



Rüdiger Follmann

Der PC als Medienzentrale

Betriebssysteme, Fernsehen, Videos,
Spiele, Office, Netzwerke, Datenbanken

EBOOK INSIDE

 **Springer** Vieweg

Der PC als Medienzentrale

Rüdiger Follmann

Der PC als Medienzentrale

Betriebssysteme, Fernsehen, Videos,
Spiele, Office, Netzwerke, Datenbanken

Rüdiger Follmann
Vice President
IMST GmbH
Kamp-Lintfort, Deutschland

Ergänzendes Material zu diesem Buch finden Sie auf <https://www.springer.com/gp/book/9783662612842>

ISBN 978-3-662-61284-2 ISBN 978-3-662-61285-9 (eBook)
<https://doi.org/10.1007/978-3-662-61285-9>

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© Springer-Verlag GmbH Deutschland, ein Teil von Springer Nature 2020

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von allgemein beschreibenden Bezeichnungen, Marken, Unternehmensnamen etc. in diesem Werk bedeutet nicht, dass diese frei durch jedermann benutzt werden dürfen. Die Berechtigung zur Benutzung unterliegt, auch ohne gesonderten Hinweis hierzu, den Regeln des Markenrechts. Die Rechte des jeweiligen Zeicheninhabers sind zu beachten.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag, noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer-Verlag GmbH, DE und ist ein Teil von Springer Nature. Die Anschrift der Gesellschaft ist: Heidelberger Platz 3, 14197 Berlin, Germany

Danksagung

Ein neues Jahr - ein neues Buch. Nach dem Buch über den Raspberry Pi habe ich festgestellt, dass viele Dinge nicht mit diesem kleinen Rechnerzweig realisiert werden können, die ich von einem Wohnzimmer-PC erwarte. Als VDR-Nutzer der ersten Stunde habe ich mich darum wieder in den Kampf mit Ausgabepugins und Co. auf PC-Basis begeben. Daher gilt ein großer Dank Klaus Schmidinger, der sofort bereit war, neben seiner unermüdlichen Arbeit am VDR das Geleitwort für dieses Buch zu schreiben. Vielen Dank dafür, Klaus!

Vor dem Schreiben und während des Schreibens eines Buches ist viel Programmier- und Bastelarbeit angesagt. Da ich die nicht alleine im Keller verrichten wollte, sah unser Wohnzimmer oft aus wie eine kleine Computerwerkstatt. Sorry, liebe Familie und danke für Euer Verständnis. Und nein, das Lattepanda-Netzteil gehörte nicht zur Weihachtsbaum-Dekoration. Sorry Jonas, dass ich schon wieder so ein Fernseh-Ding gebastelt habe, mit dem keiner zurecht kommt.

Da meine Frau inzwischen ein tiefes Verständnis für den Raspberry Pi hat, weil sie das letzte Manuskript gelesen hat, freue ich mich sehr, dass ich dieses Verständnis um X86-Architekturen erweitern konnte. Danke Nine für das Korrekturlesen des Manuskripts und für die vielen kleinen Hinweise zum sprachlichen und stilistischen Feinschliff.

Nicht zuletzt bedanke ich mich bei meinem Verleger Martin Börger für alle Diskussionen um dieses Buch und alle Hinweise. Vielen Dank Herr Börger, Sie waren mir immer eine große Hilfe.

Thank you Yiyi for the Lattepanda Alpha 864 gift, which I received from DFRobots. In addition, I would like to thank you for organizing the visit in Shanghai. Thank you Ricky, I enjoyed our meeting very much. You guys are doing a great job.

Vielen Dank, Herr Kingler, für das Zusenden der TBS-5580. Dieser DVB-Tuner funktioniert hervorragend unter LINUX und bringt sogar noch ein CI-Interface mit.

Natürlich darf auch hier der Dank an alle Programmierer freier Software nicht fehlen. Ohne sie wäre dieses Buch nicht möglich gewesen. Apropos Buch: Unter <https://www.springer.com/de/book/978-3-662-61284-2> finden Sie alle Downloads zum Buch. Außerdem gibt es dort die Errata.

Mettmann, im Februar 2020
Rüdiger Follmann



Inhaltsverzeichnis

	Geleitwort	XI
1	Einleitung	XIII
1.1	Der Inhalt des Buches	XIII
1.2	Die Zielgruppe des Buches	XIV
1.3	Der Aufbau des Buches	XV
2	Der Wohnzimmer-PC	1
2.1	Die Qual der Wahl	1
2.2	Der Eigenbau	3
2.3	X86 Embedded Boards	4
2.4	Grafik unter LINUX und Mac	7
2.4.1	Intel	10
2.4.2	NVidia	11
2.4.3	AMD	11
2.4.4	MacOS	11
2.5	Das Wichtigste in Kürze	12
3	Hackintosh	15
3.1	So ein Hack-Mac	15
3.2	Vorbereitung der Festplatte	17
3.3	Clover & Co	20
3.3.1	Erstellen eines Bootsticks unter Windows	20

3.3.2	Erstellen eines Bootsticks unter MacOS	23
3.4	Booten von USB – Die Installation	27
3.5	Der Bootloader	29
3.6	Probleme und Lösungen	34
3.7	Praxisbeispiele	35
3.7.1	Praxisbeispiel iMovie	36
3.7.2	Praxisbeispiel Fortnite	40
3.8	Fazit	43
4	Ubuntu	47
4.1	Ubuntu „LTS“ und „aktuell“	47
4.2	Vorbereitungen	47
4.3	Installation	48
4.4	Desktop	60
4.5	Grafiktreiber	61
4.5.1	Intel	62
4.5.2	NVidia	63
4.5.3	AMD	63
4.6	Einstellungen	65
4.7	Das System aktuell halten	68
5	LINUX Grundlagen	73
5.1	Netzwerke einrichten	73
5.2	Editoren	77
5.3	Fernzugriff	80
5.4	OpenVPN	89
5.5	Programme übersetzen oder interpretieren	95
5.6	Ein neuer Kernel	99
5.7	veracrypt	101
5.8	Kleider machen Leute	106
6	Spiele	111
6.1	Nintendo Wii U-Emulator unter Windows	111
6.2	Nintendo Wii U-Emulator unter LINUX	116
6.3	Nintendo Switch-Emulator	122
6.4	Nintendo Gamecube & Wii-Emulator	126

6.5	N64 Emulator	127
6.6	LINUX Spiele	131
7	Video Disk Recorder	137
7.1	DVB-Sticks	138
7.1.1	Sundtek	138
7.1.2	DVBSky S960 V2	142
7.1.3	Neuere Modelle	143
7.2	Fernbedienung mit (F)lirc	145
7.3	VDR	152
7.4	Skripte	161
7.5	Speichern und servieren	166
7.6	Schöner fernsehen	169
7.7	Stream me up!	194
7.8	Pay-TV	196
7.9	Wichtige Hinweise	199
8	KODI	209
8.1	<i>KODI</i> aus dem Repository	210
8.2	<i>KODI</i> aus dem Quelltext	210
8.3	LibreELEC	214
8.3.1	Einrichten einer <i>lirc</i> -Fernbedienung für <i>LibreELEC</i> Version 8 und kleiner	219
8.3.2	Einrichten einer <i>lirc</i> -Fernbedienung für <i>LibreELEC</i> Version 9 und neuer	220
8.3.3	Der Wetterdienst	222
8.3.4	Tvheadend zum Fernsehen	223
8.3.5	DVD und Blu-ray	227
8.3.6	Amazon Prime	228
8.4	Mobile Helfer	229
8.4.1	TvhClient	229
8.5	AirPlay und Co.	230
8.6	MP3	233
8.7	Wichtige Hinweise	233
9	Ambilight	237
9.1	Ambilight-Einkaufsliste für <i>LibreELEC</i>	237
9.2	Ambilight Hardware anschließen	238

9.3	Ambilight Software installieren	242
9.4	Ambilight Fernbedienung	252
9.5	Ambilight ein- und ausschalten	252
10	Server und Datenbanken	261
10.1	AirPrint	261
10.2	Servieren à la Python	267
10.3	Apache	268
10.3.1	PHP	270
10.3.2	MySQL	271
10.4	CMS	273
10.5	DLNA	281
10.6	Philips Hue	285
11	Schreiben mit dem Wohnzimmer-PC	289
11.1	<i>LibreOffice</i>	289
11.1.1	<i>LibreWriter</i>	291
11.2	\LaTeX	292
11.2.1	\LaTeX -Beispiel	293
11.2.2	Bewerbungen schreiben	296
11.2.3	Rechtschreib-Prüfung	300
11.2.4	Eine einseitige Bewerbung	301
11.2.5	Ein Buch schreiben	303
11.3	PlayOnLinux	304
	Anhang	311
	Fehlersuche	311
	Die wichtigsten LINUX-Befehle	313
	Literatur	315
	Abbildungsverzeichnis	325
	Tabellenverzeichnis	331
	Index	333

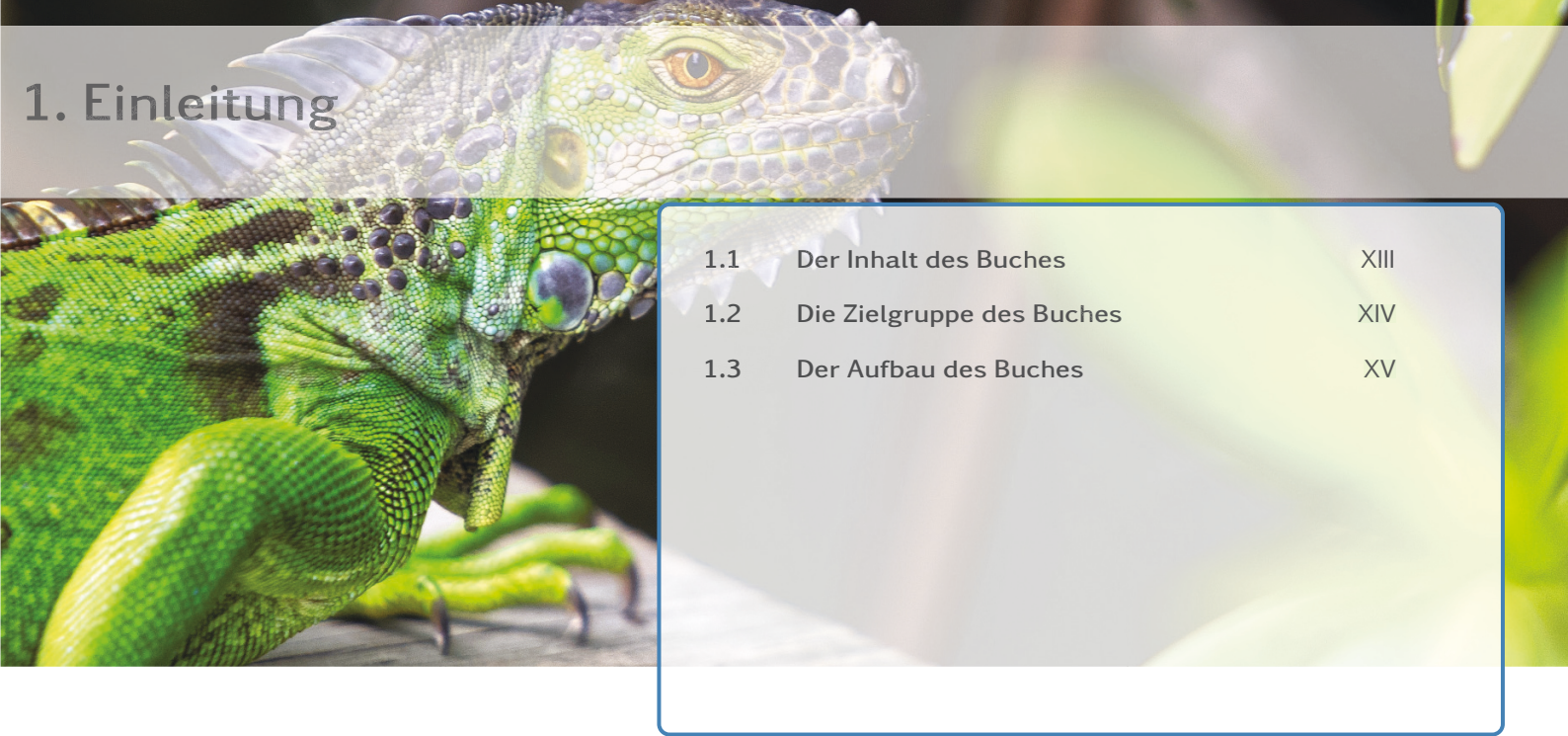


Als ich im Februar des Jahres 2000 die allererste Version der VDR-Software ins Netz stellte, war dies zum großen Teil das Resultat meines Ärgers über die grottenschlechte Firmware meines damaligen (noch analogen) Satelliten-Receivers. Auch musste man seinerzeit noch Sat-Receiver und Video-Recorder separat programmieren, wenn man eine Sendung aufnehmen wollte. Zum Glück erschienen Ende der 1990er Jahre die ersten digitalen TV-Empfangskarten für den PC. Und als dann auch noch ein Treiber für Linux verfügbar wurde, konnte ich meinen Traum von einem wirklich brauchbaren digitalen Empfangs- und Aufnahmegerät verwirklichen.

Heute bekommt man schon für wenig Geld digitale Festplatten-Receiver, die oft deutlich mehr können als der gute, alte VDR. Die Tradition der teilweise umständlichen Bedienung hat sich aber bei so manchem Hersteller erstaunlicherweise gehalten. Um so besser ist es da, wenn man die Software vollständig selber in der Hand hat und sie seinen Wünschen und Erfordernissen entsprechend anpassen kann. Im Laufe der Jahre hat VDR alle wichtigen Entwicklungen im Bereich DVB mitgemacht. DiSEqC, AC3, VPS, SCR, HDTV, UHD, CICAM (inzwischen auch mit MTD, also dem gleichzeitigen Entschlüsseln von Programmen mehrerer Transponder mit nur einem CAM) und vieles mehr konnte dank der Mithilfe zahlreicher Entwickler unter den VDR-Benutzern realisiert werden. Und auch heute, zwei Jahrzehnte nach den ersten Anfängen, erfreut sich VDR immer noch einer großen User-Gemeinde.

Dass Rüdiger Follmann in seinem neuesten Buch dem VDR ein großes Kapitel widmet, ist für mich zum zwanzigsten „Geburtstag“ des Video-Disk-Recorders ein ganz besonderes Geschenk!

Taufkirchen, Februar 2020
Klaus Schmidinger



1. Einleitung

1.1	Der Inhalt des Buches	XIII
1.2	Die Zielgruppe des Buches	XIV
1.3	Der Aufbau des Buches	XV

Dieses Buch wendet sich an alle Nutzer, die einen PC als Medienzentrale nutzen wollen. Das bezieht sich beispielsweise auf die Themen Fernsehen oder „video on demand“ und schließt auch Anwendungsbereiche ein, bei denen ein PC im Gegensatz zu ARM-Boards [1] seine volle Leistung entfalten kann. Dazu zählen die Themenbereiche Spielekonsole und Emulatoren, aber auch produktive Bereiche wie Videobearbeitung und die Vernetzung mit mobilen Endgeräten. Als Betriebssystem nutzt dieses Buch das freie LINUX-System „Ubuntu“, das sich durch seine Bedienerfreundlichkeit auszeichnet.



1.1 Der Inhalt des Buches

Personal Computer – oder kurz PCs – gibt es in beliebigen Ausprägungen. Die Palette reicht vom kleinen Einsteiger-PC für Office-Anwendungen bis hin zum Spieleboliden mit Wasserkühlung. Soll ein PC aber als Medienzentrale im Wohnzimmer sein Dasein fristen, gibt es einige Kriterien, die es zu beachten gilt. Dazu zählen unter anderem Aussehen und Lautstärke. Daher gibt das erste Kapitel dieses Buches einen Überblick über verschiedene PC-Ausführungen, die sich als Mediacenter eignen. Dabei wird ebenfalls die Grafikfähigkeit verschiedener Systeme unter LINUX beschrieben (OpenGL, Grafikbeschleunigung, Dekodieren und Enkodieren).

Bevor wir im dritten Kapitel ein Ubuntu-LINUX installieren, werfen wir im zweiten Kapitel einen Blick auf das Betriebssystem MacOS, welches sich mit einigen Kniffen auf einer Nicht-Apple-Hardware installieren lässt. Das Buch befasst sich mit der sogenannten „Hackintosh“-Methode und beschreibt sowohl die Vor- als auch die Nachteile dieser Methode. Anhand einer Filmbearbeitung (iMovie) prüfen wir die Alltagstauglichkeit dieses Systems.

In einer Schritt-für-Schritt-Anleitung erklärt das nächste Kapitel dieses Buches die Installation eines LINUX-Betriebssystems am Beispiel von „Ubuntu“. Dieses System ist kostenlos und zeichnet sich unter anderem durch seine große Bedienerfreundlichkeit aus. Das Starten mehrerer verschiedener Betriebssysteme mit einem Bootloader wird ebenso beschrieben wie die Emulation von Windows-Programmen mittels Wine.

Das Kapitel Netzwerke zeigt, wie man die Medienzentrale in das heimische Netzwerk einbindet, auf sie zugreift oder Dateien transferiert und verschlüsselt. Das Einbinden erfolgt dabei entweder mit Hilfe eines Ethernet-Kabels mit dem heimischen Router oder per WLAN. Da ein PC oft sehr leistungsfähig ist, kann er ebenfalls als Backend für Textverarbeitung oder andere Programme verwendet werden. Das Kapitel beschreibt ebenfalls, wie der Remote-Zugriff erfolgen kann.

Was wäre eine Multimedia-Zentrale ohne die Möglichkeit, fernzusehen oder „videos on demand“ zu schauen? Ein großes Kapitel dieses Buches ist dem Thema Multimedia gewidmet. Nutzen Sie Ihr Media-Center, um Fernsehsendungen aufzuzeichnen oder Blu-rays zu schauen. Verbinden Sie Ihren Fernseher mit einem Ambilight und steuern Sie dieses mit Ihrer Medien-Zentrale. Schauen Sie Netflix-Videos oder nutzen Sie Ihren Amazon-Prime oder Spotify-Account.

Wie wäre es mit einem Spiel? Ein an den Fernseher angeschlossener PC bietet hier alle Möglichkeiten. Ein weiteres großes Kapitel ist daher dem Thema „Spielen“ gewidmet. Verbinden Sie Ihre Controller mit dem Medien-PC und spielen Sie moderne Spiele auf einer Wii-Emulation. Dieses Buch zeigt Ihnen, wie Sie die entsprechende Software installieren und nutzen können. Spielen Sie die alten Klassiker wie Super Mario. Selbst viele Windows-Spiele funktionieren mit Wine einwandfrei.

Das Kapitel „Server und Datenbanken“ erklärt Schritt für Schritt das Einbinden eines Druckers für AirPrint und zeigt Ihnen, wie Sie eine Webseite servieren können. Weiterhin setzen wir in diesem Kapitel einen DLNA-Server auf, der es erlaubt, Musik, Filme und Bilder im Heimnetz zu verteilen. Weiterhin stelle ich Ihnen Skripte vor, mit denen Sie Ihre Philips Hue-Lampen steuern können.

Alle Kapitel dieses Buches werden begleitet von kleineren oder größeren Code-Ausschnitten und Programmen in verschiedenen Programmiersprachen. Daher ist ein weiteres Kapitel diesem Thema gewidmet. Lernen Sie, Programmiersprachen unter LINUX einzusetzen und anzuwenden. Suchen Sie sich für Ihre Applikation die geeignete Sprache oder übersetzen Sie freie Software aus dem Internet.

1.2 Die Zielgruppe des Buches

Dieses Buch richtet sich sowohl an LINUX-Anfänger als auch an fortgeschrittene Benutzer. In Schritt-für-Schritt-Anleitungen werden alle Inhalte des Buches verständlich gemacht. Bilder und Skizzen unterstützen Sie dabei, das Gelernte in die Tat umzusetzen. Mit Hilfe dieses Buches verwandeln Sie Ihren PC in ein Multimedia-Center, nutzen ihn zum Servieren von Webseiten und setzen ihn im produktiven Bereich ein.

Anfänger haben so schnelle Erfolgserlebnisse, Profis erhalten wertvolle Tipps und ein Nachschlagewerk.

1.3 Der Aufbau des Buches

Jedes Kapitel enthält zunächst eine kleine Übersicht darüber, was in diesem Kapitel abgehandelt wird.

Anmerkungen werden wie in diesem Beispiel eingerückt und kursiv dargestellt. Anmerkungen enthalten hilfreiche Hinweise und Tipps, die den Text in den Kapitelabschnitten dieses Buches unterstützen und die einzelnen Tipps hervorheben.



Hinweis 1.1 (Grund des Hinweises) Wichtige Hinweise stehen in diesem umrandeten Kasten. Sie gehen über Anmerkungen hinaus und sollten auf jeden Fall beachtet werden. Hierzu zählt z. B. das Ziehen des Netzsteckers vor dem Öffnen eines PC-Gehäuses und dem Einbau eines Speicher-Moduls.

Code-Ausschnitte werden in einer eigenen Box dargestellt und sehen wie folgt aus:

Code-Ausschnitt 1.1: Schleife

```
#!/bin/bash
i=0
while [ $i -le 10 ]
do
    echo $i
    let i=$i+1
done
}
```

Alle Downloads zu diesem Buch finden Sie unter <https://www.springer.com/de/book/978-3-662-61284-2>.



Zusammenfassung und Ausblick 1 Das war die Einleitung zum Buch „Der PC als Medienzentrale“. Ich hoffe, dass Sie beim Lesen des Buches ebenso viel Spaß haben wie ich beim Schreiben. Nach der Einleitung geht es los mit einer Übersicht über mögliche wohnzimmertaugliche PCs. Starten Sie mit der Wahl einer Hardware, die Ihren Bedürfnissen gerecht wird. ■

Ich bedanke mich bei der Firma DFRobot (<https://www.dfrobot.com> [2]) aus Shanghai, die so nett war und mir kostenlos einen Lattepanda Alpha 864(s) zur Verfügung gestellt hat, den ich nutzen durfte, um dieses Buch zu schreiben. Weiterhin bedanke ich mich bei der Firma TBS Technology, die mir eine TBS-5580 Empfangsbox zur Verfügung gestellt hat.

Literaturverzeichnis

- [1] Rüdiger Follmann. *Das Raspberry Pi Kompendium, 2. Auflage*. Springer Verlag, Heidelberg, 2018. ISBN: 978-3-662-58143-8 (siehe Seiten XIII, 4).
- [2] DFRobot. *Webseite*. 2019.
<https://www.dfrobot.com>
besucht am 23.12.2019 (siehe Seite XV).

2. Der Wohnzimmer-PC



2.1	Die Qual der Wahl	1
2.2	Der Eigenbau	3
2.3	X86 Embedded Boards	4
2.4	Grafik unter LINUX und Mac	7
2.5	Das Wichtigste in Kürze	12

Dieses Kapitel gibt einige Empfehlungen rund um das Thema Wohnzimmer-PC. Dabei werfen wir einen Blick auf verschiedene wohnzimmer-taugliche PCs, die sich unter anderem durch ein schickes Gehäuse auszeichnen und die sogar mit einer passiven Kühlung ohne Lüfter auskommen und dennoch sehr leistungsfähig sind. Dieses Kapitel beleuchtet ebenfalls verschiedene Grafikmöglichkeiten (Videoausgabe, OpenGL).



2.1 Die Qual der Wahl

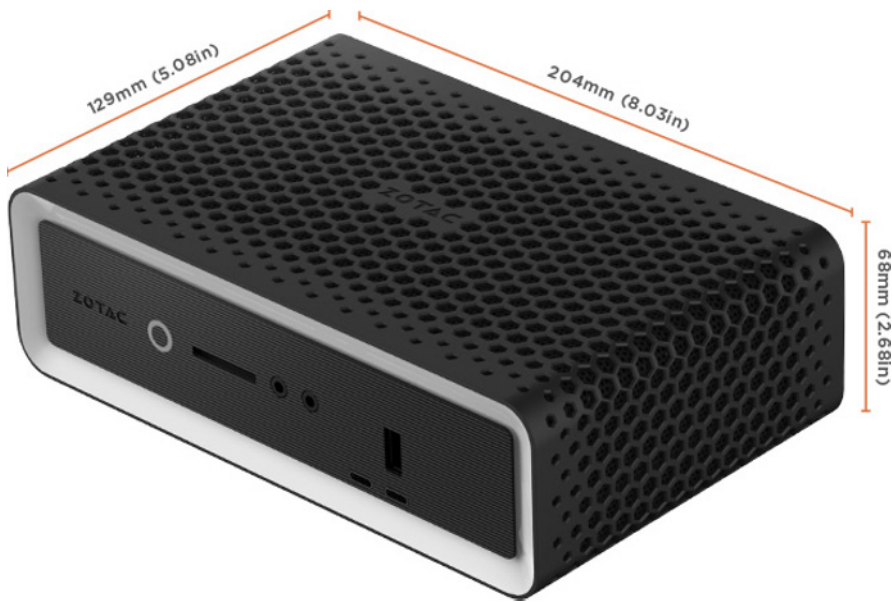


Abbildung 2.1: Der ZOTAC CI660 ist ein luftgekühlter Intel-Bolide.
Quelle: <http://zotac.com> [3].

Ein Wohnzimmer-PC sollte nett aussehen, leise sein und genügend Leistungsreserven für ein Spiel bieten. Alle diese Eigenschaften vereint der passiv gekühlte ZOTAC CI660. Trotz seiner kompakten Abmessungen von gerade einmal 204 mm × 129 mm × 68 mm (L×B×H) können sich die Daten des Rechners durchaus sehen lassen:

- Intel Core i7-8550U (quad core 1.8 GHz, bis 4.0 GHz)
- 2 × DDR4-2400/2133 SODIMM Slot (bis 32 GB)
- Intel UHD Grafik 620 (3840×2160@60Hz, 4096×2160@60Hz)
- 1 × 2.5-Zoll SATA 6.0 Gbps SSD/HDD slot
- 3-in-1 (SD/SDHC/SDXC) Kartenleser
- 2 × USB 3.1, 1 × USB 3.0 (Front), 4 × USB 3.0 (Rückseite)
- 2 × 10/100/1000 Mbps Ethernet
- 802.11ac WIFI und 4.2 Bluetooth
- Digital 8 Channel Audio (digital via HDMI), Mikrofon-Eingang

Auf dem Papier hat dieser PC also alles, was man für ein Mediacenter benötigt: Rechenpower zum Spielen, Decoder und Encoder für hochauflösendes Fernsehen (UHD) und kabelgebundene und kabellose Netzwerk-Möglichkeiten. Audio kann dabei wahlweise analog oder über HDMI digital ausgegeben werden. Sogar ein Kartenleser und ein Mikrofon-Eingang sind mit an Bord. Einzig ein Infrarot-Empfänger fehlt. Diesen kann man aber problemlos mit Hilfe einer der vielen USB-Ports nachrüsten.

Aber sind diese Daten ausreichend, um alle im Buch beschriebenen Projekte umzusetzen?

Wie so oft ist die Antwort auch dieses Mal: Das kommt darauf an. Während ein Windows-Betriebssystem überlicherweise die komplette Hardware eines PCs unterstützt - notfalls durch eigene Treiber der Hersteller - ist diese Kompatibilität unter LINUX nicht immer gegeben. Oft gibt es keine Hersteller-LINUX-Treiber, oder diese werden erst lange nach Erscheinen eines Computers durch die Internet-Gemeinschaft entwickelt. Manchmal fehlen sie auch komplett. Möchten Sie auf ihrem Computer ein MacOS installieren (Hackintosh), empfiehlt sich vor dem Kauf ein Blick auf die Kompatibilitäts-Liste von <https://www.tonymacx86.com/buyersguide/building-a-custom-mac-hackintosh-the-ultimate-buyers-guide/> [4].

Dieser Einkaufsführer gibt Tipps für mögliche Mac-kompatible Aufbauten, auf denen das Apple-eigene Betriebssystem relativ problemlos laufen soll.

Hinweis 2.1 (Die beste Wahl für Hackintosh) Wenn Sie einen PC kaufen möchten, auf dem sich Apples MacOS (Hackintosh) installieren lässt, sollten sie vor dem Kauf unbedingt einen Blick auf die Einkaufsliste von <https://www.tonymacx86.com/buyersguide/building-a-custom-mac-hackintosh-the-ultimate-buyers-guide/> [4] werfen, um spätere Kompatibilitätsprobleme auszuschließen.



Abbildung 2.2: Der Intel NUC7i7DNKE ist Hackintosh-kompatibel

Während moderne INTEL-CPUs in der Regel mit der Hackintosh-Methode funktionieren, ist das bei den verbauten WLAN/Bluetooth-Karten nicht immer der Fall. Dann ist es von Vorteil, wenn die WLAN/Bluetooth-Karten nicht auf dem Mainboard des PCs verlötet sind, sondern sich austauschen lassen. Dies ist beispielsweise beim ZOTAC CI660 der Fall. Der Austausch mit einer von MacOS unterstützen Komponente erspart dann den ein oder anderen USB-Adapter. Der Intel NUC7i7DNKE ist komplett kompatibel zur später vorgestellten Methode, MacOS auch auf nicht-Apple-Geräten zu in-

stallieren (<https://www.tonymacx86.com/threads/guide-intel-nuc7-nuc8-using-clover-uefi-nuc7i7bxx-nuc8i7bxx-etc.261711/> [5]). Das setzt aber ebenfalls einen Tausch des WLAN/Bluetooth-Moduls voraus. Allerdings führt dieser NUC seine Wärme mit Hilfe eines Lüfters ab. Dieser Vorgang ist nicht ganz geräuschlos und kann besonders beim Fernsehen oder beim Anschauen eines Videos zu unliebsamen Nebengeräuschen führen. Die in den Abbildungen 2.1 und 2.2 gezeigten Barebones sind nahezu lauffähig. Es fehlen lediglich der Speicher oder eine Festplatte bzw. ein SSD-Riegel.

Hinweis 2.2 (Netzstecker vor dem Einbau ziehen) Achten Sie vor dem Einbau von Speicher und Festplatte bitte unbedingt darauf, dass der PC (Barebone) vom Netz getrennt ist, da es andernfalls zu Kurzschlüssen während des Einbaus kommen kann und der PC so dauerhaft beschädigt wird.

Nicht alle Speicher-Module sind mit allen Barebones kompatibel. Im Zweifel hilft es, auf der Internetseite des Herstellers nach kompatiblen Modulen zu suchen.



2.2 Der Eigenbau

Wer nicht auf fertige Barebones zurückgreifen möchte, kann sich seinen Wunsch-Wohnzimmer-PC auch komplett selbst zusammenstellen. Es gibt verschiedene Firmen, die sich darauf spezialisiert haben, leise oder sogar lautlose PCs zu bauen. Dazu zählt unter anderem die Firma Silentmaxx (<https://www.silentmaxx.de>) [6]. Der Kenko® Mini kommt in einem schicken Gehäuse und kann Intel-Prozessoren der 8. Generation beherbergen. Diese Eleganz hat allerdings auch ihren Preis.

Abbildung 2.3: Silentmaxx bietet schöne und lautlose PCs, die sich auf jeden Fall für den Einsatz im Wohnzimmer eignen. Quelle: <http://silentmaxx.de> [6].



Der Spaß beginnt bei 600 € für den kleinsten Prozessor und 4 GB Arbeitsspeicher. Möchte man mehr, bezahlt man auch mehr. Wer ein wenig Geld sparen möchte, kann sich die Komponenten auch selbst aussuchen, bestellen und zusammenbauen.

2.3 X86 Embedded Boards

Nach und nach findet man immer mehr Embedded Boards, die einen X86 (Intel oder AMD) bieten. Embedded Boards sind meist kleine Platinen, bei denen neben dem Prozessor in der Regel auch der Speicher aufgelötet ist. Ein beliebtes und bekanntes Beispiel ist der Raspberry Pi [1], welcher allerdings einen ARM-Prozessor an Bord hat. Während sich der Raspberry Pi vielleicht noch geradeso als HD-Mediencenter eignet, ist er nicht in der Lage, moderne Spielekonsolen wie eine Wii U zu emulieren oder ein Windows 10-Betriebssystem zu starten. An einen Hackintosh ist gar nicht zu denken.

Während der Odroid-H2 (Abbildung 2.4) in Deutschland bereits für ca. 130 € beispielsweise bei Reichelt bestellt werden kann, fehlen ihm noch Speicherriegel oder eine Festplatte, die als NVMe-SSD oder SATA-Platte nachgerüstet werden können. Der Intel Quad-Core Prozessor J4105 (Celeron) ist aber nicht für einen Hackintosh geeignet, da dieses Betriebssystem unterstützt wird. Der H2 hat seine Stärken aber auf anderen Gebieten. Der 4-Kern-Prozessor kann durchaus halbwegs anspruchsvolle Aufgaben erledigen. Aufgrund der beiden Ethernet-Schnittstellen kann der Odroid-H2 als Firewall eingesetzt werden. Die vorhandenen SATA-Ports erlauben darüber hinaus die Verwendung als NAS (Network Attached Storage). Zudem ist der Hardkernel-Rechenzweig komplett passiv gekühlt und damit vollkommen leise.

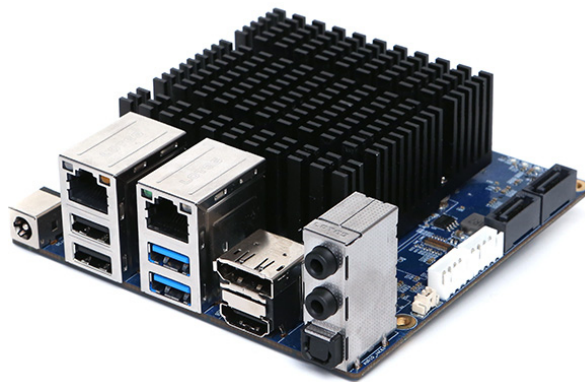


Abbildung 2.4: Der Odroid-H2 ist preiswert, unterstützt aber kein MacOS. Quelle: <https://www.hardkernel.com/shop/odroid-h2/> [7]

Bei meinen Recherchen für dieses Buch bin ich auf den LattePanda Alpha 864 gestoßen. Dieses Embedded Board hat alles Benötigte an Bord, um das in diesem Buch Beschriebene umzusetzen:



Abbildung 2.5: Der LattePanda Alpha 864 bietet alles, was man für die Projekte in diesem Buch benötigt. Quelle: <http://lattepanda.com> [8].

- **Intel Dual-Core M3-7Y30 der 7. Generation:** Dieser Prozessor findet sich noch im MacBook aus dem Jahr 2017 und ist damit hervorragend für einen Hackintosh geeignet. Er kann 4 Threads parallel verarbeiten und taktet dabei bis maximal 2.6 GHz (Single Thread). Diese Leistung reicht sogar für die Emulation einer Wii U aus.
- **Intel HD Graphics 615:** Der interne Grafikkern taktet zwischen 300 und 900 MHz und bietet mit über HDMI eine 4K-Auflösung bis 30 Hz (ohne HDCP). Mit Hilfe eines USB-C nach HDMI-Adapters lässt sich die Bildwiederholfrequenz bis 60 Hz steigern. Der kleine Rechner kann sogar zwei Bildschirme gleichzeitig ansteuern.
- **8 GB RAM:** Auf der Platine ist 8 GB LPDDR3 1866 MHz Dual-Channel fest verlötet.
- **64 GB eMMC V5.0:** In der Version LattePanda Alpha 864 bringt das X86-Board fest verlöteten eMMC-Speicher mit, den es auf Wunsch mit einem vorinstallierten Windows 10 Professional gibt.
- **PCI2:** LattePanda Alpha bringt 2 M2-Schnittstellen mit, an die NVMe-SSDs angeschlossen werden können oder mit Hilfe eines Adapters sogar Grafikkarten.
- **Funkstandards:** Der LattePanda Alpha funkt im 802.11 AC Band, wahlweise bei 2.4 GHz oder 5 GHz. Gigabit Ethernet ist genauso mit an Bord wie Bluetooth in

der Version 4.2. Das WLAN dieses Boards wird übrigens nicht von der Hackintosh-Methode unterstützt. Das ist aber nicht weiter schlimm. Ist man unter MacOS auf WLAN angewiesen, hilft ein preiswerter USB-Stick.

- **USB:** Der Lattepanada Alpha bietet insgesamt drei USB 3.0-Buchsen (Typ A) und eine vom Typ C, die PD (Power Distribution), DP (Display Port) und USB 3.0 unterstützt. Passende Adapter gibt es bei Amazon.
- **HDMI:** Displays können auch mit Hilfe eines HDMI-Kabels angeschlossen werden.
- **eDP:** Lattepanada Alpha stellt ebenfalls einen eDP-Anschluss (embedded Display Port) zur Verfügung, an den beispielsweise Touch-Displays angeschlossen werden können.
- **Audio:** Eine Audio-Buchse bietet die Möglichkeit, einen Kopfhörer mit 3,5 mm Klinkenstecker anzuschließen.

Zu den Besonderheiten des Lattepanada Alpha gehört der auf der Platine aufgelötete Co-Prozessor Arduino Leonardo, der Projekte wie ein AmbientLight ermöglicht und zahlreiche Erweiterungsmöglichkeiten zur Verfügung stellt. Darüber hinaus punktet dieses X86-Board mit weiteren Schnittstellen auf einer Jumperleiste wie GPIOs (General Purpose Input Output), I2C, I2S, USB, RS232, UART, RTC oder Power.

Der Lüfter des Board ist sehr leise und selbst unter Volllast kaum zu hören. Der Lattepanada Alpha kostet ca. 350 € mit lizenziertem Windows 10 Professional und 64 GB eMMC-Speicher, die günstigste Variante (ohne Windows und eMMC) ist ca. 100 € preiswerter. Das Board kann bei verschiedenen Distributoren oder bei DFROBOT unter <https://www.dfrobot.com/product-1728.html> [9] in China bestellt werden. Das BIOS des Lattepanada Alpha bietet neben WOL (Wake-On-Lan) auch Wake-On-Timer. Diese Funktionen helfen dabei, Strom zu sparen und das Board nur bei Bedarf zu booten.

Hinweis 2.3 (Einfuhrumsatzsteuer) Denken Sie bitte daran, dass der Zoll mit 19.5% Einfuhrumsatzsteuer zuschlägt, die üblicherweise bei Bestellungen aus China in dieser Größenordnung berechnet werden. Hinzu kommt oft noch eine kleine Bearbeitungsgebühr.

2.4 Grafik unter LINUX und Mac



Während jeder Grafikkarten-Hersteller Treiber für Windows-Betriebssysteme (DirectX) zur Verfügung stellt, ist das unter LINUX nicht immer der Fall. Für einen Hackintosh sieht das noch einmal anders aus, da Apples Geschäftsbedingungen den Betrieb von MacOS auf einer nicht-Apple-Hardware untersagen. Das bedeutet, dass die Grafik-Hardware unter diesem Betriebssystem nicht austauschbar ist und die Treiber ins Betriebssystem integriert sind. Hinzu kommt, dass verschiedene Hersteller für LINUX verschiedene Lösungen anbieten.

Man kann zwei verschiedene Einsatzfälle für Grafik unter LINUX unterscheiden: Eine 3D-Beschleunigung, wie sie beispielsweise bei Spielen oder bei der Oberfläche des Mediencenters *KODI* zum Einsatz kommt und eine Video-Beschleunigung, die verschiedene Video-Dekodierungsmöglichkeiten per Hardware zur Verfügung stellt (z. B. H264, HEVC, ...).

Unter LINUX ist OpenGL (Open Graphics Library; deutsch Offene Grafikbibliothek) die Standard 3D-Grafik API (Application Programming Interface). Die meisten Prozessoren, die eine 3D-Beschleunigung unter LINUX unterstützen, arbeiten unter OpenGL. Die meisten open-source Treiber setzen dabei auf eine Bibliothek namens *Mesa*. Proprietäre Treiber hingegen bringen ihre eigene OpenGL-Bibliothek mit. Unter LINUX besteht ein OpenGL-Treiber aus zwei Dateien:

- **libGL.so:** Diese Bibliothek stellt die OpenGL Grafikbibliothek selbst zur Verfügung und wird durch den LINUX-Loader geladen.
- **glx.so:** Diese Datei stellt den Xorg-Support unter LINUX sicher. Xorg ist eine beliebte freie Implementierung des LINUX X-Windows Systems, also dem System, das ähnlich einem Windows-Betriebssystem verschiedene Fenster mit verschiedenen Inhalten auf dem Bildschirm anzeigen kann. glx.so wird von Xorg geladen.

Viele OpenGL-Anwendungen nutzen libGLU.so als übergeordnete Grafik-Bibliothek. Während die OpenGL-Bibliothek grundlegende Funktionalitäten zur Verfügung stellt, bildet libGLU.so eine übergeordnete Instanz, die deutlich komplexere Funktionen im Vergleich zu OpenGL bietet. GLU-Operationen sind nicht Hardware-beschleunigt.

Wir werden später noch die Verwendung einer Konsole unter LINUX oder MacOS kennenlernen. Die OpenGL-Fähigkeiten Ihres Prozessors können Sie mit dem Befehl

Code-Ausschnitt 2.1: OpenGL-Abfrage

```
glxinfo | grep OpenGL
```

testen. Unter meinem LattePanda Alpha 864-Board liefert diese Abfrage das folgende Ergebnis:

Code-Ausschnitt 2.2: Das Ergebnis der OpenGL-Abfrage

```
OpenGL vendor string: Intel Open Source Technology Center
OpenGL renderer string: Mesa DRI Intel(R) HD Graphics 615 (Kaby Lake GT2)
OpenGL core profile version string: 4.5 (Core Profile) Mesa 19.0.0-devel (git-3c71ba3 ↵
    2018-12-18 cosmic-oibaf-ppa)
OpenGL core profile shading language version string: 4.50
OpenGL core profile context flags: (none)
OpenGL core profile profile mask: core profile
OpenGL core profile extensions:
OpenGL version string: 3.0 Mesa 19.0.0-devel (git-3c71ba3 2018-12-18 cosmic-oibaf-ppa)
OpenGL shading language version string: 1.30
OpenGL context flags: (none)
OpenGL extensions:
OpenGL ES profile version string: OpenGL ES 3.2 Mesa 19.0.0-devel (git-3c71ba3 2018-12-18 ↵
    cosmic-oibaf-ppa)
OpenGL ES profile shading language version string: OpenGL ES GLSL ES 3.20
OpenGL ES profile extensions:
```

Die Intel-GPU (Kaby Lake GT2) unterstützt OpenGL in der Version 4.5 und nutzt dabei den Mesa-Treiber 19.0. OpenGL ES wird ebenfalls unterstützt. OpenGL ES (Embedded Systems) stellt eine vereinfachte Version von OpenGL dar und ist oft auf Embedded-Systemen zu finden.

Neben der 3D-Beschleunigung spielt die Möglichkeit, Videos zu dekodieren, gerade für ein Mediacenter eine große Rolle. Gerade die Dekodierung hochauflösender Videos (4K) bringt eine CPU schnell an den Rand ihrer Leistungsfähigkeit, da die Dekodierung ein äußerst rechenintensiver Prozess ist. Daher bieten die meisten modernen Grafikchips eine Hardwareunterstützung zur Videode- und zur -enkodierung an. Unter dem Windows-Betriebssystem hat sich die einheitliche Schnittstelle DirectX durchgesetzt. Unter LINUX gibt es gleich drei verschiedene Schnittstellen. Jeder der großen Hersteller (Intel, AMD, NVidia) bietet eine eigene Lösung an. Die beiden verbreitetsten Lösungen sind dabei VA API (Video Acceleration API, Intel) und VDPAU (Video Decode and Presentation API for UNIX, NVidia). Durch verschiedene Treiber und weitere API-Bibliotheken ist es zwar möglich, diese beiden Lösungen herstellerübergreifend zu nutzen, allerdings bieten viele dieser Konvertierungs-APIs nicht alle Möglichkeiten des Originals (z. B. fehlende 4K-Unterstützung). Prinzipiell müssen mehrere Bedingungen erfüllt sein, damit die Hardware-Videobeschleunigung genutzt werden kann:

- Der Grafikchip an sich muss eine Videobeschleunigung bieten.
- Der Grafiktreiber selbst muss diese unterstützen. Dabei kann entweder der Treiber des Herstellers (sofern verfügbar) genutzt werden oder aber ein freier Treiber, der oft nicht alle Formate unterstützt.
- Es muss eine passende Bibliothek für die verwendete API installiert sein. Für Intel ist das beispielsweise VA API.
- Der Videoabspieler muss die jeweilige API unterstützen.

Für meine Intel GPU liefert ein Aufruf des Kommandos

Code-Ausschnitt 2.3: Abfrage der beschleunigten Codecs für Intel

```
vainfo
```

alle Formate, die mit Hilfe einer Hardware-Beschleunigung abgespielt werden können:

Code-Ausschnitt 2.4: Ergebnis der Abfrage der beschleunigten Codecs für Intel

```
libva info: VA-API version 1.2.0
libva info: va_getDriverName() returns 0
libva info: Trying to open /usr/lib/x86_64-linux-gnu/dri/i965_drv_video.so
libva info: Found init function __vaDriverInit_1_2
libva info: va_openDriver() returns 0
vainfo: VA-API version: 1.2 (libva 2.2.0)
vainfo: Driver version: Intel i965 driver for Intel(R) Kaby Lake - 2.2.0
vainfo: Supported profile and entrypoints
    VAProfileMPEG2Simple      : VAEntrypointVLD
    VAProfileMPEG2Simple      : VAEntrypointEncSlice
    VAProfileMPEG2Main        : VAEntrypointVLD
    VAProfileMPEG2Main        : VAEntrypointEncSlice
    VAProfileH264ConstrainedBaseline: VAEntrypointVLD
    VAProfileH264ConstrainedBaseline: VAEntrypointEncSlice
    VAProfileH264ConstrainedBaseline: VAEntrypointEncSliceLP
    VAProfileH264Main         : VAEntrypointVLD
    VAProfileH264Main         : VAEntrypointEncSlice
    VAProfileH264Main         : VAEntrypointEncSliceLP
    VAProfileH264High         : VAEntrypointVLD
    VAProfileH264High         : VAEntrypointEncSlice
    VAProfileH264High         : VAEntrypointEncSliceLP
    VAProfileH264MultiviewHigh : VAEntrypointVLD
    VAProfileH264MultiviewHigh : VAEntrypointEncSlice
    VAProfileH264StereoHigh    : VAEntrypointVLD
    VAProfileH264StereoHigh    : VAEntrypointEncSlice
    VAProfileVC1Simple         : VAEntrypointVLD
    VAProfileVC1Main           : VAEntrypointVLD
    VAProfileVC1Advanced       : VAEntrypointVLD
    VAProfileNone              : VAEntrypointVideoProc
    VAProfileJPEGBaseline       : VAEntrypointVLD
    VAProfileJPEGBaseline       : VAEntrypointEncPicture
    VAProfileVP8Version0_3      : VAEntrypointVLD
    VAProfileVP8Version0_3      : VAEntrypointEncSlice
    VAProfileHEVCMMain         : VAEntrypointVLD
    VAProfileHEVCMMain         : VAEntrypointEncSlice
    VAProfileHEVCMMain10       : VAEntrypointVLD
    VAProfileHEVCMMain10       : VAEntrypointEncSlice
    VAProfileVP9Profile0        : VAEntrypointVLD
    VAProfileVP9Profile0        : VAEntrypointEncSlice
    VAProfileVP9Profile2       : VAEntrypointVLD
```

Die Intel GPU des Lattepanda Alpha unterstützt neben dem H264-Codec auch HEVC oder das Format VP9.

Bitte achten Sie beim Kauf Ihres Wohnzimmer PCs darauf, welchen HDMI-Anschluss er zur Verfügung stellt. HDMI 1.4 kann in der Regel Auflösungen bis 4K in 30 Hz, unterstützt aber kein HDCP (High-bandwidth Digital Content Protection) zum Abspielen geschützter Inhalte. Abhilfe kann hier ein Display-Adapter schaffen (z. B. USB-C auf HDMI). Diese unterstützen dann auch 4K-Bildschirme mit einer Wiederholrate von 60 Hz. Genaue Auskunft hierüber erhalten Sie auf der Webseite des CPU/GPU-Herstellers.



Die Funktionalität und der Aufbau des AMD/ATI-Treibers entspricht ziemlich genau dem DirectX-Aufbau unter Windows. Das ermöglicht es dem Hersteller, einen nahezu Windows-identischen Treiber unter LINUX zu nutzen.

Für diese Grafikeinheiten erscheint daher der LINUX-Treiber (fglrx, siehe auch Abschnitt 2.4.3) häufig zeitgleich mit dem Windows-Treiber. Leider wird die XvBA API dieses Treibers nur schlecht unterstützt. Es gibt jedoch - wie so oft bei freier Software - eine Bibliothek, die XvVA-Befehle nach VA API umsetzt. Damit kann für diese Grafikkarten dieselbe Software benutzt werden, die auch für Intel-Grafikeinheiten funktioniert.

2.4.1 Intel

INTEL CPUs mit eingebauter GPU werden bereits vom Standard LINUX-Kernel unterstützt. Die nötigen Treiber liegen im Quelltext vor. Intel unterstützt VA API zur Hardware-beschleunigten Wiedergabe von Videos. VA API unterstützt die meisten heute benutzen Codecs. Die genaue Unterstützung hängt von der verwendeten Hardware und den jeweiligen Möglichkeiten des Treibers ab. Prinzipiell unterstützt werden die folgenden Formate:

- MPEG-2 decode acceleration Main Profile
- VC-1/WMV3 decode acceleration Advanced Profile
- MPEG-4 Part 2 (H.263) (auch bekannt als MPEG-4 SP/MPEG-4 ASP (Xvid)) decode acceleration
- H.264 AVC encode acceleration Main Profile
- H.264 AVC decode acceleration High Profile
 - H.264/AVC Hardware Variable Length Decoding (VLD) - CABAC
 - H.264/AVC Hardware Variable Length Decoding (VLD) - CAVLC
 - H.264/AVC Hardware Inverse Transform (IT)
 - H.264/AVC Hardware Motion Compensation (HWMC)
 - H.264/AVC Hardware In-Loop Deblocking (ILDB)
- H.265/HEVC encode acceleration
- H.265/HEVC decode acceleration
- VP9 8-bit encode acceleration
- VP9 8-bit and 10-bit decode acceleration

Zusätzlich zu VA API kann NVidia's VDPAU als sogenanntes backend benutzt werden. Die hierzu verwendeten Bibliotheken unterstützen aber nicht alle Features, die VA API bietet.



Die Webseite des Herstellers gibt Ihnen die genauen Details bezüglich der Rechen- und Grafik-Fähigkeiten der jeweiligen CPU. Für den LattePanda Alpha finden Sie diese Angaben beispielsweise unter <https://ark.intel.com/de/products/95449/Intel-Core-m3-7Y30-Processor-4M-Cache-2-60-GHz> - [10].

2.4.2 NVidia

NVidia stellt auf seiner Webseite LINUX-Treiber für seine Grafikkarten zur Verfügung. Die Quelltexte dieser Grafiktreiber sind jedoch nicht öffentlich verfügbar (<https://www.nvidia.de/object/unix-de.html> [11]).

Allerdings gibt es mit Nouveau einen brauchbaren Treiber, dessen Quelltexte offen sind. Dieser Treiber wird durch ein aufwendiges Reverse-Engineering ohne viel Unterstützung von NVidia entwickelt. Er unterstützt zwar inzwischen 3D-Funktionalität. Diese ist aber im Vergleich zu den proprietären NVidia-Treibern nicht so ausgereift und langsamer. NVidia setzt auf die API VDPAU. VDPAU ist eine offene Programmierschnittstelle, die ursprünglich sogar von NVidia geschrieben wurde.

Der proprietäre NVidia-LINUX-Treiber unterstützt VDPAU auf allen Grafikkarten, die zumindest PureVideo HD der zweiten Generation VP2 unterstützen, also ab der Geforce-8-Serie, ausgenommen der Geforce 8800 GTS, Geforce 8800 GTX und Geforce 8800 Ultra.

2.4.3 AMD

Moderne AMD-Geräte werden vom fglrx-Treiber (AMD Catalyst Display Drive) unterstützt (<https://www.amd.com/en/support> [12]). Dieser Treiber ist wie der NVidia-Treiber nicht quelloffen, unterstützt dafür aber aktuelle AMD-Grafikeinheiten. Allerdings hat AMD die Spezifikationen der AMD-Grafikchips offengelegt, so dass der quelloffene Treiber namens *radeon* auf dem Wege zum vollwertigen 2D/3D-Treiber ist. So laufen heute die meisten AMD-Chips vernünftig mit dem Radeon-Treiber.

Der fglrx-Treiber unterstützt X-Video Bitstream Acceleration (XvBA). Das ist eine proprietäre Programmierschnittstelle für Radeon Grafikkarten des Herstellers AMD, über die das Dekodieren von Videostreamen und Nachbearbeiten des dekodierten Materials auf LINUX-basierten Betriebssystemen von der CPU größtenteils auf die Grafikkarte verlagert werden kann. XvBA ermöglicht die flüssige Wiedergabe von Videos in den derzeit benutzen Formaten wie MPEG-2, MPEG-4 AVC (H.264) und VC-1.

2.4.4 MacOS

Während MacOS dafür vorgesehen ist, ausschließlich auf Apple-Hardware zu laufen, unterstützt Apple offiziell den umgekehrten Weg, LINUX oder Windows auf MacOS-Geräten zu installieren. Schaut man sich auf der Apple-Webseite (<https://www.apple.de> [13]) um, verwenden Macs aus dem Jahre 2018 die folgenden CPU/GPU-Einheiten:

- **MacBook:** Intel Core m3, i5 und i7. Der Intel Core m3-Prozessor wird übrigens auch im Lattepanda Alpha verwendet. Daher ist dieses Gerät die ideale Plattform für einen Hackintosh.
- **MacBook Air:** Intel Core i5.
- **MacBook Pro:** Intel Prozessor der 8. Generation mit Radeon Pro GPU (Radeon Pro Vega Grafik). Mit Hilfe einer externen eGPU (ebenfalls Radeon-Grafik) lässt sich die Grafik-Fähigkeit des Macbook Pro weiter steigern.

- **iMac:** Der iMac setzt ebenfalls auf Radeon-Grafik (Radeon Pro 500), ebenfalls in Kombination mit einer intel-CPU. Lediglich der kleinste iMac nutzt die Intel Iris Plus Graphics 640.
- **iMac Pro:** Der iMac Pro baut auf einen 3,2 GHz 8-Core Intel Xeon W Prozessor und nutzt eine Radeon Pro Vega 56 Grafikeinheit.
- **Mac Pro:** Alle 2018-ner Mac Pro-Varianten nutzen den Intel Xeon E5 Prozessor mit einem AMD FirePro D500/D700 Grafikprozessor.
- **Mac Mini:** Das 2018 Mac Mini-Modell baut auf einen Intel core i3/i5-Prozessor und nutzt die Intel UHD Graphics 630.

NVidia Grafik-Einheiten tauchen also im Programm des kalifornischen Computer-Herstellers gar nicht mehr auf. Möchte man also eines der Apple-Geräte als Wohnzimmer-PC nutzen, empfehle ich ein Gerät mit Intel-Grafik, da dieses mehr Möglichkeiten unter LINUX bietet.

2.5 Das Wichtigste in Kürze

Für was auch immer Sie sich beim Bau eines Wohnzimmer-PCs entscheiden, er sollte ein paar wichtige Kriterien erfüllen:

- Steht der PC im Wohnzimmer, muss er in erster Linie leise sein. Nichts trübt den Film-Genuss so sehr, wie ein lauter Lüfter.
- 4...8 GB Hauptspeicher sollten ausreichend sein. Für anspruchsvollere Aufgaben benötigen Sie entsprechend mehr Speicher.
- Falls Sie vorhaben, einen Hackintosh zu bauen, also eine nicht-Apple-Hardware, auf der MacOS als Betriebssystem läuft (Kapitel 3), informieren Sie sich vor dem Kauf im Internet, ob der PC mit der Hackintosh-Methode funktioniert oder nicht (<https://www.tonymacx86.com/buyersguide/building-a-custom-c-hackintosh-the-ultimate-buyers-guide/> [4]).
- Möchten Sie ein Apple-Gerät (z. B. den Mac Mini) als Wohnzimmer-PC nutzen, empfehle ich eines mit Intel GPU.
- Achten Sie beim Kauf darauf, dass das Bios des Rechners WOL (Wake-On-LAN) und Wake-On-Timer unterstützt. So können Sie den PC aus der Ferne einschalten, wenn Sie ihn benötigen, oder er kann sich selbst einschalten, um beispielsweise eine Fernsehsendung aufzuzeichnen.
- Ein Wohnzimmer-PC sollte mit einer Fernbedienung bedienbar sein. Achten Sie also beim Kauf darauf, dass der PC entweder einen Infrarot-Empfänger besitzt oder USB-Anschlüsse in ausreichender Anzahl zur Verfügung stellt. Denken Sie daran, dass Sie gegebenenfalls noch einige USB-Geräte anschließen wollen (Festplatte, DVB-Empfänger, Arduino für ein Ambilight, usw.). Ein USB-Hub entschärft dieses Problem zwar, erhöht aber den Verkabelungsaufwand und senkt damit erheblich den WAF (Woman Acceptance Factor). Das gilt natürlich auch umgekehrt für den MAF.

- CPU- und GPU-Anforderungen steigen mit den Anforderungen an moderne Spiele. Für eine Wii U-Emulation (Kapitel 6) stellt ein Intel Core m3 (z. B. Lattepanda Alpha) ein Minimum dar.

Für relativ kleines Geld (ab ca. 300 €) stellt der Lattepanda Alpha (864) eine hervorragende Basis für einen Wohnzimmer-PC dar. Er bietet genug Rechenpower für eine Wii U-Emulation unter Windows 10, er ist eine hervorragende Ausgangsbasis für einen Hackintosh, Ubuntu 18.10 läuft „out of the box“, und LibreElec lässt sich ebenfalls installieren. Alles das und noch vieles mehr werden wir in diesem Buch anschauen. Sie erhalten einen Leitfaden für ein Multi-Boot-System (Windows 10 Professional, MacOS Mojave 14.10, Ubuntu 18.10 und LibreElec 9.0) und lernen, die verschiedenen Betriebssysteme für verschiedene wohnzimmer-taugliche Aufgaben einzusetzen.

Zusammenfassung und Ausblick 2 Das war das Kapitel „Der Wohnzimmer-PC“, das Ihnen einige Tipps zur Auswahl eines leisen und schicken Rechners gegeben hat. Je nachdem, welche Anforderungen Sie an Spiele haben, die Sie auf Ihrem Mediacenter spielen möchten, haben Sie die Wahl zwischen verschiedenen mehr oder weniger leistungsfähigen CPUs.

Möchten Sie nur einen Fernseher mit dem Wohnzimmer-PC verbinden, reicht ein Grafikausgang. Möchten Sie aber noch zusätzlich ein LCD anschließen, das Ihnen weitere Informationen anzeigt (Kapitel 7), sollten Sie darauf achten, dass Ihr Wohnzimmer-PC weitere HDMI-Ausgänge zur Verfügung stellt.

Nachdem die Hardware nun vorgestellt und ausgewählt worden ist, widmen sich die nächsten Kapitel der Software. Lernen Sie, wie Sie ein MacOS auf Ihrem Wohnzimmer-PC installieren können, wo die Vorteile eines selbstgebaute Macs liegen und mit welchen Nachteilen Sie leben müssen. ■

Literaturverzeichnis

- [1] Rüdiger Follmann. *Das Raspberry Pi Kompendium, 2. Auflage*. Springer Verlag, Heidelberg, 2018. ISBN: 978-3-662-58143-8 (siehe Seiten XIII, 4).
- [3] Zotac. *Webseite*. 2019.
<https://www.zotac.com/>
besucht am 23. 12. 2019 (siehe Seite 1).
- [4] tonymacx86. *Building a CustoMac Hackintosh: The Ultimate Buyer's Guide*. 2019.
<https://www.tonymacx86.com/buyersguide/building-a-customac-hackintosh-the-ultimate-buyers-guide/>
besucht am 23. 12. 2019 (siehe Seiten 2, 12, 16).
- [5] tonymacx86. *[Guide] Intel NUC7/NUC8 using Clover UEFI (NUC7i7Bxx,etc)*. 2019.
<https://www.tonymacx86.com/threads/guide-intel-nuc7-nuc8-using-clover-uefi-nuc7i7bxx-nuc8i7bxx-etc.261711/>
besucht am 23. 12. 2019 (siehe Seiten 3, 16).
- [6] silentmaxx. *Webseite*. 2019.
<https://www.silentmaxx.de>
besucht am 23. 12. 2019 (siehe Seiten 3, 4).
- [7] Hardkernel. *ODROID-H2*. 2019.
<https://www.hardkernel.com/shop/odroid-h2/>
besucht am 23. 12. 2019 (siehe Seite 4).
- [8] lattepanda. *Webseite*. 2019.
<https://www.lattepanda.com>
besucht am 23. 12. 2019 (siehe Seite 5).
- [9] dfrobot. *Webseite*. 2019.
<https://www.dfrobot.com/product-1728.html>
besucht am 23. 12. 2019 (siehe Seite 6).
- [10] Intel. *Intel® Core™ m3-7Y30 Prozessor*. 2019.
<https://ark.intel.com/de/products/95449/Intel-Core-m3-7Y30-Processor-4M-Cache-2-60-GHz>
besucht am 23. 12. 2019 (siehe Seite 10).
- [11] NVIDIA. *Unix-Treiber*. 2019.
<https://www.nvidia.de/object/unix-de.html>
besucht am 23. 12. 2019 (siehe Seiten 11, 63).
- [12] AMD. *AMD Drivers and Support*. 2019.
<https://www.amd.com/en/support>
besucht am 23. 12. 2019 (siehe Seite 11).
- [13] Apple. *Webseite*. 2019.
<https://www.apple.de>
besucht am 12. 12. 2018 (siehe Seite 11).

3. Hackintosh

3.1	So ein Hack-Mac	15
3.2	Vorbereitung der Festplatte	17
3.3	Clover & Co	20
3.4	Booten von USB – Die Installation	27
3.5	Der Bootloader	29
3.6	Probleme und Lösungen	34
3.7	Praxisbeispiele	35
3.8	Fazit	43

Dieses Kapitel beschreibt die Installation eines MacOS auf einem PC. Diese Methode nennt sich in Anlehnung an den Namen Macintosh „Hackintosh“. Rechtlich gesehen bewegt man sich hier in einer Grauzone. Die Apple AGBs untersagen das Installieren von MacOS auf einer nicht-Apple-Hardware. Ich gehe daher davon aus, dass Sie, ähnlich wie ich, bereits einen MacOS besitzen, ebenso wie das dazugehörige Betriebssystem. Dieses Kapitel hat reinen Lehr-Charakter. Es beleuchtet die Vorteile der Hackintosh-Methode und auch die Nachteile. Kaufen Sie im Zweifel lieber echte Apple-Hardware, denn nur dort erhalten Sie vollen Support und eventuell benötigte Unterstützung.



3.1 So ein Hack-Mac

Warum so viel Aufwand, um ein Apple-Betriebssystem auf nicht-Apple-Hardware zu installieren? Diese oder ähnliche Fragen werden Sie sich vielleicht stellen. Was sind die Vorteile dieser Methode, was sind die Nachteile? Gibt es Einschränkungen und wenn ja, kann und will ich diese in Kauf nehmen? Dieses Kapitel versucht, alle diese Fragen zu beantworten.

Neben allgemeinen Installationshinweisen schauen wir uns zunächst die verschiedenen Methoden an, Apples Betriebssystem Mojave auf der Festplatte unseres PCs zu installieren. Wir werfen einen näheren Blick auf den Bootloader *Clover*, den wir später noch dazu verwenden werden, alle weiteren installierten Betriebssysteme zu starten. Das Kapitel zeigt Ihnen, was mit der Hackintosh-Methode möglich ist und wo die Grenzen liegen. In einem Test überprüfen wir die Alltagstauglichkeit von Apples bekannter Videobearbeitungs-Software iMovie und wir testen das Spiel Fortnite. Außerdem werden wir weitere originale Apple-Hardware wie Bluetooth-Tastatur oder -Maus sowie eine Fernbedienung mit unserem Hackintosh verwenden.

Apples MacOS kannte ich bisher nur von meinem betagten Mac Mini (2009) oder von meinem MacBook. Den Mini habe ich immer sehr gerne genutzt, um mit iMovie meine Urlaubsfilme zu bearbeiten. Irgendwann aber hat der kleine Apple-Computer meine Fotos nicht mehr eingelesen, da sich das Format geändert hat. Ein Update auf ein neueres MacOS wie Mojave war aufgrund des Computeralters nicht mehr möglich.

Ein Blick auf Apples Webseite machte dann ziemlich schnell klar, dass für einen neuen, halbwegs gut ausgestatteten Mac Mini mehr als 1.200 € fällig werden; zu viel, um einmal im Jahr iMovie anzuwerfen. Da meinen MacBook noch Update-fähig war, habe ich den Versuch unternommen, Mojave auf einer nicht-Apple-Hardware zu installieren, weil ich wissen wollte, wie gut das funktioniert und ob das wirklich einen reinen Mac ersetzen kann. Meine Versuche habe ich mit dem Lattepanda Alpha 864-Board unternommen. Die hier vorgestellten Methoden lassen sich aber ohne Probleme auf weitere, unterstützte Hardware übertragen.

Normalerweise läuft Apples Betriebssystem ausschließlich auf den hauseigenen Computern. Es ist auf diese Hardware abgestimmt, und der Benutzer kann sich sicher sein, dass alles so funktioniert, wie es sollte. Da Apple an der Hardware nicht schlecht verdient, gibt es gar kein Interesse, die MacOS-Software auch für fremde Hardware lauffähig zu machen. Am Beginn jedes Hackintosh-Projekts steht also das Komponenten-Roulette. Die Internetseite <https://www.tonymacx86.com/buyersguide/building-a-custom-c-hackintosh-the-ultimate-buyers-guide/> [4] gibt einen guten Überblick darüber, welche Komponenten für das Hackintosh-Vorhaben funktionieren. Hier gibt es verschiedene Leistungsklassen in Punkto Prozessor, Grafik, Hauptspeicher und Mainboard. Wer das Zusammenstellen und Aufbauen eines kompletten PCs scheut, wird bei den sogenannten Barebones fündig. Das sind vorkonfigurierte Lösungen, wie sie im Kapitel 2 vorgestellt wurden. Viele der funktionierenden Lösungen setzen auf Intel-Chips, da diese sehr häufig in originaler Apple-Hardware zu finden sind. Leider werden aber keine Intel-WLAN-Chips unterstützt. Wer also auf WLAN (und auch Bluetooth) angewiesen ist, aber keinen unterstützen USB-Dongle verwenden möchte, sollte sich nach einem Barebone umschauen, dessen WLAN/Bluetooth-Karte nicht fest verlötet ist, sondern getauscht werden kann. Dies trifft beispielsweise auf den Intel NUC7i7DNKE zu (<https://www.tonymacx86.com/threads/guide-intel-nuc7-nuc8-using-clover-uefi-nuc7i7bxx-nuc8i7bxx-etc.261711/> [5]).

Für meine Experimente habe ich das Lattepanda Alpha (864)-Board verwendet. Dieses Board ist mit einem Einstiegspreis von 300 € günstig und erfordert lediglich eine externe Festplatte zur Installation von MacOS. Der interne eMMC-Speicher kann hierfür nicht verwendet werden. Auch auf diesem Board ist ein WLAN/Bluetooth-Kombichip verlötet. Während der WLAN-Teil nicht von MacOS unterstützt wird, funktioniert die Bluetooth-Einheit nahezu reibungslos. Ethernet wird unterstützt ebenso wie die Grafikbeschleunigung der Intel HD615-Grafikeinheit. Insgesamt erhält man hier ein hervorragendes Preis-Leistungsverhältnis.

Um Mojave installieren zu können, benötigen wir neben einer Festplatte noch einen USB-Stick, von dem der Rechner booten kann. 8 GB sind für die Installation von Mojave ausreichend.



Wichtig ist, dass das BIOS des Rechners das Booten von USB-Laufwerken unterstützt. Das sollte aber bei allen moderneren Rechnern der Fall sein.

3.2 Vorbereitung der Festplatte

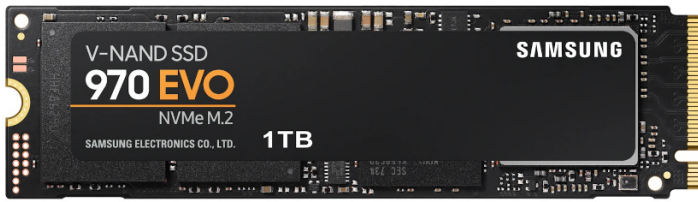


Abbildung 3.1: Die Samsung MZ-V7E500BW 970 EVO ist eine beliebte NVMe-Festplatte. Quelle: <https://www.samsung.com> [14].

Das Lattepanda Alpha-Board unterstützt wie viele andere Barebones die Verwendung einer NVMe SSD. Der Begriff SSD bedeutet Solid State Disk und ist eine Festplatte, die ohne mechanische Teile, sondern nur mit Elektronik auskommt. Damit ist sie verschleißarm und sehr schnell. NVMe steht für Non-

Volatile Memory Express, also nicht-flüchtigen Speicher, der extrem schnell beschrieben und gelesen werden kann. Je nach Ausführung erreichen diese Festplatten Datenraten von mehreren GB/s, kosten allerdings auch 120 € in der 500 GB-Version.

Hinweis 3.1 (Netzstecker vor dem Einbau ziehen) Achten Sie vor dem der Festplatte bitte unbedingt darauf, dass der PC (Barebone) vom Netz getrennt ist, da es andernfalls zu Kurzschlüssen während des Einbaus kommen kann und der PC so dauerhaft beschädigt wird.

Es gibt beliebig viele Installationsszenarien für die Festplatte. Einige Nutzer möchten lediglich ein einziges Betriebssystem auf die Festplatte installieren, andere kaufen einen Barebone, auf dem bereits ein Windows installiert ist, welches die ganze Festplatte in Anspruch nimmt und wollen die verwendete Partition verkleinern, um ein weiteres Betriebssystem zu installieren. Wir gehen hier vom schlimmsten Fall aus: Auf der Festplatte befindet sich vielleicht bereits ein Betriebssystem, das wir behalten möchten. Trotzdem soll Platz für weitere Betriebssysteme geschaffen werden. Partitions-Zuordnungen von Festplatten sollte man nicht im laufenden Betrieb verändern, da das einen Datenverlust mit sich bringen kann. Es ist allerdings auch unpraktisch, eine Festplatte aus einem Computer auszubauen, sie in einem anderen Rechner zu partitionieren, um sie dann wieder in den ursprünglichen Rechner einzubauen. Praktischer ist es, die Festplatte eingebaut zu lassen, den Rechner von einem USB-Betriebssystem zu starten und die Festplatte von diesem System aus zu partitionieren.

Hinweis 3.2 (Datensicherung) Bitte führen Sie vor dem Umpartitionieren einer Festplatte eine Datensicherung durch, um Datenverlust zu vermeiden.

Ein beliebtes Programm zur Festplattenpartitionierung ist *gparted*. Der Name steht für GNU Parted und ist ein GTK+ Front-End zum Erstellen, Löschen, Ändern der Größe, Verschieben, Prüfen und Kopieren von Festplattenpartitionen und deren Dateisystemen. Es gibt lauffähige ISO-Images, die von einem USB-Stick starten. Ein solches werden wir nun herunterladen und auf einem USB-Stick installieren, um die Festplatte nach unseren Wünschen zu partitionieren. Falls Ihre Festplatte leer ist oder Sie auf deren Inhalt verzichten können, können Sie diesen Schritt auch auslassen und während der Installation von MacOS Apples Festplatten-Dienstprogramm verwenden. Die hier gezeigte Anleitung gilt für NVMe-Festplatten genau so wie für SATA-Platten. *gparted* erlaubt es Ihnen, die zu ändernde Festplatte auszuwählen. Anstelle von *gparted* können Sie natürlich auch mit einem anderen Partitionierer Ihrer Wahl arbeiten.

Fertige *gparted*-ISO-Images kann man unter <https://gparted.org/download.php> [15] herunterladen.



