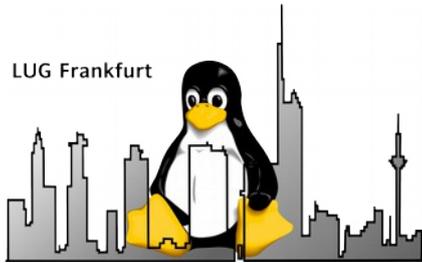
Sonstige Quellen für (HD) Videos



Sonstige Quellen für (HD) Videos

- Mediathek View - Zugriff auf Mediatheken der ÖR:
<http://zdfmediathk.sourceforge.net/>
- Youtube (und andere Videoportale):
Youtube-dl - Textbasiertes Tool:
<http://rg3.github.io/youtube-dl/>

Frontend für Youtube-dl:
<https://code.google.com/p/ytd-gtk/>
- Podcasts



Danke
für's
Zuhören
und
euer
Interesse !