Die Webseite <https://gparted.org/download.php> [15] bietet ein i686- und ein amd64-Image an. Die Architektur der modernen CPUs wird als i686 bezeichnet (Intel X86, 6. Generation). Distributionen oder Pakete, die für 64-Bit kompiliert wurden, werden im Allgemeinen mit dem Hinweis „amd64“ gekennzeichnet. Dabei sind aber Intel- und AMD-Prozessoren normalerweise kompatibel. Für moderne PCs können sie also getrost das amd64-Image herunterladen.

Nachdem das *gparted*-Image heruntergeladen wurde, schreiben wir es auf einen USB-Stick. Da das Image selbst nur etwa 350 MB groß ist, reicht ein kleiner 1 GB USB-Stick aus. Unter Windows können Sie das Programm *win32diskimager* verwenden, um das Image auf den USB-Stick zu bringen. Sie finden es unter <https://sourceforge.net/projects/win32diskimager/> [16] als kostenlosen Download. Installieren Sie es und führen Sie es aus.

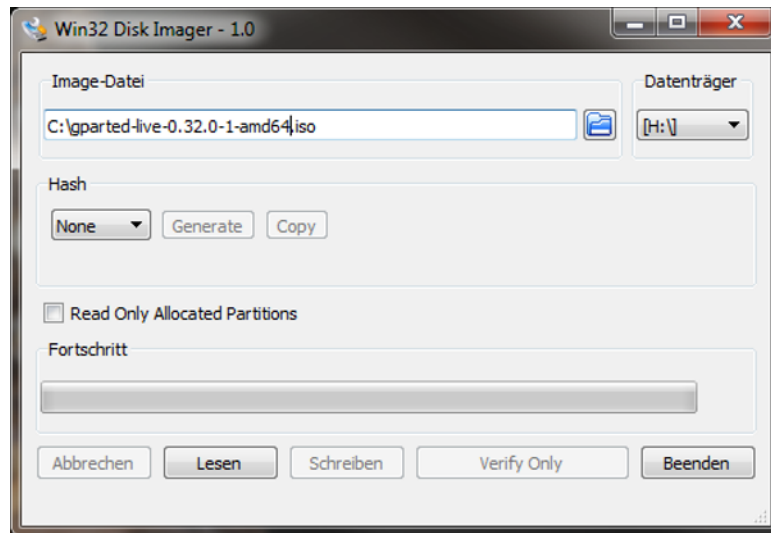


Abbildung 3.2: Win32diskimager schreibt unter Windows komfortabel ISO-Images auf den USB Stick.

Wählen Sie das zu schreibende Image ebenso wie den USB-Stick, auf dem Sie es installieren möchten, aus.

Hinweis 3.3 (Datensicherung) Das Beschreiben des USB-Sticks mit Win32disk-imager löscht alle Daten, die sich auf dem USB-Stick befinden. Bitte stellen Sie sicher, dass Sie eventuell noch vorhandene Daten gesichert haben oder nicht mehr benötigen.

Starten Sie danach den Schreib-Vorgang. Nach kurzer Zeit befindet sich das Image auf dem USB-Stick, von dem dann auch gebootet werden kann.

Wenn ihr Computer mit einer UEFI-Bios (Unified Extensible Firmware Interface) ausgestattet ist, kann es nötig sein, die Option Secure Boot auszuschalten. Falls diese Option aktiviert ist, kann es sein, dass das gparted-Livesystem nicht startet.



Booten Sie Ihren Rechner also von dem gerade beschriebenen USB-Stick und starten Sie das Programm *gparted*.

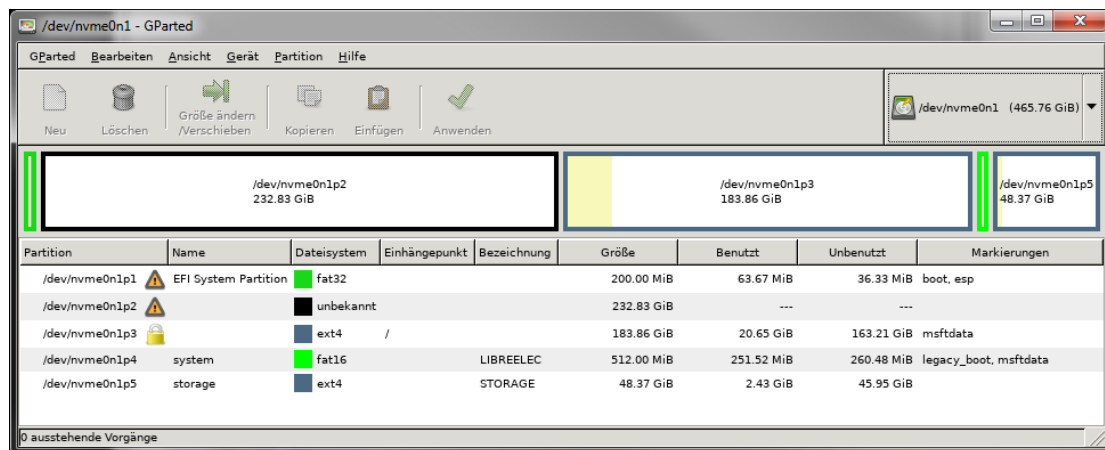


Abbildung 3.3: gparted erlaubt es auf einfache Weise, Festplattenpartitionen anzulegen oder zu ändern.

Abbildung 3.3 zeigt die Partitionierung meiner NVMe-Festplatte. Die enthält zunächst eine EFI System-Partition (/dev/nvme0n1p1 200 MB), gefolgt von einer unbekannten Partition (/dev/nvme0n1p2, 230 GB, MacOS). Diese Partitionen werden wir später mit dem MacOS Festplatten-Dienstprogramm anlegen und **nicht** mit *gparted*. Eine Partition, die ich angelegt habe, ist /dev/nvme0n1p3 (180 GB). Diese Partition ist mit dem Dateisystem ext-4 formatiert und beherbergt ein Ubuntu 18.10 mit dem Einhängepunkt „/“ (root). Eine Erklärung hierzu gibt es im Kapitel 4. Die beiden letzten Partitionen /dev/nvme0n1p4 und -5 (512 MB und 50 MB) bilden ein LibreElec-Bootsystem (siehe Kapitel 8).

Nutzen Sie *gparted* an dieser Stelle, um Partitionen zu reservieren oder zu verkleinern. Wir werden - wie bereits gesagt - im Laufe des MacOS-Installationsprozesses das MacOS Festplatten-Dienstprogramm verwenden, um die erforderlichen MacOS-Partitionen anzulegen. Alle durchgeführten Vorgänge müssen durch einen Klick auf den Knopf „Anwenden“ aktiviert werden.

Bei meinem LattePanda Alpha 864 ist das Windows-Betriebssystem auf dem internen eMMC-Speicher (`/dev/mmcblk0`) installiert und taucht daher in Abbildung 3.3 nicht auf.



Falls Sie eine USB-Festplatte angeschlossen haben, finden Sie diese unter dem Device `/dev/sdX`, $X = a, b, c, \dots$. MacOS nennt die Partitionen übrigens ähnlich, aber doch anders. Die einzelnen Partitionen heißen hier `disk`, gefolgt von der Sektion `s`, also beispielsweise `disk0s1`.

Sie sollten Ihre Festplatte nun grob vorbereitet haben. Dazu zählt ein evtl. Verkleinern einer Windows-Partition und das Anlegen von Partitionen für weitere Betriebssysteme. Das Dateisystem spielt zunächst keine Rolle. Die jeweiligen Installationsprogramme der verschiedenen Betriebssysteme können das noch anpassen. Das Festplatten-Dienstprogramm von MacOS kümmert sich ebenfalls darum, eine EFI-Partition anzulegen, diese müssen Sie nicht separat vorsehen. Wir werden nun einen USB-Installations-Stick für MacOS vorbereiten, von dem aus wir denn die Installation des Hackintosh starten können.



Unter LINUX und MacOS können Sie den Befehl `dd` verwenden, um beispielsweise ISO-Images auf einen USB-Stick zu kopieren. Dieses Vorgehen wird in Abschnitt 3.5 noch genauer beschrieben.

3.3 Clover & Co

Clover ist eine Software, die einem MacOS während des Bootvorganges vorgaukelt, auf echter Mac-Hardware zu laufen. Es gibt auch alternative Optionen, die in das BIOS des Mainboards eingreifen und MacOS so vorspielen, auf echter Hardware zu laufen [17]. Da in einigen Foren davon abgeraten wird, die *Beast-Tools* (Unibeast und Multibeast) zu verwenden und *Clover* bei mir einwandfrei funktioniert hat, beschreibe ich in diesem Abschnitt die Installation mittels *Clover*. Am Ende einer Installation kann man dann den *Clover Configurator* verwenden, um alles sauber einzurichten.

Prinzipiell gibt es die Möglichkeit, einen MacOS-Bootstick unter Windows oder unter MacOS selbst zu installieren. Nennt man bisher keinen Macintosh sein eigen, bleibt wahrscheinlich nur die Windows-Methode übrig oder der Besuch eines Freundes, der einen Mac besitzt.

3.3.1 Erstellen eines Bootsticks unter Windows

Die Webseite <https://www.aioboot.com/en/clover-boot-disk/> [18] stellt alle Dateien zur Verfügung, die man für das Erstellen eines bootbarem USB-Sticks unter Windows benötigt. Während der *Clover Boot Disk Creator* sicherlich noch ein legales Werkzeug ist, verstößt es mindestens gegen die AGBs von Apple, ein HFS-Image (Hierarchical File System) der neuesten MacOS Mojave-Version herunterzuladen. Sie müssen also selbst entscheiden, ob Sie dieses Risiko eingehen möchten oder ob Sie ein Mojave, das Sie vielleicht noch auf einem Mac-Computer zu Hause haben, nutzen und den originalen Macintosh während dieser Zeit lahm legen. Neben dem MacOS Image benötigen Sie weiterhin die Datei *Clover.img*, die ein bootbares EFI inklusive der Clover-Dateien zur Verfügung stellt. Diese EFI-Partition erlaubt es Ihnen später auch, beliebige Festplatten-Partitionen vom USB-Stick aus zu starten.

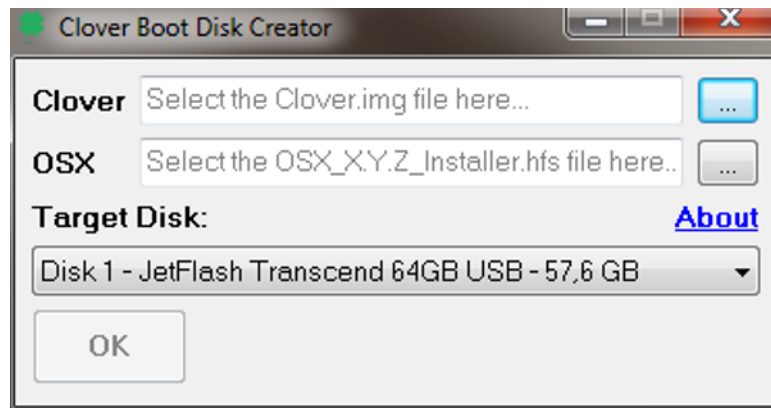


Abbildung 3.4: Der *Clover Booth Disk Creator* erlaubt es, einen USB-Bootstick mit MacOS unter Windows zu erstellen.

Der *Clover Boot Disk Creator* (Abbildung 3.4) erwartet drei Eingaben:

1. Geben Sie den Speicherort der heruntergeladenen Datei *Clover.img* an.
2. Geben Sie den Speicherort der MacOS-Images an.
3. Wählen Sie den USB-Stick aus, der als Boot-Stick verwendet werden soll. Achtung: Alle Dateien auf diesem Stick werden gelöscht.
4. Klicken Sie auf OK, um den Schreibvorgang zu starten.

Nicht jeder USB-Stick ist geeignet, um ihn als Bootstick zu verwenden. Sollten Sie mit einer Marke Pech haben, versuchen Sie eine andere. Ebenso kann es sein, dass sich ein bestimmter Stick von einem anderen USB-Port besser booten lässt. Auch hier kann Tauschen helfen.



Nachdem der Bootstick erfolgreich erstellt worden ist, geht es an die Konfiguration. bei der Konfiguration werden sogenannte *KEXT*-Dateien in das Verzeichnis */EFI/CLOVER/kext/Other* kopiert. Bei *KEXT*-Dateien handelt es sich um Kernel-Module (Extensions). Das sind Dateien, die dem Betriebssystem (MacOS) erlauben, mit der Hardware zu kommunizieren und sie damit einzubinden. Wird also auf einem Hackintosh ein Teil der Hardware nicht out-of-the-box von MacOS unterstützt, helfen die Kernel-Dateien, diesen Teil der Hardware in Betrieb zu nehmen. *KEXT*-Dateien kann man von <https://bitbucket.org/RehabMan/> [19] oder <https://www.tonymacx86.com/resources/categories/kexts.11/> [20] herunterladen.

Für viele Boards gibt es bereits fertige Clover-Dateien, so auch für den Lattepanda Alpha. Diese findet man unter <https://github.com/novaspirit/macpanda> [21] in der Rubrik *Releases*. Für den Lattepanda Alpha müssen Sie also nur die Datei *lattepanda.clover.zip* herunterladen und auf einem Windows- oder LINUX/MacOS-System entpacken. Im entpackten Verzeichnis *Clover* befinden sich alle benötigten Dateien. Ihr USB-Bootstick enthält eine FAT32-Partition, die von Windows- oder LINUX/MacOS-Rechnern gelesen werden kann. Falls Sie also Clover-Dateien für Ihr Board gefunden

haben, stecken Sie den USB-Stick in den USB-Port eines Rechners, auf dem bereits ein Betriebssystem läuft und ersetzen Sie den Order `/EFI/CLOVER` des Sticks mit dem heruntergeladenen Clover-Ordner.

Falls Sie keine fertigen Clover-Ordner für Ihren PC finden, ist Handarbeit angesagt. Bei „kernel panic“-Fehlern kann die Datei `NullCPUPowerManagement.kext` helfen. `IntelMausiEthernet.kext` hilft beispielsweise, um Unterstützung für Intels 82579 Gigabit-Ethernet-Controller hinzuzufügen. Weiterhin müssen Sie eventuell Änderungen an der Datei `/EFI/CLOVER/config.plist` vornehmen. In dieser Datei kann das Clover-Theme - also das Aussehen des Bootloaders - geändert werden genau so wie Einstellungen für Grafikkarten und weitere Einstellungen. So können NVidia-Grafikkarteneinstellungen beispielsweise an der Stelle

Code-Ausschnitt 3.1: Deaktivieren einer NVidia-Grafikkarte

```
<key>Arguments</key>
<string>nv_disable=1</string>
```

getätigt werden. Wesentlich komfortabler können diese Einstellungen mit dem Programm *Clover Configurator* geändert werden, welches wir später noch näher kennenlernen. Die Foren der oben genannten Webseiten sind ein guter Anhaltspunkt. Sollte Ihre Hardware nicht booten oder sollten Sie Probleme mit Clover und den benötigten Dateien haben, wird man Ihnen sicherlich gerne helfen.



Die Webseite <https://www.aioboot.com> [18] gibt einen wichtigen Ratschlag: *If the latest version does not work, try the older versions.* Wenn also die neueste Version des MacOS-Betriebssystems nicht funktioniert, probieren Sie eine ältere. Netterweise stellt die Seite auch gleich die älteren Versionen zur Verfügung. Aber Sie wissen ja: Apple-AGB-konform ist das nicht, erst recht nicht, wenn Sie die Software nicht zusammen mit einem Macintosh erworben haben. Dieses Buch ist aber lediglich ein Lehr-Beispiel und geht daher nicht weiter auf juristische Fragestellungen ein. Kaufen Sie im Zweifel lieber einen originalen Apple-Computer.

Bevor Sie Ihren Wohnzimmer-PC mit Hilfe des oben erstellten Bootsticks booten, müssen noch ein paar Einstellungen im BIOS/UEFI Ihres Rechners vorgenommen werden. Booten Sie Ihren Rechner mit angesteckter USB-Tastatur und drücken und halten Sie während des Boot-Vorganges die *Entfernen*-Taste auf der Tastatur (auch *DEL*-Taste genannt).

- Laden Sie die BIOS-Default-Werte.
- Wenn Ihre CPU VT-d (Intel Virtualization Technology for Directed I/O) unterstützt, deaktivieren Sie diese Funktion bitte, indem Sie den BIOS-Eintrag auf *disabled* setzen.
- Wenn Ihr System CFG-Lock (Powermanagement) hat, deaktivieren Sie diese Funktion bitte.
- Wenn ihr System *Secure Boot Mode* unterstützt, deaktivieren Sie diese Funktion bitte. Sie ist zwar normalerweise wichtig, um nur signierte Software auszuführen, für den Hackintosh gibt es aber keine Signatur.
- Setzen Sie den Betriebssystem-Typ bitte auf *Other OS*.

- Wenn ihr System einen seriellen Port anbietet, deaktivieren Sie diesen bitte in den BIOS-Einstellungen.
- Aktivieren Sie *XHCI Handoff* (Extensible Host Controller Interface).
- Wenn Ihr Rechner AWARD BIOS der Serie 6 oder ein X58-System hat, deaktivieren Sie bitte USB 3.0.

Speichern Sie anschließend die geänderten BIOS-Einstellungen.

Für den LattePanda Alpha musste ich abweichend von den Standard-BIOS-Einstellungen folgende Einstellungen vornehmen:

- Als Erstes habe ich alle Einstellungen auf die Default-Werte gesetzt.
- *VT-d* muss auch auf diesem Board auf *Disabled* gestellt werden.
- *iDisplay Audio Connect* muss auf *Enabled* stehen, damit Audio an HDMI oder den Display-Port (USB-C-Adapter) weitergeleitet wird.
- *Wake system from S5* steht auf *Disabled*.
- *Primary Display* steht auf *IGFX*.
- *Internal Graphics* steht auf *Enabled*.
- *GTT Size* steht auf 4 MB.
- *Aperture Size* steht auf 1024 MB.
- *XHCI Hand-off* steht auf *Enabled*.
- *Secure Boot Enable* steht auf *Disabled*.

3.3.2 Erstellen eines Bootsticks unter MacOS

Falls Sie bereits einen USB-Bootstick unter Windows erstellt haben, können Sie diesen Abschnitt überspringen. Im Folgenden wird erklärt, wie Sie einen USB-Bootstick unter MacOS erstellen können.

Booten Sie zunächst Ihr MacOS und laden Sie sich dann im App Store den Installer für MacOS Mojave herunter. Nach dem Download fängt dieser automatisch mit der Installation an. Brechen Sie die Installation ab. Der Installer selbst sollte im Ordner */Applications* liegen. Falls Sie den Installer bereits in einen anderen Ordner verschoben haben, schieben Sie ihn bitte zurück in den */Application*-Ordner. Bitte ändern Sie auch nicht den Namen, sondern lassen Sie ihn, wie er ist: *Install macOS Mojave.app*. Falls Sie sich für das Beta-Programm bei Apple registriert haben, heißt die Datei *Install macOS Mojave Beta.app*.

Stecken Sie nun Ihren USB-Stick, der mindestens 8 GB hat, ein. Der nächste Schritt wird alle Daten auf dem USB-Stick löschen. Stellen Sie also bitte vorher sicher, dass

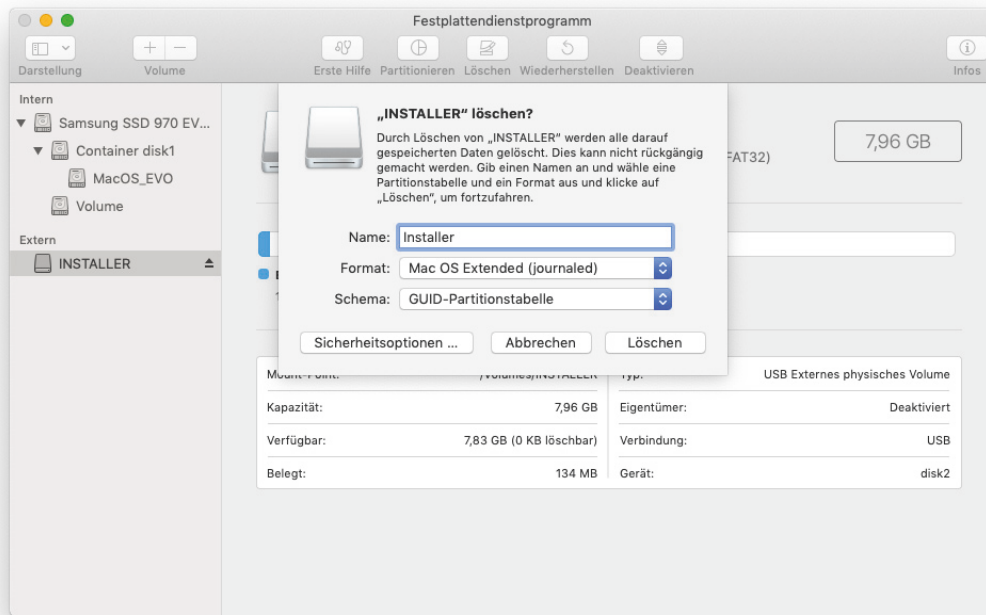


Abbildung 3.5: So formatieren Sie den USB-Bootstick unter MacOS richtig.

der Stick keine Daten mehr enthält, die Sie noch benötigen. Starten Sie dann bitte das MacOS Festplatten-Dienstprogramm und wählen Sie Ihren USB-Stick aus. Wählen Sie *Löschen* und setzen Sie die anderen Einträge wie folgt:

- Vergeben Sie unter *Name* einen aussagekräftigen Namen für den Boot-Stick, beispielsweise *usb-boot*.
- Wählen Sie unter *Format* den Eintrag *MacOS Extended (Journaled)* aus.
- Wählen Sie unter *Schema* den Eintrag *GUID Partition map* aus.

Drücken Sie anschließend den Knopf *Löschen*. Das formatiert den Stick wie angegeben. Nach erfolgreicher Formatierung bindet MacOS den Stick automatisch ein und zeigt ihn unter seinem Namen an.

Öffnen Sie nun ein Terminal und geben Sie den folgenden Befehl ein:

Code-Ausschnitt 3.2: Erstellen eines Boot-Sticks unter MacOS

```
sudo /Application/install\ macOS\ Mojave.app/Contents/Resources/createinstallmedia --volume /Volumes/usb-boot/ --nointeraction
```

Bitte ersetzen Sie im Codeausschnitt 3.2 den Namen *usb-boot* durch den Namen, den Sie beim Formatieren des USB-Sticks angegeben haben. Verfahren Sie analog mit dem Namen des MacOS-Installers, falls Sie die Beta-Version heruntergeladen haben.



Wie unter LINUX können Sie auch in einem MacOS-Terminal die TAB-Taste Ihrer Tastatur nutzen, um Befehle automatisch vervollständigen zu lassen.

Das Terminal wird Sie nach Ihrem Kennwort fragen und nach korrekter Eingabe damit starten, alle Installationsdateien auf den USB-Stick zu schreiben. Dieser Boot-Stick enthält noch keine Clover-Dateien. Um diese Dateien auf den Stick zu bekommen, benötigen wir noch den Clover EFI Bootloader für MacOS. Diesen können Sie sich unter <https://github.com/CloverHackyColor/CloverBootloader/releases> [22] herunterladen. Starten Sie ihn bitte trotz der Warnung, dass es sich um fremde Software handelt.



Abbildung 3.6: Die Installation des Clover EFI Bootmanagers ist schnell erledigt.

Hinweis 3.4 (EFI Bootloader Installation) Wahrscheinlich sind Sie es gewohnt, bei Installationsprogrammen immer auf *weiter* zu klicken, um eine Installation abzuschließen. Machen Sie das hier bitte nicht, da der EFI Loader sonst auf Ihrer Festplatte installiert wird, was wir noch nicht wollen.



Abbildung 3.7: Ändern Sie den Speicherort des Bootloaders, da er ansonsten auf der Festplatte landet.

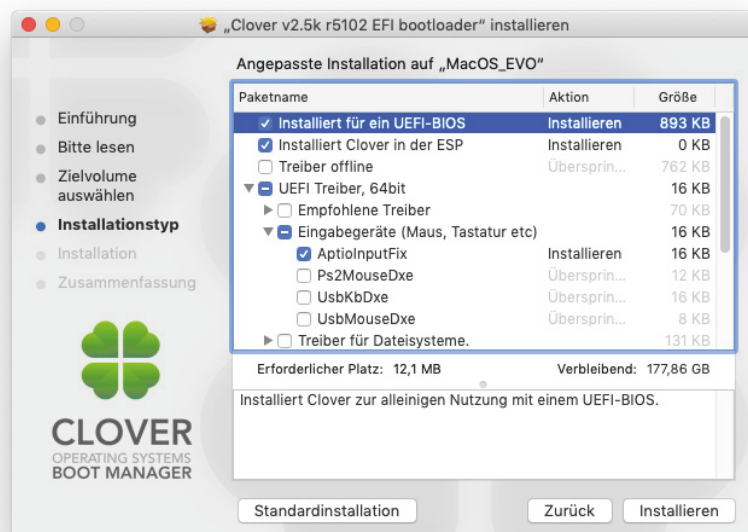


Abbildung 3.8: Ändern Sie bitte die Optionen, die Sie für Ihr Board benötigen.

Wählen Sie bitte *Ort für die Installation ändern/Change install location* und wählen Sie danach den USB-Stick aus. Stellen Sie bitte sicher, dass die Optionen aus Abbildung 3.8 angewählt sind. Nach einem Klick auf *Install* und nach Eingabe Ihres Kennworts startet die Installation des Bootloaders auf den USB-Stick.

Nach erfolgreicher Installation erscheint der USB-Stick mit dem Namen *EFI* eingebunden auf dem Desktop des MacOS. Wenn Sie diese Partition mit einem Doppelklick öffnen, landen Sie im Order *EFI*, der unter anderem den Clover-Ordner enthält. Ab hier gilt das unter Abschnitt 3.3.1 Gesagte.

Sollten Sie für Ihren PC keine vorkonfigurierten Clover-Dateien im Internet finden, ist Handarbeit angesagt. Bei dieser kann Ihnen der Clover Configurator helfen. Wir werden ihn später noch benutzen, um weitere Betriebssysteme in den Bootloader einzutragen. Sie finden ihn auf der Webseite <https://mackie100projects.altervista.org/download-clover-configurator/> [23]. Er läuft unter MacOS und benötigt als Eingabedatei die Datei *config.plist*, welche sich im Order */EFI/CLOVER* befindet. Hat man diese Datei einmal geladen, kann man Clover komplett grafisch administrieren.

3.4 Booten von USB – Die Installation

Stecken Sie Ihren USB-Stick in einen USB-Port und booten Sie Ihren Rechner von USB. Das kann entweder dadurch geschehen, dass dieser Boot-Eintrag im BIOS vorgenommen wird oder indem Sie während des Boot-Vorganges die Taste *F7* so lange drücken, bis ein Boot-Menü erscheint, in dem Sie Ihren USB-Stick als Boot-Medium auswählen können. Machen Sie das und booten Sie Ihren Rechner vom erstellten USB-Stick.

In einem ersten Schritt müssen wir eine Partition für die Installation von MacOS Mojave vorbereiten. Wählen Sie daher bitte nach dem Einstellen der Sprache im Boot-Menü von MacOS das Festplatten-Dienstprogramm (Disk Utility) aus.

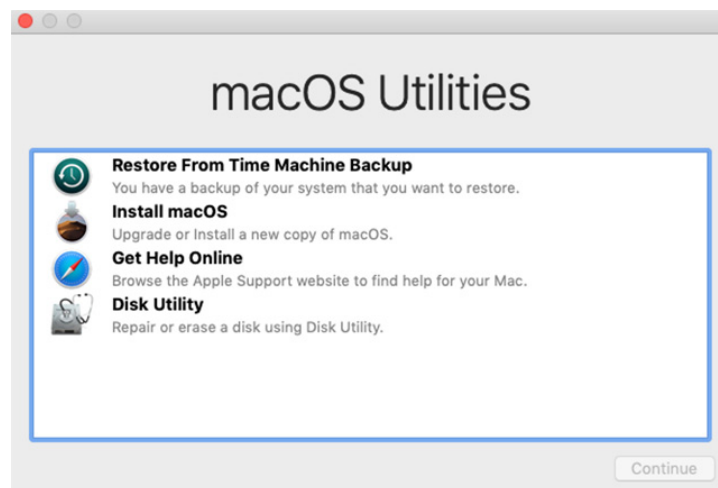


Abbildung 3.9: Das Festplatten-Dienstprogramm unter MacOS erlaubt das Partitionieren der Festplatte(n).

Die Installation von MacOS Mojave setzt das GUID (Globally Unique Identifier) Partitionsschema voraus. Diese Option ist im Festplatten-Dienstprogramm von Apple nur sehr schwer zu finden.

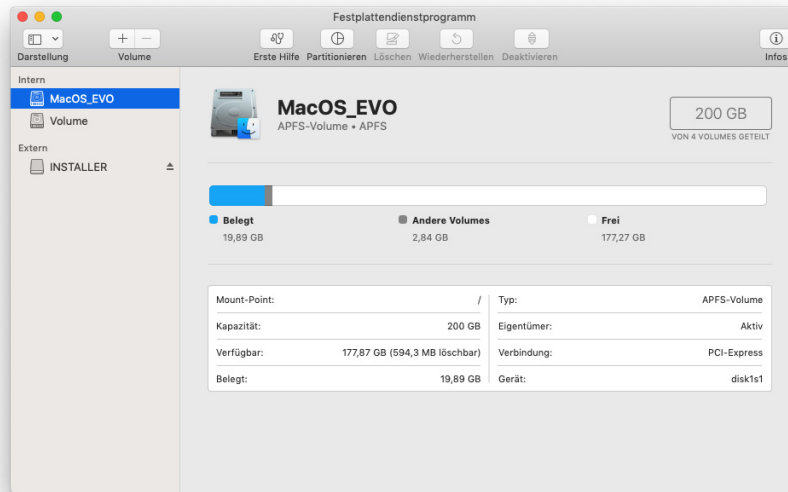


Abbildung 3.10: Das Festplatten-Dienstprogramm gibt nicht sofort alle Einstellungen preis.

In den Abbildungen 3.10 und folgende verwende ich meine SSD als Beispiel. Klicken Sie in der oberen linken Ecke des Festplatten-Dienstprogramms auf *Darstellung/View* und wählen Sie *Alle Geräte einblenden/Show All Devices*. Sie sollten nun die Laufwerksbeschreibungen sehen und die verschiedenen Partitionsnamen eingerückt darunter. Bitte wählen Sie die Partition, die Sie für den späteren Gebrauch von MacOS Mojave formatieren wollen.

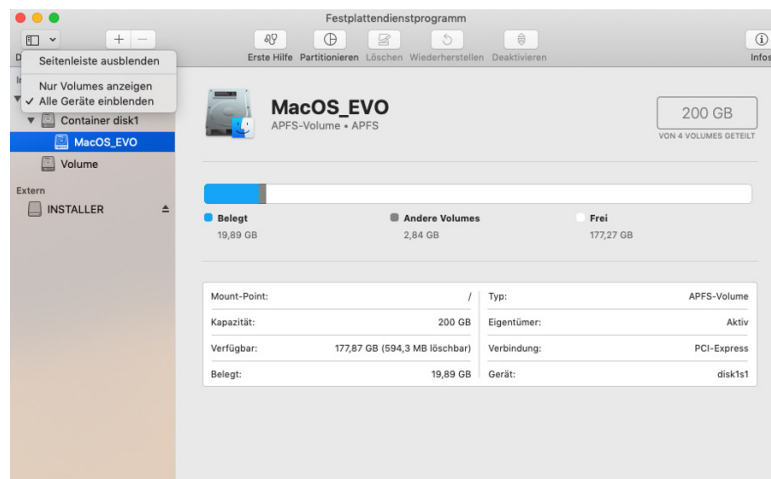


Abbildung 3.11: Ein Klick auf *View* und *Show All Devices* gibt weitere Informationen aus.

Klicken Sie nun auf den *Löschen/Erase*-Reiter. Setzen Sie das Format auf *MacOS Extended (Journaled)* und wählen Sie als Schema *GUID Partition Map*.

Wenn Sie den Laufwerksnamen anstatt seiner Beschreibung ausgewählt haben, wird das Schema nicht angezeigt.



Klicken Sie nun auf *Löschen/Erase*, um die Festplatte zu formatieren und alle darauf befindlichen Dateien zu löschen. Die Partition selbst wird mit dem Dateisystem HFS+ (Hierarchical File System) formatiert. Die Installation von MacOS Mojave ändert dieses Format später nach APFS (Apple File System), vorausgesetzt, Sie benutzen eine SSD. APFS löste HFS+ Mitte 2016 ab und ist auf die Verwendung von SSDs optimiert. Daher kommt es auch in Apples mobilen Geräten unter iOS zum Einsatz.

Beenden Sie anschließend das Festplatten-Dienstprogramm und wählen Sie (Abbildung 3.9) *Install MacOS*. Der Installer sollte Ihnen nun die frisch formatierte Festplatte zur Installation von Mojave anbieten.

Benennen Sie bitte die Installationspartition anders als die Partitionen auf dem USB-Stick. Falls beides beispielsweise MacOS Install heißt, kann das zu Verwirrungen bei der eigentlichen Installation führen. MacOS bootet mehrfach neu und Sie sollten wissen, von wo nach wo Sie installieren.



Der eigentliche Installationsprozess ist selbsterklärend und nimmt 15-30 Minuten in Anspruch.

3.5 Der Bootloader



Abbildung 3.12: Das Bootmenü des Clover Bootloaders lässt sich einfach konfigurieren.

Falls Sie alle erforderlichen Clover-Einstellungen im Vorfeld der Installation beim Erstellen des USB-Bootsticks vorgenommen haben, sollte Mojave bereits weitestgehend funktionieren; falls nicht, ist weitere Handarbeit mit dem Clover-Konfigurator erforderlich.

Nach der Installation bootet Mojave noch nicht von der Festplatte. Es ist immer noch

der USB-Stick erforderlich, der im Clover-Bootmenü jetzt nicht nur die Installation anzeigt, sondern darüberhinaus das frisch installierte Mojave auf der Festplatte sowie eine Rettungspartition. Letztere hat das Installationsprogramm automatisch angelegt. Auf meinem Rechner war bereits ein Windows-System installiert, welches der Clover Bootloader auf dem USB-Stick ebenfalls korrekt angezeigt hat. Im Bootmenü selbst können Sie sich mit den Pfeiltasten auf der Tastatur nach rechts und links bewegen. Die *RETURN*-Taste wählt einen Booteintrag aus. Später werden wir noch den Infrarot-Empfänger *Flirc* installieren. *Flirc* wird als USB-Tastatur erkannt. Sie können dann sogar mit Hilfe einer Fernbedienung den gewünschten Booteintrag auswählen.

Zunächst aber kümmern wir uns darum, dass der Bootloader auf die Festplatte kommt. Hierzu können Sie wieder den Clover EFI Bootloader selbst verwenden, oder aber ein Terminal unter MacOS oder LINUX bemühen.



Abbildung 3.13: Der Clover EFI Bootloader erlaubt es, grafisch auszuwählen, wohin der Bootloader installiert werden soll. Installieren Sie ihn dieses Mal nicht auf den USB-Stick, sondern auf die Festplatte.

Hinweis 3.5 (Manuelle Installation des Bootloaders) Sollten Sie an Terminal-Phobie leiden, also Angst im Umgang mit einem Terminal haben, dann ist das hier eine gute Stelle, um die grafische Installation auszuwählen. Ein falsch installierter Bootloader kann nämlich dazu führen, dass nicht mehr alle Betriebssysteme auf der Festplatte gestartet werden können. Schauen Sie also bei den nachfolgenden Befehlen ganz besonders gut hin, was *von wo nach wo* kopiert werden soll.

Da auf dem USB-Stick, von dem Sie gerade ihr MacOS installiert haben oder ihr installiertes Mojave gestartet haben, bereits ein funktionierender Bootloader vorliegt, können Sie diesen sehr einfach vom USB-Stick auf die Festplatte kopieren. Das Terminal-Programm finden Sie unter Mojave im Launchpad unter *Andere*.



Abbildung 3.14: Das Launchpad zeigt unter MacOS alle installierten Programme.



Abbildung 3.15: Das Terminal-Programm finden Sie unter *Andere*.

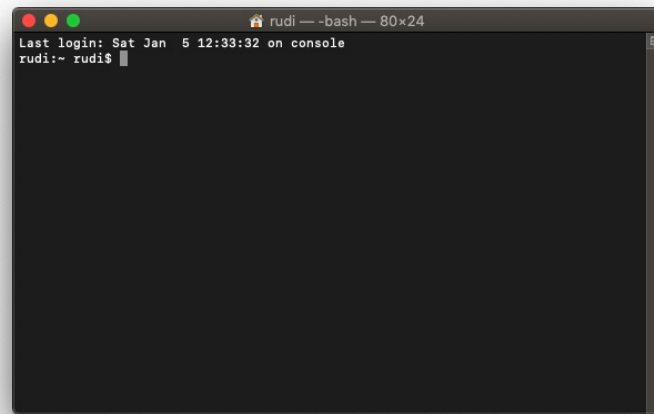


Abbildung 3.16: Im MacOS-Terminal können Sie Befehle eingeben.

Öffnen Sie bitte ein Terminal und geben Sie bitte den folgenden Befehl ein, der Ihnen alle verfügbaren Partitionen auflistet:

Code-Ausschnitt 3.3: Auflisten aller Partitionen unter MacOS

```
diskutil list
```

Finden Sie die EFI-Partition des USB-Sticks (*von hier werden wir gleich kopieren*) und die EFI-Partition der MacOS-Festplattenpartition (*nach hier werden wir gleich kopieren*). In meinem Beispiel nehme ich an, dass die EFI-Partition des USB-Sticks den Namen `/dev/disk2s1` trägt (Code-Ausschnitt 3.3). Weiterhin gehe ich davon aus, dass die EFI-Partition von MacOS den Namen `/dev/disk0` trägt. In diesem Fall lautet der Befehl zum Kopieren der gesamten EFI-Partition des USB-Sticks auf die Festplatte

Code-Ausschnitt 3.4: Kopieren der EFI Partion USB-Stick nach Festplatte

```
sudo dd if=/dev/disk2s1 of=/dev/disk0
```

Noch einmal zur Warnung: Bitte ersetzen Sie sowohl den Namen `disk2s1` als auch den Namen `disk0` mit den jeweiligen EFI-Partitionen Ihres USB-Sticks und Ihrer MacOS Mojave-Partition. Das Terminal fragt Sie nach Ihrem *root*-Kennwort. Geben Sie dieses bitte ein, sofern Sie bei der Installation eines festgelegt haben. Anschließend wird der Bootloader vom USB-Stick auf die Festplatte übertragen. Fahren Sie Ihr MacOS herunter, ziehen Sie den USB-Stick aus der USB-Buchse und starten Sie Ihr System neu. Wenn alles funktioniert hat, sollte Sie der Clover Bootloader begrüßen und innerhalb von 5 Sekunden MacOS starten. Der Befehl `dd` steht übrigens für Disk Dump, also ein bit-weises Kopieren der Festplatte. `dd` fertigt eine 1:1 Kopie des Input Files (*if*) an, indem es dieses auf das Output File (*of*) kopiert.

Den Bootloader können Sie auch mit einem LINUX-Betriebssystem vom USB-Stick auf die Festplatte kopieren. LINUX kennt ähnlich wie MacOS einen Device-Tree, in dem Sie sowohl die EFI-Partition des USB-Sticks (z. B. `/dev/sda1`) als auch die EFI-Partition von MacOS (z. B. `/dev/nvme0n1p1`) finden. Unter LINUX lautet der Befehl dann analog

Code-Ausschnitt 3.5: Kopieren der EFI Partion USB-Stick nach Festplatte

```
sudo dd if=/dev/sda1 of=/dev/nvme0n1p1
```

für den Fall, dass sich die EFI-Partition von MacOS in der dritten Partition einer NVME-SSD befindet und der Bootloader des USB-Sticks in der ersten Partition des Sticks. Theoretisch können Sie LINUX auch von einem 2. USB-Stick aus starten (*/dev/sdb1*) und dieses System zum Kopieren benutzen.

Ich möchte Ihnen noch eine weitere, manuelle Einstellmöglichkeit für den Bootloader zeigen. Die zentrale Datei, die alle Einstellungen von Clover steuert, nennt sich *config.plist*. Sie befindet sich - wie bereits erwähnt - im Verzeichnis *EFI/CLOVER*. Sie können die als Beispiel angegebene Partition von einem LINUX-System, das sich entweder auf Ihrer Festplatte (Kap. 4) oder einem USB-Stick befindet, wie folgt einbinden:

Code-Ausschnitt 3.6: Einbinden der EFI-Partition unter LINUX

```
sudo mount /dev/nvme0n1p1 /mnt
```

Dieser Befehl bindet die MacOS-Partition (*/dev/nvme0n1p1*) in das sogenannte Mount-Verzeichnis (*/mnt*) ein und macht dort die eingebundenen Dateien sichtbar. Der Befehl

Code-Ausschnitt 3.7: Wechseln in das Clover-Verzeichnis

```
cd /mnt/EFI/CLOVER
```

wechselt in das Verzeichnis, in dem sich die zu editierende Datei *config.plist* befindet. Im Kapitel 4 wird der Umgang mit Editoren noch genauer erklärt. Ich habe den Clover-Bootloader so eingestellt, dass das zuletzt geladene Betriebssystem automatisch nach fünf Sekunden wieder geladen wird. Der Benutzer hat also fünf Sekunden Zeit, ein anderes als das zuletzt geladene Betriebssystem zu starten, welches ab da das neue Default-Betriebssystem ist. Die folgenden Zeilen sind dafür in der Datei *config.plist* verantwortlich:

Code-Ausschnitt 3.8: Einstellen des Default-Betriebssystems für Clover

```
<dict>
  <key>Arguments</key>
  <string>nv_disable=1 dart=0 -cdfon -igfxnohdmi lilucpu=8</string>
  <key>DefaultVolume</key>
  <string>LastBootedVolume</string>
  <key>NeverHibernate</key>
  <true/>
  <key>Secure</key>
  <false/>
  <key>Timeout</key>
  <integer>5</integer>
</dict>
```

Um *LastBootedVolume* zu aktivieren, musste ich das #-Zeichen vor *DefaultVolume* entfernen. Alle hier beschriebenen Änderungen können Sie natürlich auch grafisch vornehmen. Wir werden das im Kapitel 8 noch nutzen, um ein installiertes LibreELEC-Betriebssystem in den Clover-Bootloader einzubinden.

Hinweis 3.6 (Vorsicht beim manuellen Editieren der config.plist-Datei) Passen Sie bitte auf, dass Sie die Syntax der Datei *config.plist* nicht zerstören. Der Clover-Bootloader funktioniert dann eventuell nicht mehr. Das Anfertigen einer Sicherheitskopie vor jeder Änderung ist sehr ratsam.

Haben Sie alles richtig gemacht, sollte Mojave jetzt von Ihrer Festplatte aus starten, ohne dass Sie den USB-Stick mit Bootloader benötigen. Auf meinem Lattepanda Alpha 864 denkt Mojave, ein MacBook aus dem Jahre 2016 vorzufinden; Prozessor (Core m3), Speicher (8 GB DDR3) sowie die Grafikkarte (Intel HD Graphics 615) werden korrekt erkannt. Da ich beim Apple-Betaprogramm angemeldet bin, konnte ich die Mojave-Version 10.14.3 Beta updaten, nachdem ich die Version 10.14.0 installiert hatte.



Abbildung 3.17: Startbildschirm von Mojave mit der Information *Über diesen Mac*.

3.6 Probleme und Lösungen

Mein Lattepanda Alpha 864 benötigt zum Booten von MacOS weniger als 40 Sekunden, bis ich mit dem Anmeldebildschirm von MacOS Mojave begrüßt werde. Die gesamte Hardware inklusive Bluetooth funktioniert mit folgenden Einschränkungen:

- **WLAN:** Der eingebaute WLAN-Adapter wird nicht von MacOS Mojave unterstützt. Abhilfe kann hier ein unterstützter USB-WLAN-Stick schaffen. Ethernet funktioniert hingegen problemlos.

- **Bildschirmauflösung:** Obwohl der Lattepanda Alpha 864 4K-Auflösungen unterstützt (sowohl über HDMI bis 30 Hz als auch über einen externen USB-C-HDMI-Konverter bis 60 Hz), ist die Bildschirmauflösung fest auf 1920 × 1080 Punkte eingestellt. Zum Zeitpunkt, als ich dieses Buch geschrieben habe, war noch keine Lösung in Sicht (<https://github.com/novaspirit/macpanda>) [21].
- **iTunes-Filme:** Die Wiedergabe von iTunes-Filmen wird nicht unterstützt. Das Programm bricht bei den Inhalten mit einem HDCP-Fehler (High-bandwidth Digital Content Protection) ab. Der HDMI 1.4-Ausgang des Lattepanda Alpha unterstützt zwar kein HDCP. Es gibt aber passende USB-C auf HDMI-Adapter, die HDCP 2.2 unterstützen. Trotzdem funktioniert die Wiedergabe von iTunes-Filmen hier nicht. Es gibt aber ein sehr aktives Forum, das dieses Problem diskutiert (<https://www.tonymacx86.com/threads/itunes-protected-video-black-screen-solution-hdcp.195083/> [24]).
- **Bluetooth nach Kaltstart:** Manchmal kommt es nach einem Warmstart vor, dass der Bluetooth-Adapter meines Lattepanda Alpha nicht erkannt wird. Ein Ausschalten und anschließender Kaltstart des Rechners brachte den Adapter jedes Mal zuverlässig zum Vorschein.

Noch ein paar Hinweise zum Thema „Update“. Apple bringt hin und wieder neue Versionen seines Betriebssystems heraus. So hat Mojave beispielsweise mit der Version 14.10.0 gestartet, gefolgt von 14.10.1 und .2. Inzwischen sind sicherlich weitere Versionen erschienen. Jedes Update birgt das Risiko, dass ein Hackintosh nicht mehr oder in Teilen nicht mehr läuft. Das ist der Preis, den man bezahlen muss, wenn man den Preis, den man bezahlen soll, nicht bezahlen will. Das sind zunächst alle Unzulänglichkeiten, die mir aufgefallen sind. Einige werden im Laufe der Zeit durch die Internet-Gemeinschaft gelöst werden, andere sicherlich nie. Alles andere lief bei mir problemlos. So spielte *Safari* beispielsweise 4K-Videos von Youtube ruckelfrei ab. Zum Testen, ob Ihr Rechner HDCP unterstützt, kann ich Ihnen das kostenlose Windows-Tools CyberLink Ultra HD Blu-ray Advisor empfehlen (<https://de.cyberlink.com/stat/bd-support/deu/index.jsp> [25]).

3.7 Praxisbeispiele

Vielleicht werden Sie sich gefragt haben, welchen Sinn es hat, MacOS auf einem Medien-Center-PC zu installieren. Ein guter Grund ist iMovie. Dieses kostenlose Videoschnitt-Programm ist eine abgespeckte Version des Profi-Tools Final Cut Pro und bietet alles, was man zum Schneiden, Vertonen und Veredeln eines Films benötigt. Ich selbst nutze dieses Programm gerne, um meine Urlaubsfilme zu schneiden und mit Musik zu unterlegen.

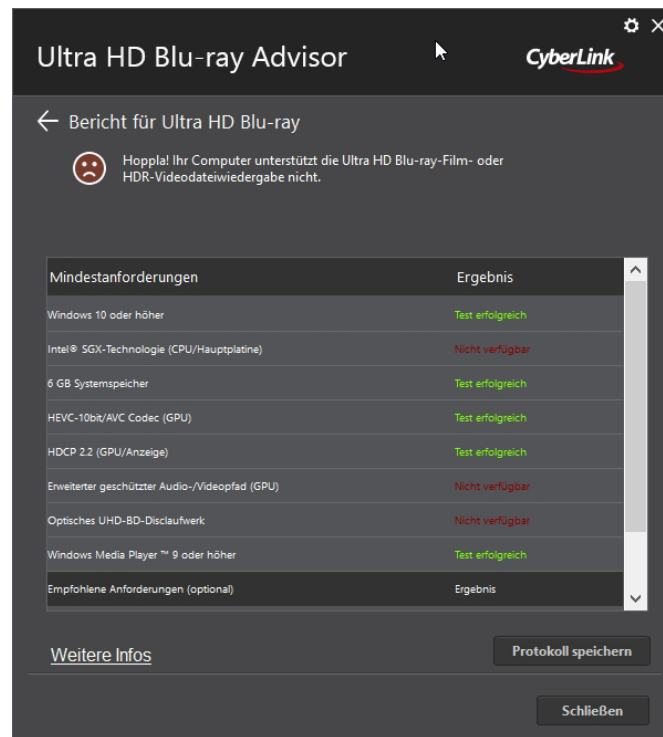


Abbildung 3.18: Cyberlinks Blue-ray Advisor sagt Ihnen unter Windows, ob Ihr System HDCP unterstützt und wenn ja, in welcher Version.

Ein zweites Praxis-Beispiel beschäftigt sich mit einem Spiel unter MacOS. Zum Testen habe ich mir zusammen mit meinem Sohn, einem Fortnite-Profi, Fortnite angeschaut. Im Spiele-Kapitel (Kap. 6) gibt es dann noch einen Vergleich mit der Windows-Version.

3.7.1 Praxisbeispiel iMovie



Das Schneiden und Bearbeiten von Urlaubsfilmen ist aufwendig, schwierig und nur Profis vorbehalten? Stimmt nicht. Apples iMovie ist der beste Beweis und bietet eine Fülle an Optionen, mit denen auch ein Anfänger gut zurecht kommt. Suchen Sie sich aus einer Fülle an Trailern den passenden für Ihren Film aus oder unterlegen Sie den Film mit Musik. Selbst kleinere Nachbearbeitungen wie das Beseitigen von Verwacklungen sind kein Problem. Die Software selbst können Sie kostenlos im Appstore herunterladen.

Sie benötigen dazu eine Apple-ID. Diese können Sie ebenfalls kostenlos anlegen.

Nach dem Öffnen des Programms begrüßt iMovie Sie mit der Projektübersicht. Zu dieser gelangen Sie immer durch einen Klick auf die Schaltfläche *Projekte*. Einen neuen Film können Sie durch einen Klick auf *Neues Projekt* und das anschließende Auswählen von *Film* erstellen. Die Arbeitsfläche des Programms ist in drei Teile aufgeteilt. Zwei Teile befinden sich in der oberen Hälfte des Bildschirms, ein Teil in der unteren Hälfte.

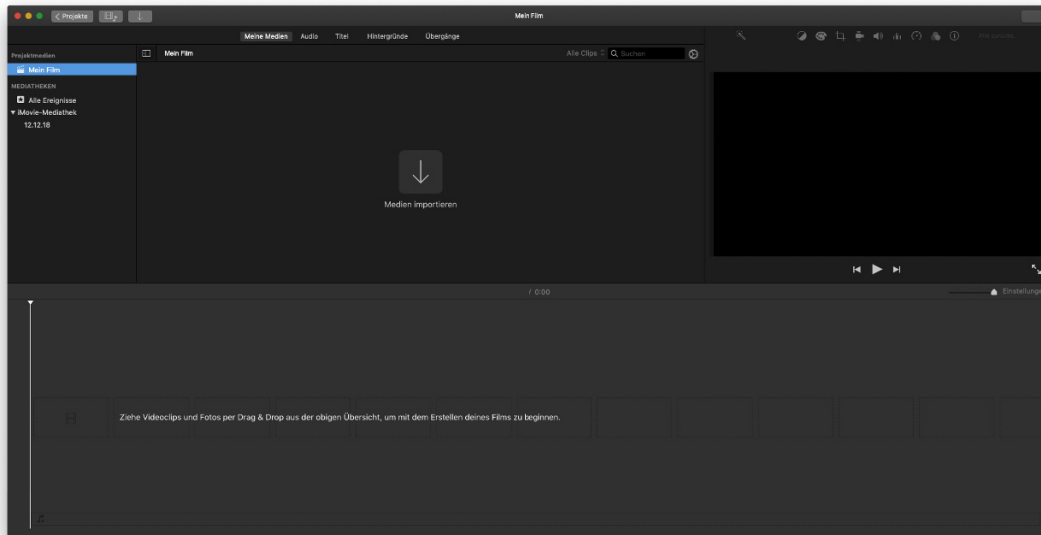


Abbildung 3.19: iMovie ist in drei Bereiche aufgeteilt: Medien, Vorschau und Zeitleiste.

Oben links befinden sich Ihre Medien. Dazu zählen Fotos, Videosequenzen, Musik oder Toneffekte. Hier werden auch Übergänge wie beispielsweise Blenden dargestellt. Oben rechts sehen Sie eine aktuelle Vorschau Ihres Films, die alles berücksichtigt, was Sie und wie Sie es bis dahin bearbeitet haben. Der untere Teil des Bildschirms stellt eine Zeitachse dar, auf der Sie Ihr Bild- und Tonmaterial arrangieren können.

In der Regel bringen Sie eine Fülle an Video-Material aus einem Urlaub mit: Fotos, die mit einer Kamera oder Handys aufgenommen wurden, Video-Schnipsel von der Videokamera, Video-Sequenzen, die mit einer Drohne aufgenommen wurden und so weiter. Zunächst gilt es, diese verschiedenen Medien zu sammeln und zu sortieren, um sie dann in iMovie zu verwenden. Der Import wird dadurch erleichtert, dass alle importierten Dateien einem Ereignis zugewiesen werden können, z. B. dem Ereignis „Urlaub 2019“. Das erleichtert es später ungemein, bestimmte Sequenzen wiederzufinden. Neue Ereignisse lassen sich durch einen Klick mit der rechten Maustaste auf *iMovie-Mediathek* und der Auswahl *Neues Ereignis* erstellen. beim Anschließen einer Kamera an MacOS kann das Betriebssystem alle Bilder mit der Foto-App synchronisieren. Sollten Sie also bereits Fotos in die Foto-App importiert haben, können Sie diese bequem per drag-and-drop in das zuvor erstellte Ereignis ziehen. Der Import von Video-Sequenzen gestaltet sich ähnlich einfach: Klicken Sie mit der rechten Maustaste auf das erstellte Ereignis und wählen Sie anschließend *Medien importieren* aus. Musik nimmt iMovie von iTunes entgegen. Das bedeutet, dass sämtliche Lieder, die Sie später in iMovie verwenden möchten - etwa um den Film mit Musik zu unterlegen - zunächst in iTunes importiert werden müssen.

Nach dieser Vorarbeit kann es an das eigentliche Erstellen des Films gehen. Sämtliche importierte Clips können Sie in der Vorschau anschauen um zu beurteilen, ob Teile des Materials für den Film taugen oder nicht. In der Regel werden Sie nicht den kompletten Video-Schnipsel verwenden, sondern lediglich einen Ausschnitt, der einige Sekunden lang ist, in Ausnahmefällen länger. Die für Sie interessanten Clips können Sie auf die Zeitachse ziehen und die nicht benötigten Sequenzen entfernen. Das geschieht durch

Klicken mit der linken Maustaste auf den Schnipsel. Dieser Vorgang öffnet eine gelbe Box, die Sie nach Belieben verschieben können. Ein Druck auf die Entfernen-Taste (Bearbeiten → Löschen) entfernt das markierte Filmmaterial aus dem fertigen Film.

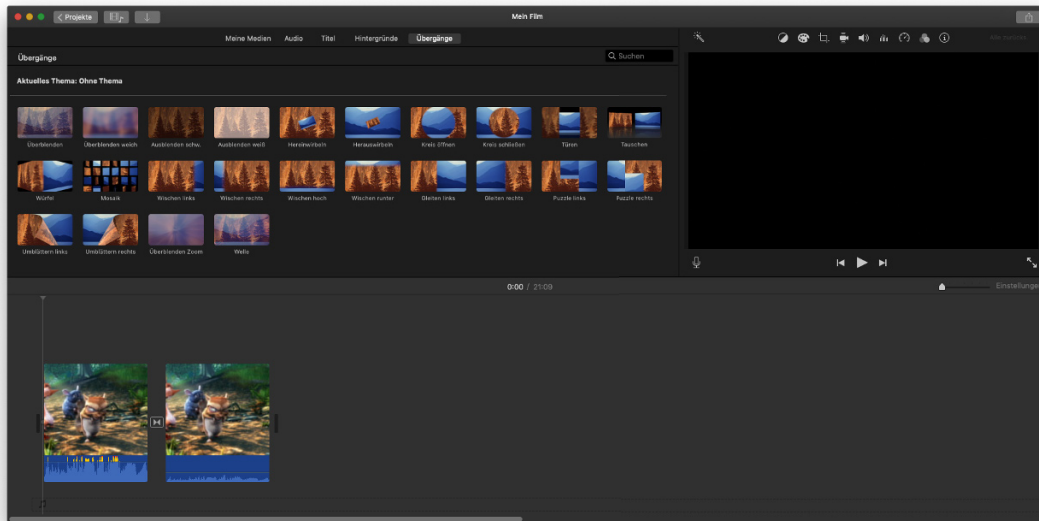


Abbildung 3.20: iMovie bietet eine Fülle an Übergängen, die dosiert eingesetzt werden sollten.

Es steht auch eine Undo-Funktion zur Verfügung, falls Sie versehentlich das Falsche gelöscht haben (Command-Z).

Nachdem Sie zwei Video-Sequenzen eingefügt haben, können Sie diese durch einen Übergang miteinander verbinden. Dazu bietet iMovie Ihnen eine Fülle an Übergängen, aus denen Sie nach belieben wählen können. Machen Sie die Übergänge sichtbar, indem Sie auf den Reiter *Übergänge* klicken und ziehen Sie den Übergang Ihrer Wahl zwischen die beiden eingefügten Video-Clips.



Bei den Übergängen ist weniger oft mehr. Sicherlich verleitet es am Anfang, alle möglichen Arten von Übergängen einzusetzen. Ich empfehle besondere Übergänge aber nur für besondere Sequenzen. In der Regel ist man mit einem einfachem Überblenden bestens bedient.

Die Länge der Übergänge lässt sich bequemen anpassen. Ein Klick auf die Leertaste zeigt Ihnen den bisher arrangierten Film.

Entscheiden Sie selbst, ob Sie die Audio-Spur der Videosequenzen benötigen oder nicht. Falls nicht, können Sie die Lautstärke der Audio-Spur auf „0“ setzen und einen Musiktitel aus iTunes zur Untermalung auswählen. Klicken Sie hierzu auf den Reiter *Audio*, wählen Sie den Titel aus und schieben Sie ihn mit der Maus auf die Zeitachse unter das Video. Auch hier können Sie die Lautstärke einzelner Titel anpassen, um sie aufeinander abzustimmen. Auch Audio-Clips können analog zu Video-Clips gekürzt oder ineinander überblendet werden.

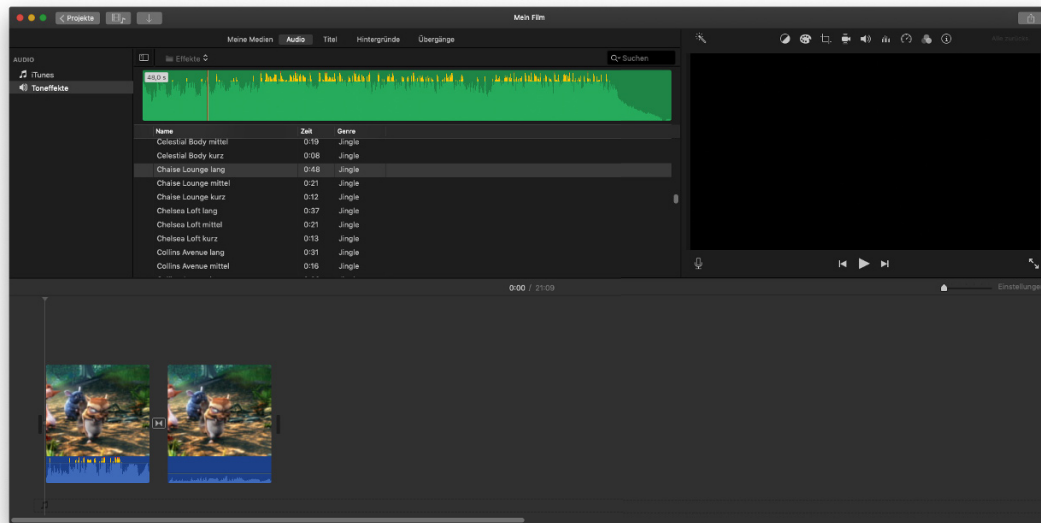


Abbildung 3.21: iMovie stellt viele Audio-Schnipsel zum Einfügen zur Verfügung. Musik kann von iTunes importiert werden.

Für Fotos hält iMovie noch eine Besonderheit parat. Nach dem Einfügen, das analog zu dem Einfügen von Video-Clips erfolgt, gelangen Sie durch einen Doppelklick auf das Foto zu dem sogenannten „Ken-Burn“-Effekt, der es Ihnen erlaubt, Bewegung in ein Foto zu bringen. Wählen Sie hierzu einen Bild-Startausschnitt und einen Bild-Endausschnitt sowie die Zeit für diese Bewegung. In der Regel reichen hier 3-5 Sekunden. iMovie animiert das Foto dann entsprechend.

Sind Sie mit dem Arrangement Ihres Films zufrieden, kann es dennoch sein, dass kleinere Nacharbeiten an einigen Sequenzen erforderlich sind, etwa um die Farbgebung anzupassen oder Verwacklungen zu entfernen. Klicken Sie in diesem Fall auf den zu bearbeitenden Schnipsel. Daraufhin erscheint am oberen Bildschirmrand eine Leiste mit Bearbeitungsmöglichkeiten, aus denen Sie die passende auswählen können. Sie können mit diesen Mitteln den Film nicht nur „aufpolieren“, sondern auch gezielt weitere Effekte wie Zeitlupen verwenden.

Den fertigen Film können Sie in verschiedenen Formaten und Auflösungen exportieren. Wählen Sie hierzu den Menüpunkt *Ablage* und *Bereitstellen*.

iMovie bietet noch weitaus mehr, als dieser kleine Abschnitt des Buches beschreiben könnte. So können Sie beispielsweise nach Belieben Texte in Filmsequenzen einfügen oder Ihrem Film ein Thema wie beispielsweise „News“ verpassen. Texte werden dann im entsprechenden Nachrichten-Stil präsentiert. Weiterhin können Sie komfortabel Trailer erstellen, die ebenfalls wieder verschiedenen Themen wie Abenteuer oder Ferien zugeordnet sind. Fertige Trailer können in eine Filmsequenz konvertiert werden, die dann wiederum im Hauptfilm selbst verwendet werden kann.

Um zu überprüfen, ob iMovie auf meinem Lattepanda Alpha 864 Hackintosh auch mit 4K-Filmen umgehen kann, habe ich mir Big Buck Bunny (4K, 60 Hz) aus dem Internet heruntergeladen (<https://4kmedia.org/big-buck-bunny-4k-demo/> [26]) und

in iMovie importiert. Obwohl der Video-Abspieler Quicktime einige Probleme beim Abspielen des Films hatte, hat iMovie den Video-Schnipsel klaglos importiert und nach einiger Zeit funktionierte auch die Wiedergabe ohne Probleme. Für einen einfachen Test habe ich das Filmsegment zweimal hintereinander eingefügt und eine einfache Blende dazwischen geschaltet. Die gesamte Bedienung des Programms verlief dabei ohne Komplikationen. Der Export in eine Datei stellt dann auch die Option 4K zur Verfügung.



Abbildung 3.22: iMovie kann auf meinem Hackintosh Filme in UHD-Auflösung exportieren.

3.7.2 Praxisbeispiel Fortnite

Fortnite ist ein Shooter-Spiel von Epic-Games, welches unter Jugendlichen besonders beliebt ist. Durch die comichaft Darstellung der Charaktere ist es frei ab 12 Jahren. Das Spiel selbst ist kostenlos, allerdings kann man im Spiel viele Dinge wie Skins kaufen. Das Spiel kann auf der Webseite <https://www.epicgames.com/fortnite/download> [27] heruntergeladen werden. Rufen Sie dazu die oben angegebene Webseite in Safari auf.

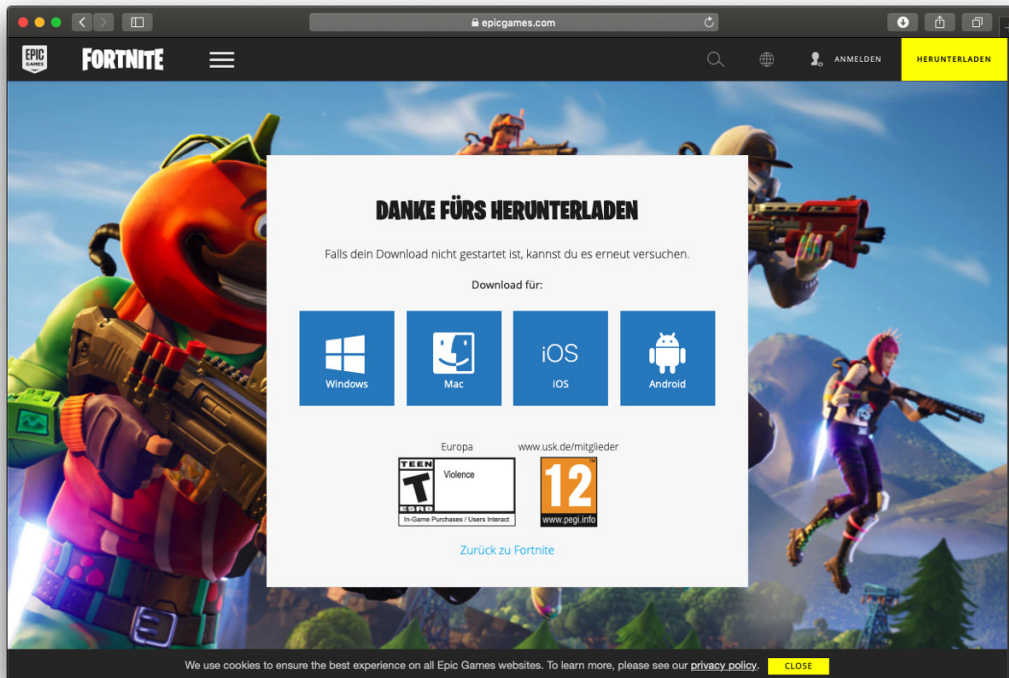


Abbildung 3.23: Fortnite kann kostenlos von der Epic-Games-Seite heruntergeladen werden.

Da der Download selbst bis zu 30 GB groß ist, ist viel Geduld gefragt. Dasselbe gilt auch für den ersten Start des Spiels. Es kann schon gerne ein paar Minuten dauern, bis man endlich spielen kann.

Auf schwächeren Prozessoren und Grafikeinheiten sollten Sie auf jeden Fall die in Abbildung 3.25 gezeigten Einstellungen vornehmen, um halbwegs vernünftig spielen zu können. Auf meinem Lattepanda Alpha erreiche ich mit diesen Einstellungen 30 bis 40 frames pro Sekunde.

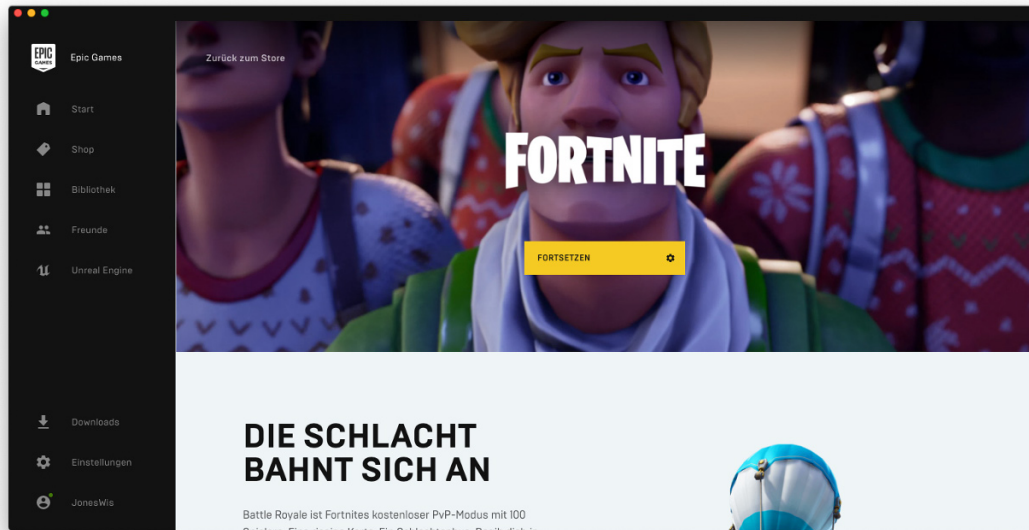


Abbildung 3.24: Der Fortnite-Download nimmt viel Zeit in Anspruch.



Abbildung 3.25: Die empfohlenen Fortnite-Einstellungen für schwächere CPUs und GPUs.

Um vernünftig spielen zu können, habe ich einen Playstation-4 Bluetooth Controller mit über Bluetooth mit meinem Hackintosh verbunden. Das hat auf Anhieb funktioniert: Der Controller wurde ohne weitere Software erkannt und eingebunden. Da sich meine Computer-Spiel-Erfahrungen auf Super-Mario beschränken, habe ich meinen Sohn gebeten, sich Fortnite unter MacOS und unter Windows anzuschauen und das Spiel im Vergleich zu dem Spiel auf seiner Playstation zu bewerten. Dabei hat er Punkte vergeben im Vergleich zum Playstation-Spiel (10 Punkte). Das sagt mein Sohn zu Fortnite unter MacOS auf dem Lattepanda im Vergleich zur Windows-Version:

Die MacOS-Version von Fortnite war nicht spielbar, da das Spiel stark geruckelt hat und es war sehr verzögert. Deshalb konnte man auch nicht „bauen“. Ich habe probiert zu bauen, doch es ging nicht. Die Geschwindigkeit beim Bauen war langsam, manchmal hat das Spiel gar nicht gebaut. Die Grafik war total schlecht. Ich gebe 2 von 10 Punkten.

Im Gegensatz zur MacOS-Version war die Windows-Version sehr gut spielbar, da das Spiel nicht stark geruckelt hat und es fast nicht verzögert war. Es war auch hier nicht perfekt, aber spielbar. Zudem konnte man auch bauen, weil die Geschwindigkeit in Ordnung war und die Grafik auch akzeptabel war. Hier gebe ich 7 von 10 Punkten.

Natürlich kann man diese Aussage nicht verallgemeinern. Sie bezieht sich auf den Prozessor und die Grafikeinheit meines Lattepanda Alpha, also Intel Core-m3 und HD615. Ein leistungsfähiger Core-i7 würde bei Fortnite unter MacOS deutlich besser abschneiden. Die Frage bleibt aber, ob ein Hackintosh in dieser Beziehung schlechter ist als das Original. Die klare Antwort ist, dass ein Hackintosh einem Original in diesem Punkt in nichts nachsteht.

3.8 Fazit

Lohnt es sich nun, einen Hackintosh zu bauen und zu nutzen? Die Antwort ist ja und nein.

Es lohnt sich, weil es Spaß mit sich bringen kann, ein System aufzubauen, das im Vergleich zur originalen Apple-Hardware deutlich preiswerter ist und in den Punkten Geschwindigkeit und Performance mit der originalen Hardware mehr als mithalten kann. Das System ist allerdings nur so lange alltagstauglich, bis ein Update von Apple dafür sorgen kann, dass einige Dinge nicht mehr funktionieren. Ein Hackintosh ist also nie fertig. Darüber hinaus muss man mit einigen Einschränkungen leben. Dazu gehört, dass beispielsweise integrierte WLAN- oder Bluetooth-Karten nicht erkannt werden, DRM bei iTunes-Filmen nicht immer funktioniert und der Bildschirm beim Abspielen von DRM-geschützten Filmen dunkel bleibt oder Bildschirm-Auflösungen sich nicht einstellen lassen. Und selbst wenn alles funktioniert, kann es sein, dass es durch das nächste Update zunichte gemacht wird.

Wenn Sie also keinen Spaß am Basteln haben und nicht damit leben können, dass ein Update Teile Ihrer Arbeit zunichte macht, kaufen Sie lieber das Original.

Da Apple in neue Hardware einen Security-Chip namens T2 einbaut, kann es gut sein, dass die Hackintosh-Systeme in absehbarer Zeit gar nicht mehr funktionieren werden.

Zusammenfassung und Ausblick 3 Dieses Kapitel hat Ihnen gezeigt, wie MacOS auf einem PC installiert werden kann. Dabei gibt es einige Grenzen aufgrund nicht zertifizierter Hardware. Dennoch ist MacOS ein tolles Betriebssystem, das eine unglaubliche Vielfalt an guten und kostenlosen Programmen mit sich bringt. Sollten Sie Spaß an diesem System haben, kaufen Sie sich das Original. Damit sind Sie nicht nur zukunftssicher, sondern auch update-sicher.

Das nächste Kapitel zeigt Ihnen, wie Sie die aktuelle Ubuntu-Version auf Ihrem Wohnzimmer-PC installieren können. Wenn Sie möchten, sogar neben einem existierenden Windows- oder MacOS-Betriebssystem. Ubuntu wird der Ausgangspunkt weiterer Kapitel sein, in denen wir nach und nach die Multimedia-Fähigkeit des Wohnzimmer-PCs auf- und ausbauen. ■

Literaturverzeichnis

- [4] tonymacx86. *Building a CustoMac Hackintosh: The Ultimate Buyer's Guide*. 2019.
<https://www.tonymacx86.com/buyersguide/building-a-customac-hackintosh-the-ultimate-buyers-guide/>
besucht am 23. 12. 2019 (siehe Seiten 2, 12, 16).
- [5] tonymacx86. *[Guide] Intel NUC7/NUC8 using Clover UEFI (NUC7i7Bxx,etc)*. 2019.
<https://www.tonymacx86.com/threads/guide-intel-nuc7-nuc8-using-clover-uefi-nuc7i7bxx-nuc8i7bxx-etc.261711/>
besucht am 23. 12. 2019 (siehe Seiten 3, 16).
- [14] Samsung. *970 EVO NVMe M.2 SSD*. 2019.
<https://www.samsung.com/de/memory-storage/970-evo-nvme-m-2-ssd/MZ-V7E1T0BW/>
besucht am 23. 12. 2019 (siehe Seite 17).
- [15] gparted. *GNOME Partition Editor*. 2019.
<https://gparted.org/download.php>
besucht am 23. 12. 2019 (siehe Seite 18).
- [16] Sourceforge. *Win32 Disk Imager*. 2019.
<https://sourceforge.net/projects/win32diskimager/>
besucht am 23. 12. 2019 (siehe Seite 18).
- [17] Jason Harder. *So ein HackMac – Projekt Hackintosh*. 2016.
<http://printpraxis.net/2016/10/07/so-ein-hackmac-projekt-hackintosh-die-installation>
besucht am 20. 12. 2018 (siehe Seite 20).
- [18] Tu Nguyen. *Create macOS Mojave bootable USB from Windows with Clover Boot Disk*. 2019.
<https://sourceforge.net/projects/win32diskimager/>
besucht am 23. 12. 2019 (siehe Seiten 20, 22).
- [19] Rehabman. *Repositories*. 2019.
<https://bitbucket.org/RehabMan/>
besucht am 23. 12. 2019 (siehe Seite 21).
- [20] tonymacx86. *Kexts*. 2019.
<https://www.tonymacx86.com/resources/categories/kexts.11/>
besucht am 23. 12. 2019 (siehe Seite 21).
- [21] novaspirit. *macpanda*. 2019.
<https://github.com/novaspirit/macpanda>
besucht am 23. 12. 2019 (siehe Seiten 21, 35).
- [22] Github. *CloverBootloader*. 2019.
<https://github.com/CloverHackyColor/CloverBootloader/releases>
besucht am 23. 12. 2019 (siehe Seite 25).
- [23] mackie Projects. *Download Clover Configurator*. 2019.
<https://mackie100projects.altervista.org/download-clover-configurator/>
besucht am 23. 12. 2019 (siehe Seite 27).

- [24] tonymacx86.com. *iTunes Protected Video Black Screen Solution (HDCP)*. 2019.
<https://www.tonymacx86.com/threads/itunes-protected-video-black-screen-solution-hdcp.195083/>
besucht am 23. 12. 2019 (siehe Seite 35).
- [25] Cyberlink. *Webseite*. 2019.
<https://de.cyberlink.com/stat/bd-support/deu/index.jsp>
besucht am 23. 12. 2019 (siehe Seite 35).
- [26] 4kmedia. *Webseite*. 2019.
<https://4kmedia.org/big-buck-bunny-4k-demo/>
besucht am 23. 12. 2019 (siehe Seite 39).
- [27] Epicgames. *Fortnite*. 2019.
<https://www.epicgames.com/fortnite/download>
besucht am 23. 12. 2019 (siehe Seite 40).

4. Ubuntu

4.1	Ubuntu „LTS“ und „aktuell“	47
4.2	Vorbereitungen	47
4.3	Installation	48
4.4	Desktop	60
4.5	Grafiktreiber	61
4.6	Einstellungen	65
4.7	Das System aktuell halten	68

In diesem Kapitel werden wir Ubuntu 18.10 (Cosmic Cuttlefish) auf unserem Medien-center-PC installieren. Dieses System ist Ausgangspunkt für viele weitere Kapitel dieses Buches. Es ist Grundlage für den Video Disk Recorder VDR und Basis für Netzwerke, Textverarbeitung oder Spiele. Die Installation selbst wird Schritt für Schritt erklärt und berücksichtigt dabei das Einbinden aktueller Grafiktreiber.



4.1 Ubuntu „LTS“ und „aktuell“

Zur Installation eines LINUX-Systems stehen viele Distributionen zur Verfügung. Ubuntu ist für Stabilität bekannt und in der Regel funktioniert sämtliche Hardware out-of-the-box. Darüber hinaus bietet es einen großen Bedienkomfort. Seine Stabilität hat Ubuntu übrigens Debian zu verdanken, auf dem es basiert.

Ubuntu wird in zwei Varianten angeboten, einer sogenannten LTS-Version (Long Term Support) und einer aktuellen Version. Zum Zeitpunkt, als ich dieses Buch geschrieben habe, hatte die LTS-Version die Versionsnummer 18.04 und die aktuelle Version die Nummer 18.10. Wie der Name schon sagt, bietet Ubuntu für die Long Term Support-Version einen langen Support, der in der Regel zwei Jahre währt. Prinzipiell sagt man den LTS-Versionen größere Stabilität nach im Vergleich zu den aktuellen Versionen. Ich konnte bei meinen Experimenten aber keine Probleme mit der aktuellen Version feststellen, so dass ich mich entschieden habe, die Version 18.10 zur Installation und später das Update auf die Version 19.10 für die weiteren Kapitel dieses Buches zu verwenden.

4.2 Vorbereitungen

In der Standard-Version benötigt Ubuntu einen Rechner mit mindestens 1 GB Hauptspeicher und eine 3D-fähige Grafikkarte. Das sind allerdings die absolut minimalen Anforderungen. 2-4 GB Hauptspeicher sind als Minimum auf jeden Fall empfehlenswert. Ein komplettes System belegt auf der Festplatte gerne einmal 3-4 GB. Ich empfehle aber auf jeden Fall 32 GB oder besser noch 64 GB Festplattenkapazität, um überhaupt sinnvoll arbeiten zu können.



Bitte führen Sie vor der Installation von Ubuntu auf Ihrer Festplatte eine Datensicherung durch. Falls Sie aus Versehen die gesamte Festplatte formatieren oder eine falsche Partition auswählen, sind Ihre Daten unwiederbringlich verloren.

Die weitere Beschreibung der Installation geht davon aus, dass Ubuntu von einem USB-Stick aus installiert wird. Hierzu reicht eine Größe von 4 GB aus. Weiterhin wird vorausgesetzt, dass die Festplatte bereits wie in Kapitel 3 beschrieben, korrekt partitioniert ist. Falls das nicht der Fall ist, holen Sie das bitte nach, es sei denn, Sie möchten die komplette Festplatte ausschließlich für Ubuntu nutzen.

Ein Ubuntu ISO-Image können Sie von der Webseite <http://releases.ubuntu.com> [28] herunterladen. Wählen sie die Version *ubuntu-18.10-desktop-amd64.iso*. Zum Schreiben des ISO-Images auf den USB-Stick können Sie, wie im Kapitel 3 beschrieben, das Programm *win32diskimager* benutzen.

4.3 Installation



Abbildung 4.1: Vor der Installation können Sie Ubuntu ausprobieren.

Dieser Abschnitt beschreibt Schritt für Schritt die Installation von Ubuntu mit Hilfe eines USB Sticks. Stellen Sie das BIOS Ihres PCs so ein, dass der Rechner von einem USB Stick booten kann. Stecken Sie dann den USB-Stick in einen USB-Port und booten Sie den Rechner von diesem Stick. Nach dem Booten begrüßt Ubuntu Sie mit dem in Abbildung 4.1 gezeigten Bildschirm. Hier können Sie Ihre Sprache einstellen und

auswählen, ob Sie Ubuntu ausprobieren oder direkt installieren möchten. Wählen Sie bitte die Installation aus.

Zur Installation benötigen Sie nicht zwangsläufig eine Internetverbindung. Sollte Ihnen aber eine zur Verfügung stehen – egal ob kabellos oder kabelgebunden –, können Sie bereits während der Installation zusätzliche Pakete wie Sprachpakete oder Updates installieren.

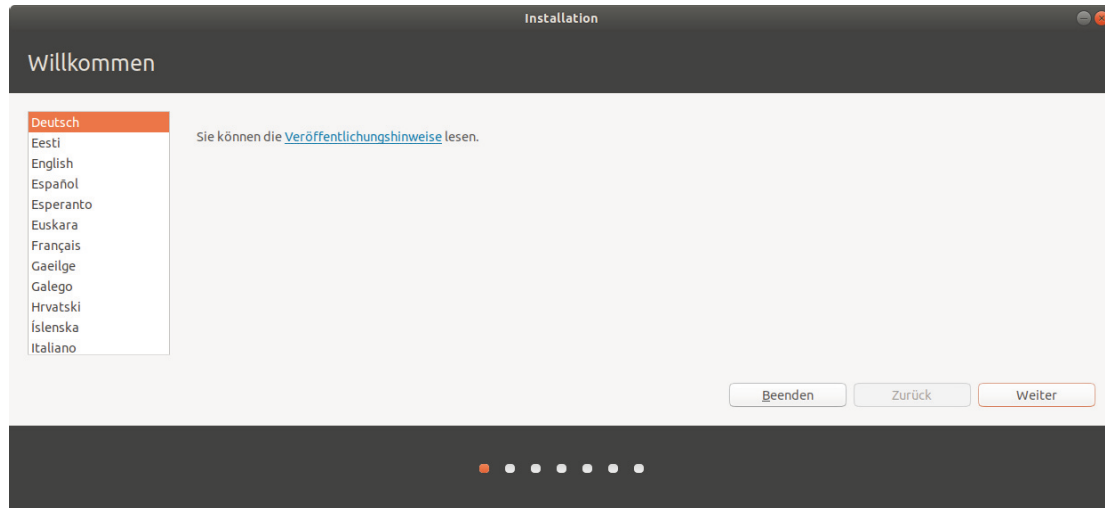


Abbildung 4.2: Ubuntu–Installation: Auswahl der Sprache.

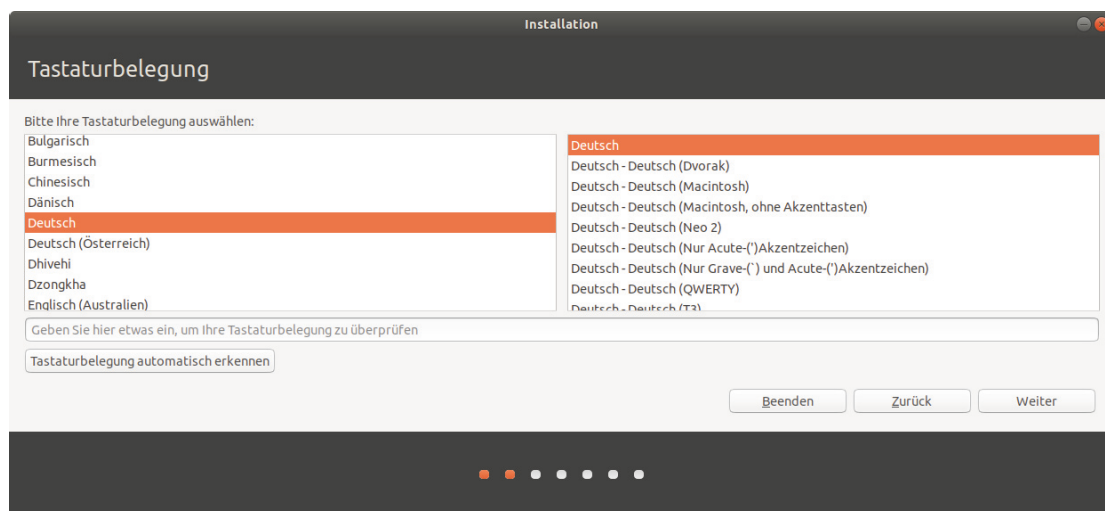


Abbildung 4.3: Ubuntu–Installation: Auswahl der Tastatur.

Nach Ändern der Sprache auf Deutsch erscheinen alle weiteren Installations-Seiten auf Deutsch. Im nächsten Schritt wird der korrekte Tastaturtreiber ausgewählt. Bei einer üblichen PC-Tastatur wählt man die Einstellung *Deutschland* und *Deutsch*. Verwenden Sie eine andere Tastatur, passen Sie diese Einstellung bitte an.

Falls Ihr PC nicht mit einem LAN verbunden ist, haben Sie anschließend die Möglichkeit, ihn mit einem verfügbaren WLAN zu verbinden.

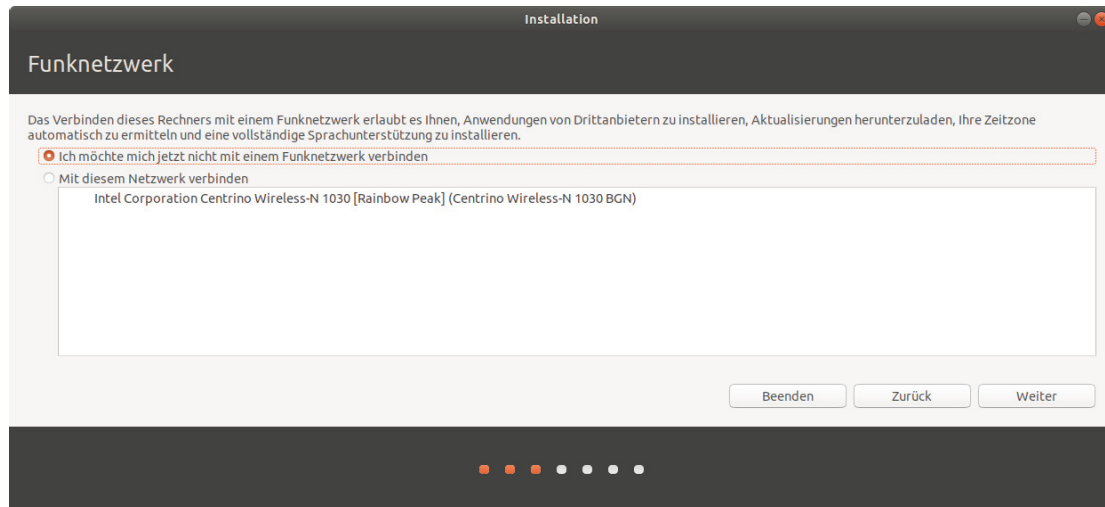


Abbildung 4.4: Ubuntu–Installation: Auswahl des Netzwerkes.

Im folgenden Schritt müssen Sie sich entscheiden, ob Sie Ubuntu in einer Minimal-Version oder in der Standard-Version installieren möchten. In der Minimal-Version bringt Ubuntu nur einen Browser und die wichtigsten Systemprogramme mit. Ich rate daher auch im Hinblick auf die weiteren Kapitel dieses Buches zur Standard-Version.

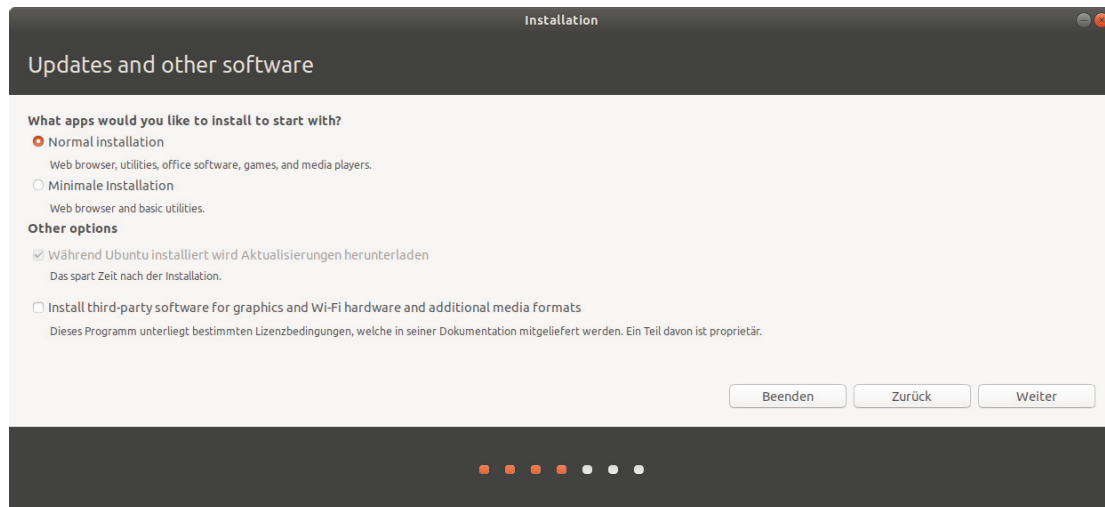


Abbildung 4.5: Ubuntu–Installation: Updates und Software.

Wählen Sie auch die beiden unteren Check-Boxen in Abbildung 4.5 an. Das stellt sicher, dass nach der Installation die neuesten Updates sowie alle benötigten Treiber geladen werden, vorausgesetzt, Sie haben eine funktionierende Internetverbindung. Der Third-Party-Download enthält auch verschiedene Codecs (z. B. mp3), die Sie für das Abspielen von Multimedia-Inhalten benötigen werden. Dieser Schritt kann aber auch ausgelassen werden. Ubuntu bietet Ihnen die Möglichkeit, zu einem späteren Zeitpunkt noch fehlende Programme (und auch die Codes und Drittanbieter-Apps) zu installieren.

Nach diesen Voreinstellungen müssen Sie entscheiden, wo auf der Festplatte Ubuntu Platz finden soll. Falls Sie lediglich Ubuntu als Betriebssystem nutzen möchten, können Sie den Punkt *Festplatte löschen und Ubuntu installieren* auswählen.

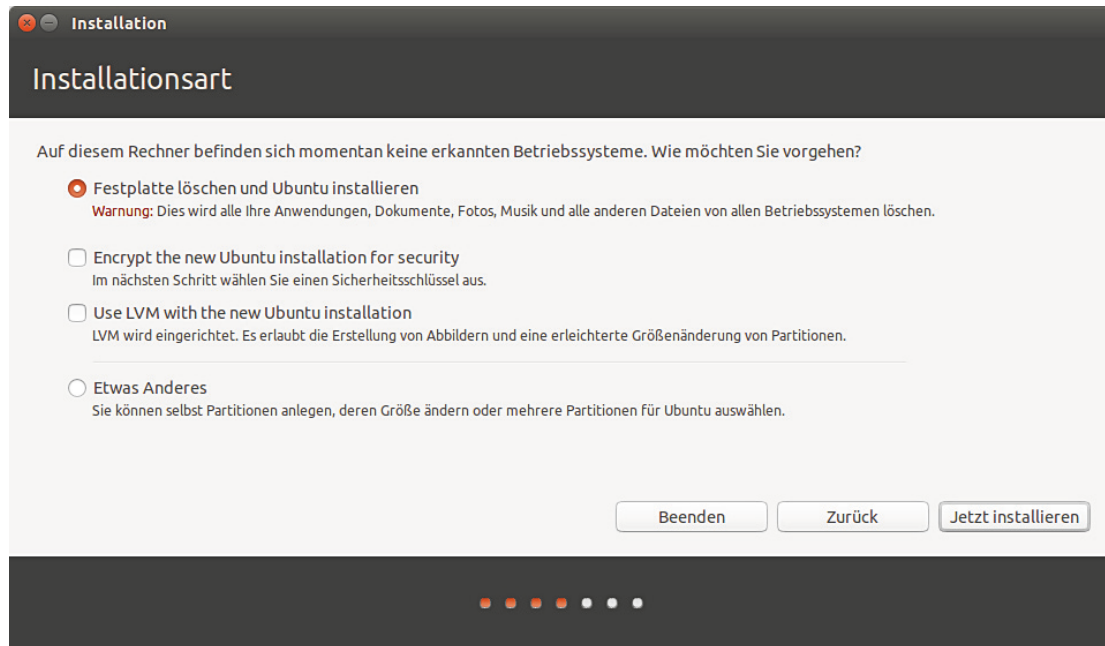


Abbildung 4.6: Ubuntu–Installation: Festlegen der Installationsart 1.

Das Fenster der Abbildung 4.6 wird Ihnen nur bei einer leeren Festplatte gezeigt. Sollten Sie bereits andere Betriebssystem installiert haben oder die Festplatte für Ubuntu bereits mit einer eigenen Partition versehen haben, sieht das Installationsart-Fenster etwas anders aus.

Das Installationsprogramm erkennt auch bereits installierte Ubuntu-Versionen (Abb. 4.7), die man dann erneut installieren kann, durch eine neuere Version von Ubuntu ersetzen kann oder zu denen man ein neues Ubuntu parallel installieren kann.

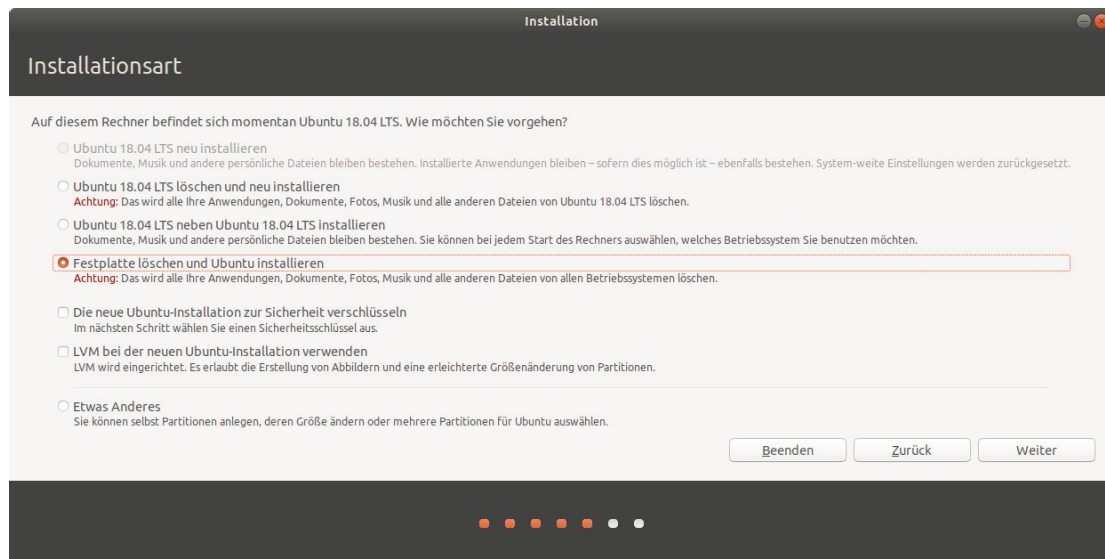


Abbildung 4.7: Ubuntu–Installation: Festlegen der Installationsart 2.

Je nachdem, was bereits auf der Festplatte vorhanden ist, werden die folgenden Optionen angeboten:

- **Festplatte löschen:** Diesen Punkt sollte man nur auswählen, wenn man neben Ubuntu kein anderes Betriebssystem installieren möchte. Hier wird in einem späteren Schritt die gesamte Festplatte unwiderruflich gelöscht und alle Daten, die sich eventuell noch darauf befinden, sind verloren. Sollte Ihr Rechner mehrere Festplatten haben, können Sie eine davon zur Installation auswählen. Achtung: Der Bootloader wird dabei in den Mast-Boot-Record (MBR) der ersten Festplatte geschrieben und überschreibt dabei vielleicht einen schon vorhandenen Bootloader, z. B. Clover. Nutzen Sie ein EFI-Bootmanagement, schreibt Ubuntu eine EFI-Partition auf die erste Festplatte.
- **Ubuntu Versionsnummer löschen und neu installieren:** Bei diesem Punkt wird eine bereits vorhandene Ubuntu-Version mit der angegebenen Versionsnummer durch eine neue Installation derselben Versionsnummer ersetzt. Die alte Version wird dabei komplett gelöscht. Das beinhaltet auch eventuell angelegte Nutzerdaten.
- **Ubuntu Versionsnummer alt auf Ubuntu Versionsnummer neu aktualisieren:** Wenn man bereits eine ältere Ubuntu-Version installiert hat, kann diese durch eine neuere Ubuntu-Version ersetzt werden. Auch dieser Vorgang löscht alle vorhandenen Nutzerdaten.
- **Ubuntu Versionsnummer neben Anderes Betriebssystem:** Hier kann Ubuntu neben einem bereits installierten Betriebssystem installiert werden. Eventuell wird Ihnen noch ein Schieberegler angeboten, der den Festplattenspeicherplatz zwischen dem bereits installierten System und der neuen Ubuntu-Version verteilt. Das kann unter Umständen sogar eine Verkleinerung des bereits installierten Systems einschließen.

Fortgeschrittene Optionen: Für alle Varianten kann das zu installierende System auch verschlüsselt werden oder die Partition in einem LVM abgelegt werden (Logical Volume Manager). Ein LVM ist ein Partitionsschema, das eine Abstraktionsebene zwischen Festplatten, Partitionen und Dateisystemen erlaubt. So kann man beispielsweise veränderbare Partitionen anlegen, die sich auch über verschiedene Festplatten erstrecken können.

Etwas Anderes: Dieser Punkt erlaubt das manuelle Partitionieren von Festplatten. Das ist ratsam, wenn Sie Ubuntu neben einem bereits installierten System wie Windows oder MacOS installieren wollen und der Bootloader eben nicht in den MBR oder die EFI-Partition der ersten Festplatte geschrieben werden soll. Zur manuellen Partitionierung wird das Programm *gparted* verwendet, welches in die Installation integriert wird. Dieses Programm haben Sie bereits im Kapitel 3 kennengelernt. Bei einer leeren Festplatte sehen die Einstellungen in *gparted* aus, wie in Abbildung 4.8 gezeigt wird.

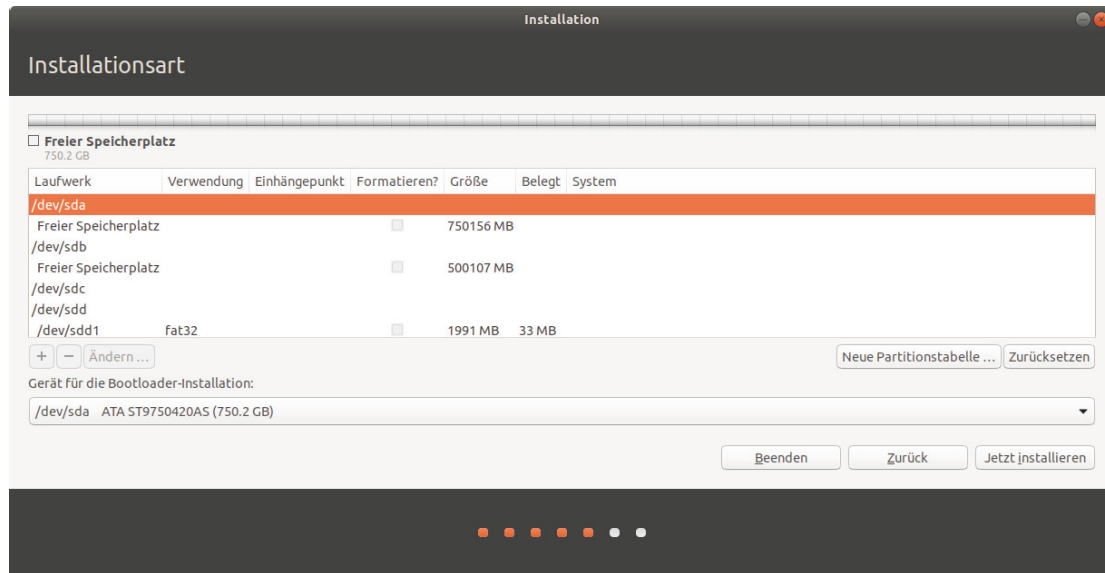


Abbildung 4.8: Ubuntu-Installation: Festlegen der Installationsart 3.

Sollte Ihre Festplatte bereits Partitionen beinhalten, ändert sich das *gparted*-Er-scheinungsbild. Festplatten werden dabei als Device angegeben. Die erste Festplatte heißt */dev/sda*, die zweite */dev/sdb* und so weiter. Partitionen innerhalb der Datenträger werden einfach durchnummeriert, also beispielsweise */dev/sda1* oder */dev/sda2* für die erste und zweite Partition auf einem Datenträger */dev/sda*.

Partitionstabellen können unter der Rubrik *Neue Partitionstabelle...* angelegt werden, um dann Partitionen darin zu erstellen. Der Befehl erlaubt auch das Löschen bereits vorhandener Partitionen. Falls Sie sich beim Einstellen von Partitionen vertan haben, können Sie mit *Zurücksetzen* sämtliche Änderungen, die Sie vorgenommen haben, verwerfen. Das gilt aber nur, wenn Sie die Änderungen noch nicht mit *Jetzt installieren* und anschließender Bestätigung der vorgenommenen Einstellungen endgültig anwenden.

Um vorhandene Partitionen zu bearbeiten, muss die entsprechende Partition angeklickt werden. Das gilt auch für (noch) freie Bereiche. Es können die folgenden Operationen ausgeführt werden.

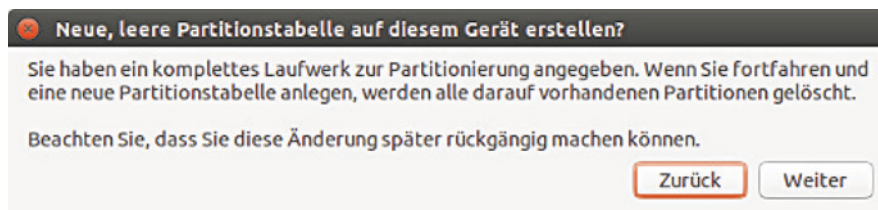


Abbildung 4.9: Ubuntu-Installation: Erstellen einer neuen Partition 1.

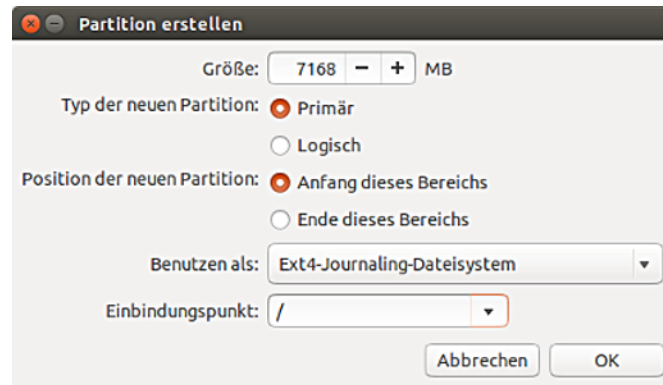


Abbildung 4.10: Ubuntu-Installation: Erstellen einer neuen Partition 2.

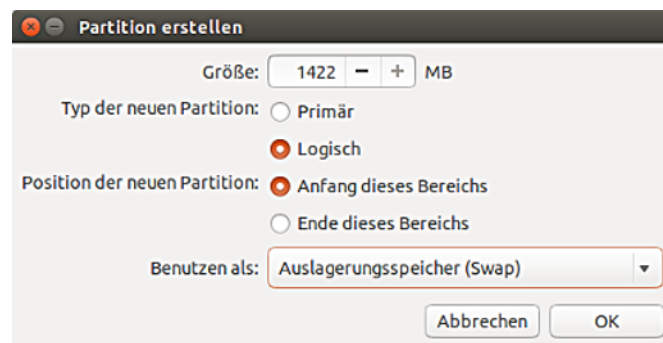


Abbildung 4.11: Ubuntu-Installation: Erstellen einer neuen Partition 3.

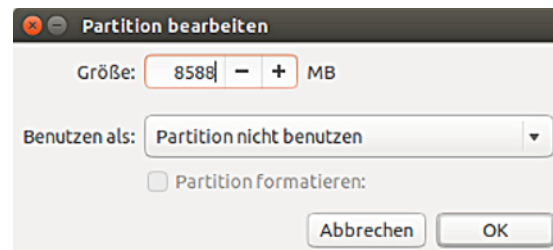


Abbildung 4.12: Ubuntu-Installation: Bearbeiten einer Partition.

- **Partition hinzufügen:** Mit einem „+“ wird der in Abbildung 4.9 gezeigte Dialog aufgerufen, bei dem Sie mit einem anschließenden *OK* alle Einstellungen vornehmen können. Wiederholen Sie diesen Schritt solange, bis Sie alle gewünschten Partitionen angelegt haben. Folgende Einstellungen können vorgenommen werden:
 - Die Größe der Partition in MB kann eingestellt werden.
 - Das anzulegende Dateisystem kann angegeben werden. Ich empfehle für Ubuntu die Verwendung des *ext4*-Dateisystems.
 - Der Einhängepunkt (Abb. 4.9) muss angegeben werden. Dieser sog. „mount point“ ist der Ursprung des LINUX-Dateisystems und lautet in der Regel „/“.

- Der Partitionstyp kann angegeben werden. Hier muss ausgewählt werden, ob eine primäre Partition oder eine logische Partition erstellt werden soll. Das gilt allerdings nur für Tabellen im MSDOS-Schema.
 - Die Position der neuen Partition muss angegeben werden. In der Regel beginnen Partitionen am Anfang des freien Festplattenspeichers.
- **Partition löschen:** Mit „-“ (bzw. *Löschen*) kann eine Partition entfernt werden. Im anschließenden Dialog müssen Sie bestätigen, dass Sie die Partition wirklich komplett löschen möchten.
 - **Partition ändern:** In diesem Menüpunkt kann eine vorhandene Partition verändert werden. Das schließt sogar ein mögliches Verkleinern der Partition ein, sofern der Datenbestand dieser Partition das zulässt. Es können auch fremde Partitionen, wie beispielsweise eine Windows-Partition, verkleinert werden. Da eine Windows-Installation üblicherweise den kompletten Festplattenspeicher einnimmt, besteht durch Verkleinern dieser Partition die Möglichkeit, zusätzlich zu Windows ein Dual-Boot-System aufzusetzen.

Bitte denken Sie daran, vor dem Verkleinern einer anderen Partition eine Datensicherung durchzuführen. Geht hier etwas schief, sind die Daten unwiederbringlich verloren.



- **Korrektur bei neu angelegten Partitionen:** Hier können Sie noch einmal für neu angelegte Partitionen nachträglich das Dateisystem und den Einhängepunkt ändern, jedoch nicht die Größe der Partition.

Möchten Sie nachträglich die Größe einer Partition ändern, löschen Sie diese einfach und legen Sie sie danach wieder an.



Hinweis 4.1 (Hinweise für die Ubuntu-Installation) Für die Installation eines Ubuntu-Systems ist die Root-Partition („/“) dringend erforderlich. Achten Sie bitte darauf, dass dieser Einhängpunkt existiert.

Bei früheren LINUX-Installationen war es erforderlich, eine sogenannte Swap-Partition anzulegen. Diese ist mit der Windows-Auslagerungsdatei zu vergleichen und sollte unter LINUX die 1,3-fache Speichergröße besitzen. Bei einem Hauptspeicher von 8 GB wären hier also 11 GB angeraten. Neuere Ubuntu-Systeme nutzen aber eine Auslagerungs-Datei, so dass hier keine Swap-Partition mehr erforderlich ist.

Es macht evtl. Sinn, Ihre eigenen Dateien, also das `/home`-Verzeichnis auf einer eigenen Partition anzulegen. Der Vorteil liegt darin, dass Sie diese Daten dann auch bei Änderungen am Betriebssystem behalten können.

Falls Sie das System mit EFI einrichten, müssen Sie zwingend eine EFI-Boot-Partition anlegen. Das ist insbesondere bei einer leeren Festplatte der Fall. Bei einem installierten Windows im EFI-Mode existiert diese Partition aber schon.

Im Auswahlmenü *Bootloader* können Sie auswählen, wo der Bootloader *grub2* installiert werden soll. Legen Sie hier den Startbereich der Ubuntu-Installation fest, um einen eventuell vorhandenen Clover-Bootloader nicht zu überschreiben. Die Voreinstellung legt den MBR der Festplatte fest.

Noch ein paar Worte zum Bootloader. Dieser erkennt zwar in der Regel viele andere Betriebssysteme wie Windows und kann diese im Anschluss booten. Allerdings hat er mein MacOS (Hackintosh) nicht erkannt. Daher noch einmal die dringende Warnung: Überschreiben Sie nicht den evtl. installierten Clover-Bootloader im MBR der ersten Festplatte, sondern installieren Sie den Grub-Loader in der Partition, die Sie für die Installation von Ubuntu vorgesehen haben. Clover erkennt das installierte Ubuntu später automatisch und startet dann den Bootloader der Ubuntu-Partition, der dann Ubuntu selbst startet. Defaultmäßig lässt sich der Grub-Bootloader 25 Sekunden Zeit, bevor Ubuntu gestartet wird. Diese Zeit werden wir später noch auf einen erträglichen Wert (1 s) reduzieren.

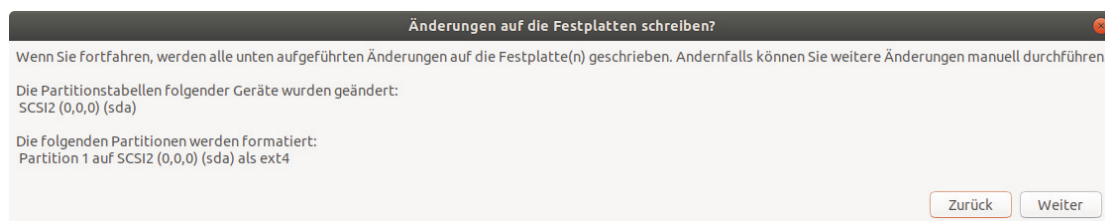


Abbildung 4.13: Ubuntu-Installation: Speichern der Änderungen an der Partitionstabelle.

Erst wenn alle Einstellungen richtig vorgenommen worden sind, sollten Sie auf *Jetzt installieren* klicken. Sie werden noch einmal gewarnt, dass alle Änderungen gespeichert werden und evtl. die Partition, die Sie für Ubuntu verwenden möchten, formatiert wird. Im nächsten Installations-Schritt wird ein Dialog gezeigt, der das Einstellen der Zeitzone zulässt. Wählen Sie hier *Berlin* aus, falls ihr Standort sich in Deutschland befindet. Falls nicht, passen Sie die Auswahl entsprechend an.

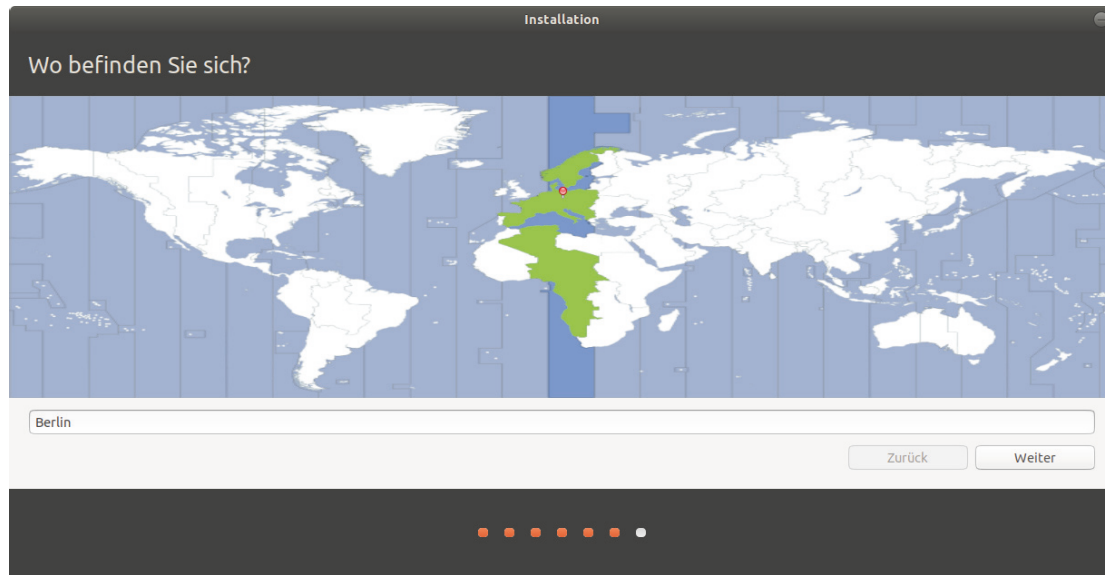


Abbildung 4.14: Ubuntu-Installation: Auswahl der Zeitzone.

Nach Einstellen der Zeitzone müssen wichtige Daten zum Hauptbenutzer des Ubuntu-Systems angelegt werden. Im Einzelnen sind das:

- **Ihr Name:** Geben Sie hier Ihren Namen ein.
- **Name Ihres Rechners:** Vergeben Sie hier einen Namen für Ihren PC. Unter diesem Namen kann der Rechner später neben seiner IP-Adresse im Netzwerk angesprochen werden.
- **Wählen Sie einen Benutzernamen:** Geben Sie hier den Benutzernamen ein, unter dem Sie sich später am Rechner anmelden möchten. Der Benutzername für die Anmeldung am System darf Kleinbuchstaben, Zahlen, Bindestriche und Unterstriche enthalten, jedoch keine Umlaute.
- **Ein Passwort auswählen:** Geben Sie hier ein schwer zu erratendes Kennwort ein, das möglichst Klein- und Großbuchstaben enthält sowie Sonderzeichen. Der Hauptbenutzer hat später volle Gewalt über das System. Das hier angegebene Kennwort sollte also wirklich schwer zu erraten sein. Das System zeigt Ihnen an, ob ihr gewähltes Kennwort sicher ist (**Gutes Passwort**).
- **Passwort wiederholen:** Geben Sie hier dasselbe Kennwort noch einmal an. Dieser Schritt dient nur der Verifikation des von Ihnen vergebenen Kennworts.

Abbildung 4.15: Ubuntu–Installation: Anlegen eines Benutzers.

Weiterhin können Sie angeben, ob der Benutzer automatisch im System angemeldet werden soll oder ob er jedes Mal nach einem Neustart des Rechners sein Kennwort zur Anmeldung eingeben muss. Wählen Sie hier bitte *Automatisch anmelden*. Das ist von Vorteil, wenn wir später einen VDR (Video Disk Recorder) unter Ubuntu einrichten und Sie möchten, dass z. B. eine Fernsehaufnahme nach einem Timer-Start automatisch startet, ohne dass ein Kennwort eingegeben wird.



Weitere Benutzer können später ohne Probleme hinzugefügt werden. Per Default können alle Benutzer die Home-Verzeichnisse anderer Benutzer lesen, aber nicht schreiben. Diese Einstellung kann man aber leicht ändern.

Während der Installation, die von nun an selbständig verläuft, ist eine Slide-Show zu sehen.



Abbildung 4.16: Ubuntu-Installation: Während der Installation.

Nach erfolgreicher Installation muss der Rechner einmal neu gestartet werden, bevor er komplett einsatzbereit ist.

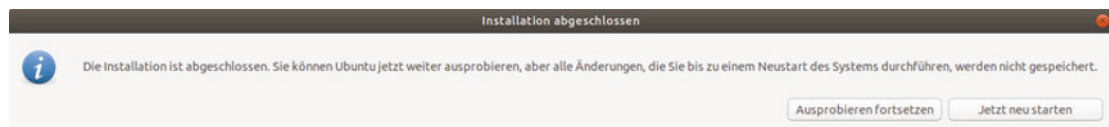


Abbildung 4.17: Ubuntu-Installation: Abschluss der Installation.

Falls der Rechner, auf dem Sie gerade Ubuntu installiert haben, während der Installation nicht Online gewesen sein sollte, haben Sie nun die Möglichkeit, bei der ersten Internetverbindung (LAN oder WLAN) die fehlenden Sprachpakete für die deutsche Sprachunterstützung nachzuinstallieren.

4.4 Desktop

Nach dem Neustart des Rechners sollte Ihr Schreibtisch so ähnlich aussehen wie in der folgenden Abbildung.

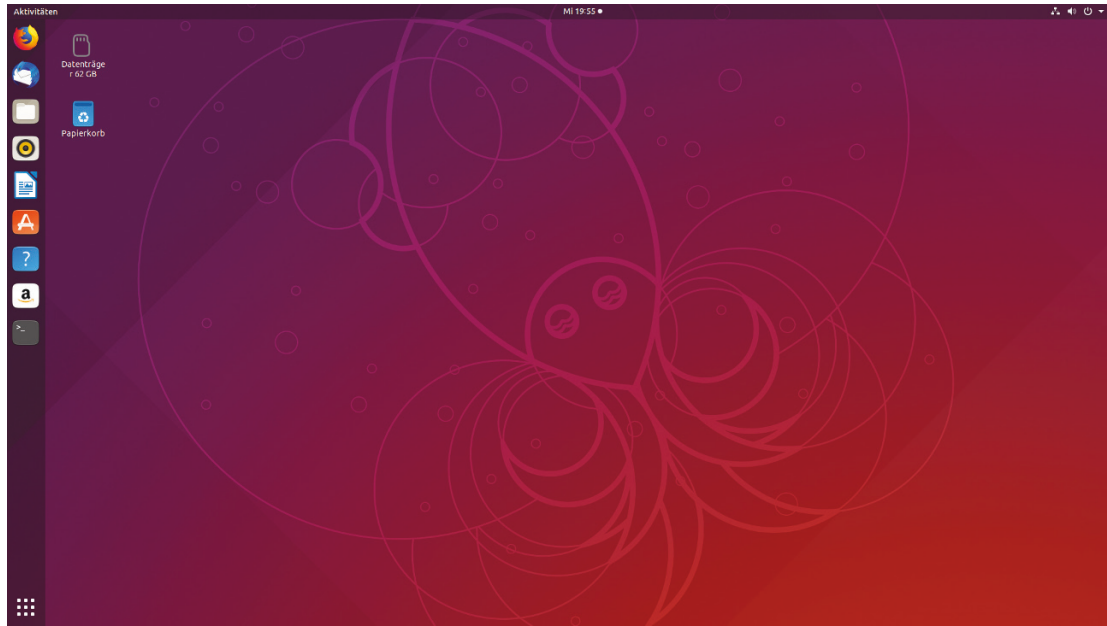


Abbildung 4.18: Der Ubuntu-Schreibtisch nach der Installation.

Bei meinem Lattepanda bindet Ubuntu 18.10 automatisch meine Windows-Partition ein, die als Datenträger auf dem Schreibtisch zu sehen ist. Außerdem habe ich bereits in der linken App-Leiste unten das Terminal hinzugefügt.

Die Leiste auf der linken Seite beinhaltet Programme, die häufig benutzt werden. Programme, die noch nicht in diesem Dock sind, können einfach hinzugefügt werden.



Ein Klick auf das Symbol mit den neun Punkten, das sich unten links auf dem Schreibtisch im Dock befindet, öffnet einen Bildschirm, in dem alle installierten Programme gezeigt werden. Hier gibt es auch eine Suchfunktion, mit der Sie beispielsweise nach *Terminal* suchen können. Ein Klick auf *Aktivitäten* stellt alle Arbeitsflächen neben- und untereinander dar. Die obere Leiste zeigt in der Mitte den Wochentag und die aktuelle Uhrzeit an. In der rechten oberen Ecke können die Netzwerkeinstellungen vorgenommen, die Lautstärke eingestellt oder der Rechner neu gestartet oder heruntergefahren werden.



Abbildung 4.19: Die Programmvorschau unter Ubuntu.

Wenn Sie ein Programm geöffnet haben, erscheint das Logo des Programms im linken Dock. Klicken Sie mit der rechten Maustaste auf das Logo und wählen Sie „Zu Favoriten hinzufügen“ aus, um das Programm dauerhaft im linken Dock zu fixieren.



4.5 Grafiktreiber

Dieser Abschnitt beschreibt, wie Sie aktuelle Grafiktreiber für Ubuntu 18.10 installieren können. Die Grafiktreiber decken alle 2D-Anwendungen ab, wie beispielsweise die beschleunigte Wiedergabe von Videos, aber auch 3D-Anwendungen, wie OpenGL. Die Installation erfolgt über ein Terminal. Dieses können Sie direkt von Ubuntu aus mit der Tastenkombination **CTRL-ALT-T** starten oder indem Sie auf *Aktivitäten* in der oberen linken Ecke des Bildschirms klicken, in der Suche *terminal* eingeben und anschließend auf das Terminal-Icon klicken, das dann erscheint. Alternativ können Sie sich auch mit einem anderen Computer unter Verwendung von *ssh* einloggen (Kapitel 5). Das setzt allerdings voraus, dass Sie auf Ihrem Ubuntu der *ssh*-Server läuft, den Sie von einem Terminal aus so installieren können:

Code-Ausschnitt 4.1: Installation des openssh-Servers

```
sudo apt-get install ssh
```

Der Befehl *sudo* verleiht dem normalen Benutzer zunächst einmal *root*-Rechte. Diese entsprechen den Administrator-Rechten unter einem Windows-Betriebssystem. Der *sudo*-Befehl fordert Sie zur Eingabe des *root*-Kennwortes auf. *apt-get install* installiert Pakete, in unserem Fall das Paket *ssh*, das unter anderem den *ssh*-Server enthält.

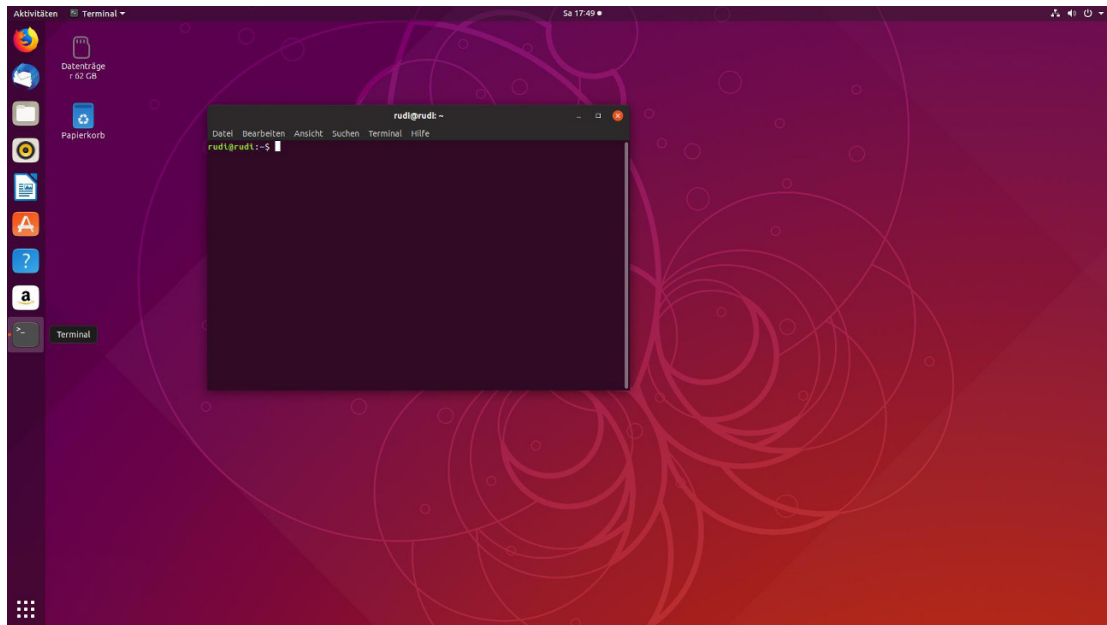


Abbildung 4.20: Das Terminal-Fenster unter Ubuntu.

4.5.1 Intel

Fertige, das heißt kompilierte, Programmpakete können in sogenannten Repositories zur Verfügung gestellt werden. Das Repository *oibaf/graphics-drivers* enthält aktuelle 2D- und 3D-Grafiktreiber für Radeon-, Intel- und NVidia-Hardware. Mit den folgenden Befehlen können Sie dieses Repository zu Ihrem Ubuntu-System hinzufügen und die aktuellen Treiber installieren:

Code-Ausschnitt 4.2: Installation der aktuellen Intel-Grafiktreiber

```
sudo add-apt-repository ppa:oibaf/graphics-drivers
sudo apt update
sudo apt upgrade -y
```

Der erste Befehl des Code-Ausschnitts 4.2 fügt das Repository *oibaf/graphics-drivers* zu Ihrem Ubuntu hinzu, der nächste Befehl schaut nach, ob Updates vorhanden sind und der letzte Befehl installiert sie. Der Parameter *-y* bedeutet dabei, dass auf alle gestellten Fragen, etwa auf die, ob aktuelle Grafiktreiber installiert werden sollen, mit *Ja* geantwortet wird. *y* steht dabei für das englische *yes*. Sollten Sie doch lieber die alten Grafiktreiber behalten wollen, können Sie die oben gemachten Änderungen am System wieder rückgängig machen:

Code-Ausschnitt 4.3: De-Installation der aktuellen Intel-Grafiktreiber

```
sudo apt-get install ppa-purge
sudo ppa-purge ppa:oibaf/graphics-drivers
```


4.5.2 NVidia

NVidia stellt auf der Webseite <https://www.nvidia.de/object/unix-de.html> [11] aktuelle Treiber für LINUX-Systeme zur Verfügung. Grafiktreiber müssen als *root*, also als Administrator, installiert werden. Haben Sie beispielsweise den NVidia-Treiber *NVidia-Linux-x86_64-410.93.run* heruntergeladen, installieren Sie ihn so:

Code-Ausschnitt 4.4: Installation des NVidia-Grafiktreibers

```
sudo su
sh ./NVidia-Linux-x86_64-410.93.run
```

Die oben genannten Befehle gehen davon aus, dass das Terminal in dem Verzeichnis steht, in das Sie den Treiber heruntergeladen haben. Sollte das nicht der Fall sein, können Sie mit dem Befehl

Code-Ausschnitt 4.5: Wechseln des Verzeichnisses unter LINUX

```
cd verzeichnis
```

in dieses Verzeichnis wechseln.

Da der Treiber selbst ein sogenanntes *Shell*-Skript ist, wird er mit dem Befehl *sh* ausgeführt. Die NVidia-Webseite stellt auch eine *README*-Datei zur Verfügung, die eine Beschreibung zum jeweiligen Grafiktreiber enthält. Der Treiber selbst kann nach der Installation mit dem Befehl

Code-Ausschnitt 4.6: De-Installation des NVidia-Grafiktreibers

```
sudo su
nvidia-installer --uninstall
```

deinstalliert werden. Sollte der Treiber auf Ihrem System nicht richtig installiert werden können, kann Ihnen die folgende Befehlsfolge helfen:

Code-Ausschnitt 4.7: Manuelle Installation des NVidia-Grafiktreibers

```
sudo su
sh ./NVidia-Linux-x86_64-410.93.run --extract-only
cd NVidia-Linux-x86_64-410.93-pkg1
make install
```

Die oben angegebene Befehlsfolge entpackt den Treiber zuerst, bevor er dann mit *make install* installiert wird.

4.5.3 AMD

Auch AMD stellt LINUX-Treiber zur Verfügung. Diese finden Sie unter <https://www.amd.com/de/support>. Die AMD-Treiber können nach dem Herunterladen wie folgt installiert werden:

Code-Ausschnitt 4.8: Installation des AMD-Grafiktreibers

```
cd ~/Downloads
tar -Jxvf amdgpu-pro-YY.XX-NNNNNN.tar.xz
cd ~/Downloads/amdgpu-pro-YY.XX-NNNNNN
./amdgpu-install
```

Ersetzen Sie dabei bitte *YY.XX* mit der aktuellen Release-Nummer und die sechs Zahlen *NNNNNN* mit der Build-Nummer des Downloads.

Die AMD-Treiber werden Sie so wieder los:

Code-Ausschnitt 4.9: Deinstallation des AMD-Grafiktreibers

```
sudo su
amdgpu-uninstall
```

Nachdem alle Grafiktreiber erfolgreich installiert worden sind, testen wir die Videobeschleunigung an einem Beispiel. Laden Sie einen Filmschnipsel aus dem Internet herunter, beispielsweise von <http://4ksamples.com> [29]. Hierzu können Sie den Firefox Browser benutzen, der bereits auf Ubuntu installiert ist. Firefox speichert die heruntergeladenen Dateien im *Download*-Verzeichnis des aktuellen Benutzers. Bei meinen Experimenten habe ich mich für den Elysium-Sampler entschieden. Falls Sie ein anderes Beispiel heruntergeladen haben, ändern Sie bitte den Namen in den folgenden Aufrufen entsprechend.

Bevor das Video abgespielt werden kann, muss ein Abspieler installiert werden. Hierzu bietet sich *mpv* an, das wie folgt aus einem Terminal heraus installiert wird:

Code-Ausschnitt 4.10: Installation von mpv

```
sudo apt-get install mpv
```

mpv wird auf einem Intel-System so aufgerufen:

Code-Ausschnitt 4.11: Aufruf von mpv

```
mpv --vo vaapi /home/rudi/Videos/sample-Elysium.2013.2160p.mkv
```

Der Parameter *--vo vaapi* nutzt für den Video Output den beschleunigten Intel *vaapi*-Treiber. Falls Sie ein NVidia-System oder eine AMD-Grafik verwenden, zeigt Ihnen der folgende *mpv*-Aufruf die Video-Optionen Ihres Systems:

Code-Ausschnitt 4.12: Video-Optionen von mpv

```
mpv --vo help
```

Bei einem NVidia-System sollte der Treiber *vdpa* auftauchen, den Sie dann einfach anstelle von *vaapi* einsetzen können. Als Alternative zu *mpv* kann ich noch die *gstreamer*-Tools empfehlen:

Code-Ausschnitt 4.13: Installation der gstreamer-Tools

```
sudo apt-get install libgstreamer1.0-0 gstreamer1.0-tools libxvidcore4 gstreamer1.0-plugins-↵
base gstreamer1.0-plugins-good gstreamer1.0-plugins-ugly gstreamer1.0-plugins-bad ↵
gstreamer1.0-alsa gstreamer1.0-fluendo-mp3 gstreamer1.0-libav
```

Diese Tools verwenden automatisch den korrekten Grafiktreiber für Ihr System. Der heruntergeladene Film kann dann so abgespielt werden:

Code-Ausschnitt 4.14: Abspielen eines Films mit den gstreamer-Tools

```
gst-play-1.0 /home/rudi/Videos/sample-Elysium.2013.2160p.mkv
```

Falls der Film nicht direkt von Ihrem PC aus gestartet wird, sondern remote über eine Shell, muss der *DISPLAY*-Parameter mit angegeben werden, damit der Film korrekt dargestellt werden kann. Der Aufruf lautet dann so:

Code-Ausschnitt 4.15: Abspielen eines Films mit den gstreamer-Tools

```
DISPLAY=:0.0 gst-play-1.0 /home/rudi/Videos/sample-Elysium.2013.2160p.mkv
```

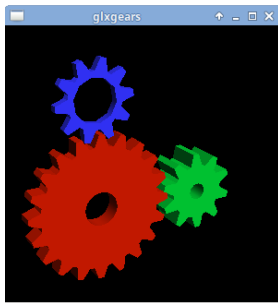


Abbildung 4.21: glxgears zeigt OpenGL-beschleunigte Zahnräder.

Zum Abschluss dieses Abschnitts werfen wir noch einen kurzen Blick auf die OpenGL-Beschleunigung. Dazu installieren wir das Paket *mesa-utils*:

Code-Ausschnitt 4.16: Installation mesa-utils

```
sudo apt-get install mesa-utils
```

und rufen in einem Terminal das OpenGL-Testprogramm *glxgears* so auf:

Code-Ausschnitt 4.17: Test OpenGL

```
glxgears
```

Es sollten die in Abbildung 4.21 gezeigten Zahnräder zum Vorschein kommen, die sich mindestens mit einer Bildwiederholrate drehen sollten, die der Wiederholrate Ihres Bildschirms entspricht, also 50 Hz oder 60 Hz (fps, frames per second).

4.6 Einstellungen

Ich habe an meinem System noch die Wartezeit im Bootloader von zehn Sekunden auf eine Sekunde geändert, da ich Clover als Bootloader nutze und nach Auswahl von Ubuntu nicht noch einmal zehn Sekunden warten wollte. Dazu habe ich in der Datei */etc/default/grub* den Eintrag *GRUB_TIMEOUT=10* so geändert:

Code-Ausschnitt 4.18: /etc/default/grub

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
# info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=1
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command 'vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
```

Zur Änderung können Sie einen Editor Ihrer Wahl benutzen. Standardmäßig ist bereits der Editor *vi* auf dem System installiert. Die Datei */etc/default/grub* muss mit Root-Rechten editiert werden:

Code-Ausschnitt 4.19: Editieren von /etc/default/grub

```
sudo vi /etc/default/grub
```

Bewegen Sie den Cursor mit den Pfeiltasten Ihrer Tastatur auf die *1* in der Zeile *GRUB_TIMEOUT=10* und drücken Sie dann *x* zum Löschen der *1*. Anschließend drücken Sie bitte die Tastenkombination *ESC:wq*. *ESC* beendet den aktuellen Modus, der *:* schaltet in den Befehlsmodus des Editors um und die Tasten *w* und *q* stehen für *write*, also Speichern der Datei, und *quit*, dem Verlassen des Editors.

Nach dem Ändern der Datei müssen wir dem *grub*-Bootloader die Änderungen noch mitteilen. Dies geschieht durch einen Aufruf von

Code-Ausschnitt 4.20: Aktualisieren von grub

```
sudo update-grub
```

Weiterhin habe ich für mein Ubuntu 18.10-System die Quelltext-Pakete aktiviert. Das hat den Vorteil, dass ich später Abhängigkeiten sehr schnell auflösen kann. So installiert der Befehl

Code-Ausschnitt 4.21: Installation der Abhängigkeiten für VDR

```
sudo apt-get build-dep vdr
```

beispielsweise alle Pakete, die zum Übersetzen des Programms *VDR* erforderlich sind. Am schnellsten aktivieren Sie die Sourcecode-Pakete, in dem Sie das Programm *Anwendungen und Aktualisierungen* starten und dort die Sourcecode-Pakete aktivieren.

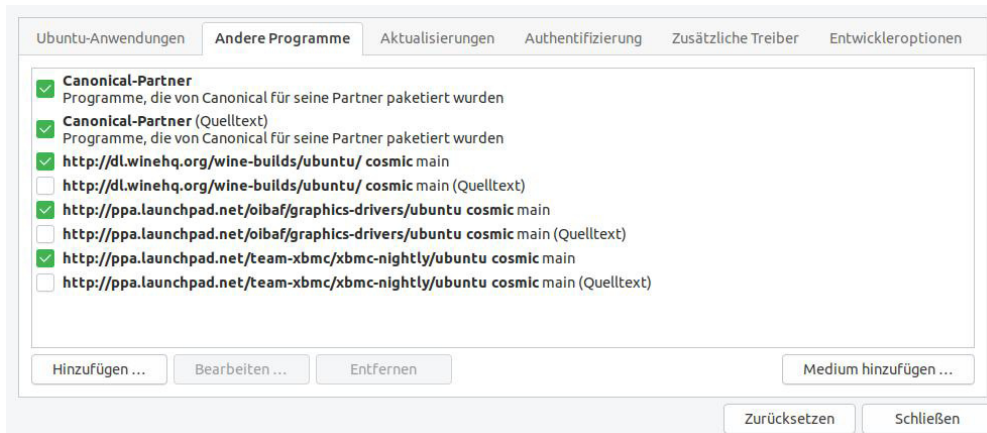


Abbildung 4.22: Aktivieren der Quelltext-Pakete in Ubuntu.

Alternativ können Sie die Pakete auch manuell in die Datei */etc/apt/sources.list* einpflegen. Bei mir sieht diese Datei so aus:

Code-Ausschnitt 4.22: Inhalt der Datei */etc/apt/sources.list*

```
# deb cdrom:[Ubuntu 18.10 _Cosmic Cuttlefish_ - Release amd64 (20181017.3)]/ cosmic main ↵
# restricted
deb-src http://de.archive.ubuntu.com/ubuntu/ cosmic main restricted #Added by software-↵
# properties

# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://de.archive.ubuntu.com/ubuntu/ cosmic main restricted
deb-src http://de.archive.ubuntu.com/ubuntu/ cosmic multiverse universe #Added by software-↵
# properties

## Major bug fix updates produced after the final release of the
## distribution.
deb http://de.archive.ubuntu.com/ubuntu/ cosmic-updates main restricted
deb-src http://de.archive.ubuntu.com/ubuntu/ cosmic-updates multiverse universe main restricted↵
#Added by software-properties

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://de.archive.ubuntu.com/ubuntu/ cosmic universe
# deb-src http://de.archive.ubuntu.com/ubuntu/ cosmic universe
deb http://de.archive.ubuntu.com/ubuntu/ cosmic-updates universe
# deb-src http://de.archive.ubuntu.com/ubuntu/ cosmic-updates universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://de.archive.ubuntu.com/ubuntu/ cosmic multiverse
# deb-src http://de.archive.ubuntu.com/ubuntu/ cosmic multiverse
deb http://de.archive.ubuntu.com/ubuntu/ cosmic-updates multiverse
```

```
# deb-src http://de.archive.ubuntu.com/ubuntu/ cosmic-updates multiverse

## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
deb http://de.archive.ubuntu.com/ubuntu/ cosmic-backports main universe multiverse restricted
deb-src http://de.archive.ubuntu.com/ubuntu/ cosmic-backports main universe multiverse ←
    restricted #Added by software-properties

## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and the
## respective vendors as a service to Ubuntu users.
deb http://archive.canonical.com/ubuntu cosmic partner
deb-src http://archive.canonical.com/ubuntu cosmic partner

deb http://de.archive.ubuntu.com/ubuntu/ cosmic-security main restricted
deb-src http://de.archive.ubuntu.com/ubuntu/ cosmic-security multiverse universe main ←
    restricted #Added by software-properties
deb http://de.archive.ubuntu.com/ubuntu/ cosmic-security universe
# deb-src http://security.ubuntu.com/ubuntu cosmic-security universe
deb http://de.archive.ubuntu.com/ubuntu/ cosmic-security multiverse
# deb-src http://security.ubuntu.com/ubuntu cosmic-security multiverse

# This system was installed using small removable media
# (e.g. netinst, live or single CD). The matching "deb cdrom"
# entries were disabled at the end of the installation process.
# For information about how to configure apt package sources,
deb http://de.archive.ubuntu.com/ubuntu/ cosmic-proposed main universe multiverse restricted
deb-src http://de.archive.ubuntu.com/ubuntu/ cosmic-proposed main universe multiverse ←
    restricted #Added by software-properties
# see the sources.list(5) manual.

deb http://dl.winehq.org/wine-builds/ubuntu/ cosmic main
# deb-src http://dl.winehq.org/wine-builds/ubuntu/ cosmic main
```

Sie enthält auch Quellen für den Windows-Emulator *wine*, den wir uns im Spiele-Kapitel 6 noch näher anschauen werden.

4.7 Das System aktuell halten

Ubuntu bietet regelmäßige Updates, ähnlich wie Windows. Diese Updates können Sicherheitslücken schließen oder verbessern die Funktionalität von Programmen. Das Programm *Aktualisierungsverwaltung* erlaubt es, Updates auf einfache Art und Weise zu installieren. In einem Terminal können dazu die Befehle

Code-Ausschnitt 4.23: Aktualisieren des System mit Hilfe eines Terminals

```
sudo apt-get update
sudo apt-get upgrade
```

verwendet werden.



Abbildung 4.23: Grafische Installation von Ubuntu-Aktualisierungen.

Sollte ein Release-Upgrade der Ubuntu-Version anstehen, kann dieses im Terminal mit dem Befehl

Code-Ausschnitt 4.24: Installation eines Release-Upgrades

```
sudo apt-get dist-upgrade
```

installiert werden.

In der Regel können Sie einfache Updates problemlos installieren. Eventuell müssen Sie selbst kompilierte Programme neu kompilieren. Bei Release-Updates kann es hingegen zu Inkompatibilitäten kommen. Überlegen Sie daher vorher, ob Sie dieses Risiko eingehen möchten oder nicht.



Werfen wir zum Abschluss dieses Abschnitts noch einen Blick darauf, wie neue Programme installiert oder auch deinstalliert werden können. Auch hierzu bietet Ubuntu eine grafische Möglichkeit neben der Möglichkeit, das in einem Terminal zu erledigen.

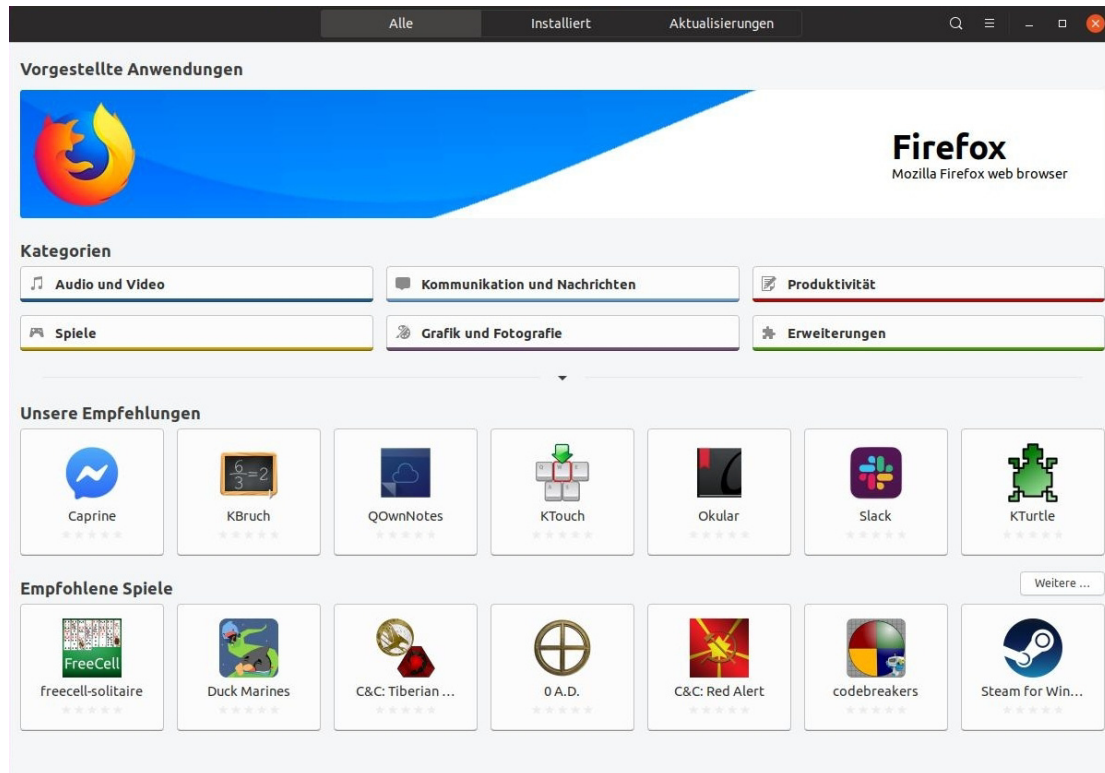


Abbildung 4.24: Mit dem Programm *Ubuntu-Software* kann man Programme suchen, installieren und deinstallieren.

Den Befehl `apt-get install` zum Installieren von Paketen haben Sie bereits kennengelernt. Der Parameter `purge` löscht ein installiertes Paket wieder:

Code-Ausschnitt 4.25: Löschen eines installierten Paketes

```
sudo apt-get purge paketname
```

Sollten beim Löschen eines Programmpaketes abhängige Pakete übrig bleiben, die von keinem anderen Programm mehr verwendet werden, hilft ein

Code-Ausschnitt 4.26: Löschen nicht mehr benötigter Abhängigkeiten

```
sudo apt-get autoremove
```

um diese loszuwerden.

Im Terminal können Sie ebenfalls nach Programmpaketen suchen:

Code-Ausschnitt 4.27: Suchen von Programmpaketen

```
sudo apt search paketname
```

Zum Suchen von Paketen verwende ich gerne das Programm *aptitude*, das ähnlich wie *apt* arbeitet. Es wird so installiert:

Code-Ausschnitt 4.28: Installation von aptitude

```
sudo apt-get install aptitude
```

Ein anschließender Aufruf von

Code-Ausschnitt 4.29: Suchen von Paketen mit aptitude

```
sudo aptitude search paketname
```

sucht alle Pakete, die das Wort *paketname* im Namen tragen.

Zusammenfassung und Ausblick 4 Mit der Installation von Ubuntu haben wir die Grundlage für die weiteren Kapitel geschaffen. Ubuntu ist ein einfach zu bedienendes, vollwertiges Desktop-System. Sie können also eine Tastatur und eine Maus anschließen und Ihre täglichen Arbeiten damit verrichten.

Da Wohnzimmer-PCs aber nicht immer gut zugänglich sind, weil sie beispielsweise in einem Regal verbaut sind, lernen Sie im nächsten Kapitel, wie Sie Ihr Ubuntu einrichten können, wie Netzwerkverbindungen funktionieren und wie Sie von außen - eben ohne Tastatur und Maus - auf dieses System zugreifen können.

Auf diese Art und Weise können Sie mit Hilfe eines Laptop-Computers oder eines Tablets weitere Installationen bequem von der Couch aus erledigen, während Ihr Wohnzimmer-PC bereits in einem Regal verbaut ist. ■

Literaturverzeichnis

- [11] NVIDIA. *Unix-Treiber*. 2019.
<https://www.nvidia.de/object/unix-de.html>
besucht am 23. 12. 2019 (siehe Seiten 11, 63).
- [28] ubuntu releases. *These releases of Ubuntu are available*. 2019.
<http://releases.ubuntu.com>
besucht am 27. 12. 2019 (siehe Seite 48).
- [29] 4ksamples. *Free Downloads 4k Sample Content*. 2019.
<http:4ksamples.com>
besucht am 27. 12. 2019 (siehe Seite 64).

5. LINUX Grundlagen

5.1	Netzwerke einrichten	73
5.2	Editoren	77
5.3	Fernzugriff	80
5.4	OpenVPN	89
5.5	Programme übersetzen oder interpretieren	95
5.6	Ein neuer Kernel	99
5.7	veracrypt	101
5.8	Kleider machen Leute	106

In diesem Kapitel werden LINUX-Grundlagen am Beispiel eines Ubuntu-Systems erklärt. Sie lernen, Internet-Verbindungen nach innen und außen herzustellen oder Ihren PC als Hotspot zu nutzen. Weiterhin möchte ich Ihnen verschiedene Editoren nahe bringen, die Sie zum Programmieren oder Schreiben nutzen können. Nach einem kurzen Ausflug in die Welt der Computerprogramme werden wir einen VPN-Server aufsetzen und ich zeige Ihnen, wie Sie Ihre Daten verschlüsseln können. Ein weiterer Abschnitt zeigt, wie Sie das Aussehen Ihres Ubuntu-Systems ändern und an Ihre Bedürfnisse anpassen können. Das Kapitel schließt mit einer Erklärung, wie Sie WINDOWS-Programme unter LINUX laufen lassen können.



5.1 Netzwerke einrichten

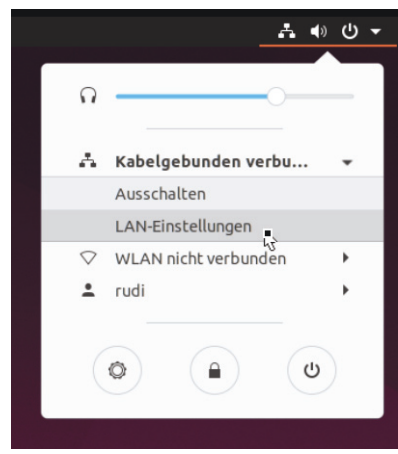


Abbildung 5.1: Die Netzwerkeinstellungen sind in der oberen Ubuntu-Leiste zu finden.

Ein PC - und erst recht ein Wohnzimmer-PC - benötigt eine Netzwerkverbindung. Diese kann nicht nur genutzt werden, um beispielsweise YouTube-Videos abzuspielen oder den TV-Stream im Haus zu verteilen. Sie kann auch genutzt werden, um Programme zu installieren oder das System selbst zu warten.

Wahrscheinlich vergibt Ihr Router zu Hause dynamische IP-Adressen über DHCP (Dynamic Host Configuration Protocol) und Ihr Ubuntu-System ist bereits automatisch mit dem Internet verbunden. Falls Sie eine feste IP-Adresse vergeben wollen, klicken Sie bitte auf das Netzwerksymbol, öffnen Sie die LAN-Einstellungen und geben Sie die feste IP-Adresse und gegebenenfalls die Router-Adresse sowie die Nameserver-Adresse(n) so ein, wie in 5.3. Nach einem Klick auf „Anwenden“ werden die neuen Einstellungen übernommen.

Abbildung 5.2: Alle LAN-Einstellungen können nach einem Klick auf das Zahnrad vorgenommen werden.

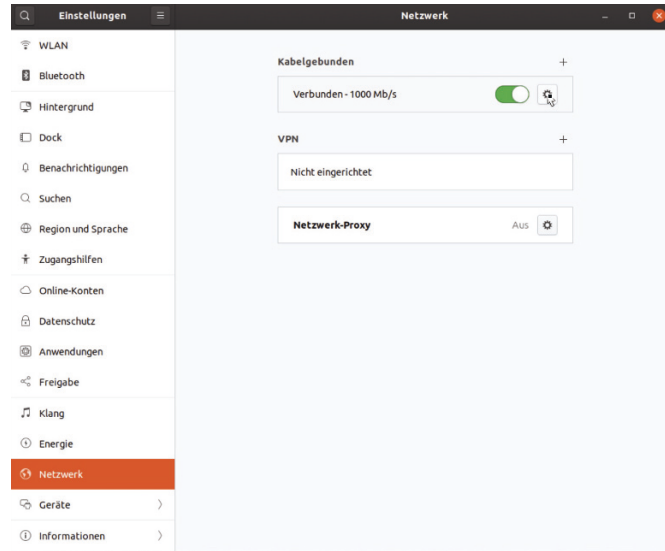
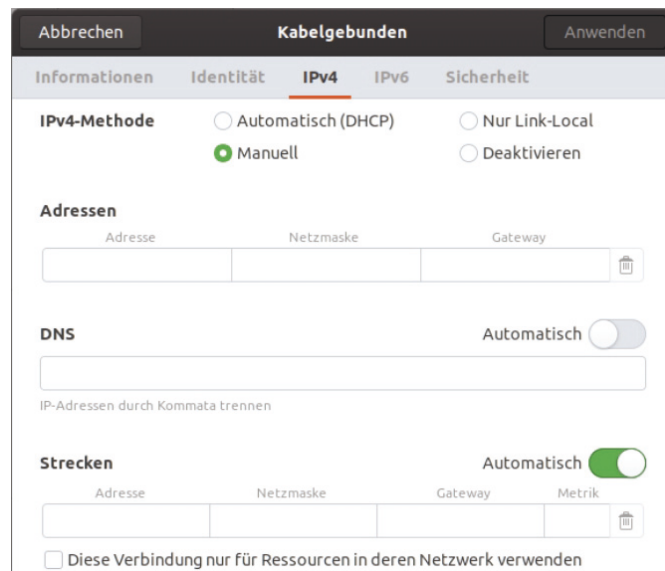


Abbildung 5.3: So stellt man unter Ubuntu eine feste IP-Adresse ein.



Wenn der Wohnzimmer-PC erst einmal im Regal verbaut ist, kann es aufwändig sein, eine Tastatur und eine Maus anzuschließen, um Programme zu installieren, Updates einzuspielen oder den Funktionsumfang des PCs zu erweitern. Daher macht es Sinn, die Multimedia-Zentrale mit der Möglichkeit auszustatten, per Terminal Zugriff zu erhalten. Zu diesem Zweck öffnen wir ein Terminal auf dem Wohnzimmer-PC.

Sie erreichen das Terminal durch einen Klick auf das rechts gezeigte Symbol und das anschließende Eingeben von *Terminal* im Suchfenster. Installieren Sie dann im Terminal den *ssh*-Server so:



Code-Ausschnitt 5.1: *ssh*-Server installieren

```
openssh-server
```


ssh steht dabei für secure shell und bezeichnet einen sicheren Terminal-Zugang von einem Rechner zu einem anderen. Die Verbindung kann dabei entweder über den Rechnernamen oder die IP-Adresse erfolgen. Den Rechnernamen selbst haben Sie beispielsweise bei der Installation von Ubuntu vergeben. Er kann über den Befehl

Code-Ausschnitt 5.2: Abfragen des Rechnernamens

```
hostname
```

ermittelt werden. Die IP-Adresse eines Rechners kann mit dem Befehl *ifconfig* ermittelt werden. *ifconfig* gehört zum Paket *net-tools*.

Code-Ausschnitt 5.3: Installation der *net-tools* und Abfrage der IP-Adressen

```
sudo apt install net-tools
ifconfig
```

Bei mir liefert der Aufruf folgendes Ergebnis

Code-Ausschnitt 5.4: Anzeigen der IP-Adressen

```
enp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.178.75 netmask 255.255.255.0 broadcast 192.168.178.255
    inet6 2001:16b8:9:dc00:fddb:1511:26fe:919e prefixlen 64 scopeid 0x0<global>
    inet6 2001:16b8:9:dc00:2d51:3ab9:404c:5abe prefixlen 64 scopeid 0x0<global>
    inet6 fe80::7aa3:7220:d88f:ff9c prefixlen 64 scopeid 0x20<link>
    ether 00:e0:4c:00:90:8a txqueuelen 1000 (Ethernet)
    RX packets 24154 bytes 5193926 (5.1 MB)
    RX errors 0 dropped 9357 overruns 0 frame 0
    TX packets 6021 bytes 828318 (828.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Lokale Schleife)
    RX packets 5883 bytes 930952 (930.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5883 bytes 930952 (930.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp1s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 50:eb:71:40:24:a5 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Mein Ethernet-Anschluss hat also die IP4-Adresse 192.168.178.75. Möchte ich nun also von einem anderen LINUX-Rechner aus Verbindung zu meinem Wohnzimmer-PC aufnehmen, kann ich das durch den Aufruf von

Code-Ausschnitt 5.5: Terminal-Verbindung zu einem anderen Rechner aufbauen

```
ssh 192.168.178.75
```

erledigen. Falls Sie sich mit einem anderen Benutzernamen *name* einwählen möchten, können Sie der IP-Adresse ein *name@* voranstellen.

Sollte Ihnen keine kabelgebundene Internet-Verbindung zur Verfügung stehen, können Sie Ihren Wohnzimmer-PC auch per WLAN mit dem Internet verbinden - vorausgesetzt, Ihr PC besitzt einen WLAN-Adapter. Sollte das nicht der Fall sein, können Sie diesen auch per USB nachrüsten. Die WLAN-Verbindung können Sie ebenfalls über das Netzwerk-Logo öffnen.

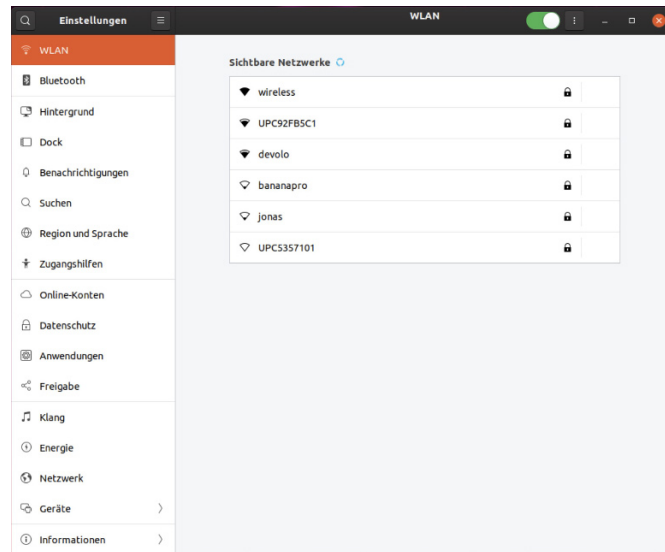


Abbildung 5.4: Ubuntu stellt alle verfügbaren WLAN-Netze übersichtlich dar.

Bei Netzwerkzugriff über Ethernet kann das WLAN auch als Zugangspunkt geschaltet werden. Voraussetzung hierfür ist, dass die verbaute Netzwerkkarte diesen Modus unterstützt. In der Regel sollte das aber der Fall sein. Die Internet-Verbindung über Ethernet wird dann automatisch auf den WLAN-Klienten durchgestellt. Der Zugangspunkt kann ebenfalls über die Netzwerkeinstellungen eingeschaltet werden. WPA/WPA2 wird dabei automatisch eingestellt.



Abbildung 5.5: Für einen Zugangspunkt vergibt Ubuntu automatisch den Hotspot-Namen und den Zugangs-Code.

Falls Sie die Ubuntu-Vorgaben ändern möchten (Name des Hotspots, Kennwort etc.), können Sie dies in der Datei *Hotspot.nmconnection* erledigen, die Sie im Verzeichnis */etc/NetworkManager/system-connections* finden. Nach Änderungen in dieser Datei muss der Netzwerk-Manager neu gestartet werden.

Code-Ausschnitt 5.6: Neustart des Netzwerk-Managers

```
sudo systemctl restart NetworkManager
```

Auch die Verbindung zu einem VPN-Netzwerk gestaltet sich sehr einfach. Installieren Sie bitte zunächst zwei erforderliche Pakete.

Code-Ausschnitt 5.7: VPN-Klienten-Pakete installieren

```
sudo apt install vpnc network-manager-vpnc
```

Öffnen Sie dann wie oben gezeigt die Netzwerkverbindungen und klicken Sie auf die Registerkarte *VPN*. Wählen Sie dort die Schaltfläche *Hinzufügen* aus und anschließend „Cisco-kompatibler VPN-Client (vpnc)“. Diesen haben Sie eben installiert. Nach Eingabe der Serverdaten sowie Ihrem Benutzernamen und dem dazugehörigen Kennwort können alle Daten wieder durch einen Klick auf das Feld *Anwenden* gespeichert werden.

Nach einem Neustart des Netzwerk-Managers können Sie sich über das Netzwerksymbol in der oberen Leiste mit dem gerade eingerichteten VPN-Server verbinden. Später zeige ich Ihnen noch, wie Sie einen eigenen VPN-Server einrichten können.

5.2 Editoren

Im ersten Abschnitt dieses Kapitels habe ich davon gesprochen, dass Änderungen in Dateien vorgenommen werden können. Vielleicht haben Sie sich gefragt, wie das geschehen kann. Die Antwort liefert dieser Abschnitt, in dem ich Ihnen einige Editoren unter LINUX (Ubuntu) vorstellen möchte.

Der einfachste Editor unter LINUX heißt *vi*. Installiert wird er so:

Code-Ausschnitt 5.8: *vi*-Editor installieren

```
sudo apt install vim-common
```

Eine hilfreiche Referenz für den Editor *vi* finden Sie in der folgenden Tabelle:

vi Befehl	Bedeutung
i	Fügt an der aktuellen Cursorposition ein
a	Fügt hinter der aktuellen Cursorposition ein
x	Löscht das Zeichen rechts vom Cursor
dd	Löscht eine ganze Zeile
yy	Kopiert die aktuelle Zeile und fügt sie unterhalb des Cursors ein
ESC	Verlässt den Editor-Modus
:w	Schreibt die Datei
:q	Verlässt den Editor
:q!	Beendet <i>vi</i> , ohne das aktuelle Dokument zu speichern
123G	Springt zur Zeile 123

Tabelle 5.1: Die wichtigsten *vi*-Befehle lernt man schnell.

Sieht kryptisch aus? Ist es auch. Aber glauben Sie mir: Dadurch, dass *vi* auf jedem Kühlschrank läuft, ist es hervorragend geeignet, um schnell Änderungen an Dateien durchzuführen. Zusätzlich können Sie auch die Hilfsseite zum Editor aufrufen, die sogenannte Manual Page oder Handbuch-Seite.

Code-Ausschnitt 5.9: Manual-Page zum *vi*-Editor aufrufen

```
man vi
```

Hier erhalten Sie eine detaillierte Auflistung aller Parameter, die man beim Aufruf übergeben kann. Diese Hilfe ist für die meisten Programme verfügbar. Wenn Sie es nicht ganz so kryptisch mögen, kann ich Ihnen den kleinen Editor *joe* ans Herz legen. Raten Sie, was kommt? Genau, wir installieren ihn mit dem Befehl

Code-Ausschnitt 5.10: *joe*-Editor installieren

```
sudo apt-get install joe
```

Sie können übrigens auch *aptitude* zum Installieren von Software verwenden (*sudo apt install aptitude*). Der entsprechende Befehl lautet dann

Code-Ausschnitt 5.11: Installation von *joe* mit *aptitude*

```
sudo aptitude install joe
```

aptitude hat den Vorteil, dass es komplizierte Installations-Situationen besser auflösen kann als *apt-get*. Aber zurück zu unserem Editor *joe*. Rufen Sie ein mit *vi* erstelltes Dokument *test* auf oder erstellen Sie ein neues Dokument:

Code-Ausschnitt 5.12: Dokument mit dem Editor *joe* bearbeiten

```
joe test
```

Im Editor *joe* können Sie sich frei bewegen, einfügen und löschen, ganz so wie Sie es von einem Textverarbeitungsprogramm gewohnt sind. Aber das Beste kommt noch: Man muss sich keine kryptischen Befehle merken! *joe*'s Befehle kann man sich mit dem *CTRL-h*-Kommando anzeigen lassen. Das sieht dann ungefähr so aus wie in Abbildung 5.6. Der wichtigste Befehl beim Editor *joe* ist *CTRL-x*. Dieses Kommando speichert die Datei und verlässt den Editor.

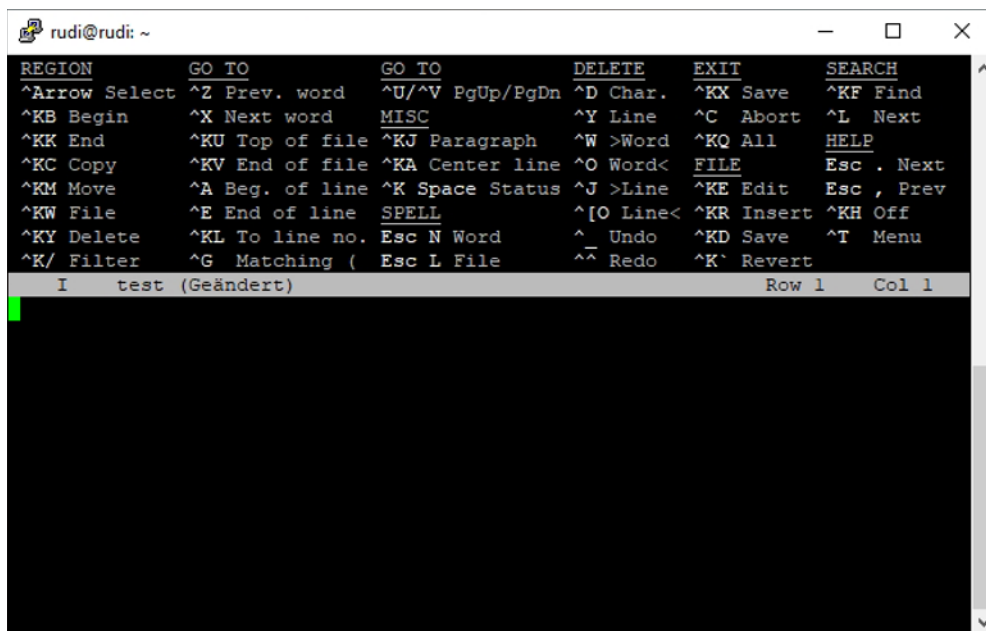


Abbildung 5.6: Die Hilfe zum Editor *joe* kann man im Editor selbst aufrufen.

Beide bisher vorgestellten Editoren *vi* und *joe* arbeiten innerhalb der aufgerufenen Konsole. Für diejenigen, die lieber grafisch arbeiten, ist der Editor *emacs* eine gute Wahl. Er beherrscht unter anderem das Syntax-Highlighting, stellt verschiedene Schriftstücke wie Quelltext-Code oder Python-Code farblich passend dar.

Code-Ausschnitt 5.13: Den Editor *xemacs* installieren

```
sudo apt install xemacs21
```



Bitte denken Sie daran, dass *xemacs* im Gegensatz zu *vi* und *joe* eine grafische Benutzeroberfläche benötigt, auf der er läuft. Ruft man beispielsweise unter Windows (nächster Abschnitt) in einem Terminal LINUX-Programme auf, die eine grafische Benutzeroberfläche (X11) voraussetzen, quittieren diese Programme ihren Dienst mit der Fehlermeldung *Error: Can't open display*. Abhilfe schafft hier ein Fernzugriff unter VNC oder die Weiterleitung des X11-Displays.

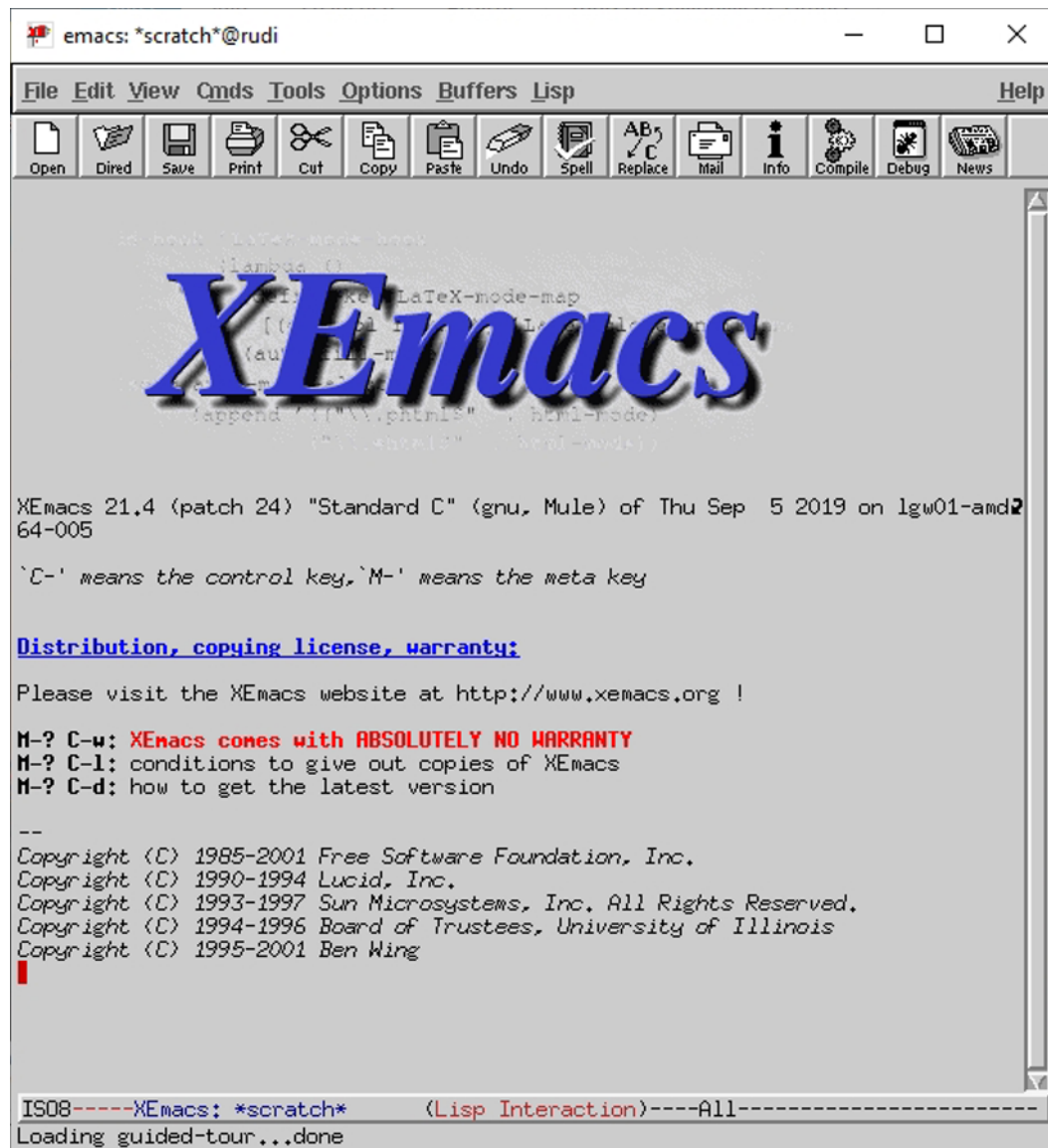


Abbildung 5.7: *xemacs* ist ein Editor, der von der grafischen Oberfläche Gebrauch machen kann.

5.3 Fernzugriff

Viele werden wahrscheinlich einen Windows-Laptop ihr Eigen nennen. Dieser eignet sich ebenfalls, um mit einer kleinen Konsole auf den Wohnzimmer-PC zuzugreifen. Für diesen Zugriff ist *putty* hervorragend geeignet. *putty* ist eine kleine, lauffähige Datei und erfordert keine weitere Installation. Es findet sich unter <https://www.putty.org> [30].

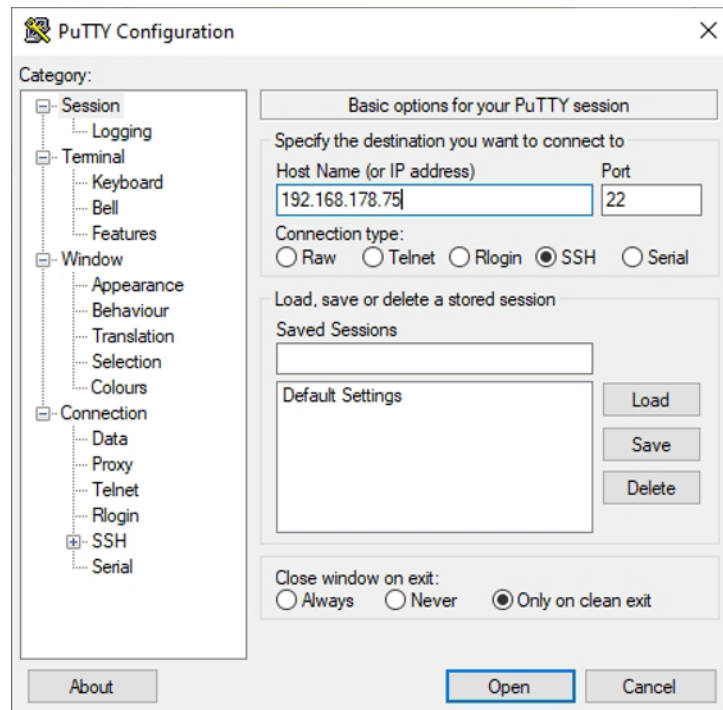


Abbildung 5.8: Mit *putty* kann man sich von einem Windows-System mit dem Wohnzimmer-PC verbinden.

Um sich per *putty* mit dem Wohnzimmer-PC zu verbinden, reicht es, im Feld *Host Name (or IP address)* den Namen oder die IP-Adresse des Wohnzimmer-PCs anzugeben und auf *Open* zu klicken. Aber *putty* kann noch mehr. So gestattet es *putty* beispielsweise, grafische Programme auf eine X11-Oberfläche weiterzuleiten (X-forwarding). Voraussetzung dafür ist natürlich, dass diese auf dem Windows-System läuft. Ein Beispiel für eine solche Oberfläche ist *Xming* <https://de.wikipedia.org/wiki/Xming> [31].

Ein Programm für Windows, welches die grafische Oberfläche gleich mitbringt, ist *MobaXterm*. Es kann unter <https://mobaxterm.mobatek.net/download.html> [32] in einer kostenlosen Version heruntergeladen werden. Diese unterliegt zwar einigen Einschränkungen, reicht aber im Normalfall völlig aus. Die *DISPLAY*-Variable wird nach dem Einwählen per Terminal automatisch gesetzt, so dass Sie grafische Programme einfach mit Namen aufrufen können. Die grafische Oberfläche unterstützt sogar OpenGL-beschleunigte Ausgaben.

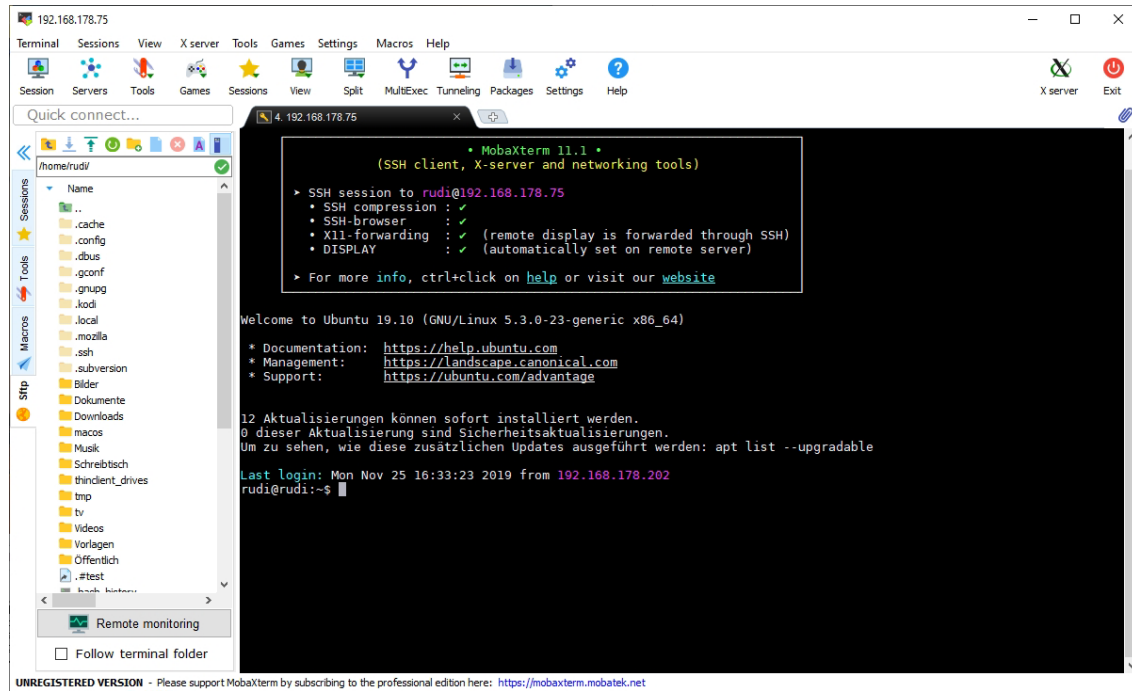


Abbildung 5.9: MobaXterm bringt eine praktische X11-Oberfläche mit.

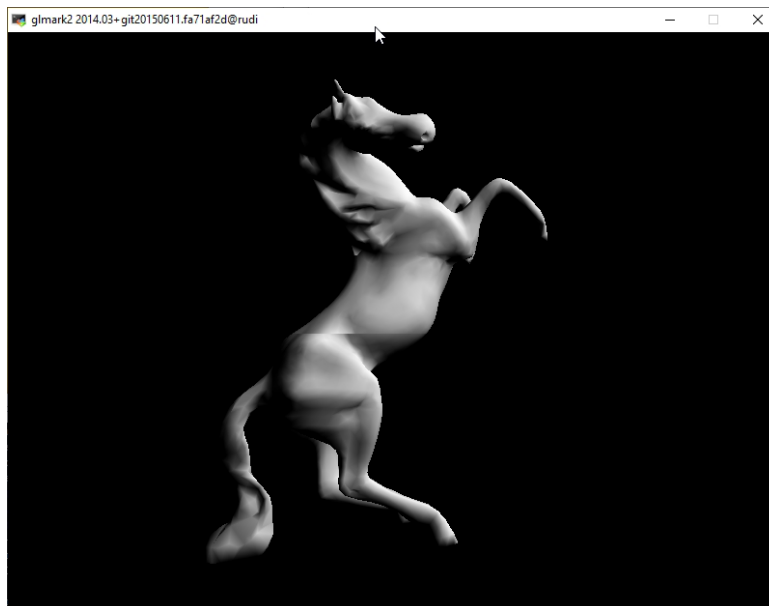


Abbildung 5.10: Die grafische Oberfläche von MobaXterm unterstützt sogar OpenGL-beschleunigte Ausgaben, hier am Beispiel von *glmark2*.

Terminals gibt es auch für viele Tablets. Sie finden Sie, indem Sie in den entsprechenden Stores nach dem Begriff „Terminal“ suchen. Diese Terminals eignen sich ebenfalls, um sich per IP-Adresse mit dem Wohnzimmer-PC zu verbinden.

Eine weitere Möglichkeit, den Wohnzimmer-PC remote zu steuern, bietet VNC (Virtual Network Computing). Dazu muss der VNC-Server auf dem Wohnzimmer-Ubuntu aktiviert werden. Standardmäßig ist der Server namens *vino* bereits installiert. Er kann entweder über die Oberfläche direkt aktiviert werden oder von einem Terminal aus. Entscheiden Sie sich für das Terminal, lautet die Befehlsfolge

Code-Ausschnitt 5.14: VNC-Server aktivieren

```
gsettings set org.gnome.Vino prompt-enabled false
gsettings set org.gnome.Vino authentication-methods "['vnc']"
gsettings set org.gnome.Vino require-encryption false
gsettings set org.gnome.Vino vnc-password $(echo -n 'kennwort'|base64)
gsettings set org.gnome.settings-daemon.plugins.sharing active true
eths=$(nmcli -t -f uuid,type c s --active | grep 802 | awk -F ":" '{ print "\'"$1"\'" }' ↵
| paste -s -d, -)
gsettings set org.gnome.settings-daemon.plugins.sharing.service:/org/gnome/settings-daemon/
plugins/sharing/vino-server/ enabled-connections "[ $eths ]"
```

Bitte ersetzen Sie dabei *kennwort* durch ein Kennwort Ihrer Wahl. Sie müssen dieses Kennwort dann zukünftig eingeben, wenn Sie sich per VNC mit Ihrem Wohnzimmer-PC verbinden wollen. Als VNC-Betrachter kommen wieder viele Programme in Frage. Eines der bekanntesten, das darüber hinaus für jede Plattform verfügbar ist, ist der *vncviewer*, der von der Webseite <https://www.realvnc.com/de/connect/download/viewer/> [33] heruntergeladen und installiert werden kann. Im *vncviewer* muss dann lediglich die IP-Adresse des Wohnzimmer-PCs angegeben werden.

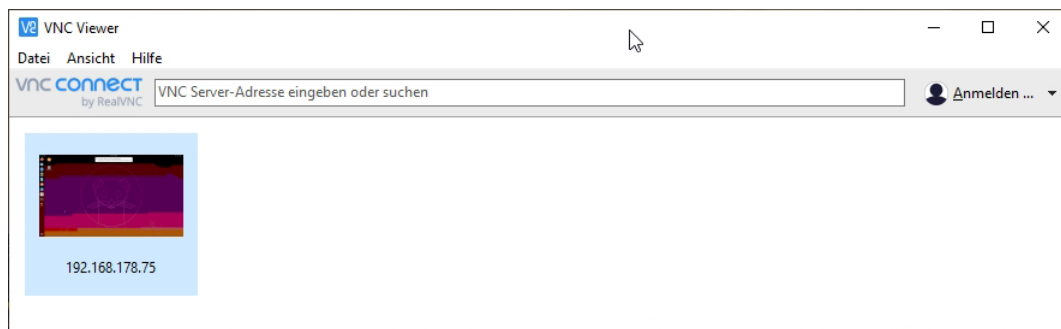


Abbildung 5.11: Der *vncviewer* speichert die einmal verbundenen Rechner.



Der *vncviewer* nutzt die Ports 5900 und 5800. Sollten Sie außerhalb Ihres eigenen Netzwerkes auf den Wohnzimmer-PC zugreifen wollen, müssen diese Ports freigegeben werden. Andernfalls kommt keine Verbindung zustande.

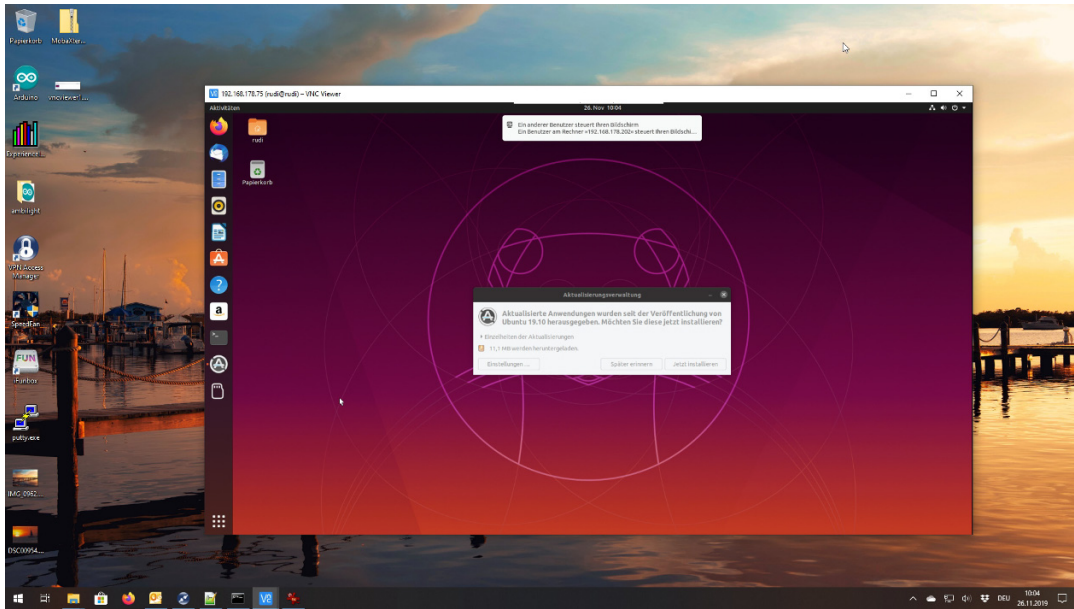


Abbildung 5.12: Der *vncviewer* macht den Ubuntu-Desktop unter anderen Betriebssystemen sichtbar.

Der letzte Teil dieses Abschnitts beschäftigt sich mit dem Thema Austausch von Dateien. Hierzu schauen wir uns die Installation von *dropbox* genauer an und werden *samba* einrichten.

dropbox ist ein sogenannter Filehosting-Dienst. Das Unternehmen Dropbox Inc. aus den USA stellt hier online-Speicherplatz zur Verfügung, auf den von verschiedenen Betriebssystemen aus lesend und schreibend zugegriffen werden kann. Bis zu 2 GB online-Speicher erhält ein registrierter Nutzer kostenlos. Das sollte ausreichen, um Dateien zwischenzulagern oder um ein Arbeitsverzeichnis zu haben, auf das mit allen Geräten zugegriffen werden kann. Unter Ubuntu wird *dropbox* so installiert:

Code-Ausschnitt 5.15: *dropbox* installieren

```
sudo apt install nautilus-dropbox
```

Geben Sie als nächstes den Befehl

Code-Ausschnitt 5.16: *dropbox* installieren – 2

```
dropbox start -i
```

in einem Terminal ein. Das sorgt dafür, dass zunächst noch fehlende Dateien heruntergeladen und installiert werden.

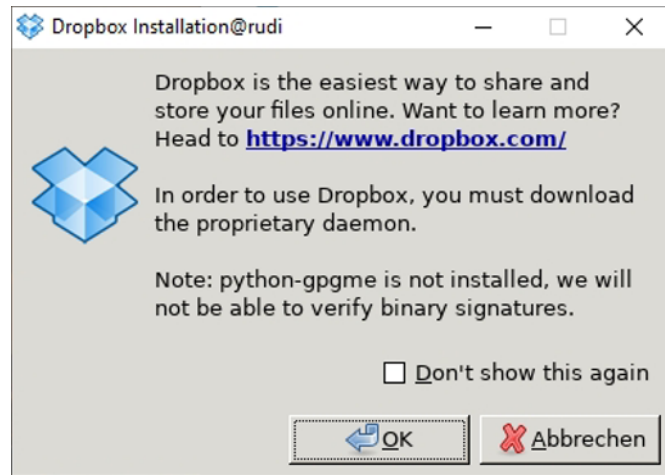


Abbildung 5.13: Die Installation von *dropbox* lädt weitere Pakete aus dem Internet nach.

Nach der Installation des *dropbox*-Paketes öffnet sich ein Browser-Fenster, in dem Sie Ihre Anmeldedaten eingeben können.

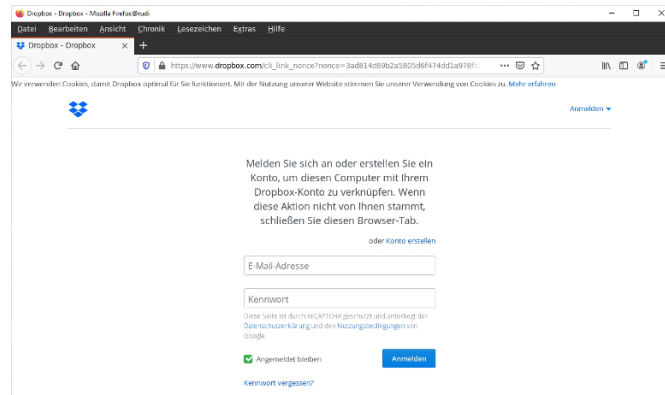


Abbildung 5.14: Die Eingabe der *dropbox*-Benutzerdaten erfolgt per Web-Interface.



In der freien Version von dropbox können maximal drei verschiedene Geräte angemeldet sein. Eventuell müssen daher erst einmal Geräte abgemeldet werden, bevor die Anmeldung auf dem Wohnzimmer-PC erfolgen kann.

Ab sofort erscheint das *dropbox*-Logo in der oberen Ubuntu-Leiste.

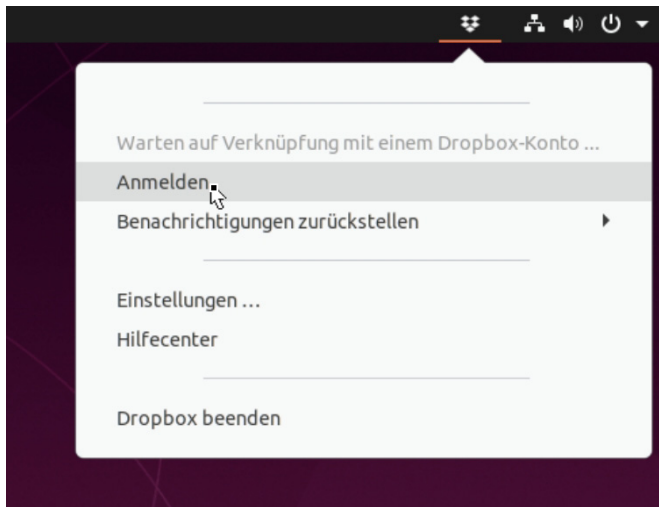


Abbildung 5.15: Nach der Installation erscheint das *dropbox*-Logo in der oberen Ubuntu-Leiste.

Alle *dropbox*-Dateien werden ebenfalls im LINUX-Dateimanager *nautlius* angezeigt.

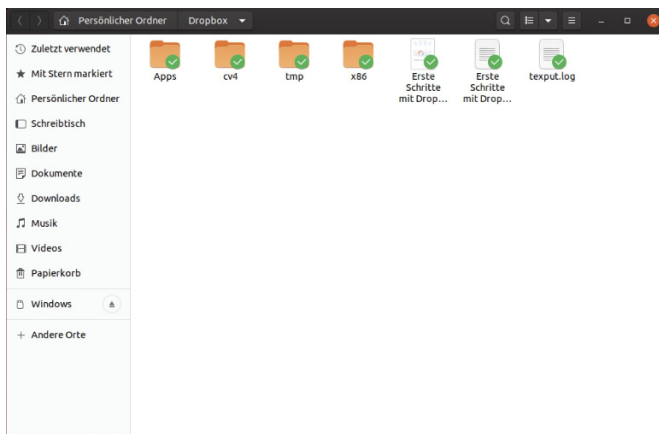


Abbildung 5.16: Der LINUX-Dateimanager *nautlius* zeigt alle *dropbox*-Dateien an.

Während *dropbox* einen externen Dienstleister benutzt, um Dateien auf ein und von einem Online-Netzwerk zu kopieren, kann *samba* dazu benutzt werden, Dateien direkt auf den oder vom Wohnzimmer-PC zu kopieren. Bevor *samba* benutzt werden kann, muss es so installiert werden:

Code-Ausschnitt 5.17: *samba* installieren

```
sudo apt-get install samba
```

Die Installation von *samba* installiert weitere Pakete, die für den Betrieb erforderlich sind. *samba* setzt ähnlich wie das LINUX-System selbst auf einer Benutzerverwaltung auf. Die Benutzerverwaltung von *samba* ist allerdings unabhängig von der des Ubuntu-Systems. Die *samba*-Kennwörter werden in der Datei `/etc/samba/smbpasswd` gespeichert. Für jeden Benutzer, der *samba* und damit beispielsweise eine Dateifreigabe nutzen soll, muss ein separates Kennwort angelegt werden. In meinem Beispiel heißt der *samba*-Account *rudi*. Das Kennwort für diesen Account wird wie folgt erstellt:

Code-Ausschnitt 5.18: *samba*-Kennwort erstellen

```
sudo smbpasswd -a rudi
```



Für den *samba*-Zugang sollte ein sicheres Kennwort gewählt werden, erst recht, wenn der Account außerhalb des eigenen Netzes freigeschaltet wird.

Möchten Sie lediglich einen Order für den Datenaustausch vorsehen, dann erstellen Sie diesen Ordner bitte, z. B. so:

Code-Ausschnitt 5.19: Ein Verzeichnis erstellen

```
cd ~
mkdir ordnername
```



Bitte erstellen Sie den freizugebenden Ordner nicht mit einem voran gestellten `sudo`. Das würde den Ordner mit Root-Rechten anlegen und der *samba*-Benutzer könnte später nicht darauf zugreifen, weil ihm die entsprechenden Rechte fehlen.

Im weiteren Verlauf beschreibe ich, wie das gesamte *home*-Verzeichnis eines Benutzers (in meinem Fall */home/rudi*) freigegeben werden kann. Zur Freigabe selbst muss die Datei */etc/samba/smb.conf* editiert werden. Es ist sicherlich eine gute Idee, ein Backup dieser Datei vor dem Bearbeiten anzulegen:

Code-Ausschnitt 5.20: *samba*-Backup erstellen

```
sudo cp /etc/samba/smb.conf ~
```

Editieren Sie die Datei */etc/samba/smb.conf* nun (z. B. `sudo vi /etc/samba/smb.conf`) und fügen Sie die folgenden Zeilen am Ende der Datei ein:

Code-Ausschnitt 5.21: *samba* konfigurieren

```
[rudi]
path = /home/rudi
available = yes
valid users = rudi
read only = no
browseable = yes
public = yes
writable = yes
```

Die Bedeutung der einzelnen Einträge ist wie folgt:

- **[rudi]**: Das ist der Name der Freigabe. Er kann willkürlich gewählt werden.
- **path**: Gibt den Pfad zur Freigabe an.
- **available**: Gibt an, ob die Freigabe für Klienten im Netzwerk zur Verfügung gestellt wird.
- **valid users**: Gibt eine mit Kommata getrennte Liste von *samba*-Benutzern an, welche die Freigabe benutzen dürfen.
- **read only**: Gibt an, ob die Dateien nur gelesen oder auch geschrieben werden dürfen.

- **browseable:** Gibt an, ob alle Dateien im Browser des *samba*-Benutzers angezeigt werden.
- **public:** Gibt an, ob die Freigabe öffentlich zur Verfügung steht.
- **writeable:** Gibt an, ob neue Daten in die Freigabe geschrieben werden dürfen.

Nach den Änderungen an der Datei */etc/samba/smb.conf* muss der *samba*-Dienst neu gestartet werden, damit die editierten Änderungen berücksichtigt werden.

Code-Ausschnitt 5.22: *samba*-Dienst neu starten

```
sudo /etc/init.d/smbd restart
```

samba bietet ebenfalls ein Werkzeug zum Testen der korrekten Syntax der Datei */etc/samba/smb.conf*:

Code-Ausschnitt 5.23: *samba*: Überprüfen der korrekten Syntax

```
sudo testparm
```

Nach diesen Schritten kann als Beispiel der Windows-Dateimanager durch Eingabe von `\\IP-Adresse` dazu verwendet werden, Dateien mit dem Wohnzimmer-PC auszutauschen. Bitte ersetzen Sie in dem oben genannten Aufruf *IP-Adresse* mit der IP-Adresse Ihres Wohnzimmer-PCs.

Zum Abschluss dieses Abschnitts möchte ich Ihnen noch einen kleinen Tipp mit auf den Weg geben. Viele PCs unterstützen das WOL (Wake On LAN) Protokoll, das es erlaubt, den PC aus dem Standby zu booten. Das spart Energie, da man den Rechner über ein WOL-Kommando einschalten kann, wenn man ihn benötigt und er ansonsten ausgeschaltet bleiben kann. Trotzdem kann es vorkommen, dass man den Rechner von unterwegs aus einschalten möchte, um beispielsweise die Aufnahme einer Fernsehsendung zu programmieren. Ist WOL im BIOS des Wohnzimmer-PCs aktiviert, kann der Computer beispielsweise von unterwegs mit Hilfe des DSL-Routers (z. B. einer FritzBox!) gestartet werden. Es gibt ebenfalls Applikationen für Mobilfunkgeräte, die das Einschalten eines PCs durch Versenden des sogenannten Magic Packets realisieren. Unter LINUX kann man *ethtool* verwenden um zu prüfen, ob der Ethernet-Adapter WOL unterstützt.

Code-Ausschnitt 5.24: Installation von *ethtool* und Abfragen der Ethernetkarte

```
sudo apt install ethtool  
sudo ethtool enp2s0
```

Bei meinem Lattepanda Alpha lautet die Ausgabe:

Code-Ausschnitt 5.25: Ausgabe von *ethtool*

```

Settings for enp2s0:
  Supported ports: [ TP MII ]
  Supported link modes: 10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Full
  Supported pause frame use: Symmetric Receive-only
  Supports auto-negotiation: Yes
  Supported FEC modes: Not reported
  Advertised link modes: 10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Full
  Advertised pause frame use: Symmetric Receive-only
  Advertised auto-negotiation: Yes
  Advertised FEC modes: Not reported
  Link partner advertised link modes: 10baseT/Half 10baseT/Full
                                    100baseT/Half 100baseT/Full
                                    1000baseT/Half 1000baseT/Full
  Link partner advertised pause frame use: No
  Link partner advertised auto-negotiation: Yes
  Link partner advertised FEC modes: Not reported
  Speed: 1000Mb/s
  Duplex: Full
  Port: MII
  PHYAD: 0
  Transceiver: internal
  Auto-negotiation: on
  Supports Wake-on: pumbg
  Wake-on: pumbg
  Current message level: 0x00000033 (51)
                        drv probe ifdown ifup
  Link detected: yes

```

Unter *Wake-on* finden sich alle unterstützten Optionen, mit deren Hilfe die Karte und damit der Rechner gebootet werden kann. Dabei bedeuten die einzelnen Buchstaben:

- **p**: Wake on PHY activity
- **u**: Wake on unicast messages
- **m**: Wake on multicast messages
- **b**: Wake on broadcast messages
- **a**: Wake on ARP
- **g**: Wake on Magic Packet™
- **s**: Enable SecureOn™ password for Magic Packet™
- **d**: Disable (wake on nothing). This option clears all previous options.

WOL funktioniert also, wenn der Buchstabe **g** unter den *Wake-on*-Optionen auftaucht. Um WOL zu aktivieren, muss der Befehl

Code-Ausschnitt 5.26: WOL setzen

```
sudo ethtool -s enp2s0 wol g
```

nach jedem Booten aufgerufen werden. Danach steht die Option *Wake-on* nur auf **g** und der Rechner kann nach dem Ausschalten mit

Code-Ausschnitt 5.27: Rechner herunterfahren

```
sudo halt -p
```

mit dem sogenannten Magic Packet eingeschaltet werden. Unter Ubuntu kann das Programm *wakeonlan* verwendet werden, um im selben Netzwerk einen Rechner mittels WOL zu booten. Installieren Sie es so:

Code-Ausschnitt 5.28: *wakeonlan* installieren

```
sudo apt install wakeonlan
```

Das Programm selbst wird dann mit der MAC-Adresse des zu startenden Rechners aufgerufen. Diese erhalten Sie nach dem Aufruf von

Code-Ausschnitt 5.29: Ethernet MAC-Adresse ausgeben

```
ip link show enp2s0 | grep ether
```

Bei mir lautet die Ausgabe

Code-Ausschnitt 5.30: Ausgabe der Ethernet MAC-Adresse

```
link/ether 00:e0:4c:00:90:8a brd ff:ff:ff:ff:ff:ff
```

Damit lautet mein WOL-Aufruf wie folgt:

Code-Ausschnitt 5.31: WOL Beispiel unter LINUX

```
wakeonlan 00:e0:4c:00:90:8a
```

Um die Option *Wake-on g* dauerhaft zu aktivieren, können Sie den Aufruf aus Code-Ausschnitt 5.26 in der Datei */etc/rc.local* speichern.

5.4 OpenVPN

Dieser Abschnitt erklärt das Aufsetzen eines OpenVPN-Servers auf dem Wohnzimmer-PC. Falls Ihr DSL-Router zu Hause keinen VPN-Zugang bietet, könnte der hier erklärte VPN-Server auf dem Wohnzimmer-PC eine gute Möglichkeit sein, von außerhalb mit Ihrem privaten Netzwerk zu verbinden.

Die Konfiguration eines OpenVPN-Servers kann schon einmal einige Zeit in Anspruch nehmen. Es müssen verschiedene Zertifikate und Dateien erstellt werden, Dateien müssen umkopiert oder zusammengefasst werden und es müssen Konfigurationen angelegt werden. Glücklicherweise findet sich auf Github ein Skript, welches alle diese Punkte erledigt, ohne dass sich der Benutzer viele Gedanken über Details machen muss. Das Skript selbst klonen wir mit dem *git* Befehl. Dazu stellen wir zunächst einmal sicher, dass *git* auch installiert ist.

Code-Ausschnitt 5.32: Installation von *git*

```
sudo apt-get install git
```

Danach wechseln wir in das aktuelle Home-Verzeichnis und klonen das Installations-Skript wie folgt:

Code-Ausschnitt 5.33: Klonen des *openvpn*-Installations-Skriptes

```
cd ~  
git clone https://github.com/Nyr/openvpn-install.git
```

Das Klonen erstellt ein Verzeichnis mit dem Namen *openvpn-install*, in das wir nun wechseln. Dort machen wir die Skript-Datei *openvpn-install.sh* ausführbar und starten die Installation als Benutzer *root*.

Code-Ausschnitt 5.34: Installation von *openvpn* mit Hilfe des Skriptes vorbereiten

```
cd openvpn-install
chmod +x openvpn-install.sh
sudo ./openvpn-install.sh
```

im weiteren Verlauf der Installation werden Ihnen einige Fragen gestellt. Die Optionen zur Beantwortung sind:

- **IP address:**

Geben Sie hier die IP-Adresse an, unter der Ihr Wohnzimmer-PC von außen zu erreichen ist. In der Regel wird dieser PC nicht sofort von außen zu erreichen sein, sondern wird seine IP-Adresse wieder aus einem privaten IP-Sub-Bereich von einem DSL-Router (z. B. einer FritzBox!) zugeteilt bekommen. Sie können sich hier so behelfen, dass Sie einen DynDNS-Dienst damit beauftragen, Ihre ständig wechselnde zu-Hause-IP mit einem festen Namen zu verknüpfen. Die meisten DSL-Router bieten eine solche Option an. Ich nehme als Beispiel den DynDNS-Namen *mein.rechner.de*. Bitte ersetzen Sie diesen Namen durch Ihren DynDNS-Namen. In Ihrem Router müssen Sie dann ebenfalls eine Portweiterleitung einrichten, die den UDP-Port 1194 an die IP-Adresse Ihres Wohnzimmer-PCs weiterleitet. Falls Ihr Wohnzimmer-PC nicht von außen zugänglich ist, gibt Ihnen das Skript folgende Meldung aus: *This server is behind NAT. What is the public IPv4 address or hostname?* Weiterhin gibt das Skript Ihnen die IP-Adresse Ihres DSL-Routers aus.

- **1) UDP (recommended)**

- 2) **TCP, Protocol [1-2]:**

Hier können Sie das UDP- und das TCP-Protokoll auswählen. UDP ist das Standard-Protokoll für VPN-Verbindungen. Wählen Sie daher hier bitte den Wert 1 aus.

- **What port do you want OpenVPN listening to?:**

Hier tragen Sie die Port-Adresse ein, auf der der VPN-Server nach neuen Verbindungen lauscht. Der Default-Port ist der Port 1194. Bitte geben Sie diesen als Port ein.

- **Which DNS do you want to use with the VPN?**

- 1) **Current system resolvers**

- 2) **1.1.1.1**

- 3) **Google**

- 4) **OpenDNS**

- 5) **Verisign**

- DNS [1-5]:**

Hier können Sie den Nameserver eintragen, der die Namen in IP-Adressen auflöst. Funktioniert Ihre Internetverbindung einwandfrei (wovon ich ausgehe), tragen Sie bitte hier 1 ein, damit der bisher verwendete Nameserver auch für die VPN-Verbindung verwendet wird.

- **Finally, tell me your name for the client certificate.**

Please, use one word only, no special characters.

Client name:

Geben Sie hier bitte einen Namen für das Zertifikat des Klienten ein. Bitte achten Sie darauf, dass der Name nur aus einem Wort bestehen und keine Sonderzeichen beinhalten darf. Geben Sie hier beispielsweise *klient* ein.

Das Skript fordert Sie im weiteren Verlauf der Installation auf, die *Enter*-Taste zu drücken. Es installiert dann selbständig weitere Pakete, die für den Betrieb von *OpenVPN* erforderlich sind. Am Ende der Installation sollten Sie folgende Nachricht erhalten:

Code-Ausschnitt 5.35: Ausgabe des Installations-Skripts nach erfolgreicher Installation

```
Using configuration from ./safessl-easyrsa.cnf
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
commonName      :ASN.1 12:'klient'
Certificate is to be certified until Nov 25 14:40:07 2029 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated

Using SSL: openssl OpenSSL 1.1.1c 28 May 2019
Using configuration from ./safessl-easyrsa.cnf

An updated CRL has been created.
CRL file: /etc/openvpn/server/easy-rsa/pki/crl.pem

Created symlink /etc/systemd/system/multi-user.target.wants/openvpn-iptables.service -> /etc/systemd/system/openvpn-iptables.service.
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn-server@server.service -> /lib/systemd/system/openvpn-server@.service.

Finished!

Your client configuration is available at: /root/klient.ovpn
If you want to add more clients, just run this script again!
```

Die Konfigurations-Datei für den *OpenVPN*-Server liegt im Verzeichnis */etc/openvpn/server* und heißt *server.conf*. Falls Sie noch Änderungen an der Konfiguration des Servers vornehmen möchten, können Sie das in dieser Datei erledigen. Bei mir sieht sie so aus:

Code-Ausschnitt 5.36: Konfigurations-Datei des *openvpn*-servers

```

local 192.168.178.75
port 1194
proto udp
dev tun
ca ca.crt
cert server.crt
key server.key
dh dh.pem
auth SHA512
tls-crypt tc.key
topology subnet
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 192.168.178.1"
keepalive 10 120
cipher AES-256-CBC
user nobody
group nogroup
persist-key
persist-tun
status openvpn-status.log
verb 3
crl-verify crl.pem
explicit-exit-notify

```

Weiterhin hat das Installations-Skript ein virtuelles Netzwerk-Interface angelegt, das sich *tun0* nennt. Sie können den Befehl

Code-Ausschnitt 5.37: Virtuelles OpenVPN-Interface anzeigen

```
ip ad | grep tun0
```

verwenden, um sich ein erfolgreich angelegtes Netzwerk-Interface anzeigen zu lassen. Bei mir sieht die Ausgabe so aus:

Code-Ausschnitt 5.38: Ausgabe beim Anzeigen des virtuellen OpenVPN-Interfaces

```

4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group ↔
   default qlen 100
   inet 10.8.0.1/24 brd 10.8.0.255 scope global tun0

```

Im letzten Schritt können Profile für weitere Benutzer angelegt werden. Das Anlegen eines Profils mündet in der Erstellung einer Datei mit der Endung *.ovpn*, die dann von OpenVPN-Klienten verwendet werden kann. Zu diesem Zweck kann das Installations-Skript ein weiteres Mal aufgerufen werden.

Code-Ausschnitt 5.39: Erneutes Aufrufen des OpenVPN-Installations-Skriptes

```
sudo ./openvpn-install.sh
```

Das Skript teilt uns nun mit, dass OpenVPN bereits installiert ist und bietet die Möglichkeit, einen neuen Benutzer hinzuzufügen (1), einen existierenden Benutzer zu löschen (2), OpenVPN zu de-installieren (3) oder das Skript zu beenden (4).

Code-Ausschnitt 5.40: Ausgabe bei erneutem Aufrufen des OpenVPN-Installations-Skriptes

```
Looks like OpenVPN is already installed.
```

```
What do you want to do?
```

- 1) Add a new user
- 2) Revoke an existing user
- 3) Remove OpenVPN
- 4) Exit

```
Select an option [1-4]: 1
```

Wir entscheiden uns, einen neuen Benutzer anzulegen, den wir *nutzer* nennen. Bitte ersetzen Sie diesen Namen durch einen sinnvollen Namen Ihrer Wahl.

Code-Ausschnitt 5.41: Anlegen eines weiteren OpenVPN-Benutzers

```
Tell me a name for the client certificate.
```

```
Client name: nutzer
```

```
Using SSL: openssl OpenSSL 1.1.1c 28 May 2019
```

```
Generating a RSA private key
```

```
.....+++++
```

```
.....+++++
writing new private key to '/etc/openvpn/server/easy-rsa/pki/private/nutzer.key.0Xka2RoDkA'
```

```
-----
Using configuration from ./safessl-easyrsa.cnf
```

```
Check that the request matches the signature
```

```
Signature ok
```

```
The Subjects Distinguished Name is as follows
```

```
commonName      :ASN.1 12:'nutzer'
```

```
Certificate is to be certified until Nov 25 15:21:48 2029 GMT (3650 days)
```

```
Write out database with 1 new entries
```

```
Data Base Updated
```

```
Client nutzer added, configuration is available at: /root/nutzer.ovpn
```

Die fertige OpenVPN-Datei liegt wieder im Verzeichnis */root* und heißt *nutzer.ovpn*.

Als normaler Benutzer dürfen Sie nicht auf das Home-Verzeichnis */root* des Benutzers Root zugreifen. Der Befehl `sudo su` verleiht Ihnen Superuser-Rechte, so dass Sie danach die Datei in ein Verzeichnis umkopieren können, auf das Sie auch zugreifen können. Alternativ können Sie die Datei auch mit dem Befehl `sudo cp /root/nutzer.ovpn ~` in Ihr Home-Verzeichnis kopieren.



Die *.ovpn*-Datei nutzen wir nun im Klienten, der mit dem OpenVPN-Server verbinden soll. Das kann ein Tablet sein oder ein anderer PC. Im Folgenden werde ich die Verbindung von einem Windows-PC aus beschreiben. Auf diesem laden wir zunächst den OpenVPN-Klienten herunter. Den findet man unter <https://openvpn.net/client-connect-vpn-for-windows/> [34]. Laden Sie ihn herunter und installieren Sie ihn. Übertragen Sie die Datei *nutzer.ovpn* ebenfalls auf den Windows-PC. Dazu können Sie z. B. *samba* nutzen oder sich die Datei einfach per E-Mail senden.

Nach der Installation des OpenVPN-Klienten unter Windows können Sie mit der rechten Maustaste auf die *.ovpn*-Datei klicken. Wählen Sie dann *Öffnen mit...* und anschließend *OpenVPN connect*. bestätigen Sie das Laden des Profils. Danach können Sie die VPN-Verbindung zu Ihrem Wohnzimmer-PC aufbauen.

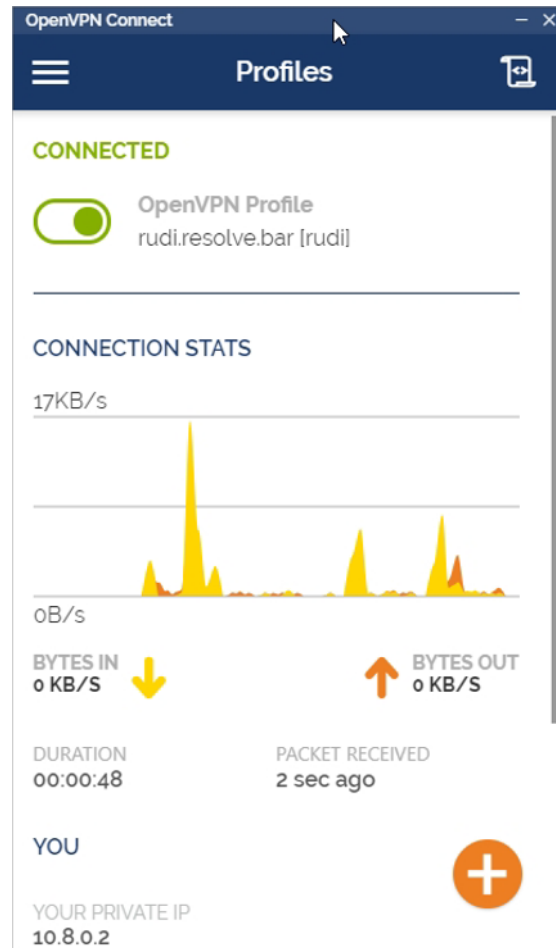


Abbildung 5.17: *OpenVPN Connect* unter Windows zeigt neben dem Verbindungsstatus auch die übertragene Datenmenge grafisch an.

Ähnlich einfach funktioniert das Importieren der *.ovpn*-Datei auf einem Tablet oder mit einem LINUX-Rechner.

Am Ende dieses Abschnitts möchte ich Ihnen noch einen Tipp zum Transfer von Dateien geben. Unter LINUX existiert das Kommando *scp* (Secure Copy), mit dem Sie Dateien von einem Rechner zu einem anderen Rechner kopieren können. Ein Beispielaufruf sieht so aus:

Code-Ausschnitt 5.42: *scp* kann Dateien von einem Rechner zu einem anderen Rechner kopieren

```
scp dateiname nutzer@rechner.ip:/verzeichnis
```

Ersetzen Sie dabei *dateiname* mit dem Dateinamen, den Sie kopieren möchten, *nutzer* mit dem Benutzernamen auf dem Rechner, auf den Sie kopieren möchten, *rechner.ip* mit der IP-Adresse des Rechners, auf den Sie kopieren möchten und *verzeichnis* mit dem Verzeichnisnamen auf dem Zielrechner, in den Sie kopieren möchten. Die Oberfläche von *MobaXterm* erlaubt Ihnen ebenfalls ein secure copy per drag & drop. Für Windows existiert ein weiteres Programm, das Sie zum Kopieren von Daten nutzen können. Es heißt *Winscp* und kann unter <https://winscp.net/eng/index.php> [35] heruntergeladen werden.

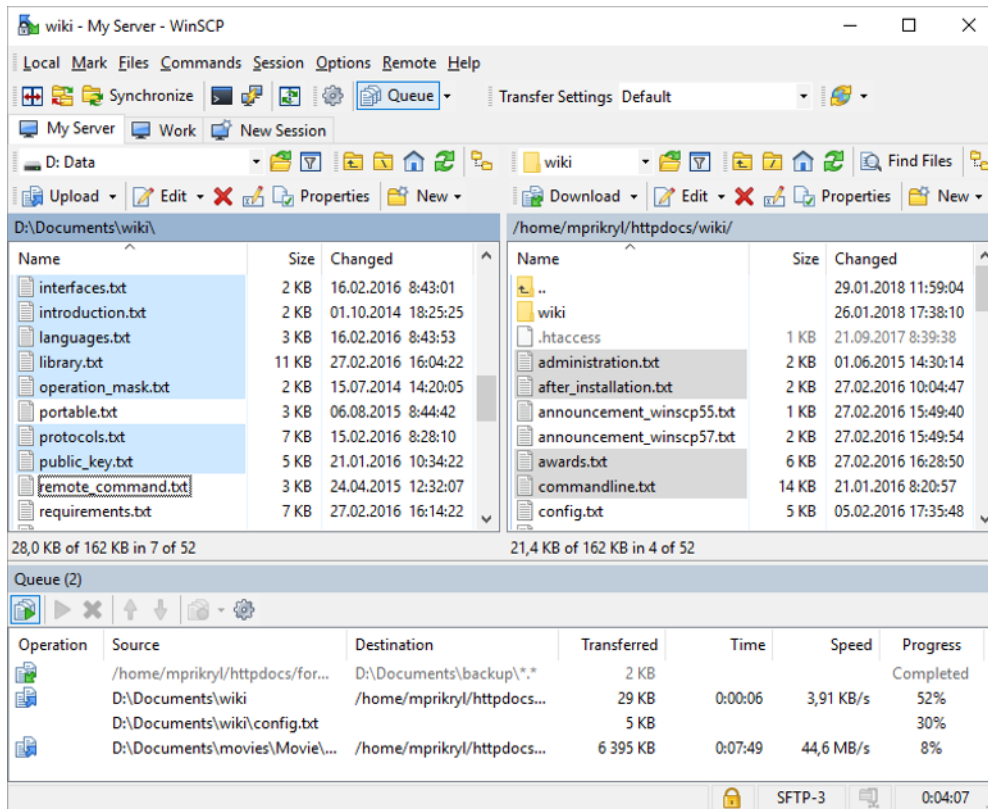


Abbildung 5.18: Mit *Winscp* können Dateien von Ubuntu nach Windows kopiert werden (Quelle: winscp.net [35]).

5.5 Programme übersetzen oder interpretieren

Dieser Abschnitt gibt einen kleinen Einblick in das Gebiet der C/C++- und Python-Programme. Er hat nicht den Anspruch, ein umfassendes Nachschlagewerk darzustellen, sondern gibt lediglich einen minimalen Überblick über Compiler, Makefiles und Interpreter, da viele der nachfolgenden Beispiele und Programme auf diesen Grundlagen beruhen.

Viele LINUX-Programme und besonders zeitkritische Programme basieren auf C/C++-Code. Der Grund liegt darin, dass bei manchen Anwendungen Interpreter wie Python deutlich langsamer sind als übersetzte C/C++-Programme. In diesem Abschnitt schauen wir uns die LINUX C/C++-Compiler näher an, den *gcc* (Gnu Compiler Collection). Der *gcc* enthält sowohl ein Frontend für den C- und den C++-Compiler als auch die entsprechenden Bibliotheken für diese Programmiersprachen. Die Entwicklungsumgebungen können mit dem Befehl

Code-Ausschnitt 5.43: Die Installation von *gcc*

```
sudo apt-get install gcc g++ make
```

installiert werden.



Vielleicht haben Sie sich gefragt, wie Sie Programme, die Sie per Terminal installiert haben, wieder loswerden können. Der Befehl hierzu lautet

Code-Ausschnitt 5.44: Die De-Installation von Programmen

```
sudo apt-get purge programmname
```

Bitte ersetzen Sie programmname dabei durch den Namen des Programm-Paketes, das Sie de-installieren möchten.

Neben den eigentlichen Compilern installiert der Befehl 5.43 auch das Paket *make*. *make* ist ein Hilfsmittel, welches beim Übersetzen, Linken, Installieren und De-installieren von Quelltext-Paketen behilflich ist. Wir beginnen mit einem einfachen „Hallo Welt!“-Beispiel. Wechseln Sie bitte in ein Verzeichnis Ihrer Wahl und erstellen Sie eine Datei mit dem Namen *hello.c* und füllen Sie diese Datei mit dem folgenden Inhalt:

Code-Ausschnitt 5.45: Hallo-Welt Beispiel in C

```
#include <stdio.h>

int main(void)
{
    puts("Hallo Welt!");
    return 0;
}
```

Das Beispiel inkludiert zunächst die sogenannte Header-Datei *stdio.h*, die Funktionen für Ein- und Ausgaben zur Verfügung stellt. Das Hauptprogramm (*main*) hat keine Übergabeargumente (*void*) und gibt einen ganzzahligen Wert zurück (*int*). Der Standard-Rückgabewert ist 0 (falls kein Fehler auftritt). Das Programm kann mit dem Befehl

Code-Ausschnitt 5.46: Übersetzen eines C-Programms

```
gcc -o hello hello.c
```

zu einer ausführbaren Datei mit dem *hello* übersetzt werden.



*Bei Windows-Systemen sind ausführbare Dateien mit der Endung **.exe** (**executable**) gekennzeichnet. Dies ist unter LINUX (also z. B. Ubuntu) nicht erforderlich. Ausführbare Dateien besitzen das Attribut **x** (**executable**).*

Die übersetzte und gelinkte Datei kann nun wie folgt aufgerufen werden:

Code-Ausschnitt 5.47: Aufruf eines ausführbaren Programms unter LINUX

```
./hello
```

Es liefert die erwartete Ausgabe *Hallo Welt!*.

Falls viele verschiedene Programme übersetzt und zu einem großen Programm gelinkt (also zusammengebunden) werden sollen, würde die Befehlszeile hierzu schnell sehr groß und damit unübersichtlich werden. Um dieses Vorgehen zu vereinfachen, wurde das *Makefile* erfunden. Bitte editieren Sie in dem Verzeichnis, in dem Sie die Datei *hello.c* angelegt haben, eine Datei mit dem Namen *Makefile* und füllen Sie diese Datei mit dem folgenden Inhalt:

Code-Ausschnitt 5.48: Erstellen eines Makefiles

```

VERSION = 1.0
CPP = /usr/bin/g++
CFLAGS = -Wall -DVERSION=\"$(VERSION)\"
LDFLAGS =

OBJ = hello.o

BIN = hello

all: $(OBJ)
    $(CPP) $(CFLAGS) -o $(BIN) $(OBJ) $(LDFLAGS)

%.o: %.c
    $(CC) $(CFLAGS) -c $<

clean:
    rm -rf $(BIN) $(OBJ) *~

```

Zunächst rufen wir das *Makefile* auf und löschen die alten Dateien, die beim Übersetzen entstanden sind:

Code-Ausschnitt 5.49: Aufrufen eines Makefiles

```
make clean
```

Der Befehl *make* ruft die Datei *Makefile* auf und springt dann in die Sektion *clean*. Alles, was in dieser Sektion steht, wird dann Schritt für Schritt aufgerufen. Die ausführbare Datei *hello* wird gelöscht (*rm*), ebenso wie die Objekt-Datei *hello.o* und mögliche Editor Backup-Dateien, die mit einer Tilde (~) beginnen. Der Aufruf von

Code-Ausschnitt 5.50: Aufrufen eines Makefiles – 2

```
make
```

erstellt wieder die ausführbare Datei *hello*. Das oben angegebene *Makefile*-Beispiel zeigt auch, wie Parameter für die Quelltext-Dateien übergeben werden können (*-DVERSION*).

Wenn sich ein Quelltext ändert, wird das automatisch vom make-System erkannt und die ausführbaren Dateien werden nach einem Aufruf von make neu erstellt. Damit das funktioniert, muss der Rechner aber das aktuelle Datum haben. Falsch gesetzte Daten oder Uhrzeiten führen im make-System zu Verwirrungen und gegebenenfalls zu Fehlermeldungen. Moderne Wohnzimmer-PCs sollten aber eine sogenannte RTC (Real Time Clock) besitzen, so dass permanent die richtige Uhrzeit und das richtige Datum eingestellt sind.



Im Gegensatz zu C/C++ ist Python ein Interpreter. Auf der Python-Webseite <http://www.python.org> [36] steht: „Python is a programming language that lets you work quickly and integrate systems more effectively“. Und das stimmt. Python ist wirklich eine Sprache, mit der man schnell und effizient programmieren kann. Durch unzählige Bibliotheken können selbst anspruchsvolle Aufgaben mit wenigen

Codezeilen erfüllt werden. Wir schauen uns in diesem Abschnitt nur einige grundlegende Dinge an.

Zunächst installieren wir Python und die Python-Entwicklungsumgebung:

Code-Ausschnitt 5.51: Installation von *Python*

```
sudo apt-get install python python-dev
```

Da Python ein Interpreter ist, also einen Code Schritt für Schritt ausführt, gibt es zwei Möglichkeiten, ein „Hallo Welt!“-Beispielprogramm zum Leben zu erwecken. Die erste Option ist, *Python* in einem Terminal aufzurufen durch einfach Eingabe von

Code-Ausschnitt 5.52: Den Python-Interpreter aufrufen

```
python

Python 2.7.17rc1 (default, Oct 10 2019, 10:26:01)
[GCC 9.2.1 20191008] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Python begrüßt Sie daraufhin mit Angabe der Version (2.7) und gibt einige Hinweise. Danach wartet es auf Eingaben. Wir geben das „Hallo Welt!“-Beispiel ein und schauen uns die Ausgabe an:

Code-Ausschnitt 5.53: „Hallo-Welt!“-Beispiel in Python

```
>>> print "Hallo Welt!"
Hallo Welt!
>>>
```

Der Interpreter kann durch die Eingabe von *CTRL-d* verlassen werden.

Bei der zweiten Option werden die Python-Befehle in einer Datei gespeichert. Zur Demonstration nehme ich die Datei *hallo.py*, die ich mit folgendem Inhalt fülle:

Code-Ausschnitt 5.54: Ein simples Python-Programm

```
print "Hallo Welt!"
```

Der Aufruf erfolgt im Terminal durch Eingabe von

Code-Ausschnitt 5.55: *Python*-Programm aufrufen

```
python ./hallo.py
```

Die Ausgabe ist identisch zur ersten Ausgabe.



Die Python Syntax kennt keine Anfangs- oder End-Markierungen wie ein Semikolon oder geschweifte Klammern. Die Absetzung erfolgt durch eine sogenannte Indentierung, also eine Einrückung. Als Indents können entweder Leerzeichen oder Tabulatoren verwendet werden. Das folgende Beispiel zeigt die Ausgabe der Zahlen 0 bis 9 in einer Schleife.

Code-Ausschnitt 5.56: Ausgabe der Zahlen 0 bis 9 in einer Schleife

```
for i in xrange(0,10):
    print(i)
    print ("Hallo")
```

Mit der oben gezeigten Einrückung wird nach jeder Zahl das Wort Hallo ausgegeben. Würde man die Einrückung der letzten Zeile weglassen, würde Hallo nur einmal nach allen Zahlen ausgegeben werden.

5.6 Ein neuer Kernel

Der LINUX-Kernel ist das Herzstück des Betriebssystems. Er stellt die Verbindung zwischen einem Programm und den Schnittstellen eines PCs zur Verfügung. Der LINUX-Kernel ist fast ausschließlich in der Programmiersprache C geschrieben. Lediglich einige zeitkritische Treiber sind in Assembler integriert. Dieser Abschnitt erklärt, wie der Ubuntu-Kernel aus dem Quelltext heraus übersetzt, installiert und verwendet werden kann. Vielleicht fragen Sie sich, warum das überhaupt erforderlich ist? Es ist z. B. dann erforderlich, wenn ein Treiber nicht im aktuell benutzten Kernel vorhanden ist oder beispielsweise fehlerhaft ist, so dass Sie ihn korrigieren (patchen) können. Die hier vorgestellte Vorgehensweise geht davon aus, dass alle benötigten Source-Code-Quellen in der Datei `/etc/apt/sources.list` eingetragen sind. Bei einer Ubuntu 19.10-Installation sind dies in meinem Fall:

Code-Ausschnitt 5.57: Benötigte Source-Code-Einträge in der Datei `sources.list`

```
...
deb-src http://de.archive.ubuntu.com/ubuntu/ eoan main restricted
...
deb-src http://de.archive.ubuntu.com/ubuntu/ eoan-updates main restricted
...
deb-src http://de.archive.ubuntu.com/ubuntu/ eoan-updates universe
---
deb-src http://de.archive.ubuntu.com/ubuntu/ eoan multiverse
deb-src http://de.archive.ubuntu.com/ubuntu/ eoan-updates multiverse
deb-src http://de.archive.ubuntu.com/ubuntu/ eoan-backports main restricted universe multiverse
...
```

Wenn Sie eine andere Ubuntu-Version installiert haben, steht bei Ihnen vielleicht `disco` statt `eoan`. Sie können das hier vorgestellte Verfahren dennoch nutzen.



Nach Änderungen an der Datei `/etc/apt/source.list` müssen dem System die Änderung durch Aufruf von

Code-Ausschnitt 5.58: Update des Ubuntu-Systems

```
sudo apt update
sudo apt upgrade
```

mitgeteilt werden, damit sie berücksichtigt werden. Gleichzeitig können wir das System auf den neuesten Stand bringen. Das folgende Kommando installiert alle Abhängigkeiten, die zum Übersetzen des Ubuntu-Kernels benötigt werden.

Code-Ausschnitt 5.59: Abhängigkeiten zum Übersetzen des Kernel installieren

```
sudo apt-get build-dep linux linux-image-$(uname -r)
sudo apt-get install libncurses-dev flex bison openssl libssl-dev dkms libelf-dev libudev-dev ↵
libpci-dev libiberty-dev autoconf git
```

Der aktuelle Ubuntu-Kernel kann mit Hilfe von `git` aus einem Repository ausgecheckt werden. Wechseln Sie mit dem Terminal in ein Verzeichnis Ihrer Wahl und laden Sie dann die Kernel-Quelltexte herunter.

Code-Ausschnitt 5.60: Kernel-Quelltexte herunterladen

```
git clone git://kernel.ubuntu.com/ubuntu/ubuntu-eoan.git kernel-source
```


Für andere Ubuntu-Versionen ändern Sie beispielsweise *eoan* in *disco* um. Haben Sie Geduld. Bis sich alle Daten im Verzeichnis *kernel-source* befinden, vergeht eine ganze Weile. Die Ubuntu-Kernel-Konfiguration ist bereits vorkonfiguriert. Das bedeutet, dass bestimmte Treiber im Kernel vorhanden sind und andere fehlen. Falls Sie die Konfiguration ohne Änderungen übernehmen möchten, überspringen Sie bitte den nächsten Schritt. Falls nicht, führen Sie die folgenden Kommandos aus:

Code-Ausschnitt 5.61: Konfiguration des Ubuntu-Kernels

```
chmod a+x debian/rules
chmod a+x debian/scripts/*
chmod a+x debian/scripts/misc/*
fakeroot debian/rules clean
fakeroot debian/rules editconfigs
```

Der Befehl *chmod a+x* erlaubt Benutzern, die oben angegebenen Skripte auszuführen. Der Befehl *fakeroot* führt einen Befehl in einer Umgebung aus, in der es scheint, als habe er Root-Rechte zur Dateimanipulation. Dies ist nützlich, um Benutzern zu erlauben, Archive (tar, ar, .deb etc.) mit Dateien darin zu erstellen, die Root-Rechte haben oder Root gehören. In unserem Fall erstellen wir Installations-Pakete für den Kernel, der in einer Root-Umgebung funktionieren muss. Das letzte Kommando leitet Sie durch viele Abfragen in Bezug auf den Kernel. Jede einzelne dieser Abfragen muss beantwortet werden. Andernfalls erhalten Sie später beim Übersetzen Beschwerden bezüglich der Kernel-Konfiguration.

Um Ihren Kernel als eigene Version zu kennzeichnen, empfiehlt es sich, dem Namen des Kernels eine eigene Bezeichnung mitzugeben, indem Sie z. B. die Erweiterung *+test* anhängen. Dieser Identifier erlaubt es Ihnen nach der Installation des eigenen Kernels, diesen von einem originalen Kernel zu unterscheiden, etwa beim Aufruf von

Code-Ausschnitt 5.62: Abfragen der Kernel-Version

```
uname -a
```

Das eigentliche Übersetzen des Kernels startet nach dem Aufruf von

Code-Ausschnitt 5.63: Übersetzen des Kernels starten

```
fakeroot debian/rules clean
fakeroot debian/rules binary-headers binary-generic binary-perarch
```

Das Bauen des Kernels kann schon einmal über eine Stunde dauern, je nach Prozessor Ihres Wohnzimmer-PCs. Nach dem erfolgreichen Übersetzen des Kernels finden Sie die folgenden Pakete auf Ihrem Rechner:

Code-Ausschnitt 5.64: Anzeigen der übersetzten und gepackten Kernel-Pakete

```
cd ..
ls *.deb
./
../
kernel-source/
linux-buildinfo-5.3.0-24-generic_5.3.0-24.26_amd64.deb
linux-cloud-tools-5.3.0-24_5.3.0-24.26_amd64.deb
linux-cloud-tools-5.3.0-24-generic_5.3.0-24.26_amd64.deb
linux-headers-5.3.0-24_5.3.0-24.26_all.deb
linux-headers-5.3.0-24-generic_5.3.0-24.26_amd64.deb
linux-image-unsigned-5.3.0-24-generic_5.3.0-24.26_amd64.deb
linux-modules-5.3.0-24-generic_5.3.0-24.26_amd64.deb
linux-modules-extra-5.3.0-24-generic_5.3.0-24.26_amd64.deb
linux-tools-5.3.0-24_5.3.0-24.26_amd64.deb
linux-tools-5.3.0-24-generic_5.3.0-24.26_amd64.deb
```

Der neu übersetzte Kernel kann so getestet werden:

Code-Ausschnitt 5.65: Installation des neu übersetzten Kernels

```
sudo dpkg -i linux*5.3.0.24*.deb
sudo reboot
```

Falls Ihre Kernel-Nummer eine andere ist, setzen Sie diese entsprechend im oben gezeigten Aufruf ein. Nach der Installation des neuen Kernels ist ein Neustart erforderlich, damit der neue Kernel aktiviert wird.

Durch einen Aufruf von

Code-Ausschnitt 5.66: Kernel-Konfiguration ändern

```
make menuconfig
```

im Verzeichnis kernel-source können Sie den Kernel ebenfalls individuell konfigurieren.



5.7 veracrypt

Nicht immer möchte man, dass seine Daten offen auf der Festplatte liegen. Zum Schutz der Daten kann eine Verschlüsselung Sinn machen. Das Programm *veracrypt* kann zur Verschlüsselung von Daten benutzt werden. Im Laufe dieses Abschnitts werden wir es aus dem Quelltext heraus übersetzen und dazu nutzen, einen verschlüsselten Container zu erstellen, in dem wir sensitive Daten lagern können.

Zu Beginn müssen einige Abhängigkeiten installiert werden:

Code-Ausschnitt 5.67: Installation der Abhängigkeiten für *veracrypt*

```
sudo apt-get install libfuse-dev libgtk-3-dev yasm
```

veracrypt erfordert zum Kompilieren *wxwidgets*. Da nicht alle Versionen von *veracrypt* mit allen Versionen von *wxwidgets* kompatibel sind, stelle ich Ihnen hier eine Kombination vor, die auf jeden Fall funktioniert. Laden Sie bitte zunächst die Version 3.1.2 von *wxwidgets* herunter. Diese erhalten Sie von <https://github.com/wxWidgets/wxWidgets/releases/tag/v3.1.2> [37]. Die Datei heißt *wxWidgets-3.1.2.tar.bz2*. Laden Sie in dasselbe Verzeichnis die Version 1.23 von *veracrypt* herunter. Diese finden Sie unter der URL [https://sourceforge.net/projects/veracrypt/files/VeraCrypt%](https://sourceforge.net/projects/veracrypt/files/VeraCrypt%201.23/)

201.23/ [38] unter dem Namen *VeraCrypt_1.23_Source.tar.bz2*. Danach wechseln Sie bitte mit einem Terminal in das Verzeichnis, in dem sich die beiden Dateien befinden und führen die folgenden Befehle aus. Ich gehe davon aus, dass sich die Dateien im Home-Verzeichnis befinden.

Code-Ausschnitt 5.68: Entpacken der Quelltexte zum Übersetzen von *veracrypt*

```
cd ~
tar xfvj wxWidgets-3.1.2.tar.bz2
mkdir veracrypt
cp VeraCrypt_1.23_Source.tar.bz2 veracrypt
cd veracrypt
tar xvfj VeraCrypt_1.23_Source.tar.bz2
cd src
```

Im Verzeichnis *src*, in dem Sie sich gerade befinden, gibt es ein *Makefile*. Bitte öffnen Sie diese Datei und entfernen Sie den Eintrag *--disable-compat26*. Mit diesem Eintrag gibt es sonst beim Übersetzen einen Fehler. Die korrekte Makefile-Zeile lautet dann:

Code-Ausschnitt 5.69: Korrektur des Makefiles

```
WX_CONFIGURE_FLAGS += --enable-unicode -disable-shared --disable-dependency-tracking --enable-exceptions --enable-std_string --enable-dataobj --enable-mimetype \
```

Nach der Vorarbeit geht es nun an das Übersetzen der Quelltexte. Zunächst wird *wxwidgets* übersetzt, und zwar statisch und ohne grafische Oberfläche. Wir beschränken uns auf den Terminal-Modus von *veracrypt*, das gleich im Anschluss übersetzt wird.

Code-Ausschnitt 5.70: Übersetzen der *veracrypt*-Quelltexte

```
sudo make WXSTATIC=1 NOGUI=1 WX_ROOT=~ /wxWidgets-3.1.2 wxbuild
sudo make WXSTATIC=1 NOGUI=1
```

Das fertig übersetzte Programm ist im Verzeichnis *veracrypt/src/Main* zu finden und heißt *veracrypt*. Rufen Sie es doch einmal auf um zu sehen, welche Parameter man ihm übergeben kann.

Erstellen wir nun einen Container, in dem wir eine zu schützende Beispieldatei unterbringen. Der Befehl

Code-Ausschnitt 5.71: Neuen *veracrypt*-Container erstellen

```
sudo ./veracrypt -c
```

erstellt einen neuen Container (*-c* steht für *create*). Während des Erstellens werden einige Fragen gestellt.

- **Volume type:**

- 1) Normal

- 2) Hidden

- Select [1]:

- Drücken Sie an dieser Stelle *ENTER*, um einen normalen, verschlüsselten Container zu erzeugen und keinen versteckten.

- **Enter volume path:**

- Geben Sie hier den Pfad und den Namen Ihres Containers an, der später die verschlüsselten Daten enthalten soll. Für einen ersten Test geben Sie bitte einfach *container* ein. Das erzeugt den Container mit dem Namen *container* direkt im aktuellen Verzeichnis.

- **Enter volume size (sizeK/size[M]/sizeG):**

Hier können Sie die Größe des Containers angeben. Da dieser nicht mit der Menge der Daten mitwächst, sollten Sie eine sinnvolle Größe wählen. Für den Test habe ich 1G eingegeben. Ich reserviere also 1 GB an Festplattenspeicher für den Container.

- **Encryption Algorithm:**

- 1) AES
- 2) Serpent
- 3) Twofish
- 4) Camellia
- 5) Kuznyechik
- 6) AES(Twofish)
- 7) AES(Twofish(Serpent))
- 8) Camellia(Kuznyechik)
- 9) Camellia(Serpent)
- 10) Kuznyechik(AES)
- 11) Kuznyechik(Serpent(Camellia))
- 12) Kuznyechik(Twofish)
- 13) Serpent(AES)
- 14) Serpent(Twofish(AES))
- 15) Twofish(Serpent)

Select [1]:

An dieser Stelle können Sie den Algorithmus wählen, mit dem der Container verschlüsselt wird. Ein simples *RETURN* wählt den standardmäßig eingestellten AES-Algorithmus.

- **Hash algorithm:**

- 1) SHA-512
- 2) Whirlpool
- 3) SHA-256
- 4) Streebog

Select [1]:

An dieser Stelle können Sie den Hash-Algorithmus wählen. Auch hier reicht das Drücken der *RETURN*-Taste aus, um den Defaultwert (SHA-512) zu aktivieren.

- **Filesystem:**

- 1) None
- 2) FAT
- 3) Linux Ext2
- 4) Linux Ext3
- 5) Linux Ext4
- 6) NTFS
- 7) exFAT

Select [2]:

Hier geben Sie das Dateisystem des Containers an. Die Voreinstellung ist FAT, damit der Container auch mit *veracrypt* unter Windows gelesen werden kann.

- **Enter password:**

Vergeben Sie hier ein sicheres Kennwort. Möchten Sie den Container wieder

öffnen, fragt *veracrypt* nach diesem Kennwort. Sollten Sie es vergessen, kommen Sie nie wieder an die Daten im Container.

- **Re-enter password:**

Aus Sicherheitsgründen müssen Sie hier Ihr Kennwort noch einmal angeben. So vermeiden Sie, dass Sie sich aus Versehen bei der ersten Eingabe vertippt haben.

- **Enter PIM:**

PIM steht für "Personal Iterations Multiplier". Dieser Parameter ist ab *veracrypt* Version 1.12 verfügbar. Sie können an dieser Stelle getrost *ENTER* drücken.

- **Enter keyfile path [none]:**

Hier können Sie eine Datei angeben, die Sie zusätzlich zur Verschlüsselung benutzen. Auch hier können Sie bedenkenlos *ENTER* drücken, um diese zusätzliche Verschlüsselungsdatei auszulassen.

- **Please type at least 320 randomly chosen characters and then press Enter:**

An dieser Stelle müssen Sie 320 willkürliche Tasten Ihrer Tastatur drücken. *veracrypt* nimmt es hier sehr genau. Sollten es weniger sein, erscheint eine Meldung ähnlich *Characters remaining: 52* und weist Sie darauf hin, dass z. B. noch 52 Tasteneingaben fehlen.

Veracrypt quittiert das Fertigstellen des Containers so:

Code-Ausschnitt 5.72: Erfolgsmeldung beim Fertigstellen des *veracrypt*-Containers

```
Done: 100,000% Speed: 279 MB/s Left: 0 s
```

```
The VeraCrypt volume has been successfully created.
```

Der Befehl

Code-Ausschnitt 5.73: Dateiparameter auf der Festplatte anzeigen

```
ls -l container
```

zeigt den 1 GB großen Container an:

Code-Ausschnitt 5.74: Dateiparameter für den *veracrypt*-Container

```
-rw----- 1 root root 1073741824 Nov 30 13:34 container
```

Um den Container mit Daten zu füllen, muss er eingebunden werden, d.h. in ein Verzeichnis eingebunden werden. Dateien, die dann in dieses Verzeichnis kopiert werden, landen dann automatisch im Container, solange dieser in diesem Verzeichnis eingebunden ist. Das Einbinden geschieht mit Hilfe des folgenden Kommandos:

Code-Ausschnitt 5.75: Mounten des *veracrypt*-Containers

```
sudo ./veracrypt --mount
```

veracrypt stellt wiederum einige Fragen.

- **Enter volume path:**

Geben Sie hier den Container an, den Sie einbinden wollen, also beispielsweise *container*.

- **Enter mount directory [default]:**
An dieser Stelle wird das Verzeichnis angegeben, in das der Container eingebunden werden soll. Ubuntu bietet das Verzeichnis `/mnt` dafür an, das bereits existiert.
- **Enter password for container:**
Geben Sie bitte an dieser Stelle das Kennwort ein, das Sie beim Erstellen des Containers vergeben haben.
- **Enter PIM for container:**
Wenn Sie keine PIM beim Erstellen des Containers vergeben haben, drücken Sie bitte an dieser Stelle `RETURN`.
- **Enter keyfile [none]:**
Wenn Sie keine Schlüsseldatei beim Erstellen des Containers angegeben haben, drücken Sie bitte an dieser Stelle `RETURN`.
- **Protect hidden volume (if any)? (y=Yes/n=No) [No]:**
Drücken Sie an dieser Stelle bitte `RETURN`.

Wir wechseln nun in das Verzeichnis, in dem der Container eingebunden ist

Code-Ausschnitt 5.76: In das `/mnt`-Verzeichnis wechseln

```
cd /mnt
```

und schauen uns den Inhalt des Containers an.

Code-Ausschnitt 5.77: Anzeigen des Inhalts eines Verzeichnisses

```
ls -l
```

Bei mir ist die Ausgabe leer.

Code-Ausschnitt 5.78: Inhalt des Containers anzeigen

```
insgesamt 0
```

Das ist eigentlich klar, da wir eben erst einen leeren Container erstellt haben. Im nächsten Schritt erstellen wir mit einem Editor eine Datei im Container, die nur den Text „geheim“ enthält.

Code-Ausschnitt 5.79: Datei mit geheimen Inhalt erstellen

```
vi datei.txt
```

Geben Sie dann `i` für *insert* ein, tippen Sie *geheim* und speichern Sie die Datei mit `ESC:wq` für *write* und *quit*. Schauen wir uns den in `/mnt` eingebundenen Container nun mit `ls -l` an, zeigt er Folgendes an:

Code-Ausschnitt 5.80: Containerinhalt erneut anzeigen

```
insgesamt 4
-rwx----- 1 rudi rudi 7 Nov 30 13:38 datei.txt
```

Die Datei `datei.txt` ist im Container vorhanden. Verlassen wir nun das Verzeichnis, in dem der Container eingebunden ist

Code-Ausschnitt 5.81: In das letzte Verzeichnis zurück wechseln

```
cd -
```

und hängen den Container so aus:

Code-Ausschnitt 5.82: *veracrypt*-Container aushängen

```
sudo ./veracrypt -d
```

Wenn wir uns nun das Verzeichnis */mnt* wieder mit dem Befehl *ls -l* anschauen, ist es leer.

Code-Ausschnitt 5.83: Anzeige des leeren Mount-Verzeichnisses */mnt*

```
ls -l /mnt
insgesamt 0
```

Die geheime Datei *datei.txt* steckt verschlüsselt im Container. Die Containergröße hat sich dabei nicht verändert. Alle im Container gespeicherten Daten werden erst nach erneutem Einbinden des Containers in ein Verzeichnis und nach Eingabe des Kennwortes zur Entschlüsselung wieder sichtbar.



Sie können komplette Container auch auf andere Rechner kopieren und dort ebenfalls mit Hilfe von *veracrypt* sichtbar machen oder ergänzen.

5.8 Kleider machen Leute

Viele finden den Look von Ubuntu trist oder mögen die Farbgebung nicht. Unter LINUX ist es aber kein Problem, das komplette Aussehen des Ubuntu-Schreibtisches zu ändern. Der Phantasie sind hier keine Grenzen gesetzt. Im Internet gibt es unzählige Themen, die das Aussehen der LINUX-Oberfläche verändern. In diesem Abschnitt stelle ich Ihnen ein Thema vor, das den Ubuntu-Desktop wie MacOS Mojave aussehen lässt und erkläre Ihnen alle Schritte auf dem Weg zu diesem neuen Look.

Im ersten Schritt installieren wir die *gnome-tweaks*, die es uns erlauben, das prinzipielle Ubuntu-Thema zu ändern.

Code-Ausschnitt 5.84: Installieren der *gnome-tweaks*

```
sudo apt install gnome-tweaks gnome-tweak-tool
```

Falls Sie sich noch nicht mit dem Terminal angefreundet haben, können Sie auch in der Ubuntu-Software-Verwaltung nach *gnome tweaks* suchen und das Programm von dort aus installieren. Die Seite <https://www.gnome-look.org/> [39] stellt verschiedene Themen und Looks für LINUX zur Verfügung. Ich habe mich für das *McMojave*-Thema entschieden, das Sie unter <https://www.gnome-look.org/p/1275087/> [40] finden. Wechseln Sie bitte zur Rubrik *Files* und wählen Sie ein Thema nach Ihrem Geschmack. Ich habe mich für das *Mojave-light*-Thema entschieden. Keine Sorge. Sollten Ihnen das Thema nicht zusagen, können Sie es jederzeit wieder entfernen und ein anderes oder das Original installieren.

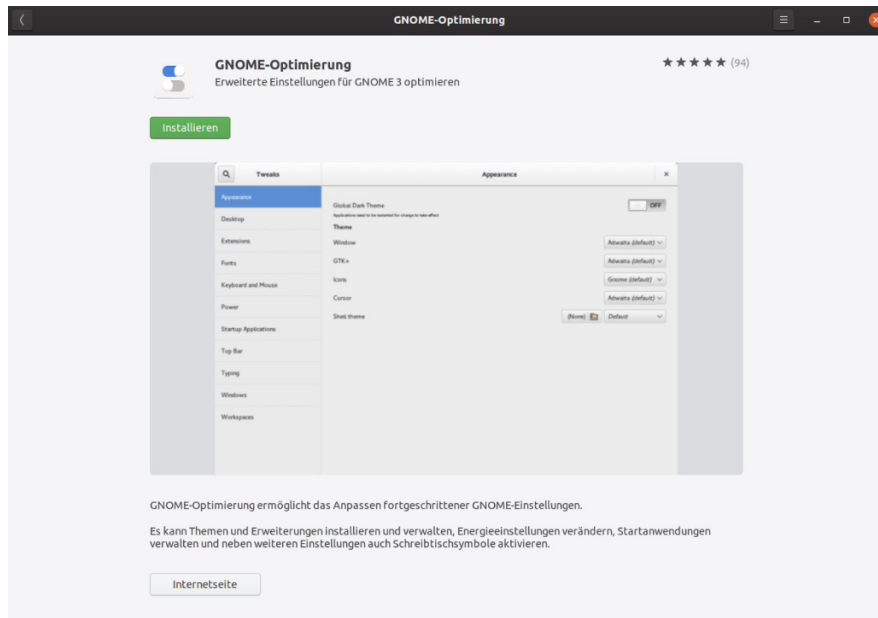


Abbildung 5.19: Mit Hilfe der *gnome-tweaks* kann das Aussehen der Ubuntu-Oberfläche komplett geändert werden.

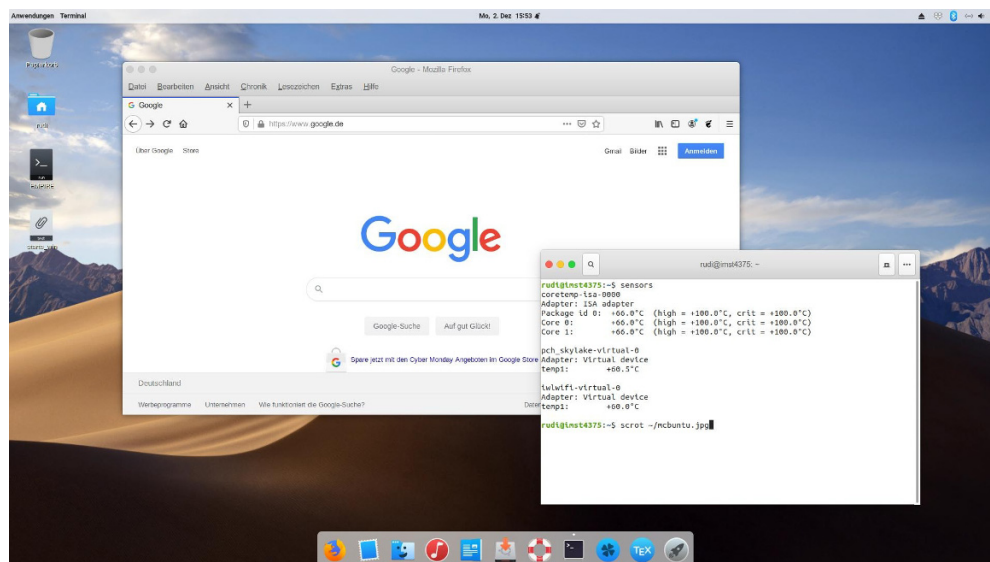


Abbildung 5.20: Mit *McMojave* sieht der Ubuntu-Desktop aus wie *MacOS Mojave*.

Ich gehe davon aus, dass sich die Themen-Datei nach dem Download in Ihrem Home-Verzeichnis im Verzeichnis *Downloads* befindet. Bitte entpacken Sie diese Datei und kopieren Sie das Thema anschließend in den dafür neu erstellen, vorgesehenen Order.

Code-Ausschnitt 5.85: Entpacken der neuen Themendatei und Kopieren in den Themen-Ordner

```
cd ~/Downloads
tar xvf Mojave-*.tar.xz
mkdir -p ~/.themes
mv Mojave-* ~/.themes
```

Öffnen Sie nun die zuvor installierten *gnome-tweaks* unter der Ubuntu-Software und navigieren Sie zum Reiter *Erscheinungsbild*.

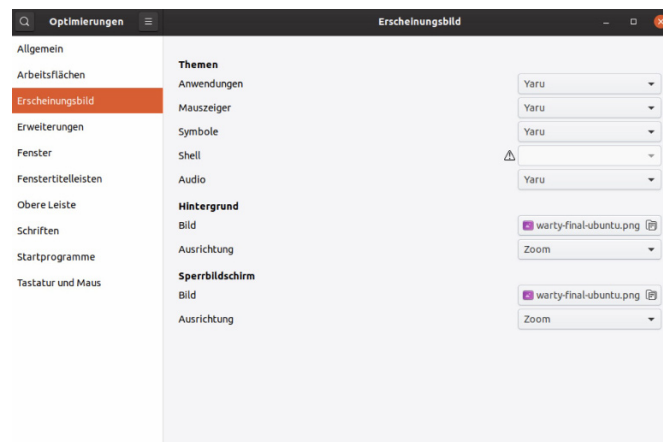


Abbildung 5.21: Wählen Sie im Erscheinungsbild das soeben heruntergeladene MacOS-Thema aus.

Unter *Anwendungen*, wo jetzt *Yaru* steht, wählen Sie bitte das MacOS-Thema aus.

Die linke Ubuntu-Leiste sieht noch nicht so aus, wie man das unter MacOS gewohnt ist. Ziehen Sie die Leiste also bitte mit der Maus an den unteren Bildschirmrand. Als Nächstes können Sie die *dash-to-dock*-Erweiterung installieren. Diese müssen Sie über die Ubuntu-Software-Verwaltung installieren. Suchen Sie dort nach „dash to dock“. Diese Erweiterung erlaubt es Ihnen, das Ubuntu-Dock so einzustellen, dass es dem MacOS-Dock mehr ähnelt. Dazu können Sie die Farbe auf *weiß* stellen und die Transparenz nach Ihrem Geschmack ändern.

Ein passendes MacOS Icon-Thema finden Sie unter <https://www.gnome-look.org/p/1102582>. Laden Sie es herunter, kopieren Sie es in das Verzeichnis */.icons* und nutzen Sie danach wieder das *gnome-tweaks*-Tool, um die entsprechenden Icons zu wählen.

Sie können ebenfalls das *gnome tweaks*-Tool verwenden, um die Schriftarten zu ändern. Die Open-Source-Schriftart, die der in MacOS benutzten nahe kommt, nennt sich „Garuda“. Nutzen Sie diese Schrift für Fenstertitel- und Interface-Fonts.

Um die bunten Icons von *LibreOffice* gegen Icons auszutauschen, die denen von MacOS eher entsprechen, können Sie das folgende Paket installieren:

Code-Ausschnitt 5.86: Ändern des Styles von *LibreOffice*

```
sudo apt-get install libreoffice-style-sifr
```

Nach der Installation muss der Style *Sifr* noch in LibreOffice aktiviert werden (*Optionen* → *LibreOffice* → *Ansicht* → *Icon Größe und Style*).

MacOS Mojave stellt dynamische Wallpaper zur Verfügung: Das Desktop-Bild ändert seine Helligkeit mit der Uhrzeit. Auch dieses Verhalten können Sie unter Ubuntu nachstellen. Die Webseite <https://gist.github.com/trongthanh/7d632e90687e1bc219e1f3262d337702> [41] stellt das Skript *mojave.xml* zur Verfügung, welches die entsprechenden Wallpaper laden kann. Geben Sie *mojave.xml* als Desktop-Hintergrund an. Die Datei *mojave.xml* verweist dann wiederum auf die entsprechenden Wallpaper. Aus rechtlichen Gründen darf ich hier den Link nicht angeben, unter dem Sie Mojave-Wallpaper herunterladen können.

Je nach Geschmack können Sie die Knöpfe zum Öffnen, Verkleinern und Schließen der Fenster noch mit Hilfe der *gnome-tweaks* von der rechten auf die linke Seite schieben.

Zusammenfassung und Ausblick 5 Nach den wichtigen Grundlagen steigen wir nun in den Spielspaß ein. Lernen Sie im nächsten Kapitel, wie Ihr Wohnzimmer-PC genutzt werden kann, um aktuelle Spielekonsolen und solche vergangener Tage zu emulieren.

Wir schauen uns dabei in erster Linie Emulatoren an, die Nintendos Spielekonsolen nachbilden können. Wir werden verschiedene Emulatoren unter Windows und Ubuntu installieren und vergleichen. Weiterhin zeige ich Ihnen, wo die Grenzen der Emulatoren liegen und welche Hardware erforderlich ist, um flüssig spielen zu können.

Einige der Programme sind für das Betriebssystem Windows geschrieben, können aber dennoch mit Hilfe von *wine* unter LINUX-Betriebssystemen zum Leben erweckt werden. Doch davon mehr im nächsten Kapitel... ■

Literaturverzeichnis

- [30] putty. *Download PuTTY*. 2019.
<https://www.putty.org>
besucht am 27. 12. 2019 (siehe Seite 80).
- [31] Wikipedia. *Xming*. 2019.
<https://de.wikipedia.org/wiki/Xming>
besucht am 27. 12. 2019 (siehe Seite 80).
- [32] MobaXTerm. *Home Edition*. 2019.
<https://mobaxterm.mobatek.net/download.html>
besucht am 27. 12. 2019 (siehe Seite 80).
- [33] RealVNC. *VNC Connect*. 2019.
<https://www.realvnc.com/de/connect/download/viewer/>
besucht am 27. 12. 2019 (siehe Seite 82).
- [34] OpenVPN. *OpenVPN Connect for Windows*. 2019.
<https://openvpn.net/client-connect-vpn-for-windows/>
besucht am 27. 12. 2019 (siehe Seite 93).
- [35] WinSCP. *Free Award-Winning File Manager*. 2019.
<https://winscp.net/eng/index.php>
besucht am 27. 12. 2019 (siehe Seiten 94, 95).
- [36] Python. *Webseite*. 2019.
<http://www.python.org>
besucht am 27. 12. 2019 (siehe Seite 97).
- [37] wxWidgets. *wxWidgets 3.1.2*. 2019.
<https://github.com/wxWidgets/wxWidgets/releases/tag/v3.1.2>
besucht am 27. 12. 2019 (siehe Seite 101).
- [38] VeraCrypt. *Open source disk encryption with strong security for the Paranoid*. 2019.
<https://github.com/wxWidgets/wxWidgets/releases/tag/v3.1.2>
besucht am 27. 12. 2019 (siehe Seite 102).
- [39] gnome-look.org. *Eyecandy for your Gnome Desktop*. 2019.
<https://www.gnome-look.org/>
besucht am 27. 12. 2019 (siehe Seite 106).
- [40] McMojave. *GTK3 Themes*. 2019.
<https://www.gnome-look.org/p/1275087/>
besucht am 27. 12. 2019 (siehe Seite 106).
- [41] trongthanh. *mojave.xml*. 2019.
<https://gist.github.com/trongthanh/7d632e90687e1bc219e1f3262d337702>
besucht am 27. 12. 2019 (siehe Seite 109).

6. Spiele



6.1	Nintendo Wii U-Emulator unter Windows	111
6.2	Nintendo Wii U-Emulator unter LINUX	116
6.3	Nintendo Switch-Emulator	122
6.4	Nintendo Gamecube & Wii-Emulator	126
6.5	N64 Emulator	127
6.6	LINUX Spiele	131

In diesem Kapitel steht der Spaß am Spielen im Vordergrund. Dabei werden einige Emulatoren vorgestellt, mit deren Hilfe man Konsolen-Spiele auf dem PC spielen kann. Rechtlich gesehen bewegt man sich hier wieder in einer Grauzone. Hat man ein Wii U-Spiel gekauft, hat Nintendo sicherlich nichts dagegen, wenn man dieses auch auf dem PC mit Hilfe eines Emulators spielt. Das Herunterladen von Spiele-ROMs aus dem Internet ist aber mit Sicherheit illegal, wenn man das Original nicht besitzt. Kaufen Sie im Zweifel also bitte eine entsprechende Spielekonsole mitsamt der Spiele, die Sie spielen möchten.



6.1 Nintendo Wii U-Emulator unter Windows



Abbildung 6.1: Quelle:
supermariorun.com [42]

Wer kennt ihn nicht, Super Mario, den italienischen Klempner, der seit vielen Jahren Prinzessin Peach aus den Händen eines Monsters befreien muss? Ob auf einer betagten SNES-Konsole, einer modernen Wii U oder einer Nintendo Switch, dieses Jump and Run-Spiel ist ein kindertauglicher Klassiker, der mit viel Liebe zum Detail daherkommt. Daher zählt Super Mario 3D World auf der Wii U und Super Mario Odyssey auf der Switch zu meinen absoluten Lieblingsspielen. *cemu* ist ein Wii U-Emulator, der unter Windows läuft. Falls Sie auf Ihrem Wohnzimmer-PC kein Windows installiert haben, ist das nicht weiter schlimm. Wir werden diesen Emulator später noch unter Ubuntu zum Einsatz bringen. *cemu* kann unter <http://cemu.info> [43] heruntergeladen werden. Das heruntergeladene Paket bringt keinen Installer mit, sondern lediglich eine zip-Datei, die mit den Windows-eigenen Bordmitteln an einem Ort Ihrer Wahl entpackt werden kann.

Für einen schnellen Zugriff bietet sich der Desktop an. Zu dem Zeitpunkt, als ich dieses Buch geschrieben habe, war die Version 1.15.2d aktuell, der entsprechende Ordner heißt nach dem Entpacken also *cemu_1.15.2*.

Damit *cemu* richtig läuft, ist die Installation des Microsoft Visual C++ 2017 X64 Redistributables erforderlich. Der Link hierzu befindet sich auch auf der Webseite von *cemu*. Jetzt könnte *cemu* prinzipiell gestartet werden. Allerdings fehlt noch ein ROM, beispielsweise das von Super Mario 3D World sowie eine Datei namens *keys.txt*, die neben dem sogenannten Wii U common key auch den Schlüssel für das Spiele-ROM enthalten muss. Die Spiele-ROMs selbst haben die Endung *.wud*, was für Wii U data steht. Die Datei *keys.txt* ist so aufgebaut und wird im *cemu*-Hauptverzeichnis platziert:

Code-Ausschnitt 6.1: *keys.txt* für *cemu*

```
# this file contains keys needed for decryption of file system data (WUD/WUX)
# 1 key per line, any text after a '#' character is considered a comment
# the emulator will automatically pick the right key

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx # Wii U Common Key
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx # Super Mario 3D World
```

Die entsprechenden Schlüssel in der Datei *keys.txt* habe ich ausge-x-t. Falls Sie nicht wissen, wie Sie ein Abbild Ihres Wii U-Spiels erstellen können, finden Sie entsprechende Dateien auch im Internet.

Hinweis 6.1 (*keys.txt* und ROMs) Bitte denken Sie daran, dass das Herunterladen einer ROM-Datei eines Spiels aus dem Internet illegal ist. Selbst, wenn Sie dieses Spiel besitzen, kann es sich hier um eine Grauzone handeln. Dasselbe gilt für den Inhalt der Datei *keys.txt*, der sich auch im Internet finden lässt.

Wii U-ROMs können gerne einmal 25 GB groß sein. Das Verzeichnis, in dem sich Ihre ROMs befinden, können Sie in *cemu* angeben. Die Spiele erscheinen dann auf der Startseite und es muss nicht bei jedem Start der Pfad zum ROM angegeben werden.

Falls Sie keinen Controller zur Hand haben, können Sie *cemu* nun starten und die Tastatur als Eingabegerät konfigurieren. Richtiger Spielspaß kommt aber erst mit einem Controller auf. Ein PS4-Controller lässt sich einfach per Bluetooth verbinden und mit Hilfe des zusätzlichen Programms *DS4Windows* für *cemu* nutzen.



Abbildung 6.2: Ein PS4-Controller eignet sich hervorragend zum Spielen mit *cemu*. Sogar das Touchpad funktioniert einwandfrei. Quelle: saturn.de

Das Programm *DS4Windows* steht auf <http://ds4windows> [44] zum Download bereit. Führen Sie nach dem Download bitte die Installation aus und starten Sie das Programm. Anschließend wird der PS4-Controller durch Drücken der Playstation-Taste in den Pairing-Modus versetzt, so dass er per Bluetooth mit Windows verbunden werden kann.

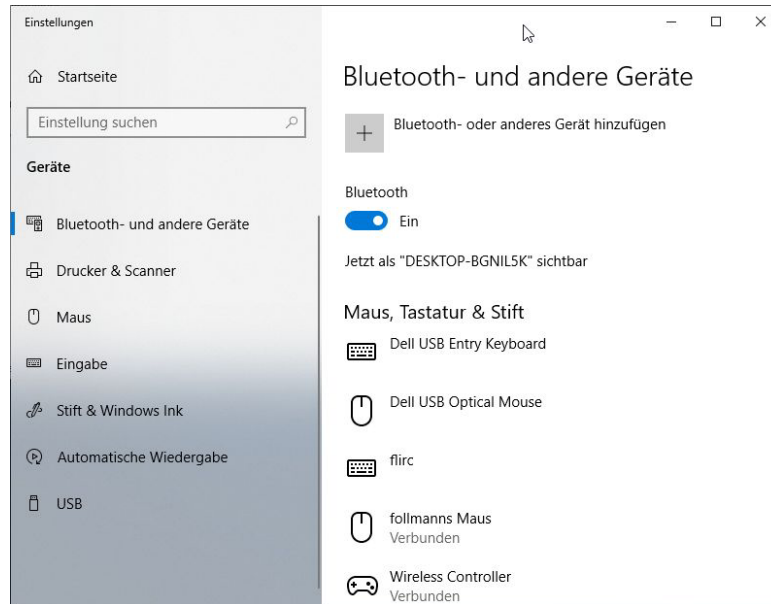


Abbildung 6.3: Verbindung eines Bluetooth-Controllers unter Windows.

Nach erfolgreicher Verbindung erscheint der Controller dann auch im Programm *DS4Windows*.

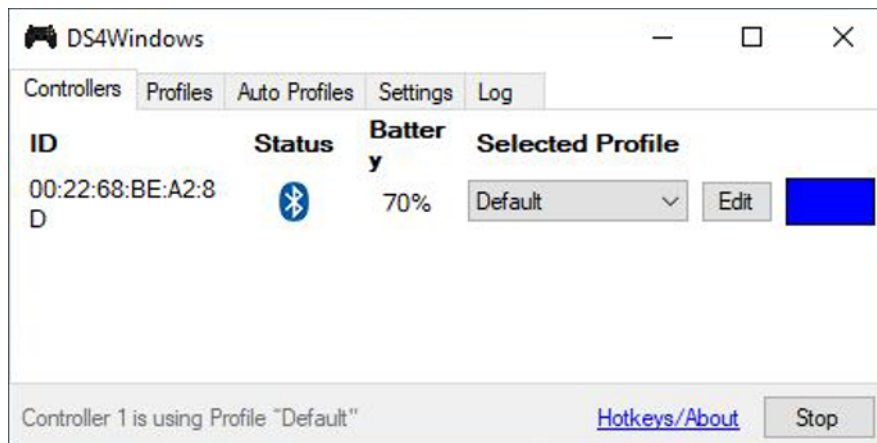


Abbildung 6.4: DS4Windows sorgt dafür, dass ein PS4-Controller unter Windows mit *cemu* verwendet werden kann.

Der Emulator *cemu* kann jetzt gestartet werden. Bevor der Spielspaß beginnt, empfiehlt es sich, den Controller so einzustellen, wie in der Abbildung 6.6 gezeigt wird.

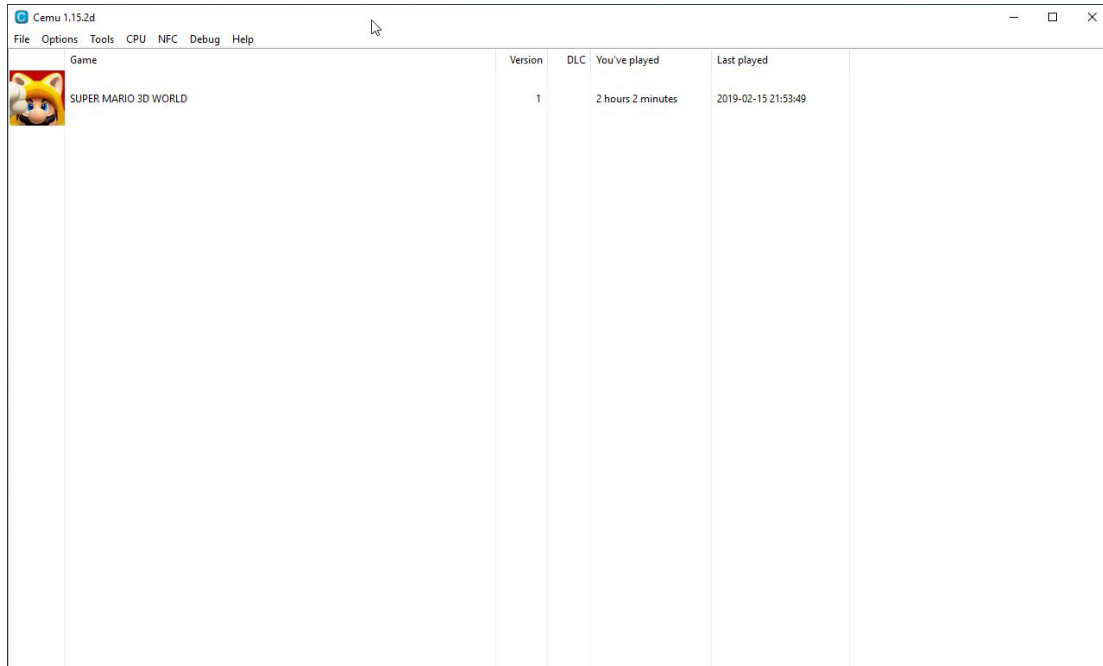


Abbildung 6.5: *cemu* macht Super Mario 3D World unter Windows spielbar.

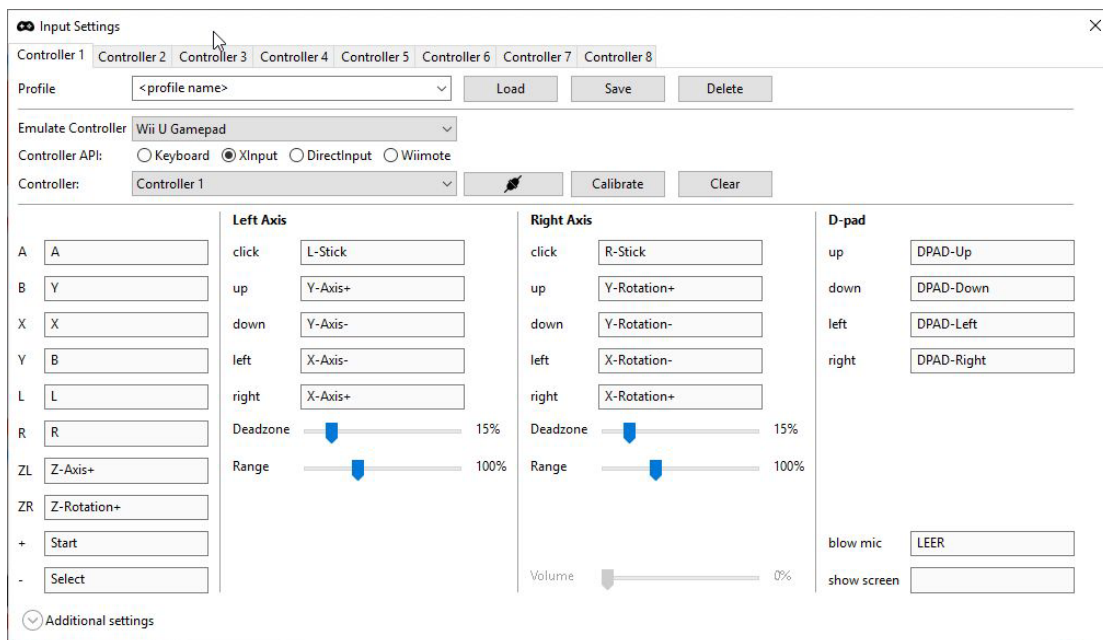


Abbildung 6.6: *cemu*-Einstellungen für einen PS4-Controller, der mit *DS4Windows* eingebunden ist.

Nach dem Einstellen des Controllers startet ein Doppelklick auf das Spiele-Logo das jeweilige Spiel.

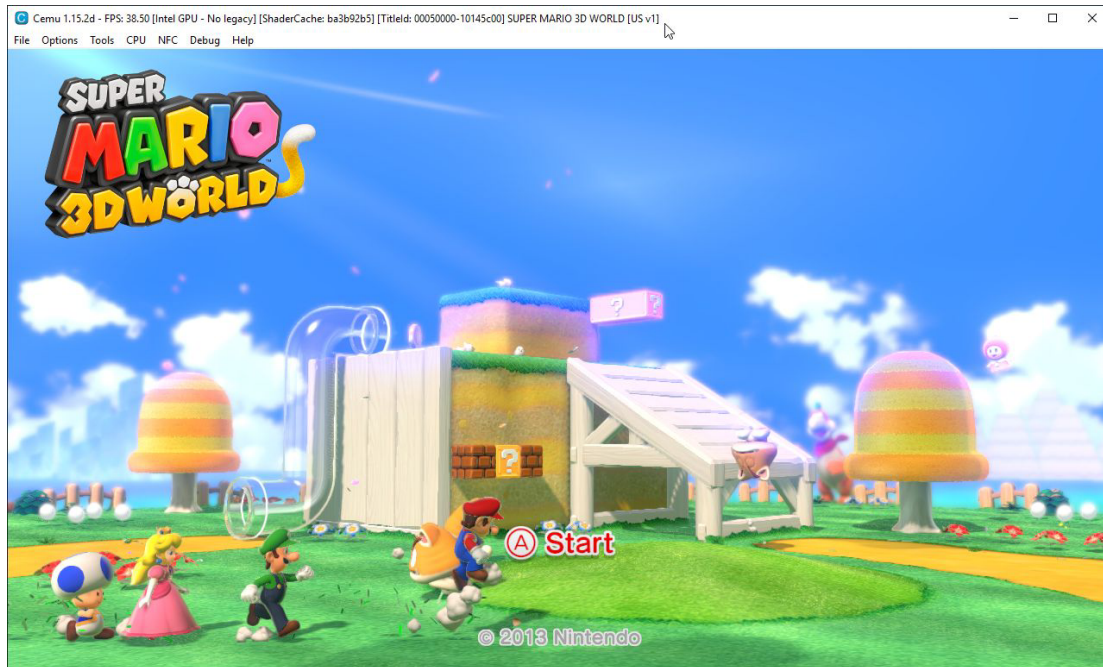


Abbildung 6.7: Der Start-Bildschirm von Super Mario 3D World in *cemu*.

Je nach Hardware-Ausstattung des verwendeten PCs kann es sein, dass das Spiel an der ein oder anderen Stelle ruckelt. Das kann man vermeiden, indem man im Internet nach vor-kompilierten Shader-Cache-Dateien sucht. Diese Datei kopiert man in das *cemu*-Unterverzeichnis *shaderCache/transferable*. Damit der Shader-Cache aktiviert wird, muss *cemu* neu gestartet werden. Für Super Mario 3D World heißt diese Datei *ba3b92b5.bin*.

Obwohl die *cemu*-Webseite vor der Verwendung einer Intel-GPU warnt, konnte ich bei meinem Test keinerlei Grafikprobleme feststellen. Die *cemu*-Webseite rät zur Verwendung von NVidia- oder AMD-Grafikeinheiten.

Auf meinem Lattepanda Alpha habe ich Frame-Raten von durchschnittlich 30-45 fps erreicht. Damit ist das Spiel ohne Probleme spielbar. Leistungsfähigere CPUs und GPUs kommen hier aber auf Frame-Raten vom 60 fps, selbst in einer 4K-Auflösung.

Manche Stellen im Spiel Super Mario 3D World verlangen die Benutzung eines Mikrofons. Hat man keines zur Hand, kann man das Mikrofon unter den Controller-Einstellungen auch auf eine Taste der Tastatur legen, beispielsweise auf die Leertaste.





Abbildung 6.8: Viel Spaß macht Super Mario 3D World erst im Vollbild-Modus.

Richtiger Spielspaß kommt aber erst im Vollbildmodus auf, den man ebenfalls in *cemu* einstellen kann.

6.2 Nintendo Wii U-Emulator unter LINUX

Der Emulator *cemu* ist leider nicht im Quelltext verfügbar. Daher gibt es keine LINUX-Version dieses Programms. Mit einem Trick ist es aber dennoch möglich, *cemu* unter LINUX zu benutzen: Mit dem Programm *wine* existiert ein Windows-Emulator, in dem wir *cemu* starten können. Ganz schön abgefahren, oder? Wir emulieren in Ubuntu ein Windows, das dann seinerseits eine Wii U emuliert und das Spielen von Wii U-Spielen unter Ubuntu ermöglicht.



Abbildung 6.9: Quelle: Wikipedia.

wine ist eigentlich gar kein Windows-Emulator. Das sagt auch schon der Name, der ein rekursives Akronym für „Wine Is Not an Emulator“ ist. Vielmehr ist *wine* eine Windows-kompatible Laufzeitumgebung, die es ermöglicht, Windows-Programme unter LINUX mit X-Windows auszuführen. *wine* kann daher ohne Windows-Installation ausgeführt werden. Auf der offiziellen *wine*-Website (<https://appdb.winehq.org/>) gibt es eine Liste mit Anwendungen, die unter *wine* laufen. Da *cemu* leider nicht mit der Ubuntu 18.10-Version von *wine* funktioniert, installieren wir die Staging-Version. Diese findet sich fertig kompiliert unter <https://dl.winehq.org/> [45]. Die Staging-Version beinhaltet viele aktuelle Ergänzungen, die in der Standard- oder in der Entwickler-Version noch nicht freigegeben sind.

Die Staging-Version von *wine* kann so installiert werden:

Code-Ausschnitt 6.2: Installation der Staging-Version von *wine*

```
sudo dpkg --add-architecture i386
sudo apt-get install wget
wget -nc https://dl.winehq.org/wine-builds/Release.key
sudo apt-key add Release.key
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 76F1A20FF987672F
sudo apt-add-repository https://dl.winehq.org/wine-builds/ubuntu/
sudo apt-get update
sudo apt-get install --install-recommends winehq-staging
```

Im Code-Ausschnitt 6.2 geschieht Folgendes: Zunächst wird die Unterstützung für 32-Bit-Software hinzugefügt. Viele Windows-Programme sind immer noch 32-Bit-Programme und nicht unbedingt 64-Bit-Programme. Danach wird *wget* installiert. Mit diesem Tool können Dateien aus dem Internet heruntergeladen werden. Das wird im nächsten Schritt getan, indem die Datei *Release.Key* heruntergeladen wird. Dieser Schlüssel wird anschließend zu der Repository-Verwaltung hinzugefügt und in einem weiteren Schritt verifiziert. Anschließend wird das *winehq*-Repository mit den *wine*-Builds für Ubuntu zu den erlaubten Repositories hinzugefügt. Das nächste *sudo apt-get update* aktualisiert die Software-Listen der Repositories. Im letzten Schritt wird dann die Staging-Version von *wine* mit allen erforderlichen Abhängigkeiten installiert.

Im nächsten Schritt kümmern wir uns um die korrekte Konfiguration von *wine*. Starten Sie aus einem Terminal heraus

Code-Ausschnitt 6.3: *wine*-Konfiguration

```
winecfg
```

und setzen Sie die Windows-Version auf die Version 10 (Abb. 6.10).

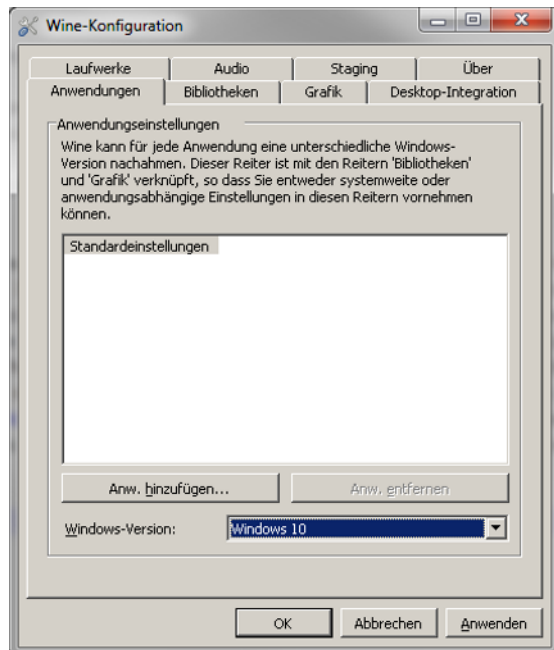


Abbildung 6.10: Konfiguration der Windows-Version für *wine*.

Im Tab *Bibliotheken* ergänzen Sie bitte *dbghelper.dll* und *keystone.dll*, wie in der Abbildung 6.11 gezeigt wird.

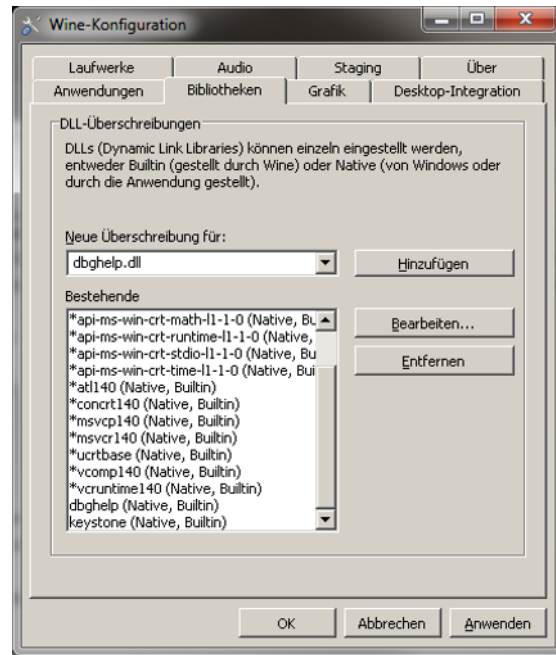


Abbildung 6.11: Konfiguration der Windows-Version für *wine* - Fortsetzung.

Die *wine*-Konfiguration wird im Verzeichnis `./wine` des aktuellen Benutzers gespeichert. Falls eine Konfiguration komplett gelöscht werden soll, kann das durch den folgenden Befehl erledigt werden.

Code-Ausschnitt 6.4: Löschen der *wine*-Konfiguration

```
rm -rf ~/.wine
```

Nachdem *wine* konfiguriert wurde, muss auch unter LINUX die Visual-Studio C++ Runtime-Umgebung (2017) hinzugefügt werden. Dies geschieht mit Hilfe des Programms *winetricks*, das Sie so installieren können:

Code-Ausschnitt 6.5: Installation von *winetricks*

```
sudo apt-get install winetricks
```

Ein anschließender Aufruf von

Code-Ausschnitt 6.6: Installation der VC++ 2017 Runtime-Umgebung

```
bash winetricks vcrun2017
```

lädt die Microsoft Visual-Studio C++ 2017-Runtime-Umgebung herunter und installiert Sie. Während der Installation werden Sie dazu aufgefordert, diese zu bestätigen. Achten Sie bitte darauf, dass Ihre Ubuntu Grafik-Treiber aktuell sind. Wie bereits weiter oben beschrieben, können Sie dazu folgendes Repository benutzen:

Code-Ausschnitt 6.7: Installation der aktuellen Grafiktreiber

```
sudo add-apt-repository ppa:oibaf/graphics-drivers
sudo apt update && sudo apt dist-upgrade
```

Da sowohl *wine-staging* als auch *cemu* permanent weiterentwickelt werden, ist es nicht immer einfach, eine *cemu*-Version zu finden, die gerade mit *wine* funktioniert. Zum Zeitpunkt, als ich dieses Buch geschrieben habe, funktionierte die *cemu*-Version 1.14 problemlos mit *wine*, neuere *cemu*-Versionen (1.15.2d) liefen hingegen nicht. Besorgen Sie sich daher bitte aus dem Internet die Version 1.14 von *cemu*, indem Sie eine

Suchmaschine Ihrer Wahl bemühen (z. B. google). *cemu* wird in der Regel als zip-Datei ausgeliefert, die Sie auch unter Ubuntu entpacken können. Nehmen wir an, Sie haben *cemu* in der Version 1.14.0 heruntergeladen. Dann können Sie es wie folgt auspacken:

Code-Ausschnitt 6.8: Entpacken von *cemu*

```
sudo apt-get install unzip
cp ~/Downloads/cemu_1.14.0.zip ~
cd ~
unzip cemu_1.14.0
```

Anschließend muss noch *cemuhook* installiert werden. *cemuhook* ist ein H264-Modul für *cemu*, das darüber hinaus auch eine Gamepad-Eingabe zur Verfügung stellt. Zusätzlich erlaubt *cemuhook* das Patchen von Daten und Code eines Spiels zur Laufzeit, so dass eventuell auftretende Fehler vermieden werden können. Die Downloads finden Sie unter <https://cemuhook.sshnuke.net/#Downloads> [46]. Achten Sie bitte darauf, die richtige *cemuhook* herunterzuladen, die zu Ihrer *cemu*-Version passt. Für die *cemu*-Version 1.14.0 ist das die *cemuhook*-Version *cemuhook_1140_0570.zip*. Kopieren Sie diese Datei bitte in das *cemu*-Verzeichnis und entpacken Sie sie.

Code-Ausschnitt 6.9: Entpacken von *cemuhook*

```
cp ~/Downloads/cemuhook_1140_0570.zip ~/cemu_1.14.0
cd ~/cemu_1.14.0
unzip cemuhook_1140_0570.zip
```

cemuhook enthält auch die beiden Dateien *dbghelper.dll* und *keystone.dll*, die weiter oben in die *wine*-Konfiguration eingetragen wurde. Möchten Sie unter Ubuntu mit mehr als 30 fps spielen, empfiehlt sich zusätzlich das Herunterladen eines sogenannten „graphics pack“. Das graphics pack kann von der Webseite https://slashiee.github.io/cemu_u_graphic_packs/ [47] heruntergeladen werden (*graphicsPacks367.zip*). Entpacken Sie diese Datei im Verzeichnis *cemu_1.14.0/graphicsPacks*.

Code-Ausschnitt 6.10: Installation der graphics packs

```
cp ~/Downloads/graphicsPacks367.zip ~/cemu_1.14.0/graphicsPacks
cd ~/cemu_1.14.0/graphicsPacks
unzip graphicsPacks367.zip
```

Auch wenn Sie *cemu* unter LINUX verwenden, müssen Sie die weiter oben beschriebene Datei *keys.txt* in das *cemu*-Verzeichnis kopieren. Weiterhin können dieselben Spiele-ROMs verwendet werden, die Sie in ein Verzeichnis Ihrer Wahl kopieren können.

Wii U-ROMs sind sehr groß. Sollten Sie auf Ihrem PC bereits ein Windows installiert haben, auf dem sich ROMs befinden, kann Ubuntu auf diese Partition zugreifen und die ROMs auch für LINUX zur Verfügung stellen. In der Regel bindet Ubuntu Windows-Partitionen automatisch ein und stellt diese im Verzeichnis /media zur Verfügung.





Auch unter Ubuntu können Sie die Shader-Cache-Datei verwenden. Diese muss in das Verzeichnis `cemu_1.14.0/shaderCache` kopiert werden.

`cemu` kann nun wie folgt aufgerufen werden:

Code-Ausschnitt 6.11: Start von `cemu` unter Ubuntu

```
MESA_VENDOR_OVERRIDE="mesa" MESA_RENDERER_OVERRIDE="mesa" MESA_GL_VERSION_OVERRIDE=4.5COMPAT ↵  
vblank_mode=0 mesa_glthread=true wine64 Cemu.exe
```

Die Parameter `MESA...` sind dabei für Intel-Grafikkarten erforderlich. Bei NVidia- oder AMD-GPUs können sie weggelassen werden. Es erscheint eine Warn-Box, bei der man *Abbrechen* drücken muss, damit `cemu` weiter lädt.

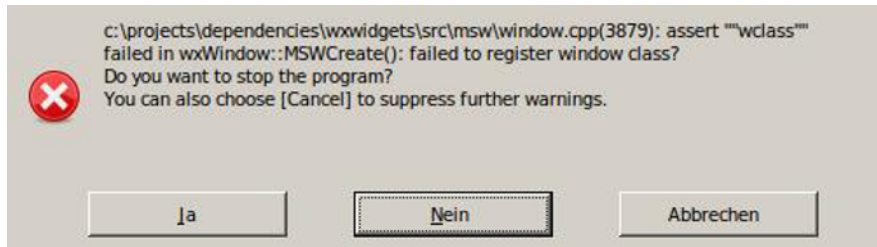


Abbildung 6.12: Diese Warnung müssen Sie mit *Abbrechen* quittieren, bevor `cemu` startet.

Folgende Einstellungen helfen, eine höhere Frame-Rate zu erhalten:

- Options → GPU buffer cache accuracy = low
- CPU → Mode = Dual core recompiler
- Debug → MM timer accuracy = 1 ms
- Debug → Custom timer = QPC

Unter Ubuntu funktioniert der Controller-Support (z. B. PS4-Controller) auf Anhieb. Der Controller muss lediglich per Bluetooth gekoppelt werden. Anschließend kann er dann als Eingabe-Gerät in `cemu` verwendet werden.

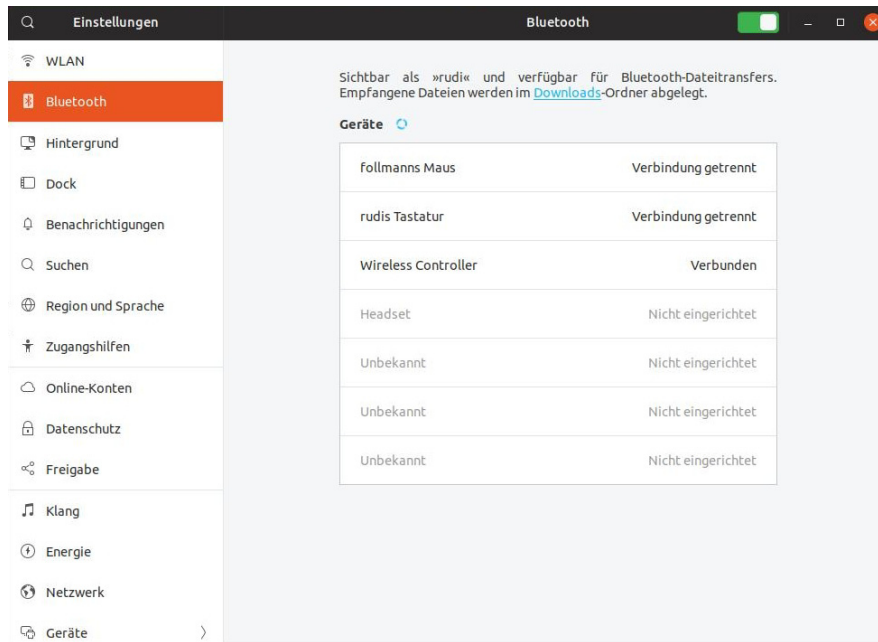


Abbildung 6.13: Unter Ubuntu kann man einen PS4-Controller ohne Probleme per Bluetooth verbinden.

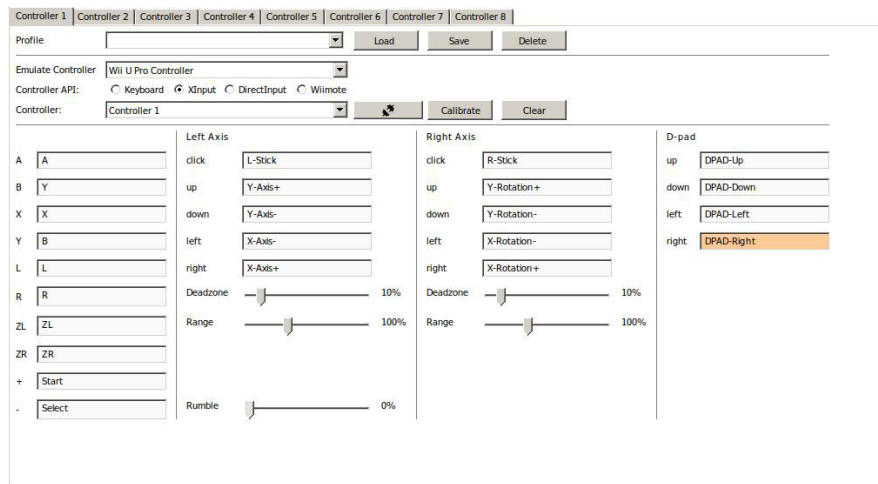


Abbildung 6.14: PS4-Controller-Einstellungen für *cemu* unter *wine*.

Auf meinem Lattepanada Alpha Core m3-Rechner läuft Super Mario 3D World unter Ubuntu mit ca. 30 fps. Manche Szenen im Spiel ruckeln ein wenig. Dennoch ist dieses Spiel auch unter Ubuntu spielbar.

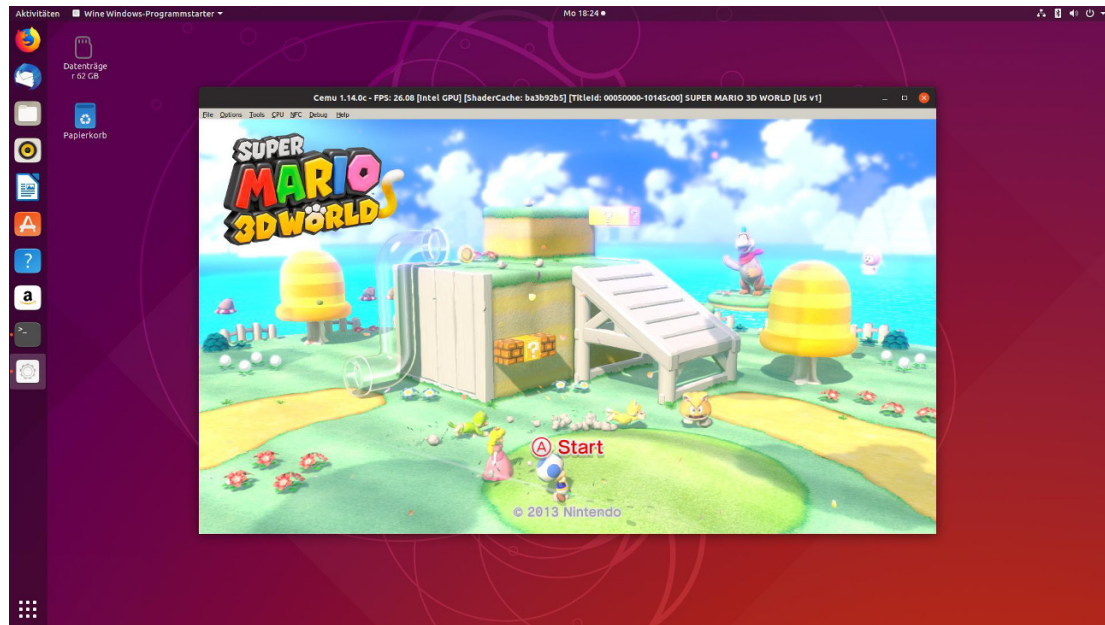


Abbildung 6.15: Super Mario 3D World für *cemu* unter *wine*.



Manche Spiele erfordern ein Update für die Wii U. Dieses Update können Sie ebenfalls für cemu installieren. Sie müssen dann im Internet nicht nur nach der Spiele-ROM suchen, sondern auch nach eventuellen Update-Dateien. Aber auch hier gilt: Das Herunterladen eines Spiele-ROM, ohne dass Sie das Original besitzen, ist illegal.

6.3 Nintendo Switch-Emulator

Darf es noch ein wenig moderner sein? Es gibt sogar einen Emulator für Nintendos aktuelle Spielekonsole, die Switch. Der Emulator *yuzu* ist sogar quelloffen und steht damit für Windows- und LINUX-basierte Systeme zur Verfügung. Der Emulator ist noch nicht so weit wie *cemu*, hat aber in den letzten Monaten rasante Fortschritte gemacht. Um es gleich vorweg zu nehmen: Mein Lattepanda Alpha mit einem Intel Core m3 7Y30-Prozessor und integrierter Intel HD615-Grafik ist zu langsam, um Switch-Spiele sinnvoll zu spielen. Bei schnelleren Prozessoren und schnellerer GPU sieht das sicherlich besser aus. Darüberhinaus gibt es häufige Updates - Geschwindigkeitsschub inklusive. Das ist sicherlich nicht weiter verwunderlich, läuft die Switch doch mit einer NVidia-Maxwell-GPU der zweiten Generation, die 256 Cuda-Kerne bietet und mit einer Frequenz von 1,6 GHz arbeitet. Es bedarf also schon schneller und moderner Hardware, um diese Konsole zu emulieren. *yuzu* selbst ist auf <https://yuzu-emu.org/> zu Hause.



Abbildung 6.16: Super Mario Odyssey unter *yuzu*. Quelle: [https://yuzu-emu.org/\[48\]](https://yuzu-emu.org/[48]).

Dieser Abschnitt erklärt, wie man den Switch-Emulator unter Ubuntu aus den Quellen übersetzt und das Spiel Super Mario Odyssey lädt und startet. Der Quelltext von *yuzu* wird auf Github bereitgestellt. Hier gibt es eine konservative Version des Emulators und eine, die viele Neuerungen beinhaltet. Im weiteren Verlauf dieses Abschnitts wird die aktuelle *yuzu-canary*-Version verwendet. Zunächst werden alle Abhängigkeiten installiert, *yuzu* benötigt.

Code-Ausschnitt 6.12: Installation der Abhängigkeiten für *yuzu*

```
sudo apt-get install git libSDL2-dev qtbase5-dev libqt5opengl5-dev build-essential cmake
```

Wechseln Sie danach bitte in ein Verzeichnis Ihrer Wahl (z. B. `cd -`) und laden Sie *yuzu* so herunter:

Code-Ausschnitt 6.13: Download des *yuzu*-Quellcodes

```
git clone --recursive git://github.com/yuzu-emu/yuzu-canary
```

Der Parameter `--recursive` sorgt dafür, dass alle erforderlichen *git*-Submodule heruntergeladen werden. Anschließend wird *yuzu* so übersetzt:

Code-Ausschnitt 6.14: Übersetzen des *yuzu*-Quellcodes

```
mkdir build && cd build
cmake .. -DCMAKE_BUILD_TYPE=Release
make -j4
```

Der Parameter `-j4` erlaubt das gleichzeitige Übersetzen auf vier CPU-Kernen. Sollte Ihre CPU mehr oder weniger Kerne haben, passen Sie den Parameter bitte an oder lassen Sie ihn ganz weg. Das Übersetzen nimmt ca. 10 Minuten in Anspruch. Das fertig kompilierte Programm finden Sie dann im Verzeichnis *yuzu-canary/build/bin*.

Nach der Installation ist *yuzu* noch nicht in der Lage, Nintendo-Switch-ROMs abzuspielen. Hierzu fehlen verschiedene Key-Dateien, wie *prod.key* oder *title.key* sowie

eine Datei namens *shared_font.bin*. Oft lassen sich diese Dateien im Internet unter dem Namen *SwitchFiles.rar* finden. Das Benutzer-Verzeichnis von *yuzu* finden Sie unter *./local/shagre/yuzu*. Es wird Ihnen auch angezeigt, wenn Sie *yuzu* starten und unter *File* den Eintrag *Open yuzu Folder* wählen. In diesem Verzeichnis müssen Sie ein Unterverzeichnis *keys* erstellen, in das die Schlüsseldateien kopiert werden müssen. Nach dem ersten Start von *yuzu* generiert der Emulator an der oben angegebenen Stelle ein weiteres Verzeichnis namens *sysdata*, in das Sie die Font-Datei kopieren müssen. Nach einem Neustart von *yuzu* werden diese Dateien dann erkannt.

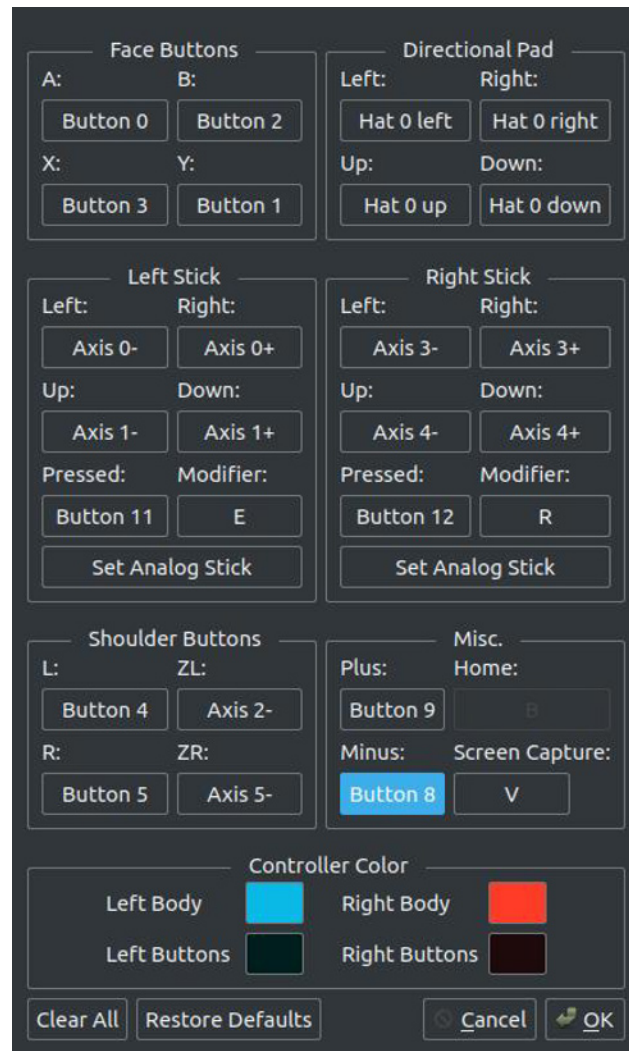


Abbildung 6.17: Auch für den Switch-Emulator *yuzu* können Bluetooth-Controller einfach konfiguriert werden.

Switch-ROMs haben die Endung *.xci*. Neben einigen „Homebrew“-Anwendungen, deren Benutzung absolut legal ist, findet man am Internet auch Abbilder von Spielen, beispielsweise auch das Spiel *Super Mario Odyssey*, das ich auf meiner Nintendo Switch sehr gerne spiele. Auch hier sei noch einmal der Hinweis gestattet, dass das Herunterladen und Benutzen von Spiele-ROMs illegal ist, sollten Sie das Original-Spiel nicht gekauft haben.

Controller können - ähnlich wie bei *cemu* - per Bluetooth mit Ubuntu verbunden werden und dann im Menüpunkt *Emulation* → *Configure* → *Controls* konfiguriert werden. Die Konfiguration einer Bluetooth-Controller funktioniert dabei ohne Probleme. Standardmäßig ist bereits eine Tastatur konfiguriert. Mit der Taste *A* starten Sie ein Spiel, Figuren können mit den Pfeiltasten auf dem Numpad bewegt werden. *yuzu* erlaubt auch das Umstellen der Sprache. Neben Englisch wird unter anderem auch Deutsch angeboten. Im Menüpunkt *File* → *Load File* kann ein ROM geladen werden.

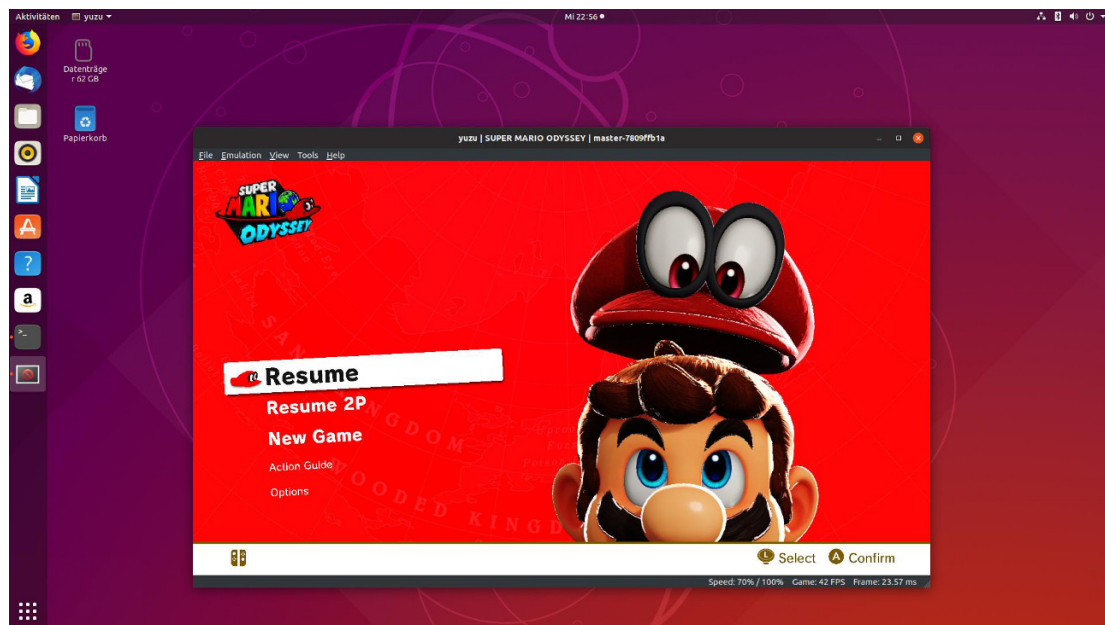


Abbildung 6.18: Super Mario Odyssey mit *yuzu* auf dem Ubuntu-Desktop.

Auf meinem Core m3-Prozessor tröpfelte das Spiel mit nicht spielbaren 4-9 fps vor sich hin. Außerdem gab es häufige Abstürze, so dass der Emulator derzeit noch nicht vernünftig zu gebrauchen ist. Dennoch ist es erstaunlich, was das *yuzu*-Team in so kurzer Zeit geschafft hat. Und der Emulator macht täglich Fortschritte. Daher ist es wahrscheinlich nur eine Frage der Zeit, bis die ersten Spiele vernünftig spielbar sind - eine leistungsstarke Hardware vorausgesetzt.

6.4 Nintendo Gamecube & Wii-Emulator



Der Wii U und Gamecube-Emulator nennt sich *dolphin*. Er hat bereits einen solchen Reifegrad erreicht, dass er einfach über die Ubuntu-Repositories installiert werden kann. Eine Windows-Version ist ebenfalls über die Webseite <https://de.dolphin-emu.org> [49] verfügbar. Der *dolphin*-Quelltext ist offen (<https://github.com/dolphin-emu/dolphin> [50]) und kann sogar für Android kompiliert werden. *dolphin* ist in C++ programmiert. Der Emulator selbst benötigt keine Schlüssel-Dateien, um ein Spiele-ROM zu starten. ROMs finden sich in der Regel als *.iso*-Dateien und wie bereits mehrfach erwähnt, gilt auch hier: Das Herunterladen von Spiele-ROMs für dieses System ist nicht legal. Unter Ubuntu verwechselt man den *dolphin*-Emulator gerne mit dem Dateimanager *dolphin*. Daher hat der Emulator den Namen *dolphin-emu* erhalten und wird so installiert:

Code-Ausschnitt 6.15: Installation des *dolphin*-Emulators

```
sudo apt-get install dolphin-emu
```



Abbildung 6.19: Super Mario Galaxy im *dolphin*-Emulator. Quelle: <http://de.dolphin-emu.org> [49].

Das war es auch schon. Danach steht der Wii-Emulator unter Ubuntu 18.10 in der Versionsnummer 5.0 zum Spielen bereit. Als Super Mario-Fan habe ich ein Abbild meiner Super Mario Galaxy-DVD erstellt (Stichwort Homebrew) und anschließend im Emulator getestet. Dieses Abbild ist knapp 5 GB groß. Wie in den anderen Emulatoren kann man auch im *dolphin*-Emulator einen Controller konfigurieren, der beispielsweise über Bluetooth verbunden ist. Mit einem PS4-Controller funktioniert das ebenfalls einwandfrei. Und wie immer gilt: Im Notfall hilft auch eine Tastatur. Ich habe bei mir ein Profil für den PS4-Controller angelegt, welches unter `~/.config/dolphin-emu/Profiles/Wiimote/PS4.ini` gespeichert wird und so aussieht:

Code-Ausschnitt 6.16: Konfigurationsdatei für den *dolphin*-Emulator

```

[[Profile]
Device = evdev/0/Wireless Controller
Buttons/A = `Button 2`
Buttons/B = `Button 0`
Buttons/1 = `Button 3`
Buttons/2 = `Button 1`
Buttons/- = 0
Buttons/+ = E
Buttons/Home = `!Alt_L` & Return
IR/Up = Cursor Y-
IR/Down = Cursor Y+
IR/Left = Cursor X-
IR/Right = Cursor X+
Shake/X = Click 2
Shake/Y = Click 2
Shake/Z = Click 2
Extension = Nunchuk
Nunchuk/Buttons/C = Control_L
Nunchuk/Buttons/Z = Shift_L
Nunchuk/Stick/Up = `Axis 1-`
Nunchuk/Stick/Down = `Axis 1+`
Nunchuk/Stick/Left = `Axis 0-`
Nunchuk/Stick/Right = `Axis 0+`
D-Pad/Up = Up
D-Pad/Down = Down
D-Pad/Left = Left
D-Pad/Right = Right

```

Beim Spielen erreicht *dolphin* auf meinem Lattepanda Alpha Frame-Raten von 50 fps.

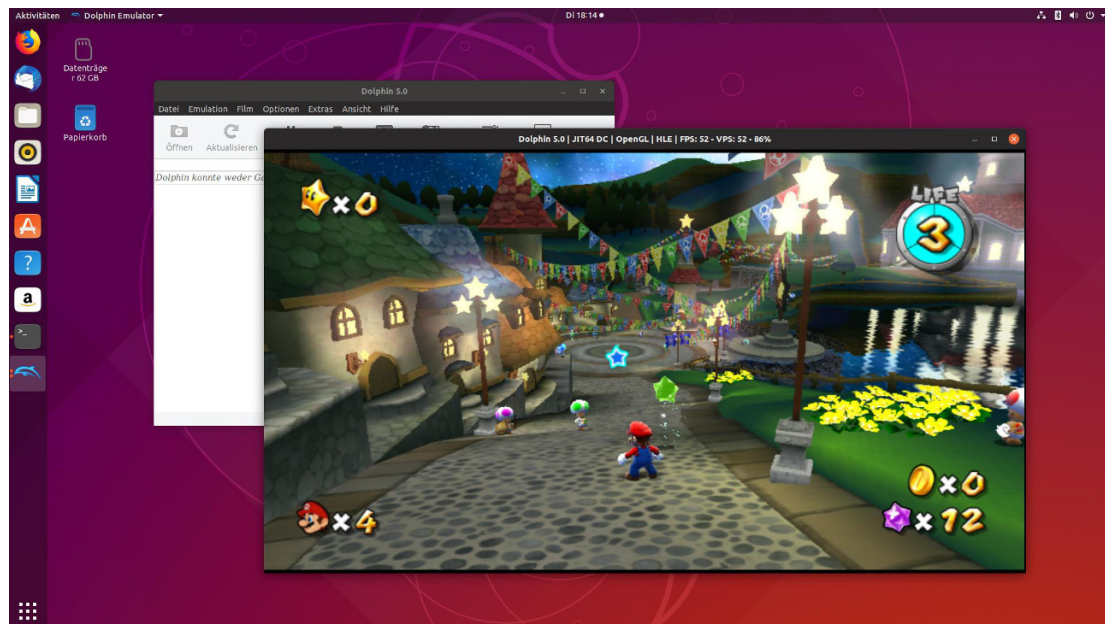


Abbildung 6.20: Der *dolphin*-Emulator auf dem Ubuntu-Desktop.

6.5 N64 Emulator

Kaum eine Spielekonsole ist häufiger emuliert worden als die Nintendo 64 aus dem Jahre 1996. Bereits drei Jahre nach dem Erscheinen dieser Konsole existierte mit UltraHLE ein Emulator, der Spiele wie Super Mario auf relativ bescheidener PC-Hardware problemlos spielbar macht. Einige Zeit später wurde der Quelltext von UltraHLE gestohlen

und tauchte dann im Internet auf. Daraufhin entwickelte sich ein ganzer Reigen an Emulatoren. Der bekannteste unter LINUX ist wohl *mupen64*. Der Code ist so effektiv programmiert, dass sich ein Nintendo 64 sogar auf einem Raspberry Pi emulieren lässt. Selbst nach über 23 Jahren hat dieser Klassiker seinen Charme nicht verloren. Die Grafik kann zwar nicht mit denen aktueller Konsolen mithalten, ist aber immer noch hübsch anzusehen.



Abbildung 6.21: Super Mario 64 ist ein Klassiker aus dem Jahre 1996. Damals servierte Nintendo noch im 4:3-Format.

Unter Ubuntu kann *mupen64plus* so installiert werden:

Code-Ausschnitt 6.17: Installation des *mupen64plus*-Emulators

```
sudo apt-get install mupen64plus-ui
```

Nintendo 64-ROMs sind sehr klein. Super Mario ist gerade einmal 8 MB groß. In der Regel haben diese ROMs die Endung *.n64*. Ein Schlüssel ist für diesen Emulator nicht erforderlich. Wenn der Emulator zum ersten Mal gestartet wird, kann es durchaus sein, dass das Bild flimmert und klein ist. Das liegt daran, dass er noch nicht konfiguriert ist. Falls Sie das ausprobieren wollen, es geht so:

Code-Ausschnitt 6.18: Start des *mupen64plus*-Emulators

```
mupen64plus name_des_roms
```

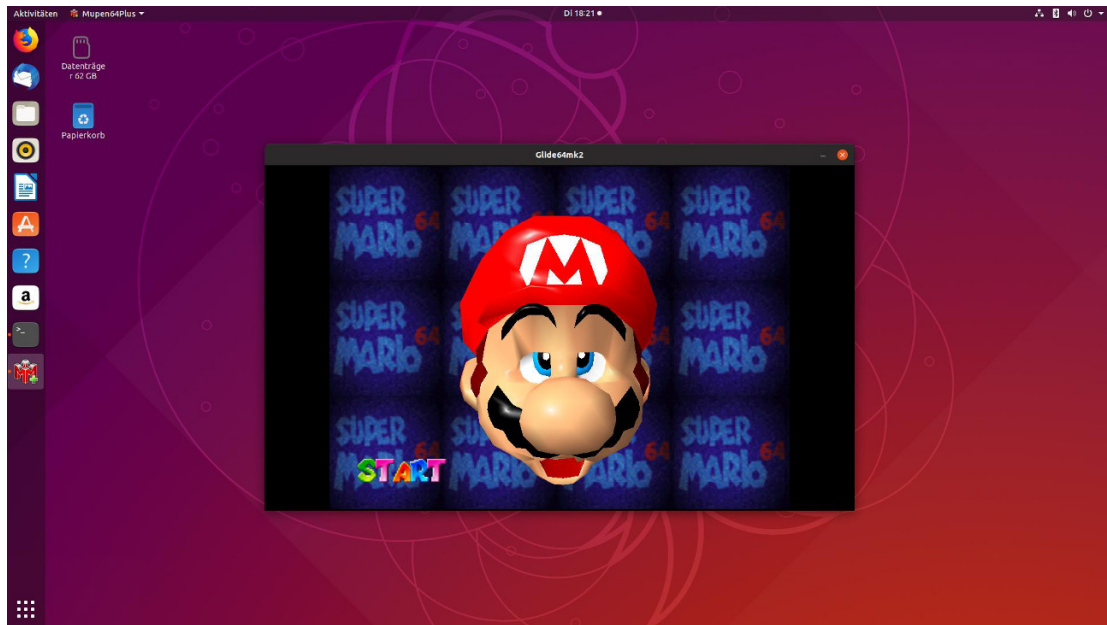


Abbildung 6.22: Der Nintendo 64-Emulator *mupen64plus* auf dem Ubuntu-Desktop.

Die komplette Konfiguration kann über Parameter erfolgen. Möchte man z. B. das Grafik-Plugin *glide64* im Vollbildmodus benutzen, sieht der Aufruf folgendermaßen aus:

Code-Ausschnitt 6.19: Start des *mupen64plus*-Emulators mit Parametern

```
mupen64plus --fullscreen --gfx mupen64plus-video-glide64 name_des_roms
```

Bitte achten Sie bei beiden oben genannten Aufrufen darauf, den korrekten Pfad zum Spiele-ROM anzugeben. Der Befehl

Code-Ausschnitt 6.20: Auflisten aller Parameter des *mupen64plus*-Emulators

```
mupen64plus --help
```

zeigt Ihnen alle Parameter, die *mupen64plus* zu bieten hat. *mupen64plus* speichert übrigens alle Parameter in der Datei `~/.config/mupen64plus/mupen64plus.cfg`.

Glücklicherweise gibt es mit *M64Py* eine grafische Oberfläche für *mupen64plus*, die das Einstellen der umfangreichen Optionen erträglich macht. Diese Oberfläche ist in Python programmiert und steht auf *Github* bereit. Bevor der Code erfolgreich ausgeführt werden kann, müssen einige Abhängigkeiten installiert werden.

Code-Ausschnitt 6.21: Installation der Abhängigkeiten für *M64Py*

```
sudo apt-get install python-pyqt5 pyqt5-dev-tools python-pyqt5.qtopengl libSDL2-dev python-pip <-> python-sdl2
```

Anschließend kann der Code in einem Verzeichnis Ihrer Wahl (z. B. im *home*-Verzeichnis) ausgecheckt werden.

Code-Ausschnitt 6.22: Installation von M64Py

```
cd ~  
git clone git://github.com/mupen64plus/mupen64plus-ui-python
```

Er wird so übersetzt:

Code-Ausschnitt 6.23: Übersetzen von M64Py

```
cd mupen64plus-ui-python/  
python setup.py build  
python setup.py install --user
```

Das ausführbare Skript steht anschließend unter `~/.local/bin/m64py` zur Verfügung und kann so

Code-Ausschnitt 6.24: Start von M64Py

```
~/local/bin/m64py
```

gestartet werden. Den Pfad vor `m64py` können Sie auch weglassen.



Abbildung 6.23: Zelda in der Oberfläche von M64Py. Quelle: <http://m64py.sourceforge.net/> [51].

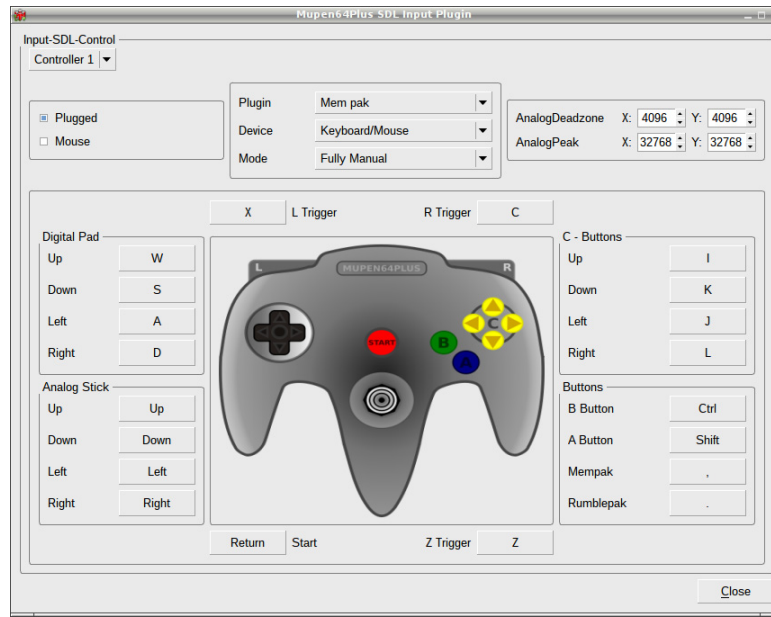


Abbildung 6.24: Einstellung des Controllers in der Oberfläche von *M64Py*.

Dank der *M64Py*-Oberfläche ist das Anlernen eines Controllers oder einer Tastatur kinderleicht. Sollten Sie mit einem Grafik-Plugin Schwierigkeiten haben, beispielsweise weil das Bild flimmert, probieren Sie einfach ein anderes.

Nintendo 64-Spiele lassen sich flüssig spielen, selbst auf alter Hardware.

6.6 LINUX Spiele

In diesem Abschnitt werden wir noch einen Blick auf native LINUX-Spiele werfen. Es gibt nicht viele professionelle Spiele-Hersteller, die Spiele für LINUX anbieten. Das liegt daran, dass LINUX lange ein Nischen-Dasein geführt hat und Mainstream-Betriebssysteme wie Windows mit DirectX Grafikschnittstellen anbieten, die so unter LINUX nicht verfügbar sind. Das ist unter anderem der Grund, warum viele Programmierer von Emulatoren *OpenGL* bevorzugen, damit diese Programme eben auf möglichst vielen Betriebssystemen funktionieren.

Native Spiele finden Sie mit dem Programm Ubuntu-Software auf Ihrem Desktop. Spiele können entweder von dort aus oder mit Hilfe eines Terminals installiert werden. Als ein Beispiel sei hier *supertuxkart* erwähnt. Das Spiel erinnert ein wenig an Mario Kart und kann so installiert werden:

Code-Ausschnitt 6.25: Installation von *supertuxkart*

```
sudo apt-get install tuxkart
```

Ein Aufruf von

Code-Ausschnitt 6.26: Parameter von *supertuxkart*

```
supertuxkart --help
```

zeigt alle Parameter an, die man diesem Spiel übergeben kann. So sorgt ein *-f* hinter dem Aufruf dafür, dass das Spiel im Vollbildmodus startet.



Abbildung 6.25: Supertuxkart erinnert ein wenig an Mario Kart.

Ein über Bluetooth angeschlossener Spiele-Controller funktioniert auf Anhieb. Die Grafik ist sicherlich nicht mit professionellen Spielen vergleichbar, macht aber dennoch Spaß.

Für *wine* gibt es inzwischen auch eine DirectX-Schnittstelle (*dxvk*, DirectX Vulkan). Wie der Name schon sagt, werden Vulkan-Treiber benötigt und der Erfolg variiert von GPU zu GPU. Einige Windows-Spiele wie Fortnite laufen nicht unter *wine*, da Epic ein sogenanntes Anti-Cheat-System eingeführt hat, das die Verwendung unter *wine* derzeit nicht erlaubt. Andere Windows-Spiele laufen mit Hilfe von *wine* und *dxvk* problemlos. Ich möchte Ihnen an dieser Stelle noch zwei Möglichkeiten vorstellen, Windows-Spiele unter LINUX zu installieren: *lutris* und *winepak*.

lutris (<https://lutris.net/> [52]) ist eine Open-Source Spiele-Plattform für LINUX. Spiele können installiert werden, ohne dass sich der Benutzer um die Details der Installation kümmern muss. Dabei können Spiele von verschiedenen Anbietern wie Steam oder Origin bezogen werden. Unter Ubuntu wird *lutris* wie folgt installiert:

Code-Ausschnitt 6.27: Installation von *lutris*

```
ver=$(lsb_release -sr); if [ $ver != "18.10" -a $ver != "18.04" -a $ver != "16.04" ]; then ver=
=18.04; fi
echo "deb http://download.opensuse.org/repositories/home:/strycore/xUbuntu_$ver/ ./" | sudo tee -
/etc/apt/sources.list.d/lutris.list
wget -q https://download.opensuse.org/repositories/home:/strycore/xUbuntu_$ver/Release.key -O- <
| sudo apt-key add -
sudo apt-get update
sudo apt-get install lutris
```

Ein anschließender Aufruf von

Code-Ausschnitt 6.28: Start von *lutris*

```
lutris
```


startet die Spiele-Plattform. *lutris* selbst setzt die Installation von *wine*-Staging voraus. Diese ist weiter oben beschrieben. Prinzipiell laufen damit sogar Spiele wie Fifa 19, allerdings berichten Benutzer unter anderem von Problemen im Online-Modus.

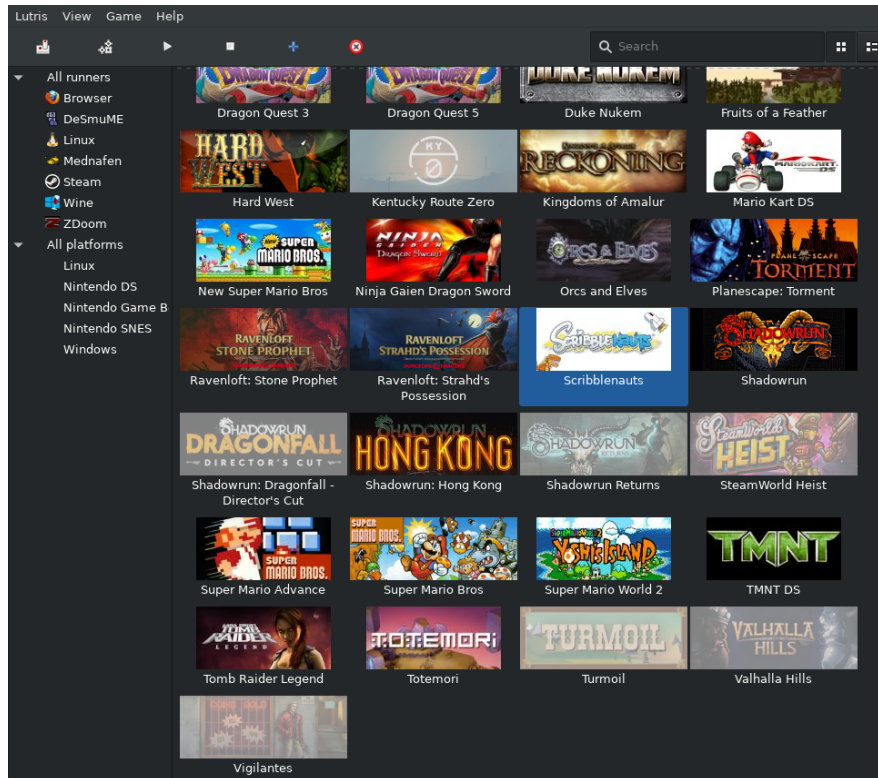


Abbildung 6.26: *lutris* ist eine Open-Source-Spieleplattform für LINUX. Quelle: <https://lutris.net/> [52].

Eine weitere Möglichkeit der Installation von Spielen bietet *flatpak*. *flatpak* ist ein Paket-Management-System für LINUX, das plattformübergreifend funktioniert (<https://flatpak.org/> [53]) und so installiert werden kann:

Code-Ausschnitt 6.29: Installation von *flatpak*

```
sudo apt install flatpak
sudo apt install gnome-software-plugin-flatpak
flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo
```

Um die Installation abzuschließen, ist ein Neustart des Systems erforderlich.

Code-Ausschnitt 6.30: Neustart des Systems

```
sudo reboot
```

winepak ist ein Repository, das neben einigen anderen Windows-basierten Programmen wie Spotify auch Spiele für LINUX anbietet, die unter Verwendung von *wine* laufen. *winepak* wird so installiert:

Code-Ausschnitt 6.31: Installation von *winepak*

```
flatpak remote-add --if-not-exists winepak https://dl.winepak.org/repo/winepak.flatpakrepo
```

Eine *winepak*-Anwendung wird so installiert:

Code-Ausschnitt 6.32: Installation einer *winepak*-Anwendung

```
flatpak install winepak tld.domain.Application
```

Möchten Sie also beispielsweise das Spiel „Overwatch“ installieren, lautet der Aufruf wie folgt:

Code-Ausschnitt 6.33: Installation des Spiels Overwatch

```
flatpak install winepak com.blizzard.Overwatch
\begin{lstlisting}
Das Spiel kann dann anschließend so gestartet werden:
\begin{lstlisting}[caption=Start des Spiels Overwatch]
flatpak run com.blizzard.Overwatch
\begin{lstlisting}
Ein Aufruf von
\begin{lstlisting}[caption=De-Installation des Spiels Overwatch]
flatpak uninstall com.blizzard.Overwatch
\begin{lstlisting}
entfernt das Spiel wieder rückstandslos.
Möchten Sie alle Programme auflisten, die sich im \textit{winepak}-Repository befinden, ist der ↔
folgende Aufruf behilflich:
\begin{lstlisting}[caption=Auflistung aller Programme im \textit{winepak}-Repository]
flatpak remote-ls winepak
```



Häufig ist es erforderlich, unter wine zusätzliche Programme wie das Visual-Studio C++-Runtime Environment zu installieren. Ein Aufruf von

Code-Ausschnitt 6.34: De-Installation von *wine*-Programmen

```
wine uninstaller
```

erlaubt es Ihnen, diese Programme im Bedarfsfall wieder zu deinstallieren.

Zusammenfassung und Ausblick 6 Nach dem Spielspaß bauen wir unseren Wohnzimmer-PC nun weiter zum Multimedia-Center aus. Lernen Sie im nächsten Kapitel, wie Sie Schritt für Schritt einen Receiver für digitales Fernsehprogramm aufbauen können, der in der Lage ist, UHD- und HD-Inhalte wiederzugeben.

Dazu werden wir Klaus Schmidingers *VDR* aus den Quelltexten heraus installieren und mit verschiedenen Plugins so erweitern, dass der Receiver Marke Eigenbau selbst professionellen Geräten in nichts nachsteht.

Individuelle Anpassungen und Erweiterungen erlauben Ihnen dabei, das Aussehen der Bedienoberfläche an Ihren eigenen Geschmack anzupassen. ■

Literaturverzeichnis

- [42] Nintendo. *Super Mario Run*. 2019.
<https://supermariorun.com/de/index.html>
besucht am 27. 12. 2019 (siehe Seite 111).
- [43] Cemu - Wii U Emulator. *Experimental software to emulate Wii U applications on PC*. 2019.
<http://cemu.info>
besucht am 27. 12. 2019 (siehe Seite 111).
- [44] Woo. *Use your PS4 Controller on your PC to its full potential*. 2019.
<http://ds4windows>
besucht am 27. 12. 2019 (siehe Seite 113).
- [45] WineHQ. *Wine Download Server*. 2019.
<https://dl.winehq.org/>
besucht am 27. 12. 2019 (siehe Seite 116).
- [46] Cemu hook. *Current versions*. 2019.
<https://cemuhook.sshnuke.net/#Downloads>
besucht am 27. 12. 2019 (siehe Seite 119).
- [47] Cemu Graphic Packs. *Official website of the Community Graphic Packs for Cemu, used to upscale and change your Wii U games*. 2019.
https://slashiee.github.io/cemu_graphic_packs/
besucht am 27. 12. 2019 (siehe Seite 119).
- [48] yuzu emulator team. *Nintendo Switch Emulator*. 2019.
<https://yuzu-emu.org/>
besucht am 27. 12. 2019 (siehe Seite 123).
- [49] Dolphin Emulator Project. *Dolphin Emulator*. 2019.
<http://de.dolphin-emu.org>
besucht am 27. 12. 2019 (siehe Seite 126).
- [50] Github. *Dolphin Quelltext*. 2019.
<https://github.com/dolphin-emu/dolphin>
besucht am 27. 12. 2019 (siehe Seite 126).
- [51] Sourceforge. *A frontend for Mupen64Plus*. 2019.
<http://m64py.sourceforge.net/>
besucht am 27. 12. 2019 (siehe Seite 130).
- [52] Lutris. *Play all your games on Linux*. 2019.
<https://lutris.net/>
besucht am 27. 12. 2019 (siehe Seiten 132, 133).
- [53] Flatpak. *The Future of Apps On Linux*. 2019.
<https://flatpak.org/>
besucht am 27. 12. 2019 (siehe Seite 133).

7. Video Disk Recorder



heute

ZDF SPORTreportage

STR: 99% SNR: 84%

7.1	DVB-Sticks	138
7.2	Fernbedienung mit (F)lirc	145
7.3	VDR	152
7.4	Skripte	161
7.5	Speichern und servieren	166
7.6	Schöner fernsehen	169
7.7	Stream me up!	194
7.8	Pay-TV	196
7.9	Wichtige Hinweise	199

Das Kapitel erklärt Schritt für Schritt den Aufbau eines DVB-Receivers (Digital Video Broadcast). Dabei spielt es keine Rolle, ob Sie Fernsehen über Satellit, über Kabel oder terrestrisch empfangen. Wir werden Klaus Schmidingers VDR (Video Disk Recorder) installieren, zusammen mit einer Infrarot-Fernbedienung und einem USB-DVB-Empfänger. Ich werde Ihnen zeigen, wie der Video-Disc-Recorder VDR mit Plugins erweitert werden kann, die das Aussehen verbessern oder Streaming des Fernseh-Programms auf ein mobiles Gerät ermöglichen. Ein Web-Frontend erlaubt es sogar, Aufnahme-Timer aus der Ferne zu setzen.



Abbildung 7.1: Das VDR-Portal ist die erste Anlaufadresse für Fragen rund um den VDR (Quelle: Jan Grell).

Der wohl bekannteste Empfänger für digitales Fernsehen ist der VDR von Klaus Schmidinger. Klaus selbst stellt ihn auf seiner Webseite <http://tvdr.de> [54] vor, es gibt ein VDR-Wiki (<http://www.vdr-wiki.de> [55]) und sogar ein sehr aktives Forum (<http://www.vdr-portal.de> [56]) zu diesem Thema. VDR ist einfach zu bedienen. Er besitzt ein OSD (On Screen Display), kann mit DiSEqC (Digital Satellite Equipment Control) umgehen, hat eine Video-Schnittfunktion und ist netzwerkfähig. Er kann durch Plugins sehr einfach erweitert werden und sogar mit verschlüsseltem Fernsehprogramm umgehen. In diesem Kapitel richten wir eine Fernbedienung für

unseren Mediacenter-PC ein, installieren einen DVB-S-Stick für Satellitenfernsehempfang und verbinden beides mit dem VDR. Für kabelgebundenen oder terrestrischen Empfang gehen Sie bitte genauso vor. Danach haben Sie einen Satellitenreceiver, der das Fernsehprogramm auf eine externe Festplatte aufzeichnen kann. Bleibt die Frage, warum wir keine fertige Distribution für den PC nehmen, etwa die MLD (Mini LINUX Distribution), die unter <http://www.minidvblinux.de> [57] verfügbar ist. Wenn Sie so schnell wie möglich fernsehen möchten, ist das in Ordnung. Wenn Sie aber das System selbstständig erweitern möchten, ist dieses Kapitel genau das richtige für Sie.

7.1 DVB-Sticks

Eine Liste mit DVB-Sticks, die unter LINUX funktionieren, finden Sie unter <http://www.vdr-wiki.de> [55]. Im Prinzip gibt es mehrere Möglichkeiten, einen DVB USB-Stick unter LINUX einzubinden:

1. Sie erhalten den Treiber direkt vom Hersteller. Dieser liefert auch ein Installationsprogramm für LINUX. Ein Beispiel hierfür sind die Sundtek Sticks. Im Folgenden werden wir uns die Installation für einen Sundtek SkyTV Ultimate näher anschauen.
2. Sie kaufen einen Stick, dessen Treiber-Support bereits im Kernel enthalten ist, sie müssen aber die Firmware des Sticks installieren. Ein Beispiel hierfür ist der TeVii S2-660 Stick. Auch seine Installation wird gleich erklärt.
3. Sie kaufen einen Stick, der zwar von LINUX unterstützt wird, dessen Treiber aber noch nicht im Kernel vorhanden ist. Auch diese Installation wird beschrieben.

Wie auch immer Sie sich entscheiden, ich empfehle den Kauf eines DVB2-fähigen Sticks. Moderne PCs sind durchaus in der Lage, UHD-auflösendes Fernsehen anzuzeigen.

7.1.1 Sundtek

Die Firma Sundtek liefert für Ihre DVB USB-Geräte komplett eigene Treiber mit, deren Installation sich sehr einfach gestaltet. Weiterhin läuft der Treiber auf vielen Embedded Boards, wie dem Raspberry Pi oder einem Odroid, selbst auf einigen NAS-Laufwerken. Natürlich lässt er sich auch auf einem PC installieren. Die Treiber für die Sundtek Sticks werden unter http://www.sundtek.de/media/sundtek_netinst.sh [58] zum Download angeboten. Da die Treiber als Systemadministrator installiert werden müssen, verschaffen wir uns zunächst *root*-Rechte. Dann wechseln wir in das temporäre Verzeichnis und laden den Treiber herunter. Beim Treiber selbst handelt es sich um ein Shell-Skript. Wir machen dieses ausführbar für alle (*chmod 777 sundtek_netinst.sh*) und rufen es dann auf. Das Skript lädt die passenden Treiberbestandteile von der Webseite des Herstellers und bindet sie so in das Betriebssystem ein, dass die Treiber selbst nach einem Neustart automatisch geladen werden.



Abbildung 7.2: Der Sundtek-Stick bringt seine eigenen Treiber mit (Quelle: Sundtek).

Code-Ausschnitt 7.1: Die Installation eines Sundtek DVB-Sticks

```
sudo su
cd /tmp
wget http://www.sundtek.de/media/sundtek_netinst.sh
chmod 777 sundtek_netinst.sh
./sundtek_netinst.sh
```

Die Treiberinstallation fordert Sie auf zu bestätigen, dass Sie fortfahren wollen. Machen Sie das. Der Treiber selbst wird im Verzeichnis */opt* installiert. Eine Liste der angeschlossenen Sticks erhalten Sie mit dem Kommando

Code-Ausschnitt 7.2: Auflisten aller angeschlossene Sundtek-Sticks

```
/opt/bin/mediaclient -e
```

Bei mir zeigt der Befehl Folgendes an:

Code-Ausschnitt 7.3: Ausgabe des Befehls zum Auflisten aller Sundtek-Sticks

```
device 0: [Sundtek SkyTV Ultimate III (USB 2.0)] DVB-S/S2, REMOTE-CONTROL
[BUS]:
  ID: 1-1.2
[SERIAL]:
  ID: U120420173106
[DVB-S/S2]:
  FRONTEND: /dev/dvb/adapter0/frontend0
  DVR: /dev/dvb/adapter0/dvr0
  DMX: /dev/dvb/adapter0/demux0
[REMOTECONTROL]:
  INPUT0: /dev/mediainput0
```

Für Embedded Boards empfiehlt Sundtek, die PID (*Packet Identifier*) Hardware-Filter zu aktivieren. Dies kann mit dem Befehl

Code-Ausschnitt 7.4: Einschalten der Hardware-Filter für Sundtek-Sticks

```
/opt/bin/mediaclient -P on
```

erledigt werden. Tauschen sie das *on* gegen ein *off*, schalten Sie den PID Hardware-Filter wieder aus. Auf einem PC spielt diese Einstellung aber keine Rolle. Alle Einstellungen für die Sundtek-Sticks können in der Datei */etc/sundtek.conf* dauerhaft gespeichert werden.

Code-Ausschnitt 7.5: Einstellungen in der Datei *sundtek.conf*

```
# ----- GLOBAL SECTION -----
#Set loglevel for logging to /var/log/mediasrv.log
#loglevel=[off|max] #default: min
#max .. little bit more debug
#Enable listening on network
#enablenetwork=[on|off] #default: off
#Lowest adapter number to start with, e.g. /dev/dvb/adapter5/frontend0
#first_adapter=5
#Call attach script when new device appears
#device_attach=[PATH_TO_SCRIPT] [PARAMETER|DEVID] #"DEVID" will
#automatically be replaced with the device ID
#Call detach script when device disappears
#device_detach=[PATH_TO_SCRIPT] [PARAMETER|DEVID] #"DEVID" will
#automatically be replaced with the device ID
# ----- Section for adapter with [SERIALNUMBER] -----
#Get adapter serial number with /opt/bin/mediaclient -e
#[SERIALNUMBER]
#Description register as dreambox tuner
#dreambox_support_fel=[on|off] #default: off
#Infrared protocol to use
#ir_protocol=[RC5|NEC|RC6] #default: NEC
#Keymap to use, e.g. "/lib/udev/rc_keymaps/vp702x"
#rcmap=[PATH_TO_KEYMAP] #default: keymap which comes with
#the device
#Choose initial DVB mode for hybrid DVB-T/DVB-C devices only
#initial_dvb_mode=[DVBC|DVBT]
#Call attach script when new device appears
#device_attach=[PATH_TO_SCRIPT] [PARAMETER|DEVID] #"DEVID" will
#automatically be replaced with the device ID
#Call detach script when device disappears
#device_detach=[PATH_TO_SCRIPT] [PARAMETER|DEVID] #"DEVID" will
#automatically be replaced with the device ID
#Volume level
#volume=[0-127] #default: 118
# WSS callback (see http://en.wikipedia.org/wiki/Widescreen\_signaling)
# trigger script when the videoformat changes between 16:9 or 4:3
#wss_callback=[scriptname]
# WSS_4_3_FULL
# WSS_14_9_LETTERBOX_CENTRE
# WSS_14_9_LETTERBOX_TOP
# WSS_16_9_LETTERBOX_CENTRE
# WSS_16_9_LETTERBOX_TOP
# WSS_16_9_LETTERBOX_DEEPER
# WSS_14_9_FULL_HEIGHT_4_3
# WSS_16_9_FULL_HEIGHT_16_9
# WSS_UNABLE_TO_DEMODULATE
#Timeout in ms after that WSS_UNABLE_TO_DEMODULATE will be called
#wss_demodulation_timeout=[MILLISECONDS]
```

Sundtek-Sticks haben einen Infrarot-Empfänger an Bord. Dieser steht unter */dev/input/...* zur Verfügung. Wir benutzen aber die *Flirc*-Methode, so dass Sie den Infrarot-Empfänger im Sundtek-Stick ohne Probleme ausschalten können. Das kann entweder bei der Installation durch Übergabe des Parameters *-nolirc* geschehen oder nach der Installation durch Aufruf von

Code-Ausschnitt 7.6: Deaktivieren des Sundtek-Infrarotempfängers

```
/opt/bin/mediaclient --disablirc
```

oder durch einen Eintrag in der Datei */etc/sundtek.conf*:

Code-Ausschnitt 7.7: Deaktivieren des Sundtek-Infrarotempfängers durch Eintrag in der Konfigurationsdatei

```
ir_disabled=1
```

Der LINUX-Treiber der Sundtek USB-Sticks unterstützt das Streamen von DVB-S/S2 über ein Netzwerk auf andere LINUX- oder Windows-PCs. Um die Netzwerkunterstützung einzuschalten, geben Sie bitte den Befehl

Code-Ausschnitt 7.8: Einschalten des Streamings für Sundtek-Sticks

```
/opt/bin/mediaclient --enablenetwork=on
```

ein. Tauschen Sie das *on* gegen ein *off*, um die Unterstützung wieder zu deaktivieren. Mit dem Befehl

Code-Ausschnitt 7.9: Bereitstellen eines Sundtek-Sticks für einen anderen Rechner

```
/opt/bin/mediaclient --mount=ip-adresse
```

stellen Sie den Stick dem Rechner mit der IP-Adresse *ip-adresse* zur Verfügung. Die Weitergabe des Sticks an einen Windows-PC ist ebenfalls möglich. Eine entsprechende Windows-Software kann man sich bei Sundtek herunterladen. Der Sundtek DVB-S2 Stick kostet übrigens 89 €. Um zu testen, ob der Stick richtig angesprochen werden kann, tunen Sie doch einfach mal auf einen Sender.

Code-Ausschnitt 7.10: Tunen eines Senders mit dem Sundtek DVB-S2-Stick

```
/opt/bin/mediaclient -m DVBS2 -f 1234000 -M QPSK -S 22000000 -V H -E 3/4
```

Die Antwort des Systems lautet:

Code-Ausschnitt 7.11: Systemantwort nach dem Tunen eines Senders mit dem Sundtek DVB-S2-Stick

```
Using device: /dev/dvb/adapter0/frontend0
Setting DVB-S/S2 tune Parameters
using real frequency: 1234000
Modulation: QPSK (DVB-S)
Symbolrate: 22000000
Voltage: 18 Volt (Horizontal)
FEC: 3/4
Tone OFF
Frequency: 1234000
Syntax OK
Checking for lock:
.....
```

Hieran sehen wir, dass ein korrekt installierter DVB-Treiber unter LINUX neue Geräte (*devices*) angelegt hat. Das DVB-Device befindet sich unter */dev/dvb/adapter0*. Auf dieses Device greift später der VDR zu.

Der Sundtek USB-Stick hat ein eigenes Netzteil, welches die Spannung für einen Multiswitch zur Verfügung stellen kann.



7.1.2 DVBSky S960 V2



Abbildung 7.3: Die DVBSky S960 V2-Box benötigt eine Firmware-Datei (Quelle: amazon.de).

DVBSky S960 V2 (<http://dvbsky.net> [59]) ist eine USB-Box, die ein wenig größer ist als der Sundtek-Stick. Dafür ist sie aber auch preiswerter. Die Box stellt ebenfalls ein Infrarot-Modul zur Verfügung, welches unter `(/dev/input/...)` zu finden ist. Der Treiber ist meistens bereits im Kernel integriert und die Box wird beim Bootvorgang korrekt erkannt. Falls die Karte nicht erkannt wird, muss der Media-Kernel-Zweig, wie im nächsten Abschnitt beschrieben wird, installiert werden. Es fehlt lediglich noch die Firmware. Firmware ist eine Software, die während des Boot-Vorganges auf die Box geladen wird und deren Funktionalität sicherstellt. Dadurch hat der Hersteller die Möglichkeit, nachträgliche Verbesserungen für den Betrieb der Box in Form einer neuen Firmware herauszugeben. Die Firmware für diese Box erhalten Sie auf der Hersteller-Webseite unter <http://www.dvbsky.net/download/linux/dvbsky-firmware.tar.gz> [59]. Bitte laden Sie die Firmware herunter. Sie können dazu den `wget`-Befehl verwenden oder den Download aus einem Browser heraus starten. Nach dem Herunterladen muss die Firmware ausgepackt werden. Diese liegt im `tar.gz`-Format vor und kann mit dem Programm `tar` entpackt werden. Das Programm ist in der Ubuntu-Distribution bereits vorinstalliert. Entpacken Sie die Firmware durch den Aufruf von

Code-Ausschnitt 7.12: Entpacken der DVBSky-Firmware

```
tar xvfz dvbsky-firmware.tar.gz
```

Die eigentliche Firmware-Datei `dvb-usb-s660.fw` kopieren Sie bitte in das LINUX Firmware-Verzeichnis, welches sich unter `/lib/firmware` befindet. Dazu benötigen Sie *root*-Rechte.

Code-Ausschnitt 7.13: Kopieren der DVBSky-Firmware in das Firmware-Verzeichnis

```
cd dvbsky-firmware
sudo ./copy-firmware.sh
```

Das kleine Skript `copy-firmware.sh` erledigt den Kopiervorgang für Sie und sorgt dafür, dass die Firmware-Dateien an ihren Bestimmungsort gelangen. Damit die Firmware-Dateien geladen werden können, ist ein Neustart erforderlich. Während des Bootens erkennen Sie die Karte in den Kernel-Logs (`dmesg`) durch folgende Einträge:

Code-Ausschnitt 7.14: Bootmeldungen des DVBSky-Sticks

```

usb 1-1.1.2: new high-speed USB device number 10 using dwc_otg
usb 1-1.1.2: New USB device found, idVendor=0572, idProduct=6831
usb 1-1.1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-1.1.2: Product: S960
usb 1-1.1.2: Manufacturer: Bestunar
usb 1-1.1.2: SerialNumber: 20120511
usb 1-1.1.2: dvb_usb_v2: found a 'DVBSky S960/S860' in warm state
[usb 1-1.1.2: dvb_usb_v2: will pass the complete MPEG2 transport stream to the software demuxer
dvbdev: DVB: registering new adapter (DVBSky S960/S860)
usb 1-1.1.2: dvb_usb_v2: MAC address: 00:17:42:54:96:0c
i2c i2c-3: Added multiplexed i2c bus 4
ts2020 4-0060: Montage Technology TS2022 successfully identified
usb 1-1.1.2: DVB: registering adapter 0 frontend 0 (Montage Technology M88DS3103)...
Registered IR keymap rc-dvbsky
rc rc0: DVBSky S960/S860 as /devices/platform/soc/3f980000.usb/usb1/1-1/1-1.1/1-1.1.2/rc/rc0
input: DVBSky S960/S860 as /devices/platform/soc/3f980000.usb/usb1/1-1/1-1.1/1-1.1.2/rc/rc0/↵
input2
usb 1-1.1.2: dvb_usb_v2: schedule remote query interval to 300 msecs
usb 1-1.1.2: dvb_usb_v2: 'DVBSky S960/S860' successfully initialized and connected
usbcore: registered new interface driver dvb_usb_dvbsky

```

Die Boot-Zeiten habe ich aus Gründen der Übersichtlichkeit entfernt. Nach der erfolgreichen Initialisierung der DVBSky-Box steht auch hier das neue Device `/dev/dvb/adapater0` zur Verfügung, welches VDR zum Empfang der Kanäle benötigt.

Eventuell müssen Sie die USB DVB-Karte noch einmal entfernen und wieder neu verbinden, damit der Treiber die passende Firmware laden kann und der gewünschte Adapter danach zur Verfügung steht.



7.1.3 Neuere Modelle

Sollten Sie sich für einen USB-Empfänger entschieden haben, für den es noch keine Unterstützung im LINUX-Kernel gibt, könnten Sie mit den LINUX-DVB-Treibern Glück haben, welche Sie unter <http://www.linuxtv.org> [60] erhalten. Die DVB-Treiber stellen einen ausgelagerten Kernel-Zweig dar. Nach ihrer Übersetzung stehen Kernel-Module neuerer Treiber zur Verfügung, die noch keinen Einzug in den offiziellen Kernel erfahren haben. Bevor das Übersetzen starten kann, müssen noch einige Pakete installiert werden.

Code-Ausschnitt 7.15: Installation der Abhängigkeiten zum Übersetzen der DVB-Treiber

```

sudo apt-get install linux-source linux-headers make gcc git-core patch patchutils libproc-processtable-perl

```

Falls Sie bei apt-get install ein Paket angeben, das Sie bereits zuvor installiert haben, ist das nicht weiter schlimm. Sie erhalten dann lediglich eine Rückmeldung, dass das bestimmte Programm bereits installiert ist.



Die meisten Pakete, die zur Installation benötigt werden, haben Sie bereits im vorigen Kapitel kennengelernt. Neu hinzugekommen sind die Header- und Quellcode-Dateien des Kernels, die Patchutils (Hilfsprogramme für das Arbeiten mit *patch*) und ein Perl-Programm, um Tabellen zu prozessieren. Machen wir uns an das Auschecken und Übersetzen der Treiber. Dazu öffnen wir ein Terminal und geben die folgenden Befehle ein:

Code-Ausschnitt 7.16: Auschecken der LINUX-Treiber

```
git clone --depth=1 git://linuxtv.org/media_build.git
cd media\_build
./build
```

Den Parameter `--depth=1` können Sie weglassen, falls die komplette Historie ausgecheckt werden soll. Das Starten der Übersetzung wird mit `./build` im Verzeichnis `media_build` initiiert.



Dieser Hinweis beschreibt gängige Makefile-Optionen, wie sie bei fast allen Programmen zu finden sind.

- `sudo make install`: Installiert das übersetzte Programm. Default-Einstellung ist meist die Installation in `/usr/local/bin`.
- `sudo make rminstall`: Löscht die Installation wieder.
- `sudo make distclean`: Reinigt das make-Environment, so dass ein anschließendes make den Übersetzungsvorgang neu startet.
- `sudo make menuconfig`: Ruft ein Menü im Terminal auf, in dem Einstellungen zum Übersetzen vorgenommen werden können.

Wenn Sie keine Fehlermeldungen beim Übersetzen erhalten haben, installieren Sie die neu erstellten Treiber mit

Code-Ausschnitt 7.17: Installation der übersetzten DVB-Treiber

```
sudo make install
```

Sollte es Fehler beim Übersetzen gegeben haben, gibt Ihnen das Kapitel 11.3 einen kleinen Überblick, was man in diesem Fall machen kann. Als Letztes muss noch die Firmware des DVB-Gerätes installiert werden. Falls diese schon bekannt (und freigegeben ist), erledigt das ein

Code-Ausschnitt 7.18: Installation der LINUX-Firmware

```
sudo apt-get install firmware-linux
```

Falls die gewünschte Firmware nicht dabei ist, muss sie - wie im vorigen Abschnitt beschrieben - von der Webseite des Herstellers heruntergeladen und in das Verzeichnis `/lib/firmware` kopiert werden.

7.2 Fernbedienung mit (F)lirc



Abbildung 7.4: Philips Universalfernbedienung (Quelle: Philips).

Der erste Schritt in Richtung Fernsehempfang ist getan. Da Zappen mit der Tastatur aber keinen Spaß macht, werden wir in diesem Abschnitt einen Infrarotempfänger mit unserem PC verbinden, den wir dann mit einer Fernbedienung ansprechen können. Sollten Sie keine Fernbedienung zur Hand haben, kann ich Ihnen die Fernbedienung aus Abb. 7.4 empfehlen. Sie ist auf beliebige Marken programmierbar und bringt alle Tasten mit, die für den sinnvollen Betrieb eines VDR erforderlich sind. Dazu zählen unter anderem vier Farbtasten (rot, grün, gelb und blau), eine Menü- und eine OK-Taste sowie Pfeiltasten. Bitte programmieren Sie für diese Fernbedienung einen RC5-Fernbedienungscode, da dieser mit den meisten Infrarot-Dioden funktioniert, indem Sie die Fernbedienung z. B. auf „Medion“ programmieren. Die Fernbedienung kostet ca. 12 €.

Die Fernbedienung benötigt noch ein Gegenstück, den Infrarot-Empfänger. Einige Wohnzimmer-PCs stellen diesen Empfänger fest eingebaut auf ihrer Hauptplatine zur Verfügung. In der Regel funktioniert dieser Empfänger dann mit der *lirc*-Methode, die im nächsten Abschnitt beschrieben wird. In diesem Abschnitt schauen wir uns *Flirc* an. *Flirc* steht für F*ck *lirc*. Die Namensgebung legt nahe, dass der Vertreiber des *Flirc*-USB-Sticks mit den *lirc* Fernbedienungs-Lösungen und deren Einrichtungen nicht so ganz einverstanden war.

Flirc ist mehr als ein einfacher Fernbedienungsempfänger. Er setzt mit Hilfe eines ARM-Prozessors Fernbedienungs-Befehle in Tastatur-Kommandos um. Am USB-Port gibt sich *Flirc* als Tastatur zu erkennen. Ist eine Fernbedienung angelernt, wirken deren Kommandos also wie direkte Tastatureingaben. So kann man beispielsweise mit einer Fernbedienung sehr einfach das zu startende Betriebssystem in einem Bootloader (etwa in *Clover*) auswählen - einfacher geht es nicht. Das Anlernen einer Fernbedienung selbst funktioniert mit einer Software, welche direkt auf der Herstellerseite (<https://flirc.tv/support/flirc-usb> [61]) für Windows, LINUX oder MacOS heruntergeladen werden kann. Unabhängig davon, unter welchem Betriebssystem Sie *Flirc* anlernen, der Stick kann danach mit der angelernten Fernbedienung in sämtlichen Programmen unter allen Betriebssystemen verwendet werden.

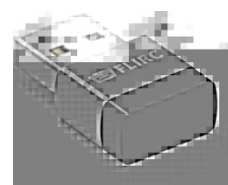


Abbildung 7.5: Der *Flirc* USB-Empfänger setzt Fernbedienungs-Befehle in Tastatur-Kommandos um. (Quelle: <https://flirc.tv/> [61]).

Unter LINUX ist das Anlernen von *Flirc* kinderleicht. Die Download-Datei heißt *flirc.latest.x86_64.tar.gz*. Packen Sie diese nach dem Herunterladen beispielsweise im *home*-Verzeichnis so aus:

Code-Ausschnitt 7.19: Auspacken der *Flirc*-GUI

```
cp ~/Downloads/flirc.latest.x86_64.tar.gz ~
cd ~
tar xvfz flirc.latest.x86_64.tar.gz
```

Nach dem Auspacken steht die grafische Konfigurationsoberfläche *Flirc* sowie ein Hilfsprogramm *flirc_util* im Verzeichnis *Flirc-3.20.4* zur Verfügung. Bitte beachten Sie, dass sich die Versionsnummer (in meinem Fall 3.20.4) ändern kann. Bevor *Flirc* gestartet werden kann, müssen noch einige Abhängigkeiten installiert werden.

Code-Ausschnitt 7.20: Installation der *Flirc*-Abhängigkeiten

```
sudo apt-get install libhidapi-hidraw0 libqt5xml5 libqt5xmlpatterns5
```

Wechseln Sie dann in das entsprechende Verzeichnis und starten Sie die GUI so:

Code-Ausschnitt 7.21: Starten der *Flirc*-GUI

```
cd Flirc-3.20.4
./Flirc
```

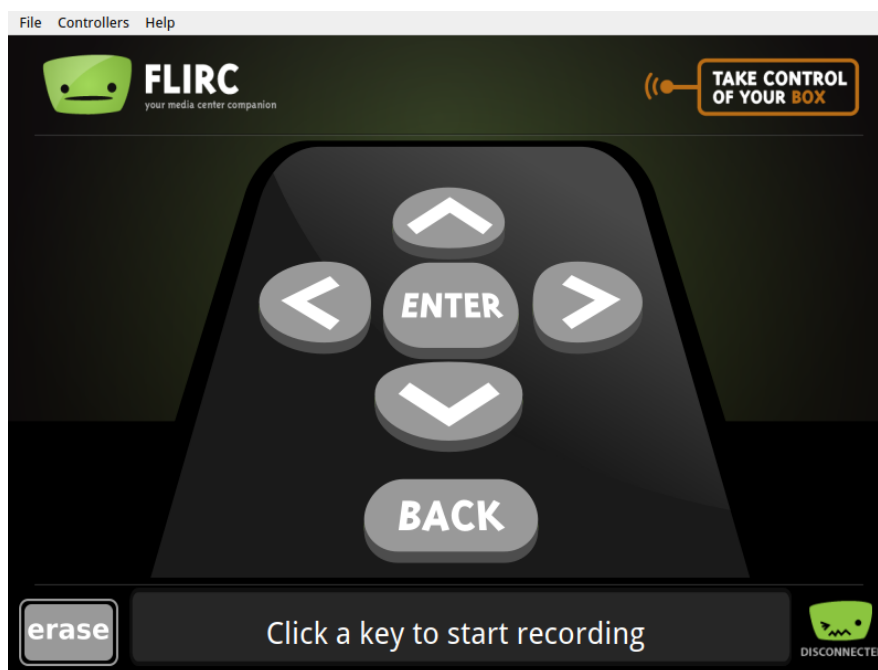


Abbildung 7.6: Die grafische *Flirc*-Oberfläche erlaubt ein einfaches Anlernen der Fernbedienung.

Die Oberfläche selbst gibt sofort Auskunft darüber, ob der *Flirc*-Empfänger erkannt wurde oder nicht. In meinem Fall (Abbildung 7.6) steckte der Empfänger nicht in einem der USB-Ports. Zur Einrichtung stehen verschiedene Fernbedienungs-Profile zur Verfügung. Sollte Ihre Fernbedienung nicht aufgelistet sein, wählen Sie als Controller „Full keyboard“ aus. So können Sie jedem einzelnen Fernbedienungs-Befehl eine einzelne Taste auf der Tastatur zuordnen. Das Anlernen der spezifischen Taste wird mit einem Mausklick auf die Taste des Tastaturbildes gestartet.

Es kann vorkommen, dass der Flirc-Empfänger träge ist und nicht immer korrekt auf die Eingaben der Fernbedienung reagiert. Abhilfe schafft hier das nochmalige Anlernen derselben Taste auf der Fernbedienung, die dann nochmals auf dieselbe Taste des Controllers (also beispielsweise der Tastatur) gemappt wird.



Der Knopf „Erase“ erlaubt es Ihnen, einzelne bereits belegte Tasten und Zuweisungen für die Fernbedienung wieder zu löschen.

Das Hilfsprogramm *flirc_util* stellt Ihnen dieselbe Funktionalität zur Verfügung wie die entsprechende GUI-Version. Hier fehlt allerdings die grafische Oberfläche, so dass Sie auf Kommandos wie „delete“ oder „record“ angewiesen sind. Die Verwendung der GUI-Version ist aber in jedem Fall zu empfehlen. Wenn Sie *Flirc* ausschließlich mit *VDR* verwenden möchten, spielt es keine Rolle, welche Taste der Fernbedienung Sie welcher Taste der Tastatur zuordnen. *VDR* erkennt *Flirc* als angeschlossene Tastatur und fragt bei der Einrichtung alle Tasten der Tastatur ab und damit automatisch alle Tasten ihrer Fernbedienung, die einer Tastatur-Taste zugewiesen sind.

Den *Flirc*-USB-Stick können Sie bei den einschlägigen Versendern für ca. 25 € erwerben.

Die Einrichtung einer Fernbedienung mit *lirc* ist dann ratsam, wenn das Mainboard Ihres Wohnzimmer-PCs bereits einen integrierten Infrarot-Empfänger besitzt oder Sie die Fernbedienung und den Empfänger Ihres USB-TV-Sticks verwenden möchten, sofern dieser mit einem solchen ausgestattet ist. Die Bezeichnung *lirc* steht übrigens für „LINUX Infrared Remote Control“. Die aktuelle *lirc*-Version installiert unter Ubuntu nicht sauber. Das Problem besteht selbst in der Version 19.04 noch. Der Aufruf von

Code-Ausschnitt 7.22: Installation von *lirc* – 1

```
sudo apt-get install lirc lirc-x liblircclient0 inputlirc
```

ergibt den folgenden Fehler:

Code-Ausschnitt 7.23: Fehlermeldung bei der Installation von *lirc*

```
dpkg: Fehler beim Bearbeiten des Paketes lirc (--configure):  
>>installiertes lirc-Skript des Paketes post-installation<< -Unterprozess gab den Fehlerwert 1 ↔  
zurück
```

Grund dieser Fehlermeldung ist die fehlende Datei *lirc_options.conf*, die bei der Konfiguration im Verzeichnis */etc/lirc* erwartet wird. Kopieren Sie diese Datei daher bitte aus einer vorhandenen Standard-Konfiguration wie folgt:

Code-Ausschnitt 7.24: Kopieren einer *Lirc*-Konfigurationsdatei

```
sudo cp /etc/lirc/lirc_options.conf.dist /etc/lirc/lirc_options.conf
```

Rufen Sie danach die Installation von *Lirc* erneut auf.

Code-Ausschnitt 7.25: Installation von *lirc* – 2

```
sudo apt-get install lirc lirc-x liblircclient0 inputlirc
```

Dieses Mal läuft sie fehlerfrei durch.

Code-Ausschnitt 7.26: Erfolgreiche Installation von *lirc*

```
lirc (0.10.1-5.2) wird eingerichtet ...
Created symlink /etc/systemd/system/multi-user.target.wants/lircd.service -> /lib/systemd/system/lircd.service.
Created symlink /etc/systemd/system/sockets.target.wants/lircd.socket -> /lib/systemd/system/lircd.socket.
```

Im nächsten Schritt editieren wir die Datei */etc/lirc/hardware.conf*. Sollte die nicht existieren, legen Sie sie bitte neu an. Eine Beispieldatei sieht so aus:

Code-Ausschnitt 7.27: Hardware-Konfiguration für *lirc*

```
# /etc/lirc/hardware.conf
#
#Chosen Remote Control
REMOTE="Linux input layer (/dev/input/eventX)"
REMOTE_MODULES=""
REMOTE_DRIVER="devinput"
REMOTE_DEVICE="/dev/input/event6"
REMOTE_LIRCD_CONF="generic/devinput.conf"
REMOTE_LIRCD_ARGS=""

#Chosen IR Transmitter
TRANSMITTER="None"
TRANSMITTER_MODULES=""
TRANSMITTER_DRIVER=""
TRANSMITTER_DEVICE=""
TRANSMITTER_LIRCD_CONF=""
TRANSMITTER_LIRCD_ARGS=""

#Enable lircd
START_LIRCD="true"

#Don't start lircmd even if there seems to be a good config file
#START_LIRCMD="false"

#Try to load appropriate kernel modules
LOAD_MODULES="true"

# Default configuration files for your hardware if any
LIRCMD_CONF=""

#Forcing noninteractive reconfiguration
#If lirc is to be reconfigured by an external application
#that doesn't have a debconf frontend available, the noninteractive
#frontend can be invoked and set to parse REMOTE and TRANSMITTER
#It will then populate all other variables without any user input
#If you would like to configure lirc via standard methods, be sure
#to leave this set to "false"
FORCE_NONINTERACTIVE_RECONFIGURATION="false"
START_LIRCMD=""
```

Die einzelnen Einträge haben die in Tabelle 7.1 angegebenen Bedeutungen. Da uns nur die Fernbedienung des VDR interessiert, können alle Einträge, die mit *TRANSMITTER* beginnen, leer bleiben.

Bestandteile der hardware.conf	
Zeile	Erklärung
<i>LIRCD_ARGS</i>	Parameter, die „lircd“ bei jedem Start ausführt.
<i>START_LIRC_CMD</i>	Der LIRC mouse daemon ermöglicht es, den Mauszeiger über die Fernbedienung zu steuern. Soll „lircmd“ beim Start von „lircd“ geladen werden, muss die Zeile einkommentiert und auf „true“ gesetzt werden.
<i>LOAD_MODULES</i>	Bei „true“ versucht „lircd“ selbstständig, ein passendes Modul für die verwendete Hardware zu laden.
<i>DRIVER</i>	Wählt einen Treiber aus. Wenn die Fernbedienung über /dev/input/„irgendwas“ angesprochen wird, sollte hier „devinput“ eingetragen werden. Eine Liste weiterer Treiber zeigt der Befehl „lircd --driver=help“ an.
<i>DEVICE</i>	Gibt an, über welches Device die Fernbedienung angesprochen wird, z. B. /dev/input/eventX.
<i>MODULES</i>	Gibt an, welches Kernelmodul „lircd“ laden soll.
<i>LIRCD_CONF</i>	Hier kann ein individueller Pfad zur „lircd.conf“ angegeben werden. Standard ist ~/.lircd.conf.
<i>LIRC_CMD_CONF</i>	Hier kann ein individueller Pfad zur „lircmd.conf“ angegeben werden. Standard ist ~/.lircmd.conf.

Tabelle 7.1: Die Tabelle zeigt alle Einstellmöglichkeiten der Datei *lirc.hardware.conf*.
 Quelle: <https://wiki.ubuntuusers.de/Lirc/> [62].

Die oben gezeigte Beispielkonfiguration (Code-Ausschnitt 7.27) geht davon aus, dass die Fernbedienung über das Event */dev/input/eventX* angesprochen werden kann. *X* steht dabei für eine Zahl, beispielsweise die 6. Diese Art Fernbedienungs-Empfänger findet man häufig in TV-USB-Sticks. Die Nummer des Events kann man wie folgt ermitteln:

Code-Ausschnitt 7.28: Ermitteln der Event-Nummer eines Eingabe-Devices

```
cat /proc/bus/input/devices
```

Die Ausgabe könnte wie folgt aussehen:

Code-Ausschnitt 7.29: Ermitteln der Event-Nummer eines Eingabe-Devices

```
I: Bus=0003 Vendor=0000 Product=0000 Version=0000
N: Name="em2880/em2870 remote control"
P: Phys=USB
S: Sysfs=/devices/virtual/input/input6
U: Uniq=
H: Handlers=kbd event6
B: EV=100003
B: KEY=4afc334 0000000 0 0 0 0000 1a0 c0000801 9e96c0 0 0 ffd
```

Die Zeile *N* enthält den Hinweis, dass es sich hier um einen Fernbedienungs-Empfänger handelt (remote-control), die Event-Nummer (6) steht in der Zeile *H*. Der Name des Eingabe-Devices lautet also */dev/input/event6*. Verfügbare *Lirc*-Treiber können nicht nur durch den Aufruf von

Code-Ausschnitt 7.30: Ermitteln der verfügbaren *Lirc*-Treiber

```
lircd --driver=help
```

aufgelistet werden. Die Webseite <http://www.lirc.org/html/table.html> [63] zeigt ebenfalls alle zur Verfügung stehenden Treiber.

Im nächsten Schritt muss eine Datei namens *lircd.conf* erstellt werden. Diese Datei stellt einen Zusammenhang her zwischen den von der Fernbedienung gesendeten Codes und den entsprechenden Tasten der Fernbedienung. Vorgefertigte *lircd.conf*-Dateien findet man unter der Adresse <http://lirc.sourceforge.net/lirc.org/> [64]. Sollten Sie das Glück haben, dass eine passende Datei für Ihre Fernbedienung vorhanden ist, kopieren Sie diese einfach mit Root-Rechten nach */etc/lirc*. Findet sich keine passende Datei, erstellen Sie selbst eine. Dazu muss *lirc* zunächst gestoppt werden.

Code-Ausschnitt 7.31: Stoppen des *Lirc*-Dienstes

```
sudo /etc/init.d/lircd stop
```

Das Programm zum Aufnehmen der Datei *lircd.conf* nennt sich *irrecord* und kann wie folgt gestartet werden:

Code-Ausschnitt 7.32: Aufnehmen der Datei *lircd.conf*

```
sudo irrecord -H treiber -d device /pfad/lircd.conf
```

Bitte ersetzen Sie im Code-Schnipsel 7.32 *treiber* mit dem zu verwendenden *Lirc*-Treiber, *device* mit dem zu verwendenden *Lirc*-Device und *pfad* mit dem Speicherort für die Datei *lircd.conf*. Sie können beispielsweise Ihr *home*-Verzeichnis auswählen und die Datei dann später nach */etc/lirc* kopieren.

Bevor wir aber die Aufnahme der *lircd.conf*-Datei starten, werfen wir einen Blick auf die Fernbedienungsschlüssel, die LINUX offiziell unterstützt, also beispielsweise den Namen *MENU* für die Menütaste Ihrer Fernbedienung. Das ist wichtig für Programme, die nur auf die richtigen KEYS reagieren. Wechseln Sie dazu in ein Verzeichnis, in dem Sie Schreibrechte haben und speichern Sie LINUX-Standardschlüsselnamen in einer Datei namens *lirc_strings.txt* ab.

Code-Ausschnitt 7.33: Auflisten aller *lirc*-Schlüssel

```
irrecord --list-namespace | grep KEY &> lirc_strings.txt
```



Der Befehl *grep* sucht das Schlüsselwort, das hinter ihm steht (also im oberen Fall das Wort „KEY“) und hat als Ergebnis die Zeile(n) der Datei oder des Befehls, die das Schlüsselwort enthalten. Der Befehl *grep* hallo datei sucht in der Datei datei alle Zeilen, die das Wort hallo enthalten.

Schauen Sie sich danach die Datei *lirc_strings.txt* an.

Code-Ausschnitt 7.34: Anschauen des Inhalts einer Datei

```
more lirc_strings.txt
```

Starten wir also das Anlernen der Fernbedienung wie oben (Code-Schnipsel 7.32) gezeigt. Das Programm fordert Sie zunächst auf, *RETURN* zu drücken und nach einer nochmaligen Bestätigung mit *RETURN* möglichst viele verschiedene Knöpfe auf der

Fernbedienung zu drücken. Jeder Druck auf der Fernbedienung sollte dabei einen Punkt (.) in das Terminal zaubern. Nachdem das Programm (hoffentlich) den prinzipiellen Code der Fernbedienung erkannt hat, geht es an das Anlernen der Tasten. Sie werden dann aufgefordert, einen Namen für die Taste einzugeben und dann die entsprechende Taste der Fernbedienung zu drücken. Verwenden Sie bitte dabei die Namen der Datei *lirc_strings.txt*, die Sie eben gespeichert haben. Nachdem Sie alle Tasten auf diese Weise angelernt haben, beenden Sie bitte das Anlernen durch nochmaliges Drücken der *RETURN*-Taste auf Ihrer Tastatur. Sie werden dann aufgefordert, eine einzelne Taste der Fernbedienung so schnell hintereinander zu drücken wie möglich. Dadurch lernt *lirc* die Wiederholrate der Fernbedienung kennen. Nach diesem letzten Schritt ist das Anlernen der Fernbedienung abgeschlossen und eine Konfigurations-Datei namens *lircd.conf* wurde in Ihrem Home-Verzeichnis erstellt. Bevor wir die Datei dauerhaft einbinden, testen wir das Ergebnis kurz. Starten Sie bitte den *lirc*-Daemon und rufen Sie das Programm *irw* auf.

Code-Ausschnitt 7.35: Anschauen der Infrarot-Befehle

```
sudo lircd -d /dev/lirc0 /pfad/lircd.conf
irw
```

Die Software *irw* (InfRared Write) reagiert auf Fernbedienungs-Tastendrucke und gibt den gefundenen KEY im Terminal aus. Haben Sie also beispielsweise für die rote Taste der Fernbedienung den Schlüssel *KEY_RED* vergeben, erscheint diese Ausgabe, wenn Sie nun die rote Taste Ihrer Fernbedienung drücken. Wenn das alles wie gewünscht funktioniert, kopieren Sie die Datei *lircd.conf* bitte nach */etc/lircd/lircd.conf.d*, und beenden Sie den *lirc*-Daemon.

Code-Ausschnitt 7.36: Kopieren der *lircd.conf*-Datei

```
sudo cp /pfad/lircd.conf /etc/lircd/lircd.conf.d
sudo kill lircd
```

Starten Sie dann den *lirc*-Dienst.

Code-Ausschnitt 7.37: *Lirc*-Dienst starten

```
sudo /etc/init.d/lirc start
```

Sollten Sie *lirc* ausschließlich für ein Programm benutzen, das nur auf Tastatureingaben reagiert und das Sie dennoch per Fernbedienung steuern möchten, ändern Sie bitte noch die folgende Zeile in der Datei */etc/lirc/lirc_options.conf*.

Code-Ausschnitt 7.38: Auslösen von Tastatur-Events

```
[lircmd]
uinput      = True
```

Hierdurch wird beim Drücken der Fernbedienung gleichzeitig ein Tastatur-Event ausgelöst.

7.3 VDR

Der Video Disk Recorder *VDR* von Klaus Schmidinger ist *die* Multimedia-Anwendung auf LINUX, mit der kaum eine andere Multimedia-Software zum Fernsehen mithalten kann. Das kostenlose Programm stellt alles zur Verfügung, was einen guten UHD- oder Full-HD-Receiver ausmacht. Darüber hinaus kann es beliebig individualisiert werden, und eine Fülle an Zusatzprogrammen - sogenannte Plugins - erfüllen auch die letzten Wünsche. In diesem Abschnitt werden wir den *VDR* in der neuesten Version aus dem Netz laden, kompilieren und installieren.

Natürlich könnten wir auch ein `sudo apt-get install vdr` benutzen und uns auf die Repository-Version beschränken. Diese hängt aber meist ein paar Versionen hinterher und bringt auch nicht alle Plugins mit, die wir selbst auf dem Wohnzimmer-PC benötigen.

Im ersten Schritt öffnen wir ein Terminal und erstellen ein Unterverzeichnis *tv*. Wir wechseln in dieses Verzeichnis und laden die aktuellen Quellen des *VDR* herunter und packen sie aus.

Code-Ausschnitt 7.39: VDR aus den Quellen installieren

```
mkdir tv
cd tv
wget ftp://ftp.tvdr.de/vdr/vdr-2.4.1.tar.bz2
tar xvfj vdr-2.4.1.tar.bz2
ln -s vdr-2.4.1 VDR
```

Diese liegen auf einem *ftp*-Server (*File Transfer Protocol*). Der *tar*-Befehl mit seinen Parametern packt den *VDR*-Quellcode in einem Verzeichnis namens *vdr-2.4.1* aus (*bzip2*). Wir erstellen nun einen symbolischen Link namens *VDR*, der auf das gerade heruntergeladene und ausgepackte Verzeichnis *vdr-2.4.1* zeigt.



Der symbolische Link `ln -s` zeigt auf das Verzeichnis, das er linkt. Man kann dann in das Verzeichnis wechseln, indem man nur den Link-Namen angibt. Im oberen Beispiel reicht dann ein `cd VDR`, um in das Verzeichnis *vdr-2.4.1* zu gelangen. Sollte es eine neuere Version des *VDR* geben, kann man diese parallel installieren und ändert nur den symbolischen Link. Schon funktionieren alle Start-Skripte wieder, die wir später noch erstellen werden. Andernfalls müssten wir in allen Skripten immer den kompletten Pfad zum *VDR* mit angeben (inkl. der Versionsnummer (-2.4.1)).

Symbolische Links sehen im Listing wie folgt aus:

Code-Ausschnitt 7.40: Symbolische Links

```
ls -l
lrwxrwxrwx 1 name name 9 Mai 4 2013 VDR -> vdr-2.4.0
drwxr-xr-x 8 name name 4096 Mai 4 16:39 vdr-2.4.0
```

Der Link wird dabei durch einen Pfeil (*-l*) symbolisiert. Links können mit `rm linkname` wieder gelöscht werden, ohne dass das Originalverzeichnis gelöscht wird. Im nächsten Schritt müssen noch ein paar Programme installiert werden, welche zum Übersetzen des *VDR* benötigt werden.

Code-Ausschnitt 7.41: Abhängigkeiten zum Übersetzen von VDR

```
sudo apt-get install mercurial libncurses5-dev libncursesw5-dev libproc-processtable-perl ↵
libfribidi-dev libcap-dev libjpeg-dev lcdproc libssl-dev libfreetype6-dev libfontconfig1-↵
dev g++ git vim build-essential libfontconfig-dev gettext libncurses-dev vdr-dev libcurl4-↵
gnutls-dev libavcodec-dev libswscale-dev libasound2-dev
```

Wechseln wir nun ins VDR-Verzeichnis und starten die Übersetzung des VDR und der Standard-Plugins.

Code-Ausschnitt 7.42: Übersetzen des VDR

```
cd VDR
make -j4
```

Das Übersetzen dauert nicht lange, wenn wir den Parameter `-j4` verwenden. Dieser startet nämlich einen Übersetzungsvorgang für bis zu vier Kerne der CPU. Das fertige Programm `vdr` liegt dann im aktuellen Verzeichnis. Bevor wir es aber starten, müssen wir noch einige Vorbereitungen treffen. VDR selbst benötigt nämlich noch die folgenden Verzeichnisse:

- Ein Konfigurations-Verzeichnis, in dem alle Konfigurations-Dateien stehen. Das Default-Verzeichnis lautet `/var/lib/vdr`. Dieses werden wir auch nutzen.
- Ein Video-Verzeichnis, in das der VDR Aufnahmen speichern kann. Für einen ersten Test erstellen wir das Verzeichnis `/video` gleich auf der aktuellen Festplatte. Es wird auch die Möglichkeit beschrieben, eine externe USB-Festplatte anzuschließen und die Aufnahmen dort zu speichern. Das Standard-Verzeichnis ist `/var/lib/video`.

Öffnen wir also ein Terminal und erstellen das Video-Verzeichnis.

Code-Ausschnitt 7.43: Erstellen des Video-Verzeichnisses

```
sudo mkdir /video
sudo chown name:name /video
sudo chmod 775 /video
```

Gleichzeitig sagen wir, dass der Besitzer des Verzeichnisses `/video` der Benutzer `name` mit der Gruppe `users` ist und erlauben, dass der Benutzer Filme schreiben und lesen darf (`chmod 775`). Ersetzen Sie den Namen `name` bitte mit Ihrem Benutzernamen. Falls ihr Wohnzimmer-PC genügend abgesichert ist, spricht zwar nichts dagegen, den VDR als Benutzer `root` laufen zu lassen, das muss aber nicht sein. Es macht aber dann Sinn, wenn wir - wie später - Skripte benötigen, die Root-Rechte brauchen. Aber auch dann könnte man diesen Skripten Root-Rechte einräumen, so dass nicht der ganze VDR als Systemadministrator laufen muss.

In der Datei `/etc/sudoers` stehen Benutzer und Programme, die als Root laufen dürfen, ohne dass z. B. ein Kennwort eingegeben werden muss. Die Datei kann man mit dem Befehl `sudo visudo` ergänzen.



Kopieren wir nun alle Standard-Konfigurations-Dateien des VDR in das Konfigurations-Verzeichnis.

Code-Ausschnitt 7.44: Kopieren der VDR-Konfigurations-Dateien

```
sudo cp ~/tv/VDR/*.conf /var/lib/vdr
```

Die Konfigurations-Dateien haben die folgenden Bedeutungen:

- *channels.conf*: Das ist die Senderliste des VDR. Unterschiedliche Empfangsarten wie DVB-S und DVB-T dürfen dabei gemischt werden.
- *remote.conf*: Hier stehen alle Möglichkeiten zur Fernbedienung des VDR. Das betrifft - Sie ahnen es - *lirc*, aber auch die Tastatur.
- *diseqc.conf*: Hier finden sich alle Diseqc-Einstellungen.
- *sources.conf*: Hier sind die Satelliten, die empfangbar sind, aufgeführt.
- *commands.conf*: Hier stehen Befehle, die der VDR auf Fernbedienungsdruck ausführen kann.
- *scr.conf*: Hierin steht das Satelliten-Kanal-Routing.
- *keymacros.conf*: Hier können Makros gespeichert werden, die dann Fernbedienungstasten zugeordnet werden können.

Bevor wir das erste Fernsehbild auf unseren Monitor zaubern, müssen noch zwei Dinge erledigt werden. Zunächst einmal füllen wir die wichtigsten Konfigurations-Dateien mit Leben, danach installieren wir ein Ausgabe-Plugin für den Wohnzimmer-PC, damit wir das Bild überhaupt (beschleunigt) sehen können. Fangen wir mit der *channels.conf* an:

Code-Ausschnitt 7.45: Beispiel einer VDR Kanal-Liste

```
:-> Satelliten-Test
Das Erste;ARD:11837:HC34M2S0:S19.2E:27500:101=2:102=deu@3,103=mis@3;106=deu@106:104;105=deu↔
:0:28106:1:1101:0
Das ErsteHD;ARD:11493:HC23M5035S1:S19.2E:22000:5101=27:5102=deu@3,5103=mis@3;5106=deu@106↔
:5104;5105=deu:0:10301:1:1019:0
:-> DVB-T2-Test
KiKa HD;ZDFmobil:578000:B8D0G19128S1T16Y0P0:T↔
:27500:2510=36:0;2520,2521:2530;2531:0:2005:8468:515:0
```

Das Beispiel enthält zwei Sender für den Satelliten-Empfang auf Astra 19.2° Ost und ein Beispiel für DVB-T2. Der genaue Aufbau der Datei ist unter <http://www.vdr-wiki.de/wiki/index.php/Channels.conf> [55] beschrieben. Dort gibt es auch Beispiele für die unterschiedlichen Empfangsarten.

Der Inhalt meiner *remote.conf* sieht wie folgt aus, damit ich den VDR schon einmal mit der Tastatur oder *Flirc* bedienen kann:

Code-Ausschnitt 7.46: Inhalt der Datei *remote.conf*

```

KBD.Up      00000000010C0039
KBD.Down    00000000001B5B42
KBD.Menu    00000000006D0039
KBD.Ok      00000000000D0039
KBD.Back    00000000007F0039
KBD.Left    000000000010E0039
KBD.Right   000000000010F0039
KBD.Red     0000000000720039
KBD.Green   0000000000670039
KBD.Yellow  0000000000000079
KBD.Blue    0000000000620039
KBD.0       0000000000300039
KBD.1       0000000000310039
KBD.2       0000000000320039
KBD.3       0000000000330039
KBD.4       0000000000340039
KBD.5       0000000000350039
KBD.6       0000000000360039
KBD.7       0000000000370039
KBD.8       0000000000380039
KBD.9       0000000000390039
KBD.Info    0000000000690039

```

Support für die Fernbedienung fügen wir gleich hinzu. Da ich Hotbird und Astra empfangen kann, sieht meine *diseqc.conf* Datei so aus:

Code-Ausschnitt 7.47: *diseqc.conf* bei Astra- und Hotbird-Empfang

```

# Full DiSEqC sequence:

S19.2E 11700 V 9750 t v W15 [E0 10 38 F0] W15 A W15 t
S19.2E 99999 V 10600 t v W15 [E0 10 38 F1] W15 A W15 T
S19.2E 11700 H 9750 t V W15 [E0 10 38 F2] W15 A W15 t
S19.2E 99999 H 10600 t V W15 [E0 10 38 F3] W15 A W15 T

S13.0E 11700 V 9750 t v W15 [E0 10 38 F4] W15 B W15 t
S13.0E 99999 V 10600 t v W15 [E0 10 38 F5] W15 B W15 T
S13.0E 11700 H 9750 t V W15 [E0 10 38 F6] W15 B W15 t
S13.0E 99999 H 10600 t V W15 [E0 10 38 F7] W15 B W15 T

```

Ich habe also die entsprechenden Kommentarzeichen (#) vor den verfügbaren Satelliten entfernt. In der *commands.conf* können Befehle eingetragen werden, die dann von VDR aus ausgeführt werden können, z. B.

Code-Ausschnitt 7.48: *commands.conf* Beispiel-Eintrag

```
Reboot : sudo /sbin/reboot
```

Der Name des Kommandos steht mit einem : getrennt vom auszuführenden Befehl. Alle anderen Dateien lassen wir auf den Default-Werten stehen. Nach diesem trockenen Teil kommen wir nun zum ersten sichtbaren Ergebnis und installieren eine Ausgabemöglichkeit für den Wohnzimmer-PC.

Es ist gar nicht so einfach, sich im Dschungel der Ausgabe-Plugins für VDR sicher zu bewegen. Hier gibt es eine Fülle an GPUs, die alle unterschiedliche Plugins benötigen. Selbst wenn man sich einmal für eine GPU (beispielsweise INTEL) entschieden hat, stößt man auf verschiedene Plugins, die ihrerseits wieder verschiedene Voraussetzungen mit sich bringen. Für eine INTEL GPU werde ich Ihnen zwei verschiedene Wege vorstellen. Einer der Wege setzt die Installation einer älteren *ffmpeg*-Version voraus, die wir selbst kompilieren, installieren und richtig verwenden müssen. Dabei lernen Sie beispielsweise den Gebrauch der System-Variable *LD_LIBRARY_PATH*, die dem

System mitteilt, aus welchem Verzeichnis Bibliotheken (beispielsweise unsere selbst kompilierte) vorzugsweise verwendet werden sollen. Der andere Weg funktioniert ohne den Kniff, Bibliothekspfade umzubiegen.

Die Entwicklung der Ausgabe-Plugins für VDR ist aber immer noch im Gange. Daher werde ich ebenfalls auf Versionen verweisen, deren Entwicklung noch im Gange war, als ich dieses Buch geschrieben habe. Wir werden aber auch einen Blick auf Ausgabe-Plugins für andere GPUs, z. B. auf NVidia-GPUs werden.

Unter <https://github.com/pesintta/vdr-plugin-vaapidevice> [65] gibt es ein VDR-Ausgabe-Plugin, das mit einer älteren *ffmpeg*-Version, beispielsweise der Version 3.3.9 auf INTEL-GPUs problemlos funktioniert.

Installieren wir zunächst die Version 3.3.9 von *ffmpeg*. *ffmpeg* ist ein Tool zur Bearbeitung von Video- und Audiostreams, das vom VDR-Ausgabe-Plugin *vaapidevice* benötigt wird. Ältere *ffmpeg*-Versionen stehen auf der Internetseite <https://www.ffmpeg.org/releases> [66] zum Download bereit. Laden Sie also bitte die Version *ffmpeg-3.3.9.tar.bz2* herunter und kopieren Sie die Datei in ein Verzeichnis Ihrer Wahl, beispielsweise nach *~/tv*. Bitte entpacken Sie die Datei danach und starten Sie den Übersetzungsvorgang.

Code-Ausschnitt 7.49: Übersetzen einer älteren *ffmpeg*-Version

```
cp Downloads/ffmpeg-3.3.9.tar.gz2 ~/tv
cd ~/tv
tar xvfj ffmpeg-3.3.9.tar.bz2
cd ffmpeg-3.3.9
./configure --prefix=/usr/local --enable-shared
make -j4
sudo make install
```

Das Prefix *--prefix=/usr/local* sorgt dafür, dass die fertige übersetzten Dateien unterhalb von */usr/local* installiert werden und damit nicht mit anderen schon vorhandenen Systeminstallationen kollidieren. Das Prefix *--enable-shared* erstellt dynamische Bibliotheken. Beim Übersetzen des VDR-Ausgabe-Plugins müssen wir nun darauf achten, den Pfad der zu nutzenden Bibliotheken so einzustellen, dass die unter */usr/local/lib* gespeicherten Bibliotheken bevorzugt verwendet werden. Dies geschieht durch das Setzen der System-Variable *LD_LIBRARY_PATH*. Bevor wir allerdings dieses Ausgabe-Plugin übersetzen, müssen weitere Pakete, die zur Übersetzung erforderlich sind, installiert werden:

Code-Ausschnitt 7.50: Übersetzen des Ausgabe-Plugins für VDR

```
sudo apt-get install libxcb-screensaver0-dev libxcb-dpms0-dev libxcb-icccm4-dev libxcb-ewmh-dev
cd ~/tv/VDR/PLUGINS/src
git clone https://github.com/pesintta/vdr-plugin-vaapidevice
cd ~/tv/VDR
LD_LIBRARY_PATH=/usr/local/lib/ make
sudo make install
```

Ich habe für dieses Ausgabe-Plugin noch drei Patches heruntergeladen. Diese Patches heißen *audio.patch*, *FixPtsLog.patch* und *speedupdown.patch*. Sie finden sich in den Github-Issues des Plugins (<https://github.com/pesintta/vdr-plugin-vaapidevice/issues> [65]). Sie können diese Patches herunterladen und in das Verzeichnis *~/tv/VDR/PLUGINS/src/vaapidevice* kopieren und wie folgt anwenden:

Code-Ausschnitt 7.51: Anwenden von Patches für das *vaapidevice*-Plugin

```
cat audio.patch | patch -sp1
cat FixPtsLog.patch | patch -sp1
cat speedupdown.patch | patch -sp1
```

Falls Sie zunächst einmal testen wollen, ob der Patch auch ohne Fehler durchläuft, übergeben Sie dem Patch-Aufruf noch den Parameter `--dry-run`. Sie erhalten alle Fehlermeldungen dann so, als wäre der Patch ausgeführt worden, die zu patchenden Dateien werden aber nicht verändert.

Wenn Sie den VDR später unter Verwendung dieses Ausgabe-Plugins starten, vergessen Sie bitte nicht, die `LD_LIBRARY_PATH`-Variable entsprechend zu setzen.



Obwohl dieses Ausgabe-Plugin nicht mehr aktiv weiterentwickelt zu werden scheint, tauchen in den Github-Issues ab und an Patches auf, die Verbesserungen bringen können. Einen Patch können Sie wie folgt anwenden: Speichern Sie die Text-Datei zunächst im Plugin-Verzeichnis und rufen Sie dann den folgenden Befehl auf:

Code-Ausschnitt 7.52: Einspielen eines Source-Code-Patches

```
cat name.diff | patch -p1
```

Der Parameter `-p1` im oben gezeigten Aufruf ignoriert einen Pfad-Eintrag bei den im Patch angegebenen Dateinamen. Falls das Patch-Kommando einen Fehler zeigt, probieren Sie den Parameter `-p2` oder lassen Sie ihn ganz weg. Im schlimmsten Fall müssen Sie den Namen der zu patchenden Datei manuell eingeben. Das Programm `patch` fragt Sie dann danach.

Unter <https://github.com/rofafor/vdr-plugin-vaapidevice> [67] steht eine Version des INTEL-Ausgabe-Plugins zur Verfügung, die auch mit neueren `ffmpeg`-Versionen funktioniert.

Achten Sie bitte darauf, funktionierende Plugin-Verzeichnisse umzubenennen, bevor Sie andere Verzeichnisse mit demselben Namen auschecken und gegebenenfalls Ihre funktionierenden Daten überschreiben. Das Umbenennen könnte mit einem einfachen `mv vaapidevice vaapidevice.old` geschehen.



Falls `ffmpeg` nicht wie oben beschrieben händisch installiert wird, erfordern die meisten Ausgabe-Plugins die Installation der Pakete `libavcodec-dev` `libavcodec-dev` `libscales-dev` `libasound2-dev`. Diese sollten Sie aber bereits vor dem Übersetzen des VDR installiert haben.

Hinweis 7.1 (Übersetzungsfehler) Manchmal ändern sich in Distributionen die Inhalte von Paketen. Sollten Sie beim Übersetzen eines VDR-Plugins auf einen Fehler treffen, liegt es meistens daran, dass ein Paket fehlt, das zusätzlich installiert werden muss. Werden Sie aus der Fehlermeldung nicht schlau, schauen Sie im Internet nach der Fehlermeldung. In der Regel finden Sie dann den Namen des fehlenden Paketes und können es nachinstallieren.

Möchten Sie das Plugin von <https://github.com/rofafor/vdr-plugin-vaapidevice> [67] installieren, gehen Sie bitte wie folgt vor:

Code-Ausschnitt 7.53: Übersetzen des Ausgabe-Plugins für VDR

```

sudo apt-get install libxcb-screensaver0-dev libxcb-dpms0-dev libxcb-icccm4-dev libxcb-ewmh-dev
cd ~/tv/VDR/PLUGINS/src
git https://github.com/rofafor/vdr-plugin-vaapidriver
cd ~/tv/VDR
make plugins
sudo make install

```



In den meisten Fällen können Sie auch in das Plugin-Verzeichnis selbst wechseln und das Übersetzen und Installieren von dort aus initiieren. Das Plugin selbst sollte sich aber immer im Verzeichnis VDR/PLUGINS/src befinden.

Beide bisher vorgestellten Plugins besitzen keine OpenGL-Beschleunigung. Das Verwenden von Skins, die wir später noch installieren werden, funktioniert damit deutlich langsamer. Eine Version des Ausgabe-Plugins, die eine OpenGL-/EGL-beschleunigte Wiedergabe des VDR-OSDs erlaubt, befindet sich gerade in der Entwicklung. Das *softhdcpuvid*-Ausgabe-Plugin kann von <https://github.com/jojo61/vdr-plugin-softhdcpuvid> [68] herunter geladen werden und funktioniert nicht nur für Intel-GPUs, sondern unterstützt auch NVidia. Die Entwicklung des Plugins und auch die erzielten Fortschritte kann man im VDR-Portal, der ersten Anlaufstelle für alle Informationen rund um den VDR, mitverfolgen (<https://www.vdr-portal.de/forum/index.php?thread/132976-softhdcpuvid-jetzt-mit-vaapi-support/> [56]). Hier gibt es auch Tipps für die Installation.



*Einige Plugins checken sich mit dem Namen *vdr-plugin-name*. Falls Sie im Verzeichnis VDR ein *make plugins* aufrufen, werden Verzeichnisse, die einen Bindestrich enthalten, ignoriert. Starten Sie in diesem Fall das Übersetzen im Verzeichnis selbst oder benennen Sie das Verzeichnis in *name* um.*

Zu dem Zeitpunkt, als ich dieses Buch geschrieben habe, ist es mir leider nicht gelungen, dieses universelle Ausgabe-Plugin, welches sowohl für INTEL als auch für NVidia GPUs funktioniert, zu übersetzen. Voraussetzungen zum Übersetzen sind die folgenden Bibliotheken:

Code-Ausschnitt 7.54: Installation von Abhängigkeiten

```

sudo apt-get install libglew-dev libglm-dev libplacebo-dev libavfilter-dev

```

Weiterhin muss für INTEL GPUs das *Makefile* so editiert werden, dass der Eintrag *CUVID=1* auskommentiert und der Eintrag *VAAPI=1* einkommentiert wird.

Code-Ausschnitt 7.55: Editieren des *Makefile*

```

# what kind of driver do we make -
# if VAAPI is enabled the drivename is softhdvaapi
# if CUVID is enabled the drivename is softhdcpuvid
VAAPI=1
#CUVID=1

```

Wenn Sie für NVidia GPUs kompilieren, ist ebenfalls die Installation von

Code-Ausschnitt 7.56: Installation der NVidia-Abhängigkeiten

```

sudo apt-get install nvidia-cuda-dev

```

Pflicht. Prinzipiell wird das Plugin wie folgt übersetzt:

Code-Ausschnitt 7.57: Installation des universellen Ausgabe-Plugins

```
cd VDR/PLUGIN/src
git clone https://github.com/jojo61/vdr-plugin-softhdcpuvid
cd vdr-plugin-softhdcpuvid
make
sudo make install
```

Nachdem Sie ein VDR-Ausgabe-Plugin erfolgreich übersetzt haben, ist eine sogenannte *shared library* erstellt worden, also eine Bibliothek, die bei Bedarf dazu geladen wird. Sie befindet sich beispielsweise im Verzeichnis `/tv/vdr/PLUGINS/src/vaapidevice` und heißt `libvdr-vaapidevice.so`. Der Aufruf von

Code-Ausschnitt 7.58: Installation des VDR und der Plugins

```
sudo make install
```

im VDR-Verzeichnis installiert die ausführbare Datei für den VDR im Verzeichnis `/usr/local/bin` und alle Plugin-Bibliotheken im Verzeichnis `/usr/local/lib/vdr`. Dabei erhalten die Bibliotheken automatisch den Zusatz der VDR-Versionsnummer, in unserem Fall also `.2.4.1`. Bei manchen Plugins kann es erforderlich sein, die Datei händisch in das Verzeichnis für die VDR-Bibliotheken zu kopieren. In diesem Fall sähe der Aufruf zum Kopieren so aus:

Code-Ausschnitt 7.59: Händisches Installieren der Plugin-Dateien

```
cp ~/tv/VDR/PLUGINS/src/vaapidevice/libvdr-vaapidevice.so /usr/local/lib/vdr/libvdr-vaapidevice↵
.so.2.4.1
```

Achten Sie bitte darauf, dass die Versionsnummer (2.4.1) identisch ist mit der des VDR. Sie erkennen das an den VDR-Plugins, die sich ebenfalls im o. g. Verzeichnis befinden. Bereit für einen ersten Start? Also los:

Code-Ausschnitt 7.60: Erster VDR-Start

```
export VDR_LANG=de_DE@euro
export VDR_CHARSET_OVERRIDE=ISO-8859-9
./vdr --lirc -c /var/lib/vdr -v /video -L /usr/local/lib/vdr P'vaapidevice -a hdmi:AES0=0x4,↵
CARD=PCH,DEV=1 -p hdmi:AES0=0x6,CARD=PCH,DEV=1 -d :0.0 -f'
```

Was geschieht hier? Zunächst werden zwei VDR-Variablen exportiert (also sichtbar gemacht und gesetzt), welche die Sprache für den VDR so einstellen, dass deutsche Umlaute korrekt dargestellt werden. Danach wird der VDR selbst mit einigen Parametern aufgerufen. Der Parameter `-c` übergibt das Konfigurations-Verzeichnis, welches wir zu Beginn erstellt haben. Der Parameter `-v` erledigt dasselbe mit dem Video-Verzeichnis. Der Parameter `-L` übergibt das Plugin-Verzeichnis. Hier nehmen wir einfach das Standard-Verzeichnis des VDR. Der Parameter `-P` übergibt ein Plugin. Wir nutzen im Moment nur ein einziges Plugin, nämlich das *vaapidevice*, über das wir das Bild auf unseren Monitor zaubern. Das Plugin *vaapidevice* selbst besitzt auch wieder einige Parameter. So steht das `-f` für fullscreen, also ein bildschirmfüllendes Bild. Der Parameter `-d` beschreibt das zu verwendende Display. In unserem Fall ist es der einzig angeschlossene Hauptbildschirm, beispielsweise also der Fernseher. Der Parameter `-a` wählt das Audio-Device aus. In meinem Fall wähle ich den Digital-Ton über HDMI aus, der bei mir über ein optisches Kabel an einen Audio-Receiver weitergegeben wird. Die möglichen Audio-Devices werden mit dem Befehl

Code-Ausschnitt 7.61: Auflisten der Audio-Devices

```
aplay -L
```

aufgelistet.

Falls Sie ein Ausgabe-Plugin installiert haben, das eigene Bibliotheken in `/usr/local/lib` voraussetzt, ändern Sie den oben gezeigten Start-Aufruf von *VDR* bitte wie folgt:

Code-Ausschnitt 7.62: Erster *VDR*-Start

```
LD_LIBRARY_PATH=/usr/local/lib ./vdr --lirc -c /var/lib/vdr -v /video -L /usr/local/lib/vdr P'↔  
vaapidevice -a hdmi:AES0=0x4,CARD=PCH,DEV=1 -p hdmi:AES0=0x6,CARD=PCH,DEV=1 -d :0.0 -f'
```



Durch den Aufruf von

Code-Ausschnitt 7.63: *VDR* mit *Lirc*-Unterstützung übersetzen

```
make REMOTE=LIRC
```

kann man den VDR auch mit lirc-Unterstützung übersetzen. Sollten Sie das beim Bauen auf dem Quelltext vergessen haben, können Sie den lirc-Support mit dem Parameter --lirc im VDR-Aufruf nachträglich einschalten.

Nach dem Start werden Sie zunächst aufgefordert, eine Taste auf Ihrer Fernbedienung zu drücken. *VDR* startet dann mit dem Zuweisen aller Fernbedienungstasten. Danach ist *VDR* nicht nur mit der Tastatur, sondern auch mit der Fernbedienung bedienbar.



Beim Beenden des VDR werden Sie sicher feststellen, dass der Cursor auf Ihrer Konsole verschwunden ist. Dieses Problem werden wir durch ein Start-Skript lösen.

Um den *VDR* so zu starten, dass die Konsole nach seinem Beenden wieder mit Cursor sichtbar ist, legen Sie bitte als *root* eine Datei mit dem Namen *runvdr* im Verzeichnis `/usr/local/bin/` mit dem folgenden Inhalt an (`sudo vi /usr/local/bin/runvdr`):

Code-Ausschnitt 7.64: Einfaches *VDR*-Start-Skript

```
#!/bin/bash  
export VDR_LANG=de_DE@euro  
export VDR_CHARSET_OVERRIDE=ISO-8859-9  
/usr/local/bin/vdr -c /var/lib/vdr -v /video -P'vaapidevice -a hdmi:AES0=0x4,CARD=PCH,DEV=1 -p ↔  
hdmi:AES0=0x6,CARD=PCH,DEV=1 -d :0.0 -f'  
# Ausgabe wieder einschalten  
stty echo
```

Die letzte Zeile sorgt dafür, dass die Konsole nach dem Beenden des *VDR* wiederhergestellt wird. Machen Sie das Skript ausführbar und starten Sie es probierhalber.

Code-Ausschnitt 7.65: Start-Skript ausführbar machen

```
chmod 775 /usr/local/bin/runvdr  
/usr/local/bin/runvdr
```

Wenn Sie die Fernbedienung nicht anlernen wollen, sondern die Tasten direkt zuweisen möchten, können Sie die *lirc*-Einträge auch direkt in der *VDR*-Konfigurationsdatei *remote.conf* vornehmen. Die Einträge sehen dann wie folgt aus:

Code-Ausschnitt 7.66: Einträge in der Datei *remote.conf*

```
...
LIRC.Left      KEY_LEFT
LIRC.Right     KEY_RIGHT
KBD.Up         00000000010C0039
KBD.Down       0000000001B5B42
...
```

lirc- und Tastatur-Einträge (KDB für *keyboard*) können einfach miteinander kombiniert werden.

Es würde den Rahmen dieses Buches sprengen, auf alle Einstellmöglichkeiten des VDR einzugehen. Probieren Sie doch einfach die Einstellungen aus, wie beispielsweise das Seitenverhältnis Ihres Bildschirms oder analoge/digitale Audioausgabe. Im weiteren Verlauf dieses Kapitels werden wir noch das eine oder andere Plugin installieren. Der Link <http://www.vdr-wiki.de/wiki/index.php/Plugins> [55] gibt eine schöne Übersicht über verfügbare VDR-Plugins.

7.4 Skripte

Im letzten Abschnitt habe ich Ihnen bereits ein kurzes Start-Skript für den VDR vorgestellt. Dieses werden wir nun ausbauen, so dass der VDR nach einem Absturz beispielsweise automatisch neu gestartet wird. Glücklicherweise geschieht das aber eher selten. Danach stelle ich Ihnen ein Skript vor, welches Ihren Wohnzimmer-PC vor einer gesetzten Timer-Aufnahme automatisch startet. Ein Skript zum Starten des VDR sieht so aus:

Code-Ausschnitt 7.67: Komplettes VDR Start-Skript

```
#!/bin/sh
#LANG=de_DE.utf8 /usr/bin/vdradmin

# set hwclock to utc in order to make acpi wakeup work
sudo hwclock --systohc --utc

export LC_MESSAGES=de_DE.UTF-8
export LC_LANG=de_DE.UTF-8
export LANG=de_DE.UTF-8
export LC_ALL=de_DE.UTF8
export VDR_LANG=de_DE@euro
export VDR_CHARSET_OVERRIDE=ISO-8859-9

VDRPRG="DISPLAY=:0.0 /usr/local/bin/vdr"
#VDRPRG="DISPLAY=:0.0 LD_LIBRARY_PATH=/usr/local/lib /usr/local/bin/vdr"

#VDR_OPTIONS="-l 0 -w 60 --lirc=/var/run/lirc/lircd -v /video -s /usr/local/bin/vdrshutdown.sh"
VDR_OPTIONS="-w 60 -v /video -c /var/lib/vdr -s /usr/local/bin/vdrshutdown.sh"

VDR_PLUGINS=" \
-P'vaapidevice -a hdmi:AES0=0x4,CARD=PCH,DEV=1 -p hdmi:AES0=0x6,CARD=PCH,DEV=1 -d :0.0 -f' \
-P'osdteletext' \
-P'externalplayer \
-P'weatherforecast \
-P'tvguideng \
-P'tvscraper -d /video/cache' \
-P'femon \
-P'skindesigner -l /var/lib/vdr/plugins/skindesigner/logos' "

#-P'osdteletext --directory=/video/cache/vtx --cache-system=packed --max-cache=128' \
#-P'scraper2vdr -i /video/cache' \
#-P'streamdev-server \
```

```

VDRCMD="$VDRPRG $VDROPTIONS $VDRPLUGINS $"

KILL="/usr/bin/killall -q -TERM"

# Detect whether the DVB driver is already loaded
# and return 0 if it is loaded, 1 if not:
DriverLoaded()
{
    return 1
}

# Load all DVB driver modules needed for your hardware:
LoadDriver()
{
    return 0
}

# Unload all DVB driver modules loaded in LoadDriver():
UnloadDriver()
{
    return 0
}

# Load driver if it hasn't been loaded already:
if ! DriverLoaded; then
    LoadDriver
fi

while (true) do
    eval "$VDRCMD"
    if test $? -eq 0 -o $? -eq 2; then exit; fi
    echo "`date` reloading DVB driver"
    $KILL $VDRPRG
    sleep 10
    UnloadDriver
    LoadDriver
    echo "`date` restarting VDR"
done

```

Das vorgestellte Skript enthält einige Zeilen, die mit einem Kommentarzeichen (#) auskommentiert sind. Durch Entfernen des #-Zeichens können diese Zeilen aktiviert werden. Das erlaubt beispielsweise das Starten von *vdradmin*. Dieses Tool stellt ein Web-Interface für Ihren VDR zur Verfügung. Ich stelle es Ihnen im weiteren Verlauf dieses Buches noch vor.

Die Zeile `sudo hwclock --systohc --utc` stellt die BIOS-Uhr Ihres Rechners auf UTC, die heute gültige Weltzeit, ein. Diese Zeit erhält man aus der MESZ (Mittleuropäische Sommerzeit), indem man zwei Stunden addiert. Das Setzen der Uhr ist erforderlich, damit der Rechner später per ACPI (Advanced Configuration and Power Interface) automatisch gestartet werden kann, falls z. B. eine Timer-Aufnahme ansteht.

Anschließend werden die Systemvariablen für Sprache auf UTF-8 gestellt. Falls Sie ein VDR-Plugin mit einer eigenen in `/usr/local/lib`-gespeicherten Bibliothek übersetzt haben, kommentieren Sie bitte die Zeile mit `LD_LIBRARY_PATH` ein. VDR wird mit weiteren Parametern gestartet. So wird der Watchdog-Timer (`-w 60`) auf 60 Sekunden gesetzt. Sollte der VDR also für 60 Sekunden kein Lebenszeichen von sich geben, wird er neu gestartet. Um Log-Ausgaben zu minimieren, wird der Log-Level auf 0 gesetzt (`-l 0`).

Alle verwendeten VDR-Plugins sind mit ihren Parametern als Nächstes aufgeführt.

Kommentieren Sie bitte die Plugins aus, die Sie nicht benötigen. Alle hier aufgeführten Plugins werden wir noch im Laufe dieses Kapitels installieren.

Die Funktionen zum Laden der DVB-Treiber sind leer. Hier können Sie nach Bedarf eigene Treiber hinzufügen.

Das hier vorgestellte Skript sollte sich mit dem Namen *runvdr* im Verzeichnis */usr/local/bin* befinden und ausführbar sein (*sudo chmod 775 runvdr*).

Hinweis 7.2 (Windows und LINUX auf einer Festplatte) Wenn Sie Windows und LINUX auf einer Festplatte als Dual-Boot-System installiert haben, kann es sein, dass Windows mit der von LINUX gesetzten UTC-Zeit nicht klarkommt. Sie erkennen das daran, dass die Uhr in Windows nach einem LINUX-Boot um 1-2 Stunden falsch geht. Sollte das Problem bei Ihnen auftauchen, können Sie es wie folgt beheben: Öffnen Sie bitte den Registry Editor in Windows. Dieser Editor kann durch eine Eingabe von „regedit“ in der Suchleiste oder durch Ausführen eines Dialogs (Windows+R-Taste) geöffnet werden. Bestätigen Sie anschließend die Meldung zur Benutzerkontensteuerung mit „ja“. Wechseln Sie zum Pfad *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation* und legen Sie dort ein neues DWORD (32-Bit) mit dem Namen „RealTimeIsUniversal“ und dem Wert 1 an. Starten Sie danach Ihr Windows-System neu.

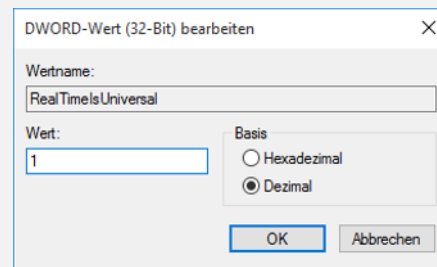


Abbildung 7.7: Mit dieser Einstellung versteht Windows UTC.

Vielleicht ist Ihnen aufgefallen, dass das VDR-Start-Skript die Zeile *-s /usr/local/bin/vdrshutdown.sh* enthält. Der Parameter *-s* übergibt das VDR Stopp-Skript *vdrshutdown.sh*, das sich ebenfalls im Verzeichnis */usr/local/bin* befindet und ebenfalls ausführbar sein soll. Das Skript hat den folgenden Inhalt:

Code-Ausschnitt 7.68: Komplettes VDR Stopp-Skript

```
#!/bin/bash
sudo hwclock --systohc --utc

NextTimer=$(( $1 - 600 )) # 10 minutes earlier

sudo bash -c "echo 0 > /sys/class/rtc/rtc0/wakealarm"
sudo bash -c "echo $NextTimer > /sys/class/rtc/rtc0/wakealarm"

sudo /sbin/poweroff
```

Eventuell wundern Sie sich, dass die Uhrzeit hier noch einmal auf UTC-Zeit gesetzt wird. Das liegt daran, dass der VDR die Möglichkeit hat, die Uhrzeit automatisch z. B. anhand der Satelliten-Transponderdaten zu korrigieren. Eine möglicherweise korrigierte Zeit wird dann ins BIOS geschrieben.

Beim Beenden übergibt VDR mehrere Parameter an das Stopp-Skript. Der erste Parameter, der im o. g. Skript in der Variablen *NextTimer* gespeichert wird, gibt die Zeit bis zur nächsten geplanten Aufnahme (plus eine eventuell eingestellte Vorlaufzeit) in Sekunden an. Damit der VDR auch garantiert zur geplanten Sendezeit gestartet ist, werden von dieser Zeit noch einmal 10 Minuten (600 Sekunden) abgezogen. Anschließend wird die neue Aufwachzeit (so sie denn vorhanden ist) ins BIOS geschrieben und der Rechner wird heruntergefahren.

Das letzte Skript, das uns zur Bedienung von VDR fehlt, ist eines, das den VDR beim Starten des Wohnzimmer-PCs automatisch startet.

Code-Ausschnitt 7.69: VDR Autostart-Skript

```
#!/bin/bash
DISPLAY=:0.0 xhost +
PATH=$PATH:/sbin:/usr/sbin
echo 'password' | sudo -S /usr/local/bin/runvdr
```



Möchten Sie nicht, dass der VDR automatisch startet, schreiben Sie einfach in die erste Zeile des Start-Skripts den Befehl `exit 1`. Die nachfolgenden Zeilen werden dann nicht mehr ausgeführt. Alternativ können Sie auch die Zeile `Stop VDR: pkill vdr` in die VDR-Konfigurations-Datei `commands.conf` aufnehmen. Sie können den VDR dann ganz bequem mit der Fernbedienung wegschalten.

Das Autostart-Skript setzt zunächst die `DISPLAY`-Variable auf den Standard-Bildschirm (also z. B. den Fernseher). Danach erhalten alle Klienten Zugriff (`xhost +`). Das ist nützlich, wenn Sie den VDR beispielsweise von einem entfernten Terminal aus starten möchten. Im Anschluss wird die Pfad-Variable auf die Binär-Verzeichnisse gesetzt, also die Verzeichnisse, die ausführbare Programme enthalten. Zuletzt wird die VDR-Start-Datei `/usr/local/bin/runvdr` aufgerufen. Dies geschieht als Benutzer `root`, damit der VDR mit Root-Rechten läuft. Geben Sie für `password` Ihr Root-Kennwort ein.



Normalerweise ist es keine gute Idee, VDR mit Root-Rechten laufen zu lassen. Bitte stellen Sie sicher, dass Ihr Rechner in Ihrem Netz geschützt ist, beispielsweise durch eine aktivierte Firewall und sichere Kennwörter. Sollte das nicht der Fall sein, ersetzen Sie die letzte Zeile durch den einfachen Aufruf des Start-Skripts (`/usr/local/bin/runvdr`).

Nachdem das Autostart-Skript erstellt worden ist, müssen wir dem System noch mitteilen, dass die Datei auch beim Start ausgeführt werden soll. Geben Sie dazu bitte auf der Konsole folgenden Befehl ein:

Code-Ausschnitt 7.70: Hinzufügen von Programmen zum Autostart

```
gnome-session-properties
```

Daraufhin öffnet sich ein Grafikfenster, welches die automatisch zu startenden Programme anzeigt (Abbildung 7.8).

Klicken Sie auf *Hinzufügen* und vergeben Sie einen Namen für das Autostart-Skript, etwa „VDR-Start“. Wählen Sie dann mit einem Klick auf *Durchsuchen* das VDR-Autostart-Skript aus. Unter *Kommentar* können Sie einen beliebigen Kommentar eingeben.

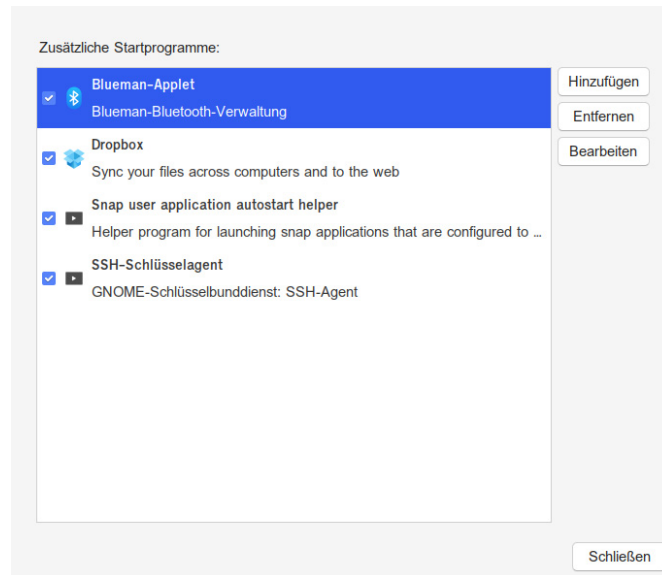


Abbildung 7.8: *gnome-session-properties* zeigt alle Programme an, die automatisch gestartet werden.

Zum Abschluss möchte ich Ihnen noch ein letztes Skript vorstellen, das wir später noch verwenden werden, um zwischen *VDR* und dem Mediacenter *KODI* hin- und herzuschalten. Dieses Skript können Sie ebenfalls im Verzeichnis */usr/local/bin* speichern und es z. B. *starte_xbmc.sh* nennen. Bitte denken Sie daran, auch dieses Skript ausführbar zu machen.

Code-Ausschnitt 7.71: XBMC Start-Skript

```
#!/bin/sh
/usr/local/bin/svdrpsend REMO off
sudo -u rudi DISPLAY=:0.0 pasuspender -- env KODI_AE_SINK=ALSA /usr/bin/kodi -fs --standalone
#sudo -u rudi DISPLAY=:0.0 /usr/local/src/xbmc/kodi.bin -fs --standalone
#sudo -u rudi DISPLAY=:0.0 /usr/lib/kodi/kodi.bin -fs --standalone
sleep 5
/usr/local/bin/svdrpsend REMO on
```

Zunächst wird das Programm *svdrpsend* dazu verwendet, die Fernbedienung für den *VDR* auszuschalten. Schließlich möchten wir nicht, dass der *VDR*, der nach dem Starten von *KODI* (Kapitel refkap:kodi) noch im Hintergrund läuft, blind Fernbedienungs-Kommandos empfängt. Danach wird *KODI* mit ALSA-Soundausgabe gestartet. Nachdem *KODI* beendet wurde, wartet das Skript fünf Sekunden, bis der *VDR* wieder läuft und schaltet dann die Fernbedienung wieder ein.

Das Programm *svdrpsend* können Sie aus dem *VDR*-Verzeichnis heraus so installieren:

Code-Ausschnitt 7.72: Installation von *svdrpsend*

```
cd ~/tv/VDR
sudo cp svdrpsend /usr/local/bin
```

Das oben gezeigte Skript werden wir später bei der Verwendung des *VDR*-Plugins *externalplayer* verwenden, in dem wir das Starten von *KODI* aus dem *VDR* heraus ermöglichen.

7.5 Speichern und servieren



Abbildung 7.9: Eine per USB eingebaundene 2,5 Zoll-Festplatte erweitert den Speicherplatz des Wohnzimmer-PCs. Aber Vorsicht: Für die externe Festplatte kann eine externe Spannungsversorgung erforderlich sein.

Im Dauer-Fernsehbetrieb wird der freie Speicherplatz auf der Festplatte eines Wohnzimmer-PCs schnell zu klein. Ein normaler HD-Film belegt gerne schon einmal 4...5 GB Speicherplatz. Es kann also Sinn machen, eine externe USB-Festplatte anzuschließen. Bitte denken Sie daran, dass Festplatten einen hohen Anlaufstrom haben und auch im Dauerbetrieb mehr als die 500 mA Strom brauchen können, die ein Wohnzimmer-PC an seinen USB-2-Anschlüssen zur Verfügung stellen kann. Sie müssen in diesem Fall also auch entweder mit einem aktiven Hub oder mit einem zusätzlichen USB-Netzteil und einem Y-Kabel arbeiten. Wenn Sie die externe Festplatte anstecken, erscheint diese unter LINUX als *sda*-Device. Zahlen hinter dem letzten Buchstaben kennzeichnen nicht fortlaufende Festplatten (wie z. B. beim *lirc*-Device), sondern verschiedene Partitionen (sofern vorhanden) auf der *a*-Platte. Weitere Platten erhalten dann den Buchstaben *b* statt *a*. Das zweite Device würde also den Namen */dev/sdb* tragen. Da wir die externe Festplatte in das */video*-Verzeichnis einhängen (*mounten*) wollen, sollte das */video*-Verzeichnis leer sein. Falls sich in Ihrem Video-Verzeichnis noch Dateien befinden, werden diese beim Einhängen der externen Festplatte verdeckt und erst wieder

sichtbar, nachdem die externe Platte ausgehängt worden ist. Sie können den Inhalt des gesamten */video*-Verzeichnisses so löschen:

Code-Ausschnitt 7.73: Löschen des Video-Verzeichnisses

```
cd /video
rm -rf *
```



Denken Sie bitte daran: Der Befehl `rm -rf *` löscht den kompletten Inhalt eines Verzeichnisses. Seien Sie also sicher, dass Sie sich im richtigen Verzeichnis befinden, bevor Sie den Befehl aufrufen. Falls Sie sich im Verzeichnis */* befinden, können Sie dabei zusehen, wie Ihr Betriebssystem sich in Luft auflöst. Nachdem das */video*-Verzeichnis leer ist, binden wir die externe Festplatte darauf ein, nachdem wir sie formatiert haben. Sie können sich das Formatieren natürlich sparen, wenn die Platte bereits ein Dateisystem enthält.

Code-Ausschnitt 7.74: Erstellen eines Dateisystems

```
sudo fdisk /dev/sda
sudo mkfs -t ext4 /dev/sda1
```

Der *fdisk*-Befehl erlaubt es Ihnen, Partitionen auf Ihrer Festplatte zu erstellen. Der *mkfs*-Befehl (*make filesystem*) erzeugt das Dateisystem, im oberen Fall vom Typ *ext4*, auf der ersten Partition */dev/sda1*.

Sie können auch NTFS-Partitionen unter LINUX mounten, lesen und schreiben. Dafür gibt es das Programm ntfs-3g. Wenn Windows NTFS-Partitionen eingebunden werden sollen, kann das mit dem Aufruf



Code-Ausschnitt 7.75: Einbinden von NTFS-Partitionen

```
sudo ntfs-3g /dev/sda1 /mnt
```

erledigt werden. Das Beispiel bindet die erste Partition einer NTFS-Platte in das Verzeichnis /mnt ein. Für den Betrieb mit VDR kann ich NTFS als Dateisystem nicht empfehlen, da es spürbar langsamer ist als ein ext-Dateisystem.

Nach diesen vorbereitenden Schritten kann die externe Festplatte im Verzeichnis `/video` eingehängt werden. Der Befehl

Code-Ausschnitt 7.76: Einbinden einer externen USB-Festplatte

```
sudo mount /dev/sda1 /video
```

erledigt das für uns. Ab sofort werden Ihre VDR-Filme auf der externen Platte gespeichert und nicht mehr auf der internen Festplatte. Nach einem Reboot des Systems ist der Mount-Punkt aber leider wieder verschwunden. Um ihn dauerhaft und reboot-sicher einzubinden, fügen Sie bitte in der Datei `/etc/fstab` die folgende Zeile hinzu:

Code-Ausschnitt 7.77: Dauerhaftes Einbinden einer externen USB-Festplatte

```
/dev/sda1 /video ext4 rw,auto,users 0 0
```

Das Dateisystem `ext4` müssen Sie natürlich mit dem Dateisystem der externen Platte ersetzen.

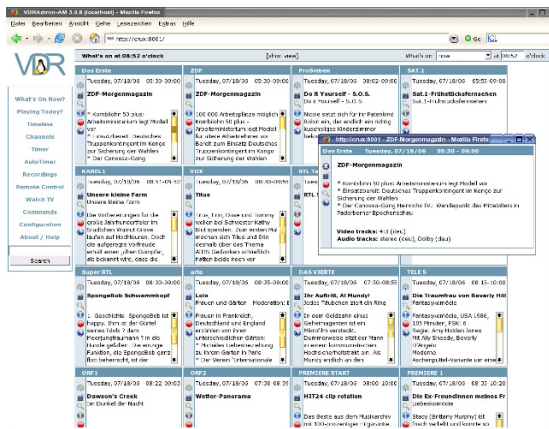


Abbildung 7.10: VDRAdmin-am erlaubt das Programmieren von Aufzeichnungen vom Internet aus (Quelle: vdr-wiki.de).

Mit VDRAdmin-am gibt es eine Software, die auf einem Webserver läuft, der mit dem VDR kommuniziert. Auf diese Art und Weise haben Sie die Möglichkeit, Timer zur Aufzeichnung von Fernsehsendungen mit Ihrem Handy zu setzen, das aktuelle Fernsehprogramm im EPG (*Electronic Program Guide*) zu sehen, Aufnahmen zu löschen und vieles mehr. Auf einem Wohnzimmer-PC läuft VDRAdmin-am ausreichend schnell - ganz im Gegensatz zu dem Betrieb auf einem Embedded Board.

VDRAdmin-am setzt das *perl*-Modul voraus, das wir zusammen mit einigen weiteren Paketen im nächsten Schritt installieren.

Code-Ausschnitt 7.78: Installation der Abhängigkeiten für VDRAdmin-am

```
sudo apt-get install libauthen-sasl-perl libdigest-hmac-perl libio-socket-inet6-perl
```

Den Quelltext von VDRAdmin-am finden Sie auf <http://projects.vdr-developer.org/git/vdradmin-am.git/> [69]. Clonen Sie die aktuelle Version mit

Code-Ausschnitt 7.79: Auschecken von VDRAdmin-am

```
git clone git://projects.vdr-developer.org/vdradmin-am.git
```

in ein Verzeichnis Ihrer Wahl. Wechseln Sie danach in das Verzeichnis *vdradmin-am.git* und führen sie die folgenden Befehle aus:

Code-Ausschnitt 7.80: Installation von VDRAdmin-am

```
cd vdradmin-am.git
sudo LANG=de_DE ./make.sh install
./make.sh cvs
./make.sh po
```

Die letzten beiden Zeilen erstellen einige Links und die Internationalisierung (Sprachen). Nach dem Installieren und Anlegen der Links und der Internationalisierung rufen Sie bitte den *vdradmin*-Daemon einmalig mit dem Parameter *--config* auf:

Code-Ausschnitt 7.81: Konfiguration von VDRAdmin-am

```
vdradmind.pl --config
```

Nach dem Konfigurieren (Sie müssen einige Fragen beantworten) können Sie den Daemon mit

Code-Ausschnitt 7.82: Starten von VDRAdmin-am

```
vdradmind.pl
```

starten. Von jetzt an können Sie sich mit einem Browser unter dem Port 8001 auf dem Wohnzimmer-PC einwählen, um die Oberfläche von VDRAdmin-am im Browser zu sehen. Bitte denken Sie daran, dass der VDR laufen muss, damit VDRAdmin-am funktioniert. Der Aufruf im Browser lautet

Code-Ausschnitt 7.83: VDRAdmin-am Browser-Aufruf

```
http://ip-adresse:8001
```

Bitte ersetzen Sie *ip-adresse* mit der IP-Adresse Ihres Wohnzimmer-PCs.

7.6 Schöner fernsehen

Der VDR stellt zwar ein paar Standard-Skins für das OSD bereit, aber keines davon sieht besonders hübsch aus. Ein Skin, das mir besonders gut gefällt, heißt Metrix HD. Da das Aussehen eines OSD und damit auch die Skins reine Geschmacksache sind, erkläre ich Ihnen in diesem Abschnitt, wie Sie zunächst das *skindesigner*-Plugin installieren können und anschließend ein Skin Ihrer Wahl (am Beispiel von Metrix HD) installieren können. Mit dem *skindesigner* hat Louis Braun ein Plugin geschaffen, das XML-basierte (Extensible Markup Language) Skins laden und darstellen kann. Das Plugin selbst ist auf <https://projects.vdr-developer.org/projects/plg-skindesigner/wiki> [70] zu Hause. Hier gibt es auch eine Beschreibung, wie die Skins aufgebaut sein müssen, damit *skindesigner* mit ihnen umgehen kann. Da es derzeit keine Webseite gibt, welche alle verfügbaren Skins für den VDR zusammenfasst, empfehle ich eine Google-Suche mit den Begriffen *vdr*, *skindesigner* und *skin*. Um den *skindesigner* zu installieren, klonen wir die neueste Version auf dem Git-Repository.



Abbildung 7.11: Metrix HD ist ein wunderschönes Skin für den VDR. (Quelle: vdr-portal.de [56]).

Code-Ausschnitt 7.84: Auschecken des Plugins *skindesigner*

```
cd ~/tv/VDR/PLUGINS/src
git clone git://projects.vdr-developer.org/vdr-plugin-skindesigner.git
mv vdr-plugin-skindesigner skindesigner
```

Das Übersetzen des *skindesigner*-Plugins setzt einige Bibliotheken voraus, die wir installieren.

Code-Ausschnitt 7.85: Abhängigkeiten für das Plugin *skindesigner* installieren

```
sudo apt-get install libcurl4-gnutls-dev libxml2-dev librsvg2-dev libcairo-dev
```

Das *skindesigner*-Plugin kann nun übersetzt und installiert werden.

Code-Ausschnitt 7.86: Übersetzen des Plugins *skindesigner*

```
cd ~/tv/VDR
make
sudo make install
```

Ein paar Skins stellt der *skindesigner* selbst zur Verfügung, so auch Metrix HD. Das Start-Skript, das ich Ihnen weiter oben vorgestellt habe, beinhaltet bereits einen Eintrag für *skindesigner*. Die entsprechende Zeile lautet.

Code-Ausschnitt 7.87: *skindesigner*-Plugin im VDR-Start-Skript

```
-P 'skindesigner -l /var/lib/vdr/plugins/skindesigner/logos' "
```

Im VDR stehen unter *Menü* → *Plugins* → *skindesigner* einige Skins zur Vorschau und Installation bereit. Diese nutzen wir aber nicht. Metrix wird bereits zusammen mit dem *skindesigner* vorinstalliert. Wählen Sie Metrix HD unter *Menü* → *Einstellungen* → *OSD* → *Oberfläche* aus. Nach dem Schließen des OSD dauert es eine Weile, bis das neue Skin initialisiert ist. Beim Aufruf werden Sie feststellen, dass einige Dinge wie die Temperaturanzeige für die CPU oder die Wettervorhersage noch nicht funktionieren. Diese Features installieren wir in Kürze. Falls Ihnen die Anordnung des Skins nicht gefällt, können Sie diese nach Belieben in den Einstellungen des *skindesigner*-Plugins ändern.



Je aufwendiger ein Skin für den VDR/skindesigner ist, desto eher ist eine OpenGL/EGL-Beschleunigung für das Ausgabe-Plugin erforderlich. Aufwendige Skins ruckeln merklich auf einem Wohnzimmer-PC, falls Sie ein Ausgabe-Plugin ohne OpenGL/EGL-Unterstützung verwenden.

Sorgen wir aber nun dafür, dass das Skin die aktuelle Temperatur des Wohnzimmer-PCs anzeigt. Das Plugin *skindesigner* erwartet das Temperaturanzeige-Skript *temperatures* im Verzeichnis */usr/local/lib/vdr/skindesigner/scripts*.

Code-Ausschnitt 7.88: Skript zum Auslesen der CPU-Temperatur für *skindesigner*

```
#!/bin/bash

OUTPUTFLDR="/tmp/skindesigner/"
mkdir -p ${OUTPUTFLDR}

# if the script is executed from system_information script set the locale back for "°C"
LANG=de_DE.UTF-8

# there can be 4 files, cpu, gpu, pccase, motherboard

rm -f ${OUTPUTFLDR}/cpu ${OUTPUTFLDR}/pccase ${OUTPUTFLDR}/gpu ${OUTPUTFLDR}/motherboard

# intel core-i cpu temp
sensors -A coretemp-isa-0000 | grep "Core 0" | awk '{print $3}' | tr -d "+" > ${OUTPUTFLDR}/cpu

# pc case temp
#sensors -A acpitz-virtual-0 | grep "temp1" | awk '{print $2}' | tr -d "+" > ${OUTPUTFLDR}/↵
pccase

# motherboard temp
sensors -A pch_skylake-virtual-0 | grep "temp1" | awk '{print $2}' | tr -d "+" > ${OUTPUTFLDR}/↵
motherboard

# nvidia gpu temp
# nvidia-settings must be run as the user of the x server
# GPU=`nvidia-settings -c :0 -t -query GPUCoreTemp | head -n 1`
#GPU=`nvidia-smi --query-gpu=temperature.gpu --format=csv,noheader`
#echo "${GPU}°C" > ${OUTPUTFLDR}/gpu
```

Kernstück dieses Skripts ist das Programm *sensors*, das die Temperatursensoren des Mainboards ausliest. Im oben angegebenen Skript habe ich die Werte für meine INTEL CPU/GPU eingetragen. Falls Sie eine NVidia-GPU haben, ändern Sie die Einträge entsprechend. Prinzipielle Auskunft über alle Temperatursensoren erhalten Sie nach der Eingabe von

Code-Ausschnitt 7.89: Auslesen der Temperatursensoren

```
sensors
```

in der Konsole. Gegebenenfalls müssen Sie dazu noch das Programm *lm-sensors* so installieren:

Code-Ausschnitt 7.90: Installation des Programms *lm-sensors*

```
sudo apt-get install lm-sensors
```

Das Skript *vdrstats* gibt über CPU- und Speicherverbrauch des VDR Auskunft.

Code-Ausschnitt 7.91: Auslesen der VDR-Auslastung für *skindesigner*

```
#!/bin/bash

# please update this script to fit your needs
# this script is call every time the according viewelement will be drawn, so keep it short and fast ;)

OUTPUTFLDR="/tmp/skindesigner/"
mkdir -p ${OUTPUTFLDR}

# there can be 2 files, vdrcpu and vdrmem
rm -f ${OUTPUTFLDR}/vdrcpu ${OUTPUTFLDR}/vdrmem

read vdr_cpu vdr_mem < < (ps -C vdr -o %cpu=%mem= | sort | tail -n1)
echo $vdr_cpu > ${OUTPUTFLDR}/vdrcpu
echo $vdr_mem > ${OUTPUTFLDR}/vdrmem
```

Auch dieses Skript wird im Verzeichnis `/usr/local/lib/vdr/skindesigner/scripts` erwartet. Falls Sie ein anderes Konfigurations-Verzeichnis verwenden, müssen Sie den Prefix `/usr/local/lib/vdr` gegebenenfalls ändern.

Im nächsten Schritt installieren wir die Wettervorhersage für Metrix HD. Diese wird auch für viele andere Skins verwendet, die das Wetter anzeigen. Das Wettervorhersage-Plugin heißt *weatherforecast*. Wir installieren es mit den benötigten Abhängigkeiten wie folgt:

Code-Ausschnitt 7.92: Installation des Plugins *weatherforecast*

```
cd ~/tv/VDR/PLUGINS/src
git clone git://projects.vdr-developer.org/vdr-plugin-weatherforecast.git weatherforecast
sudo apt-get install libcurl4-openssl-dev libjansson-dev
cd ../..
make
sudo make install
```



Vielleicht ist Ihnen aufgefallen, dass ich hinter dem git clone-Kommando einen Verzeichnisnamen weatherforecast angegeben habe. In diesem Fall kopiert der git-Befehl alle Dateien in das angegebene Verzeichnis. Wir sparen uns dann, das Verzeichnis vdr-plugin-weatherforecast umzubenennen.

Da das Plugin *weatherforecast* keine Setup-Parameter hat, können wir in unserer VDR-Startdatei `/usr/local/bin/runvdr` folgende Zeile ergänzen:

Code-Ausschnitt 7.93: Hinzufügen des *weatherforecast*-Plugins in die *runvdr.sh*

```
-Pweatherforecast \
```

Beim Start sucht das Plugin automatisch nach einem Ort in der Nähe der eigenen IP-Adresse und stellt dessen Wetterdaten dar. Um die Abfragen beim Wetter-Server zu limitieren, cached das Plugin die Abfragen. Bitte haben Sie Geduld, falls das Wetter nicht sofort angezeigt wird. Eventuell sind die 1.000 freien Abfragen beim Wetterdienst bereits aufgebraucht.

Als Nächstes machen wir uns daran, das Metrix HD-Skin mit schönen Senderlogos auszustatten. Die Logos laden wir von <https://github.com/FrodoVDR/channellogos> [71] im Verzeichnis `/var/lib/vdr/plugins/skindesigner/logos`, das wir zuvor angelegt haben, herunter. Ich verwende die light-Logos.

Code-Ausschnitt 7.94: Herunterladen der Senderlogos

```
sudo su
mkdir -p /var/lib/vdr/plugins/skindesigner/logos
cd /var/lib/vdr/plugins/skindesigner/logos
git clone https://github.com/FrodoVDR/channellogos
cp channellogos/logos-light/* .
```

Nach einem Neustart des VDR sind die Senderlogos sichtbar.

Im Folgenden werden wir uns noch ein paar weitere Plugins für den VDR näher anschauen. Dazu zählt das Plugin *tvscraper*. Dieses scannt im Hintergrund Filmdatenbanken und stellt dann für Aufnahmen Filmposter und Inhalt in einer schönen Darstellung zur Verfügung.

tvscrapper benötigt die Installation der folgenden Entwicklungsumgebungen:

Code-Ausschnitt 7.95: Abhängigkeiten für *tvscrapper*

```
sudo apt-get install libsqlite3-dev libcurl-dev libxml2-dev libjansson-dev
```

Nach der Installation dieser Abhängigkeiten kann *tvscrapper* so heruntergeladen und installiert werden:

Code-Ausschnitt 7.96: *tvscrapper* installieren

```
cd ~/tv/VDR/PLUGINS/src
git clone https://github.com/vdr-projects/vdr-plugin-tvscrapper tvscrapper
cd tvscrapper
make
sudo make install
```

Um den *tvscrapper* zu aktivieren, fügen Sie bitte die folgende Zeile in der Datei */usr/local/bin/runvdr* hinzu:

Code-Ausschnitt 7.97: Einbinden des Plugins *tvscrapper*

```
-P'tvscrapper -d /video/cache' \
```

Nach der Installation können über die Einstellungen des Plugins die Verzeichnisse und auch Sender ausgewählt werden, für die der sogenannte Scraper aktiv sein soll.



Abbildung 7.12: *tvscrapper* zeigt in einer schönen Darstellung Filmplakate und Inhalte an. Quelle: *vdrportal.de* [56]

Der Parameter *-d /video/cache* gibt an, dass der Scraper die heruntergeladenen Filmposter und Inhalte im Verzeichnis */video/cache* zwischenspeichern darf. Da dieses Verzeichnis nicht existiert, müssen wir es noch anlegen.

Code-Ausschnitt 7.98: Cache-Verzeichnis für das Plugins *tvscrapper*

```
sudo mkdir /video/cache
chmod 775 /video/cache
touch /video/cache/.update
```

Der *touch*-Befehl der letzten Zeile des o. g. Code-Ausschnitts sorgt dafür, dass eine leere Datei mit dem Namen *.update* im Verzeichnis */video/cache* erzeugt wird. Die wiederum sorgt dafür, dass der VDR das *cache*-Verzeichnis nicht versehentlich löscht, da es sich um

ein leeres Film-Verzeichnis handeln könnte. Das *externalplayer*-Plugin von Uli Eckhardt erlaubt es, andere Medienplayer wie etwa *KODI* aus dem *VDR* heraus aufzurufen. Das ist sinnvoll, wenn man vielleicht eine DVD oder eine Blu-ray schauen möchte. Der *externalplayer* benötigt keine weiteren Installationspakete. Er kann so heruntergeladen und installiert werden:

Code-Ausschnitt 7.99: Installation des Plugins *externalplayer*

```
cd ~/tv/VDR/PLUGINS/src
git clone http://git.uli-eckhardt.de/vdr-plugin-externalplayer.git externalplayer
cd externalplayer
make
sudo make install
```

Aktiviert wird der *externalplayer* durch folgenden Eintrag in der Datei */usr/local/bin/runvdr*:

Code-Ausschnitt 7.100: Aktivieren des Plugins *externalplayer*

```
-Pexternalplayer -C /usr/local/lib/vdr/plugins/externalplayer/externalplayer.conf \
```

Nach diesen Schritten ist das Plugin aber noch nicht einsatzbereit. Es erwartet noch eine kleine Konfigurationsdatei.

Code-Ausschnitt 7.101: Konfigurieren des Plugins *externalplayer*

```
sudo mkdir -p /usr/local/lib/vdr/plugins/externalplayer
sudo cp ~/VDR/PLUGINS/src/externalplayer/examples/externalplayer.conf /usr/local/lib/vdr/↵
plugins/externalplayer/
```

Die Datei *externalplayer.conf* können Sie mit dem folgenden Inhalt füllen, wenn der *VDR* auf das Mediacenter *KODI* (Kapitel 8) umschalten soll:

Code-Ausschnitt 7.102: Konfiguration des Plugins *externalplayer*

```
{
  MenuEntry = "KODI";
  Command = "/usr/local/bin/start_xbmc.sh";
  OutputMode = extern;
  InputMode = normal; # Kodi should be configured for LIRC.
}
```

Diese Konfigurationsdatei greift auf das Shell-Skript *start_xbmc.sh* zurück, das ich Ihnen bereits weiter oben vorgestellt habe. Wenn Sie nun noch *KODI* installieren (Kapitel 8), können Sie bequem mit Ihrer Fernbedienung zwischen *VDR* und *KODI* wechseln.

Ein weiteres nützliches Plugin ist *osdteletext*. Mit Hilfe dieses Plugins - Sie ahnen es - kann der Teletext eines Fernsehsenders im OSD des *VDR* angezeigt werden. Es wird so heruntergeladen und installiert:

Code-Ausschnitt 7.103: Installation des Plugins *osdteletext*

```
cd ~/tv/VDR/PLUGINS/src
git clone https://github.com/rofafor/vdr-plugin-osdteletext osdteletext
cd osdteletext
make
sudo make install
```

Die Aktivierung des Plugins läuft ähnlich ab wie die der anderen Plugins. Tragen Sie bitte die folgende Zeile in die Datei */usr/local/bin/runvdr* ein und starten Sie danach den *VDR* neu:

Code-Ausschnitt 7.104: Aktivieren des Plugins *osdteletext*

```
-P'osdteletext' \
```

Abbildung 7.13: *osdteletext* zeigt den Teletext im OSD des VDR.Quelle: `vdrportal.de`.

Beim Ausrichten von Satelliten-Schüsseln kann es hilfreich sein, die Signalstärke eines Fernsehsignals auslesen zu können. Diese Möglichkeit bietet das Plugin *femon*. Darüber hinaus zeigt es Senderinformationen wie Bitrate oder Audio-Kodierung an. Das Plugin wird wie folgt heruntergeladen und installiert:

Code-Ausschnitt 7.105: Installation des Plugins *femon*

```
cd ~/tv/VDR/PLUGINS/src
git clone https://github.com/rofafor/vdr-plugin-femon femon
cd femon
make
sudo make install
```

Auch für *femon* ist die Aktivierung durch Eintrag in der VDR-Startdatei sehr einfach. Fügen Sie bitte die folgende Zeile in der Datei `/usr/local/bin/runvdr` hinzu:

Code-Ausschnitt 7.106: Aktivierung des Plugins *femon*

```
-Pfemon \
```



Abbildung 7.14: *femon* zeigt Signaldaten eines Senders an, hier am Beispiel eines DVB-T-Senders. Quelle: vdr-wiki.de [55].

Das Plugin *tvguideng* stellt das aktuelle und zukünftige Fernsehprogramm in einer übersichtlichen Form dar.



Abbildung 7.15: *tvguideng* stellt das TV-Programm übersichtlich dar, hier zusammen mit dem Shady-Skin. Quelle: <http://anthra.de/shady-skin> [72].

Das Einbinden dieses Plugins in die VDR-Startdatei gestaltet sich sehr einfach.

Code-Ausschnitt 7.107: Aktivierung des Plugins *tvguideng*

```
-Ptvguideng \
```

Allerdings musste ich zum Kompilieren des Plugins zwei Patches aus dem VDR-Portal herunterladen und anwenden, bevor ich dieses Plugin innerhalb der neuen VDR-Version fehlerfrei übersetzen konnte. Laden wir also in einem ersten Schritt das Plugin selbst herunter.

Code-Ausschnitt 7.108: Herunterladen des Plugins *tvguideng*

```
cd ~/tv/VDR/PLUGINS/src
git clone git://projects.vdr-developer.org/vdr-plugin-tvguideng.git tvguideng
cd tvguideng
```

Der erste Patch sieht wie folgt aus:

Code-Ausschnitt 7.109: Patch 1 für das Plugin *tvguideng*

```
Index: b/definitions.h
=====
--- a/definitions.h
+++ b/definitions.h
@@ -234,7 +234,7 @@
 };

 enum class eDetailedHeaderST {
- title = eScraperHeaderST::count,
+ title = (int)eScraperHeaderST::count,
 shorttext,
 start,
 stop,
@@ -249,7 +249,7 @@
 };

 enum class eDetailedHeaderIT {
- daynumeric = eScraperHeaderIT::count,
+ daynumeric = (int)eScraperHeaderIT::count,
 month,
 year,
 running,
@@ -375,7 +375,7 @@
 };

 enum class eDetailedEpgST {
- title = eScraperST::count,
+ title = (int)eScraperST::count,
 shorttext,
 description,
 start,
@@ -393,7 +393,7 @@
 };

 enum class eDetailedEpgIT {
- daynumeric = eScraperIT::count,
+ daynumeric = (int)eScraperIT::count,
 month,
 year,
 running,
@@ -543,4 +543,4 @@
 count
 };

-#endif //__DEFINITIONS_H
\ No newline at end of file
+#endif //__DEFINITIONS_H
```


Der zweite Patch sieht so aus:

Code-Ausschnitt 7.110: Patch 2 für das Plugin *tvguideng*

```

Index: b/channelgroups.h
=====
--- a/channelgroups.h
+++ b/channelgroups.h
@@ -16,6 +16,19 @@
     int channelStop;
     string name;
 public:
+   cChannelGroup(const cChannelGroup &ChannelGroup)
+   {
+       *this = ChannelGroup;
+   };
+   cChannelGroup& operator= (const cChannelGroup &ChannelGroup)
+   {
+       this->id = ChannelGroup.id;
+       this->channelStart = ChannelGroup.channelStart;
+       this->channelStop = ChannelGroup.channelStop;
+       this->name = ChannelGroup.name;
+   }
+   return *this;
+ };
+   cChannelGroup(string name, int id);
+   virtual ~cChannelGroup(void);
+   int GetId(void) { return id; };
Index: b/switchtimer.h
=====
--- a/switchtimer.h
+++ b/switchtimer.h
@@ -11,6 +11,20 @@
     tChannelID channelID;
     int switchMinsBefore;
     int announceOnly;
+   cSwitchTimer(const cSwitchTimer &SwitchTimer)
+   {
+       *this = SwitchTimer;
+   };
+   cSwitchTimer& operator= (const cSwitchTimer &SwitchTimer)
+   {
+       this->eventID = SwitchTimer.eventID;
+       this->startTime = SwitchTimer.startTime;
+       this->channelID = SwitchTimer.channelID;
+       this->switchMinsBefore = SwitchTimer.switchMinsBefore;
+       this->announceOnly = SwitchTimer.announceOnly;
+   }
+   return *this;
+ };
+   cSwitchTimer(void);
+   cSwitchTimer(const cEvent* Event);
+   bool Parse(const char *s);

```



Die hier vorgestellten Patches finden Sie auch im Download-Bereich zu diesem Buch.

Speichern Sie die Patches unter den Namen *1.patch* und *2.patch* im *tvguideng*-Plugin-Verzeichnis, wenden Sie sie an und übersetzen Sie anschließend das Plugin.

Code-Ausschnitt 7.111: Anwenden der Patches und Übersetzen des Plugins *tvguideng*

```
cd ~/tv/VDR/PLUGINS/src/skindesigner/libskindesignerapi
make
sudo make install
cd ~/tv/VDR/PLUGINS/src/tvguideng
cat 1.patch | patch -sp1
cat 2.patch | patch -sp1
make
sudo make install
```

tvguideng benötigt die API von *skindesigner*, die ebenfalls installiert wurde.

Als nächstes Plugin dieses Kapitels installieren wir noch die EPG-Suche. *epgsearch* erlaubt es Ihnen, im EPG der Fernsehsender nach bestimmten Sendungen oder Serien zu suchen und beispielsweise Dauertimer einzurichten. *epgsearch* hat keine weiteren Abhängigkeiten. Allerdings ist auch bei diesem Plugin ein Patch erforderlich, der wie folgt aussieht:

Code-Ausschnitt 7.112: Patch für das Plugin *epgsearch*

```
--- a/conflictcheck.c
+++ b/conflictcheck.c
@@ -584,12 +584,14 @@
    LogFile.Log(3, "add pending timers");
    // if we have pending timers add them to the current start list
-   for (it = pendingTimers.begin(); it != pendingTimers.end(); ++it) {
+   it = pendingTimers.begin();
+   while (it != pendingTimers.end())
+   {
+       if ((*it) && (*it)->stop > checkTime->evaltime)
+       checkTime->startingTimers.insert(*it);
+       checkTime->startingTimers.insert(*it);
+       pendingTimers.erase(*it);
+   }
+   it = pendingTimers.begin();
+   }
    LogFile.Log(3, "attach starting timers");
    // handle starting timers
    for (it = checkTime->startingTimers.begin(); it != checkTime->startingTimers.end(); ++it) {
@@ -600,10 +602,11 @@
        if (NeedsDetachReceivers) { // but needs to detach all others?
            // disable running timers
            std::set<cConflictCheckTimerObj*, TimerObjSort::iterator it2 = devices[device].recTimers.begin();
-           for (; it2 != devices[device].recTimers.end(); ++it2) {
+           while (it2 != devices[device].recTimers.end()) {
+               LogFile.Log(3, "stopping timer '%s' (%s, channel %s) at %s on device %d ←
+               because of higher priority", (*it2)->timer->Fil
e(), DAYDATETIME((*it2)->start), CHANNELNAME((*it2)->timer->Channel()), DAYDATETIME(checkTime->evaltime), device + 1);
                AddConflict((*it2), checkTime, pendingTimers);
                devices[device].recTimers.erase(*it2);
+               it2 = devices[device].recTimers.begin();
                Conflicts++;
            }
        }
```

Speichern Sie den Patch doch ebenfalls unter dem Namen *1.patch* im Plugin-Verzeichnis von *epgsearch*, laden Sie danach das Plugin herunter und übersetzen Sie es so:

Code-Ausschnitt 7.113: Das Plugin *epgsearch* herunterladen und übersetzen

```
cd ~/tv/VDR/PLUGINS/src
git clone https://projects.vdr-developer.org/git/vdr-plugin-epgsearch.git epgsearch
cd epgsearch
cat 1.patch | patch sp1
make
sudo make install
```

Das Aktivieren von *epgsearch* erfolgt auf die übliche Art und Weise. Die Konfiguration des Plugins lässt keine Wünsche offen. Bei Bedarf erhalten Sie sogar eine E-Mail, wann welche Sendung läuft. Weitere Information gibt es unter <http://www.vdr-wiki.de/wiki/index.php/Epgsearch-plugin> [55].

Vielleicht haben Sie schon einmal einen Receiver mit einem eingebauten Display gesehen, welches das aktuelle EPG zeigt oder Informationen zu Aufnahmen auf der Festplatte. Einige teure Receiver (z. B. Vu+-Serie) lassen sich sogar komplett über ein eingebautes Display bedienen, ohne dass der Fernseher eingeschaltet werden muss. Als letztes Plugins dieses Abschnitts zeige ich Ihnen weitere Plugins, die zusammengehören. Eines lädt neben dem aktuellen EPG weitere Informationen zu Filmen oder Serien aus dem Internet und speichert diese Informationen inklusive der dazugehörigen Filmplakate in einer Datenbank. Es nennt sich *epgd*. Das steht für EPG-Daemon. Eigentlich ist der EPG-Daemon gar kein Plugin, sondern ein Dienst, der seine Tätigkeiten unauffällig im Hintergrund verrichtet.



Abbildung 7.16: 3,5 Zoll HDMI-Bildschirme sind bereits für 25 € zu haben (Quelle: <https://www.amazon.de> [73]).

Das zweite Plugin nennt sich *osd2web*. Es kann zwei Aufgaben erfüllen. Zum einen stellt es in einem Webbrowser aktuelle Informationen (z. B. das EPG samt Filmplakat) dar, zum anderen erlaubt es die Darstellung des gesamten VDR-OSDs (On Screen Display), da es als Skin für den VDR agieren kann. Beide Eigenschaften können in einem

Webbrowser umgeschaltet werden. Damit kann beispielsweise ein kleiner Browser im Vollbild auf einem 3,5 Zoll-Display betrieben werden, der genau diese Informationen anzeigt oder über den sich der VDR steuern lässt. Vielleicht haben Sie ja sogar Lust, ein eigenes Gehäuse für Ihren Receiver zu entwerfen und das Display dort einzubauen. Soll *osd2web* auch Bilder aus Film- und Seriendatenbanken anzeigen, benötigt es zusätzlich das Plugin *epg2vdr*, welches sich um die Extraktion dieser Information kümmert.

Das dritte Plugin nennt sich *scraper2vdr*. Es sorgt dafür, dass Bilder aus der Film- und Seriendatenbank nicht wie bei *epg2vdr* über *osd2web* auf dem Browser landen, sondern direkt im Skin des VDR angezeigt werden können.

Wenn Sie die hier zuletzt genannten Plugins installieren, achten Sie bitte darauf, nicht gleichzeitig ein weiteres Plugin zu installieren, das Bilder im Skin anzeigt. Die hier genannte Kombination ist also dann sinnvoll, wenn Sie das VDR-EPG und weitere Informationen in einem Browser oder einem kleinen zusätzlichen Display anzeigen wollen.



Grundlage der zuletzt genannten Plugins stellt der EPG-Daemon *epgd* dar. Daher installieren wir ihn zuerst. Grundlage für *epgd* wiederum ist eine Datenbank. Unter Ubuntu stehen zwei verschiedene Datenbanken zur Installation zur Verfügung: *mysql* und *mariadb*. *epgd* funktioniert nicht mit der aktuellen Version von *mysql*. In dieser Version sind einige Befehle nicht mehr vorhanden, die *epgd* benötigt. Installieren Sie daher bitte unbedingt *mariadb*:

Code-Ausschnitt 7.114: Installation der Datenbank *mariadb*

```
sudo apt install mariadb-server
```

Auf meinem Ubuntu 19.10 ließ sich *mariadb* nach der Installation nicht starten, da es einen Konflikt mit *Apparmor* gab. Das ist eine Software, die anderen Programmen Rechte verleiht oder diese einschränkt. Damit die Datenbank richtig läuft, habe ich *Apparmor* so deaktiviert:

Code-Ausschnitt 7.115: Deaktivieren von *apparmor* für *mysqld*

```
sudo apt install apparmor-utils
sudo aa-disable /usr/sbin/mysqld
```

Sie können die Datenbank so starten und stoppen:

Code-Ausschnitt 7.116: Starten und Stoppen der Datenbank

```
sudo /etc/init.d/mysql start
sudo /etc/init.d/mysql stop
```

Nach einem Systemneustart sollte die Datenbank automatisch gestartet werden. Im nächsten Schritt laden wir den *epgd* herunter und installieren alle benötigten Abhängigkeiten.

Code-Ausschnitt 7.117: Herunterladen von *epgd* und Installieren aller Abhängigkeiten

```
cd ~/tv/VDR/PLUGINS/src
git clone git://projects.vdr-developer.org/vdr-epg-daemon.git
sudo apt install libarchive-dev libz-dev libssl-dev libcurl4-nss-dev libxslt1-dev libxml2-dev ↵
libmysqlclient-dev libjpeg-dev uuid-dev libjansson-dev libmicrohttpd-dev libimlib2-dev ↵
python libpython libpython-dev python-dev mailutils libsystemd-dev
```

Die Übersetzung und Installation von *epgd* folgt wieder dem üblichen Muster:

Code-Ausschnitt 7.118: Übersetzen und Installieren von *epgd*

```
cd vdr-epg-daemon
make
sudo make install
```

Die Installations-Routine installiert dabei praktischerweise einige Start-Skripte, auf die wir im weiteren Verlauf noch zurückkommen werden. Vor dem ersten Start von *epgd* müssen noch Eintragungen in der Datenbank-Konfiguration vorgenommen werden. Stoppen Sie die Datenbank zuerst. Editieren Sie dann zu diesem Zweck bitte die Datei */etc/mysql/mariadb.conf.d* und achten darauf, dass sich die folgenden Zeilen in der Datei unter der Rubrik *[mysqld]* befinden. Das gilt ebenfalls für die auskommentierten Zeilen.

Code-Ausschnitt 7.119: Erforderliche Einträge in der Datenbank-Konfiguration

```
#log_bin = /var/log/mysql/mysql-bin.log
plugin_dir = /usr/lib/mysql/plugin
bind-address = 127.0.0.1
character-set-server = latin1
#collation-server = utf8mb4_general_ci
innodb_large_prefix = ON
innodb_file_format = BARRACUDA
```

Starten Sie nach diesen Änderungen die Datenbank neu.

Nun legen wir eine Datenbank mit dem Namen *epg2vdr* an und einen Benutzer mit dem selben Namen, dem wir das Kennwort *epg* geben. Geben Sie zum Erstellen die folgenden Befehle in einem Terminal ein:

Code-Ausschnitt 7.120: Erstellen der Datenbank für den EPG-Daemon

```
sudo mysql -u root -p
CREATE DATABASE epg2vdr charset utf8;
CREATE USER 'epg2vdr'@'%' IDENTIFIED BY 'epg';
GRANT ALL PRIVILEGES ON epg2vdr.* TO 'epg2vdr'@'%';
GRANT ALL PRIVILEGES ON epg2vdr.* TO 'epg2vdr'@'localhost' IDENTIFIED BY 'epg';
FLUSH PRIVILEGES;
```

Es ist an der Zeit, den *epgd* einem ersten Test zu unterziehen.

Code-Ausschnitt 7.121: Manueller Start von *epgd*

```
export LANG="de_DE.UTF-8"
ulimit -c unlimited #so you can torture the developers with back traces
/usr/local/bin/epgd -n -p /usr/local/lib/epgd/plugins
```

Schauen Sie sich parallel zum manuellen Start von *epgd* die Systemausgaben an. Dazu öffnen Sie ein zweites Terminal und geben dort ein:

Code-Ausschnitt 7.122: Überwachen der Systemausgaben

```
sudo tail -f /var/log/syslog
```

Beim ersten Start von *epgd* hatte ich die folgenden Fehler im System-Log:

Code-Ausschnitt 7.123: Mögliche *epgd*-Fehler im System-Log

```
Nov 30 20:18:12 pinguin-pvr epgd: SQL-Error in 'select epglv('123', '123')' - FUNCTION epg2vdr.↵
epglv do
Nov 30 20:18:12 pinguin-pvr epgd: SQL-Error in 'select epglvr('123', '123')' - FUNCTION epg2vdr↵
.epglvr
Nov 30 20:18:12 pinguin-pvr epgd: Error: Missing functions epglv/epglvr, please install first!
```

Die Fehler konnte ich durch folgende Änderungen in der Datenbank beseitigen. Achten Sie bitte auch darauf, dass Ihr Plugin-Verzeichnis richtig gesetzt ist (Code-Ausschnitt 7.119).

Code-Ausschnitt 7.124: Beheben der Plugin-Fehler von *epgd*

```
sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 106
Server version: 10.3.20-MariaDB-0ubuntu0.19.10.1 Ubuntu 19.10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> DROP FUNCTION IF EXISTS epglv;
Query OK, 0 rows affected, 1 warning (0.00 sec)
MariaDB [(none)]> DROP FUNCTION IF EXISTS epglvr;
Query OK, 0 rows affected, 1 warning (0.00 sec)
MariaDB [(none)]> CREATE FUNCTION epglv RETURNS INT SONAME 'mysqllepglv.so';
Query OK, 0 rows affected (0.00 sec)
MariaDB [(none)]> CREATE FUNCTION epglvr RETURNS INT SONAME 'mysqllepglv.so';
Query OK, 0 rows affected (0.00 sec)
MariaDB [(none)]> exit
```

Alle Einstellungen für den *epgd* können Sie in der Datei */etc/epgd/epgd.conf* vornehmen. Ich habe hier eine kleine Image-Größe eingestellt, um nicht zu viel Speicherplatz zu verschwenden.

Code-Ausschnitt 7.125: Konfiguration des *epgd*

```
EpgImageSize = 0
```

Nachdem der EPG-Daemon nun hoffentlich fehlerfrei läuft, kann sein Start nun automatisch bei jedem Boot-Vorgang erfolgen.

Code-Ausschnitt 7.126: Automatischer Start des *epgd* bei jedem Boot

```
sudo systemctl enable epgd.service
```

Nach dieser Vorarbeit machen wir uns an das Installieren der nächsten Plugins. Beginnen wir mit dem Plugin, welches dem VDR das EPG aus der soeben eingerichteten Datenbank zur Verfügung stellt. Es heißt *epg2vdr* und kann zusammen mit seinen Abhängigkeiten so installiert werden:

Code-Ausschnitt 7.127: Installation von *epg2vdr*

```
cd ~/tv/VDR/PLUGINS/src
git clone https://github.com/horchi/vdr-plugin-epg2vdr epg2vdr
sudo apt install libarchive-dev libmysqlclient-dev libjansson-dev libpython-dev python-dev uuid->
-dev libtinyxml2-dev
make
sudo make install
```

Der Aufruf des Plugins ist unspektakulär. Ein einfaches

Code-Ausschnitt 7.128: Einbinden von *epg2vdr* in die VDR-Startdatei

```
-Pepg2vdr
```

in der VDR-Startdatei reicht aus, um das Plugin einzubinden. Von nun an bezieht der VDR seine EPG-Daten aus dem Internet.

Installieren wir als nächstes das Plugin *scraper2vdr*, welches dafür sorgt, dass Bilder aus der Film- und Seriendatenbank in die diversen Skins eingebunden werden. Diese Bilder werden auch dem Plugin *osd2web* zur Verfügung gestellt, welches wir im Anschluss für unser kleines Display installieren. Nach Installation aller Abhängigkeiten, von denen Sie sicher schon die eine oder andere installiert haben, laden wir das Plugin herunter und übersetzen und installieren es so:

Code-Ausschnitt 7.129: *scraper2vdr* installieren und übersetzen

```
sudo apt install libmysqlclient-dev uuid-dev imagemagick
cd ~/tv/VDR/PLUGINS/src
git clone https://github.com/horchi/scraper2vdr
sudo mkdir /video/scraper2vdrimages
cd scraper2vdr
make
sudo make install
```



Abbildung 7.17: *scraper2vdr* stellt Bilder und weitere Informationen zur laufenden Sendung zur Verfügung (Quelle: <https://wiki.easy-vdr.de> [74]).

Das Einbinden dieses Plugins in die VDR-Startdatei funktioniert so:

Code-Ausschnitt 7.130: Einbinden des Plugins *scraper2vdr*

```
-P 'scraper2vdr -i /video/scraper2vdrimages -m client'
```

Das Plugin erwartet als Parameter (-i) ein Verzeichnis, in dem es alle Bilder speichern darf. Dieses haben wir in Code-Ausschnitt 7.130 bereits erstellt. Weiterhin teilen wir dem Plugin mit (-m), dass es als Klient arbeiten soll, die Bilder also darstellen und nicht nur zur Verfügung stellen soll (Master). Um die Informationen der laufenden Sendung und der Aufnahmen auf dem kleinen LCD darzustellen, fehlt das letzte Plugin *osd2web*. Dieses Plugin benötigt die Bibliothek *libwebsockets* in einer Version größer gleich 2.1. Ein aktuelles Ubuntu liefert eine Version mit, die deutlich neuer ist als die verlangte Mindestversion. *osd2web* funktioniert daher nicht mit der neuesten Version von *libwebsockets*. Das ist aber kein Problem, da wir die Version 2.1 von <https://github.com/warmcat/libwebsockets/releases> [75] herunterladen, selbst übersetzen und installieren können. Laden Sie also bitte die Version 2.1 von der angegebenen Webseite in ein Verzeichnis Ihrer Wahl, beispielsweise nach */tv*. Packen Sie sie dann aus, übersetzen und installieren Sie sie so:

Code-Ausschnitt 7.131: Installation der *libwebsockets* in der Version 2.1.0

```
cd ~/tv
wget https://github.com/warmcat/libwebsockets/archive/v2.1.0.tar.gz
tar xvfz v2.1.0.tar.gz
cd libwebsockets-2.1.0
cmake .
make
sudo make install
```

Die in Code-Ausschnitt 7.131 beschriebene Installation kopiert alle Include-Dateien und erstellten Bibliotheken automatisch in das Verzeichnis `/usr/local`. Bitte achten Sie beim Starten des VDR darauf, dass Sie die Anweisung `LD_LIBRARY_PATH=/usr/local/lib` verwenden, damit die dort installierten Bibliotheken auch gefunden werden.



Das *osd2web*-Plugin wird danach - ähnlich wie alle anderen Plugins - so übersetzt und installiert:

Code-Ausschnitt 7.132: Installation des Plugins *osd2web*

```
sudo apt install libtinyxml2-dev libexif-dev
cd ~/tv/PLUGINS/src
git clone https://github.com/horchi/vdr-plugin-osd2web osd2web
make
sudo make install
```

Das Einbinden in die VDR-Startdatei erfolgt so:

Code-Ausschnitt 7.133: Einbinden des *osd2web*-Plugins in die VDR-Startdatei

```
-P'osd2web -s svg -l /var/lib/vdr/plugins/skindesigner/logos'
```

Mit dem Parameter `-s` teilt man dem Plugin mit, in welchem Format die Kanal-Logos vorliegen. In meinem Fall sind das *svg*-Dateien. Der Parameter `-l` sagt dem Plugin, in welchem Verzeichnis die Kanal-Logos vorliegen. Bei mir ist es das Verzeichnis `/var/lib/vdr/plugins/skindesigner/logos`.

Das *osd2web*-Plugin ist ein sogenanntes Skin-Plugin, d.h. es kann im VDR als Skin ausgewählt werden. Das bedeutet dann, dass alle OSD-Ausgaben nur noch über das Plugin erfolgen und damit nur auf einer Web-Adresse im Browser sichtbar sind. Stellt man dieses Plugin nicht als Skin ein, werden lediglich die Informationen über die aktuell laufende Sendung oder Aufnahme an eine Web-Adresse übertragen. Die Webseite ermöglicht ebenfalls ein Umschalten zwischen Steuerung des VDR oder einer reinen Darstellung von Information. Bevor wir uns die Web-Adresse und die damit verbundene Information genauer anschauen, muss noch eine Einstellung vorgenommen werden. Im Plugin-Setup des Plugins *epg2vdr* muss die Option *Erweitertes EPF in AUX speichern (für Skins)* mit *ja* ausgewählt werden.

Nach dem Starten des VDR mit den soeben installierten Plugins steht Ihnen unter der IP-Adresse Ihres Wohnzimmer-PCs und dem Port 4444 ein Webserver zur Verfügung.

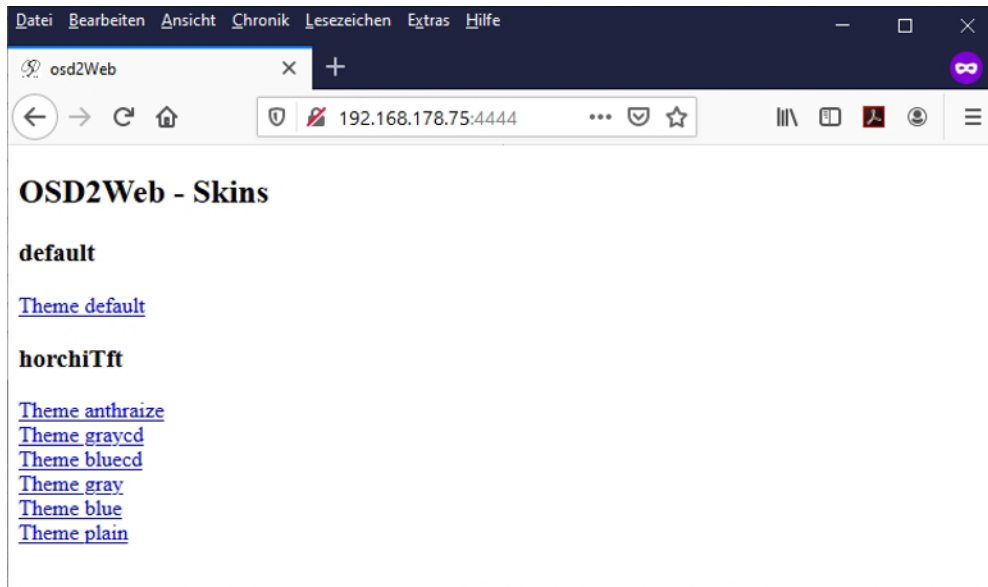


Abbildung 7.18: Beim Aufruf von *osd2web* ohne Parameter können verschiedene Farbthemen ausgewählt werden.

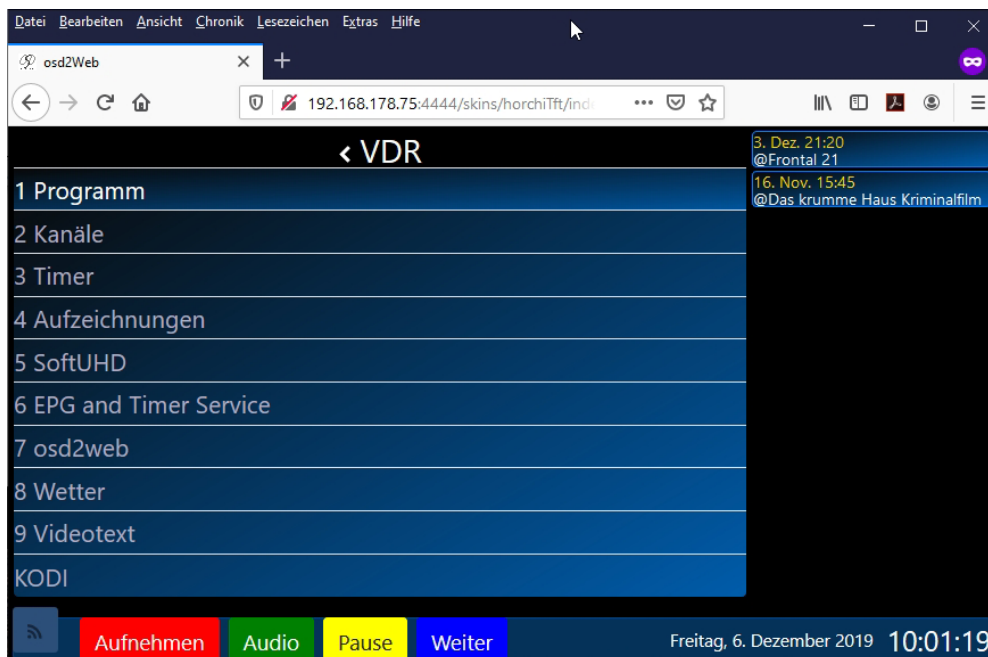


Abbildung 7.19: *osd2web* macht das OSD des VDR im Webinterface sichtbar.

Für den Betrieb von *osd2web* auf einem kleinen LCD können Sie die *osd2web*-URL wie folgt aufrufen `http://192.168.178.75:4444/skins/horchiTft/index.html?onlyView=yes&theme=blue`. Der Parameter *onlyView=yes* schaltet dabei die Steuerungsmöglichkeit ab, so dass Sie nur eine schöne Darstellung der Inhalte sehen.

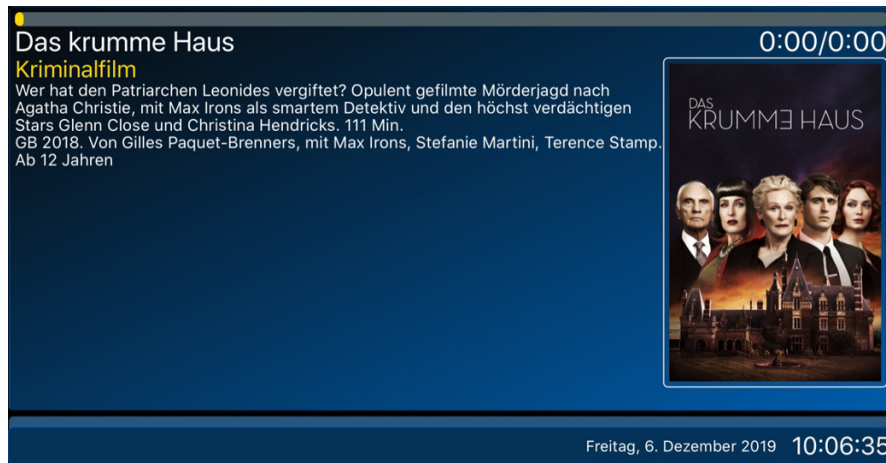


Abbildung 7.20: *osd2web*-Darstellung einer Aufnahme auf einem angeschlossenen LCD.

Der Anschluss eines kleinen LCDs an Ihren Wohnzimmer-PC kann prinzipiell über zwei Möglichkeiten erfolgen. Eine Variante nutzt einen USB-SPI-Konverter (z. B. FTDI-Chip), der dann ein SPI-Display ansteuern kann. Das ist in der Regel aber sehr langsam und daher für unsere Zwecke eher ungeeignet. Die zweite Variante bedient das LCD über ein HDMI-Interface. Diese Variante setzt voraus, dass Ihr Wohnzimmer-PC über mindestens zwei HDMI-Ausgänge verfügt: An einem hängt Ihr Fernseher, an dem anderen das LCD zur Anzeige der wichtigsten Informationen. Zur Sortierung der Bildschirme gibt es wieder zwei Möglichkeiten. Unter Ubuntu können Sie einfach einen zweiten Monitor als Erweiterung des ersten Monitors einstellen. Der Hauptmonitor (also Ihr TV) hat dann eine Auflösung von beispielsweise 1920×1080 Pixeln, während der kleine LCD eine Auflösung von z. B. 420×380 Pixeln hat. Der kleine Monitor ist dabei als Erweiterung des großen Monitors rechts oder links des großen Monitors angeordnet. Vor dem Start des VDR können Sie dann automatisch einen Browser im Vollbild-Modus starten, der als *osd2web*-Anzeige dient. Wahrscheinlich wird Ihr Wohnzimmer-PC den Browser aber auf dem Hauptbildschirm statt auf dem kleinen LCD darstellen. Eine alternative Möglichkeit ist das Starten von zwei verschiedenen X-Servern. Sie können dann über die *DISPLAY*-Variable entscheiden, auf welchem der Monitore (LCD oder TV) die Ausgabe erfolgen soll.

Das folgende Python-Skript sorgt dafür, dass ein Programm, welches auf einem bestimmten Screen geöffnet ist, auf einen anderen geschoben werden kann. Es setzt die Installation der Programme *wmctrl* und *xdotool* voraus, die zunächst installiert werden müssen.

Code-Ausschnitt 7.134: Installation von *wmctrl*

```
sudo apt install wmctrl xdotool
```

Code-Ausschnitt 7.135: Skript zum Verschieben von Fenstern

```
#!/usr/bin/env python3
import subprocess
import sys

# just a helper function, to reduce the amount of code
get = lambda cmd: subprocess.check_output(cmd).decode("utf-8")

# get the data on all currently connected screens, their x-resolution
screendata = [l.split() for l in get(["xrandr"]).splitlines() if "connected" in l]
screendata = sum([[(w[0], s.split("+")[-2]) for s in w if s.count("+") == 2] for w in screendata], [])

def get_class(classname):
    # function to get all windows that belong to a specific window class (application)
    w_list = [l.split()[0] for l in get(["wmctrl", "-l"]).splitlines()]
    return [w for w in w_list if classname in get(["xprop", "-id", w])]

scr = sys.argv[2]

try:
    # determine the left position of the targeted screen (x)
    pos = [sc for sc in screendata if sc[0] == scr][0]
except IndexError:
    # warning if the screen's name is incorrect (does not exist)
    print(scr, "does not exist. Check the screen name")
else:
    for w in get_class(sys.argv[1]):
        # first move and resize the window, to make sure it fits completely inside the targeted screen
        # else the next command will fail...
        subprocess.Popen(["wmctrl", "-ir", w, "-e", "0,"+str(int(pos[1])+100)+"100,300,300"])
        # maximize the window on its new screen
        subprocess.Popen(["xdotool", "windowize", "-sync", w, "100%", "100%"])

```

Speichern Sie das in Code-Ausschnitt 7.135 gezeigte Skript unter dem Namen *shift_window.py* in einem Verzeichnis Ihrer Wahl. Führen Sie es dann so aus:

Code-Ausschnitt 7.136: Aufruf des Programms zum Verschieben von Fenstern

```
DISPLAY=:0.0 /usr/bin/python3 /pfad/shift.py Firefox HDMI1
```

Ersetzen Sie im oben genannten Beispiel */pfad* durch den Pfad zu Ihrem Skript, das ist *shift.py* genannt habe und *Firefox* durch den Fensternamen, den Sie verschieben möchten. Ersetzen Sie ebenfalls *HDMI1* durch den Monitor, auf den Sie verschieben möchten.

Ich habe mir in der Datei */var/lib/vdr/commands.conf* ein kleines Skript abgelegt, das automatisch einen Firefox öffnet, diesen auf den zweiten Monitor (mein kleines LCD) schiebt und auf Vollbild setzt. Der Eintrag lautet wie folgt:

Code-Ausschnitt 7.137: *osd2web* automatisiert anzeigen

```
Switch Display on : at -f ~/tv/display.sh now
```

Den *at*-Befehl erkläre ich Ihnen im letzten Abschnitt dieses Kapitels näher. Das Skript *display.sh* hat folgenden Inhalt:

Code-Ausschnitt 7.138: Automatisiertes Starten der *osd2web*-Anzeige

```
#!/bin/bash
xrandr --display :0.0 --output HDMI1 --mode 1920x1080i --right-of DP1
nice -n -20 firefox --display=:0.0 http://192.168.178.75:4444/skins/TFT/index.html?theme=gray &
sleep 3
DISPLAY=:0.0 /usr/bin/python3 /home/rudi/tv/shift.py Firefox HDMI1
a=$(DISPLAY=:0.0 wmctrl -l | grep Firefox | awk '{print $1}')
DISPLAY=:0.0 xdotool windowactivate --sync $a key F11
DISPLAY=:0.0 wmctrl -a soffthdcuvid
```

Das Skript funktioniert wie folgt: Zunächst wird die Auflösung meines LCDs auf 1920×1080 interlaced gesetzt. Da mein TV ebenfalls mit 1920×1080 Pixeln Auflösung läuft, passt das perfekt zusammen. Verwende ich hier unterschiedliche Auflösungen für Haupt-Bildschirm und LCD, gibt es ein starkes Tearing im Haupt-Display. Abhilfe würden hier nur zwei verschiedene Grafikkarten schaffen. Weiterhin richten wir das LCD (HDMI1) so ein, dass der Bildschirm rechts unseres TVs (DP1) liegt. Wenn Sie also eine Maus nutzen, müssten Sie die rechts aus dem TV-Bildschirm heraus schieben, um sie im LCD zu sehen.

Als nächstes wird der Browser *firefox* gestartet mit einem Nice-Level von -20. Das bedeutet, dass er niedrige Priorität hat und andere Programme (wie eben der VDR) deutlich höher priorisiert und damit schneller im Zugriff sind. Ersetzen Sie gegebenenfalls die *osd2web*-Adresse durch Ihre eigene. Das *sleep* von 3 Sekunden ist erforderlich, weil hier gewartet werden muss, bis Firefox die Seite aufgebaut hat. Danach schieben wir das Firefox-Fenster auf das LCD und finden heraus, welche Fenster-Nummer vergeben wurde. An diese Fenster-Nummer senden wir dann die Tastatur-Eingabe *F11*, welches den Firefox in den Vollbild-Modus versetzt.

Damit der VDR im Anschluss wieder mit der Fernbedienung bedient werden kann, setzen wir zum Schluss den Fokus wieder auf das grafische VDR-Frontend, in meinem Beispiel also auf das Plugin *soffthdcuvid*. Falls der Name Ihres grafischen Ausgabe-Plugins anders lautet, ändern Sie diesen Namen bitte entsprechend ab. Der Befehl

Code-Ausschnitt 7.139: Identifizieren von Fenster-Informationen

```
DISPLAY=:0.0 wmctrl -l
```

hilft Ihnen dabei, den richtigen Namen zu identifizieren.

Die TV-Ausgabe reagiert sehr empfindlich auf unterschiedliche Auflösungen des Haupt-Bildschirms (in meinem Fall DP1) und des LCDs (in meinem Fall HDMI1), wenn beide Ausgänge an derselben Grafikkarte hängen. Prinzipiell empfehle ich Ihnen für den Hauptbildschirm (als den TV) eine Bildwiederholrate von 50 Hz. Bei meinen Versuchen mit einem LCD im erweiterten Bildschirm hatte die Wiederholrate des LCDs keinen Einfluss auf das TV-Bild, wohl aber die Auflösung des LCDs. Im einzig gut funktionierenden Fall hatte das LCD dieselbe Auflösung wie das Hauptbild, nämlich 1920×1080 Pixel.



Ist man auf verschiedene Auflösungen für LCD und TV angewiesen, gibt es zwei Lösungsmöglichkeiten. Entweder benötigt man eine zweite Grafikkarte, oder man lässt zwei Screens auf derselben Karte für unterschiedliche Bildschirme laufen. Dieses Verfahren erkläre ich Ihnen im folgenden Text. Allerdings gibt es nach meinen Experimenten auch hier Einschränkungen. Diese betrafen in meinen Experimenten die Bildwiederholrate. Eine TV-Bildschirmauflösung von UHD mit 3840×2160 Pi-

xeln bei einer Wiederholrate von 50 Hz harmonierte perfekt mit der LCD-Auflösung von 1280×720 Pixeln bei einer Bildwiederholrate von 60 Hz. Das ist schön, weil der VDR dann UHD-Fernsehen wiedergeben kann. Allerdings ist er je nach Skin und Ausgabe-Plugin sehr träge zu bedienen. Meine Versuche, das TV-Bild über den DP1-Anschluss (USB-C nach HDMI-Konverter) auf Full-HD mit einer Wiederholrate von 50 Hz zu setzen, quitierte der X-Server mit einem Hänger. In dieser Kombination funktionierten lediglich Bildwiederholraten von 29 Hz für den Hauptbildschirm, und das harmoniert nicht mit dem PAL-System. Eine Abhilfe wäre, an dieser Stelle wieder zwei verschiedene Grafikkarten zu benutzen, die sich mit ihren Timings nicht in die Quere kommen, oder aber auf einen kleinen LCD zu verzichten. Im Internet gibt es darüber hinaus den Hinweis (<https://askubuntu.com/questions/995077/screen-corruption-on-multi-monitor-displayport-using-intel-graphics?rq=1> [76]), für neuere Intel-GPUs nicht den Intel-Treiber, sondern den *modesetting*-Treiber zu verwenden. In diesem Fall müssten Sie das in der unten gezeigten Datei in der Rubrik *Driver* berücksichtigen.

Der Start von zwei verschiedenen Screens wird wieder in der X-Konfigurationsdatei eingestellt. Im Beispiel der Intel-GPU meines Lattepanda Alpha-Computers habe ich dazu die Datei *20-intel.conf* im Verzeichnis */etc/X11/xorg.conf.d* als Root mit folgendem Inhalt angelegt:

Code-Ausschnitt 7.140: X-Konfigurationsdatei zwei Bildschirme mit unterschiedlichen Screens

```

Section "Device"
    Identifier "Device0"
    Driver     "intel"
    Option     "TearFree" "true"
    Option     "ZaphodHeads" "DP1"
    Screen     0
EndSection
Section "Device"
    Identifier "Device1"
    Driver     "intel"
    Option     "TearFree" "true"
    Option     "ZaphodHeads" "HDMI1"
    Screen     1
EndSection
Section "ServerLayout"
    Identifier "Layout0"
    Screen 0 "Screen0" 0 0
    Screen 1 "Screen1" RightOf "Screen0"
EndSection
Section "Screen"
    Identifier "Screen0"
    Device     "Device0"
    Monitor     "DP1"
    DefaultDepth 24
    SubSection "Display"
        Depth     24
        #Modes     "3840x2160_50.00"
        #Modes     "1920x1080_50"
    EndSubSection
EndSection
Section "Screen"
    Identifier "Screen1"
    Device     "Device1"
    Monitor     "HDMI1"
    DefaultDepth 24
    SubSection "Display"
        Depth     24
        Modes     "1280x720_60.00"
    EndSubSection
EndSection
Section "Monitor"
    Identifier "DP1"
    Modeline "3840x2160_50.00" 587.00 3840 4144 4560 5280 2160 2163 2168 2225 -hsync +vsync
    Modeline "1920x1080_60" 148.500 1920 2008 2052 2200 1080 1084 1089 1125 +hsync +vsync
    Modeline "1920x1080_25" 74.250 1920 2448 2492 2640 1080 1082 1089 1125 +hsync +vsync interlace
    Modeline "1920x1080_50" 148.500 1920 2448 2492 2640 1080 1084 1089 1125 +hsync +vsync
    Modeline "1920x1080_24" 74.250 1920 2558 2602 2750 1080 1084 1089 1125 +hsync +vsync
    Modeline "1920x1080_30" 74.250 1920 2008 2052 2200 1080 1084 1089 1125 +hsync +vsync
EndSection
Section "Monitor"
    Identifier "HDMI1"
    #Modeline "1920x1080_60.00" 173.00 1920 2048 2248 2576 1080 1083 1088 1120 -hsync +vsync
    Modeline "1280x720_60.00" 74.50 1280 1344 1472 1664 720 723 728 748 -hsync +vsync
EndSection
Section "Monitor"
    Identifier "HDMI2"
    Option "Ignore" "true"
EndSection

```

Diese Datei müssen Sie gegebenenfalls noch in Bezug auf Ihren Wohnzimmer-PC anpassen. Bei mir hängt der TV am Ausgang DP1, der über einen USB-C nach HDMI-Adapter realisiert wird. Mein LCD hängt am Ausgang HDMI1. Während die Auflösungen für das LCD (HDMI1) immer korrekt übernommen wurden, wurden die Auflösungen für DP1 vom System ignoriert. Hier musste ich trotz der Angaben in der oben gezeigten Konfigurationsdatei den Umweg über die Ubuntu-Bildschirmeinstellungen gehen und

dort die UHD-Auflösung wählen. Wahrscheinlich liegt das daran, dass der verwendete USB-C nach HDMI-Konverter nicht von den X-Server-Einstellungen initialisiert werden kann. Die in der Datei vorhandenen *Modeline*-Einträge werden im letzten Abschnitt dieses Kapitels noch genauer erklärt.

Nach einem Neustart des Rechners gibt es zwei voneinander unabhängige DISPLAYs (0.0 und 0.1). Von einem Terminal aus kann der Erfolg der X-Server-Einstellung durch die Ausgabe einer Uhr kontrolliert werden. Auf dem Haupt-Bildschirm (DP1, TV), wird die Uhr so angezeigt.

Code-Ausschnitt 7.141: Ausgabe einer Uhr auf dem TV-Bildschirm

```
DISPLAY=:0.0 xclock
```

Auf dem LCD (HDMI1) wird die Uhr - Sie ahnen es - so angezeigt:

Code-Ausschnitt 7.142: Ausgabe einer Uhr auf dem LCD

```
DISPLAY=:0.1 xclock
```

Auf dem DISPLAY 0.1 läuft allerdings (noch) kein Fenstermanager. Sie werden also beim Beenden der Uhr auf dem LCD feststellen, dass noch „Grafikreste“ überbleiben. Weiterhin kann *wmctrl* ohne Fenstermanager keine Information über Fenster sammeln. Diese werden aber benötigt, um beispielsweise den Firefox in den Vollbild-Modus zu versetzen. Ich empfehle daher die Installation des minimalistischen Fenstermanagers *openbox*.

Code-Ausschnitt 7.143: *openbox* Fenstermanager installieren

```
sudo apt install openbox
```

Die Schritte zur Ausgabe des *osd2web*-Servers mit dieser Einstellung sind:

Code-Ausschnitt 7.144: Automatisiertes Starten der *osd2web*-Anzeige auf einem LCD

```
# Starte openbox Fenstermanager im Hintergrund auf dem LCD
DISPLAY=:0.1 openbox &
#
# Starte Firefox auf LCD
nice -n -20 firefox --display=:0.1 http://192.168.178.75:4444/skins/TFT/index.html?theme=gray &
#
# Firefox Zeit geben
sleep 3
#
# Speichern Firefox-Fenster-ID und F11 an die ID senden (Vollbild)
a=$(DISPLAY=:0.1 wmctrl -l | grep Firefox | awk '{print $1}')
DISPLAY=:0.1 xdotool windowactivate --sync $a key F11
#
# Fokus wieder auf VDR-Fenster setzen (Bildschirm 0.0)
DISPLAY=:0.0 wmctrl -a sothdcuvid
```

Alle hier gezeigten Schritte setzen natürlich voraus, dass der VDR bereits mit dem *osd2web*-Server läuft.

Falls Ihnen ein kleines LCD für den VDR noch zu groß ist, kann ich Ihnen ein VFD (Vacuum Fluorescent Display) empfehlen. Auf der Webseite <http://www.eisgold.de> [77] finden Sie ein fertiges bereits für den VDR vorkonfiguriertes Display, welches über USB angeschlossen wird.



Abbildung 7.21: Das Eisgold VFD leuchtet sehr hell und kann auch aus größerer Entfernung gut abgelsen werden [77].

Code-Ausschnitt 7.145: VFD unter VDR installieren

```
sudo apt install libusb-1.0-0-dev
cd ~/tv/VDR/PLUGINS/src
git clone https://github.com/vdr-projects/vdr-plugin-targavfd targavfd
make
sudo make install
```

Damit das Plugin ohne Fehler kompiliert hat, musste ich das Makefile ändern. In der `INCLUDES`-Zeile musste ich `-I/usr/include/freetype2` ergänzen, damit alle Include-Dateien richtig gefunden werden.

Code-Ausschnitt 7.146: Änderung im Makefile des *targavfd*-Plugins

```
INCLUDES += $(shell freetype-config --cflags) -I/usr/include/freetype2
```

Das Plugin wird mit einem einfachen

Code-Ausschnitt 7.147: Aufruf des *targavfd*-Plugins

```
-P targavfd
```

eingebunden.

7.7 Stream me up!

In diesem Abschnitt zeige ich Ihnen noch, wie Sie das aktuelle Fernsehprogramm auf ein mobiles Gerät (Telefon, Tablet oder Laptop) streamen können. Dazu verpassen wir dem VDR den *streamdev-server*.

Code-Ausschnitt 7.148: Das Plugin *streamdev-server* übersetzen

```
cd ~/tv/VDR/PLUGINS/src
git clone git://projects.vdr-developer.org/vdr-plugin-streamdev.git streamdev
cd streamdev
make
sudo make install
```

streamdev installiert praktischerweise auch einen Klienten mit, den Sie dazu nutzen können, das TV-Programm von einem VDR auf einen anderen VDR zu streamen. Wir nutzen aber den Server, um das Programm auf ein Tablet oder ein anderes mobiles Gerät zu übertragen. Tragen wir aber zunächst das Plugin in unser VDR-Startskript ein:

Code-Ausschnitt 7.149: Das Plugin *streamdev* in das Start-Skript eintragen

```
-Pstreamdev-server \
```



Üblicherweise spielt die Reihenfolge der Plugins keine Rolle. Sollte aber dennoch das eine oder andere Plugin nicht starten wollen, ändern Sie die Reihenfolge. Es gibt einige Plugins, die gerne an erster Stelle stehen wollen.

Nach diesen Vorbereitungen müssen wir dem Server noch mitteilen, welche Klienten denn überhaupt mit dem Server verbinden dürfen.

Code-Ausschnitt 7.150: *streamdev* konfigurieren

```
cd ~/tv/VDR/PLUGINS/src/streamdev/streamdev-server
sudo cp streamdevhosts.conf /var/lib/vdr/plugins/streamdev-server
```

Die Default-Einstellungen erlauben lediglich eine Verbindung vom selben Server (*localhost*). Das ist langweilig, denn wir möchten ja vielleicht ein Tablet verbinden, welches sich im selben Netz (z. B. WLAN) befindet. Ergänzen wir die Datei *streamdevhosts.conf* also um folgenden Eintrag:

Code-Ausschnitt 7.151: *streamdev* Verbindungen zulassen

```
sudo vi /var/lib/vdr/plugins/streamdev-server
0.0.0.0/0 # any host on any net (DON'T DO THAT! USE AUTHENTICATION)
```



Abbildung 7.22: Das Logo des VLC Media-Players (Quelle: <https://www.vlc.de/> [78]).

Damit erlauben wir kurzfristig jede Verbindung zum Server, egal von wo. Die Datei selbst enthält Beispiele, wie Sie nur Ihr eigenes Netzwerk erlauben. Für einen kurzen Test ist diese allgemeine Freigabe aber in Ordnung. Danach starten Sie bitte den VDR, etwa durch Aufrufen des Skripts */usr/local/bin/runvdr*. Zum Abspielen des Fernseh-Streams empfehle ich den VLC Media-Player. Der VLC Media-Player steht für fast jedes mobile und auch stationäre Device zum kostenlosen Download zur Verfügung und kann darüber hinaus sehr viele verschiedene Formate flüssig darstellen. Standardmäßig benutzt *streamdev* den Port 3000, um den Stream im TS-Format (*Transport Stream*) abzuliefern. Damit fehlt nur noch der Kanal als letzter Parameter, um die Streaming-URL perfekt zu machen. Eine typische URL ist so aufgebaut:

Code-Ausschnitt 7.152: Aufbau einer *streamdev*-URL

```
http://192.168.178.66:3000/TS/1
```

Bitte ersetzen Sie die oben angegebene IP-Adresse durch die Ihres Streaming-Wohnzimmer-PCs. Ersetzen Sie weiterhin die Kanalnummer (1) durch die Nummer des Kanals, welchen Sie streamen möchten. Eine Änderung der Kanalnummer sorgt übrigens dafür, dass das Frontend automatisch auf diesen Kanal umschaltet. *TS* bezeichnet eine Datenkodierung. Folgende Kodierungen stehen zur Verfügung: *PES*, *TS*, *PS*, *ES* und *Extern*. Falls eine nicht funktioniert, probieren Sie eine andere. *Extern* greift auf die Datei *externremux.sh* zurück und erlaubt ein neues Kodieren des Datenstreams. So könnten Sie auf einem schnellen Wohnzimmer-PC das Fernsehprogramm so komprimieren, dass Sie es selbst in einem langsamen Netzwerk außerhalb Ihres eigenen Netzes z. B. im Urlaub anschauen können. Die Zahl hinter der Kodierung bezeichnet den ausgewählten Fernsehkanal, den man schauen möchte. Er entspricht der Eintragsnummer in der VDR-Kanalliste *channels.conf*.

VDR kann man übrigens auch von einem Terminal aus bedienen. Dazu verwendet man das Programm *svdrpsend* aus dem VDR-Verzeichnis. Dieses kopiert man nach */usr/local/bin*, damit es überall verfügbar ist. Durch den Aufruf



Code-Ausschnitt 7.153: Fernsteuerung des VDR

```
svdrpsend CHAN 2
```

schaltet man auf den zweiten Kanal der Kanalliste um. Eine komplette Hilfe zu *svdrpsend* findet sich unter <http://www.vdr-wiki.de/wiki/index.php/SVDRP> [55].

7.8 Pay-TV

VDR bietet die prinzipielle Möglichkeit, bezahlte Fernsehprogramme zu entschlüsseln. Voraussetzung hierfür ist zunächst ein Common Interface (CI). Es ist gar nicht so einfach, DVB-Karten zu finden, die ein CI-Interface unterstützen. Die Firma Digital Devices (<https://digitaldevices.de> [79]) bietet ein Common Interface an, welches mit den hauseigenen DVB-Karten kompatibel ist.

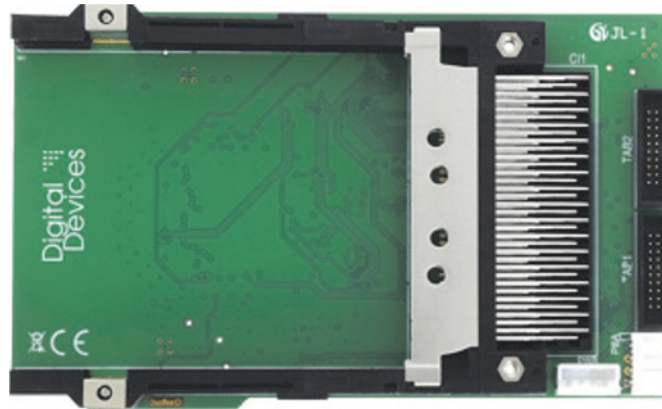


Abbildung 7.23: Die Firma Digital Devices bietet ein CI an, das mit den hauseigenen Karten DVB kompatibel ist. Quelle: <https://digitaldevices.de> [79].

Diese DVB-Karten sind aber PCI-Express-Karten und setzen damit einen Wohnzimmer-PC mit PCI-Steckplätzen voraus. Bei kompakten Wohnzimmer-PCs ist diese Voraussetzung oft nicht gegeben. Digital Devices bemüht sich aktiv darum, dass alle benötigten Treiber im LINUX-Kernel verfügbar sind. Um Das CI der Digital Devices-Karten nutzen zu können, muss das VDR-Plugin *ddci2* so übersetzt werden:

Code-Ausschnitt 7.154: Installation des *ddci2*-Plugins

```
cd ~/tv/VDR/PLUGINS/src
git clone https://github.com/jasmin-j/vdr-plugin-ddci2 ddc2
cd ddc2
make
sudo make install
```

Die Integration dieses Plugins in die VDR-Startdatei erfolgt durch ein einfaches

Code-Ausschnitt 7.155: Aufruf des *ddci2*-plugins

```
-P ddc2
```

Es gibt USB DVB-Karten, die bereits ein CI eingebaut haben. Die Firma TBS Technology bietet verschiedene DVB-S2 USB-Sticks an, die CI-Module aufnehmen können.



Abbildung 7.24: Die Firma Digital TBS Technology bietet USB-Sticks mit integriertem CI. Quelle: <https://www.tbs-technology.de> [80].

Die benötigten LINUX-Treiber stellt der Hersteller auf seiner Webseite zur Verfügung, ebenso wie die erforderliche Firmware. Die Treiber können Sie wie folgt installieren:

Code-Ausschnitt 7.156: Installation der TBS Technology-Treiber

```
cd ~/tv
sudo apt-get install libproc-processtable-perl
git clone https://github.com/tbsdtv/media_build.git
git clone --depth=1 https://github.com/tbsdtv/linux_media.git -b latest ./media
cd media_build
make dir DIR=./media
make allyesconfig
make -j4
sudo make install
```

Abschließend muss noch die Firmware heruntergeladen und installiert werden.

Code-Ausschnitt 7.157: Installation der TBS-Technology Firmware

```
wget http://www.tbsdtv.com/download/document/linux/tbs-tuner-firmwares_v1.0.tar.bz2
sudo tar jxvf tbs-tuner-firmwares_v1.0.tar.bz2 -C /lib/firmware/
```

Nach einem Neustart des Systems steht die Karte samt CI zur Entschlüsselung von Bezahl-Fernsehen bereit. Sollte es bei der Übersetzung des Treibers Probleme geben, hilft die Seite https://github.com/tbsdtv/linux_media/wiki [81] bei der Lösung.

Die letzte Möglichkeit, ein CI bereitzustellen sind USB-CIs, die man heute kaum noch findet. Die Firma Hauppauge hat ein solches Interface vor vielen Jahren für Windows-PCs angeboten.



Abbildung 7.25: USB-CI-Module sind heute kaum noch zu finden.

Quelle: <https://www.amazon.de> [82]

Diese Module haben den Vorteil, dass sie nicht an eine bestimmte DVB-Karte gebunden sind, sondern mit allen verfügbaren funktionieren. Den Treiber für diese Module gibt es auf <https://github.com/siricco/usb2ci> [83]. Er wird so geladen und übersetzt:

Code-Ausschnitt 7.158: Treiber für Hauppauge Win-TV CI USB-Modul übersetzen

```
cd ~/tv
git clone https://github.com/siricco/usb2ci
cd usb2ci
sudo ./runMake
```

Bei meinem Ubuntu-System funktionierte das Übersetzen des Treibers einwandfrei. Das Installieren hingegen funktioniert nicht, da der LINUX-Kernel nur signierte Treiber akzeptiert. Das ist aber nicht weiter schlimm, denn der Treiber kann direkt aus dem Verzeichnis *usb2ci* heraus so aktiviert werden:

Code-Ausschnitt 7.159: Aktivieren des Hauppauge Win-TV USB CI-Treibers

```
sudo insmod ./wintv-usb2ci.ko
```

Damit der Treiber richtig funktioniert, benötigt er Firmware-Dateien von Hauppauge. Diese müssen mit dem Script *WintvCI_extract_firmware.pl*, welches sich im Unterverzeichnis *scripts* befindet, extrahiert und anschließend in das Verzeichnis */lib/firmware* kopiert werden. Das weiter oben beschriebene *ddci2*-Plugin wird dazu verwendet, dieses USB-CI-Modul für verschiedene TV-Karten zuzuweisen. Vergessen Sie also bitte nicht, dieses Modul ebenfalls zu installieren, auch wenn Sie kein Digital Devices CI nutzen. Die Firmware selbst kann auch aus dem VDR-Portal geladen wer-

den ([urlhttps://www.vdr-portal.de/index.php?attachment/42443-wintv-ci-firmware-tar-gz/](https://www.vdr-portal.de/index.php?attachment/42443-wintv-ci-firmware-tar-gz/) [56]) oder aber von der Hauppauge Webseite.

Die CI-Module sollten nicht mit dem CI+-Standard verwechselt werden. Für LINUX existiert derzeit kein legal nutzbares CI+-Interface. Auf der Webseite <https://github.com/ciminus/vdr-plugin-ci-plus> gibt es zwar ein VDR-Plugin, welches prinzipiell in der Lage ist, den CI+-Standard zu bedienen. Dieses Plugin demonstriert aber eher die prinzipielle Machbarkeit und ist ohne weitere Kenntnisse nicht lauffähig.



7.9 Wichtige Hinweise

In diesem Abschnitt möchte ich Ihnen noch einige wichtige Hinweise rund um den VDR mit auf den Weg geben.

VDR und seine Plugins greifen oft auf verschiedene Skripte zurück, die verschiedene Dinge erledigen können. Dabei sind Skripte, die eine lange Laufzeit haben, allerdings eine Gefahr für den VDR: Ist die Laufzeit eines Skriptes zu lang, wird der VDR dadurch blockiert. Im Extremfall schlägt dann der Watchdog zu und startet den VDR neu. Wenn Sie das verhindern möchten, können Sie lang laufende Skripte oder Skript-Teile mit Hilfe von *at* starten.

at kann so installiert werden:

Code-Ausschnitt 7.160: VDR-Skripte sicher starten mit *at*

```
sudo apt-get install at
```

Es dient eigentlich dazu, Programme zu einem bestimmten Zeitpunkt zu starten. Dieser Zeitpunkt kann aber auch sofort (also jetzt oder im Englischen *now*) sein. Dann wird das Skript in einen eigenen Thread ausgelagert und der VDR wird nicht mehr blockiert. Interessant ist das beispielsweise innerhalb der Datei *commands.conf*, mit deren Hilfe der VDR bestimmte Befehle (oder auch Skripte) ausführen kann. Für ein Skript, das besonders lange läuft, sollte ein *commands.conf*-Eintrag so aussehen:

Code-Ausschnitt 7.161: Funktionsweise von *at*

```
Starte Skript: at now -f /pfad/script
```

Die meisten Grafikkarten schalten ihren Ausgang ab, wenn der Fernseher beim Booten des Wohnzimmer-PCs nicht eingeschaltet ist. Um in diesem Fall ein Bild auf dem TV zu sehen, müssten Sie Ihren Rechner mit eingeschaltetem Fernseher neu starten, damit die Grafikausgabe aktiviert wird. Und ehrlich, wer achtet beim Einschalten immer auf die „richtige“ Reihenfolge, erst TV und dann PC? Oder stellen Sie sich vor, Ihr PC hat eine Aufnahme bei ausgeschaltetem TV gestartet. Sie möchten diese Aufnahme zeitversetzt schauen, während sie noch läuft. Sicher war ihr Fernseher aber nicht eingeschaltet und darum sehen Sie nur ein schwarzes Bild. Daher ist es bei einem Wohnzimmer-PC sinnvoll, eine Ausgabe zu erzwingen, selbst wenn der Fernseher nicht eingeschaltet ist. Das Geheimnis der Bildausgabe liegt in der sogenannten *EDID*-Datei (Extended Display Identification Data), die - vereinfacht gesagt - die Ausgabemöglichkeiten des Ausgabegeräts (also z. B. des Fernsehers) speichert. Während NVidia-Treiber diese Möglichkeit standardmäßig vorsehen, fehlt dieser Automatismus bei INTEL-Grafiktreibern.

In diesem Abschnitt erkläre ich Ihnen, wie Sie eine EDID-Datei erstellen und beim Booten Ihres PCs so einbinden können, dass die Grafikausgabe selbst bei ausgeschaltetem TV erhalten bleibt. Voraussetzung dafür ist, dass der Kernel Ihres LINUX-Systems das Laden der EDID-Firmware während des Startvorgangs erlaubt. Das können Sie sehr einfach mit dem Befehl

Code-Ausschnitt 7.162: Auflisten einer Kernel-Option

```
grep "CONFIG_DRM_LOAD_EDID_FIRMWARE" "/boot/config-$(uname -r)"
```

testen. Bei mir liefert der Befehl folgende Ausgabe:

Code-Ausschnitt 7.163: Ausgabe der Kernel-Option

```
CONFIG_DRM_LOAD_EDID_FIRMWARE=y
```

Um eine EDID-Datei zu speichern, muss der Wohnzimmer-PC mit eingeschaltetem Fernseher gestartet werden. Viele PCs erlauben das Anschließen von mehreren Monitoren. Die zur Verfügung stehenden Anschlüsse können mit dem Befehl

Code-Ausschnitt 7.164: Auflisten aller Grafikanalysen

```
for p in /sys/class/drm/*/status; do con=${p%/status}; echo -n "${con#*/card?-}:"; cat $p; ↵  
done
```

aufgelistet werden. Bei mir lautet die Ausgabe mit eingeschaltetem Fernseher so:

Code-Ausschnitt 7.165: Ausgabe beim Auflisten aller Grafikanalysen

```
DP-1: disconnected  
HDMI-A-1: connected  
HDMI-A-2: disconnected
```

Im weiteren Verlauf gehe ich davon aus, dass der Anschluss *HDMI-A-1* verwendet wird. Falls Sie einen anderen Anschluss verwenden, ändern Sie die folgenden Eintragungen bitte dementsprechend ab.

Zum Auslesen der EDID-Daten kann das Programm *xrandr* verwendet werden. Das folgende Python-Skript wandelt die ausgelesenen Daten dann automatisch in das richtige Format um.

Code-Ausschnitt 7.166: Python-Skript zum Auslesen der EDID-Daten (<https://gist.github.com/mvollrath/84>)

```
#!/usr/bin/env python  
  
import binascii  
import re  
import subprocess  
import sys  
from os.path import basename  
  
XRANDR_BIN = 'xrandr'  
  
# re.RegexObject: expected format of xrandr's EDID ascii representation  
EDID_DATA_PATTERN = re.compile(r'^\t\t[0-9a-f]{32}$')  
  
def get_edid_for_connector(connector_name):  
    """Finds the EDID for the given connector.
```

```

Args:
    connector_name (str): Name of a connector, i.e. HDMI-0, DP-1

Returns:
    Binary EDID for the connector, or None if not found.

Raises:
    OSError: Failed to run xrandr.
"""

# re.RegexObject: pattern for this connector's xrandr --props section
connector_pattern = re.compile('^{} connected'.format(connector_name))

try:
    xrandr_output = subprocess.check_output([XRANDR_BIN, '--props'])
except OSError as e:
    sys.stderr.write('Failed to run {}\n'.format(XRANDR_BIN))
    raise e

output_lines = xrandr_output.decode('ascii').split('\n')

def slurp_edid_string(line_num):
    """Helper for getting the EDID from a line match in xrandr output."""
    edid = ''
    assert re.match(r'\tEDID:', output_lines[line_num+1])
    for i in range(line_num + 2, len(output_lines)):
        line = output_lines[i]
        if EDID_DATA_PATTERN.match(line):
            edid += line.strip()
        else:
            break
    return edid if len(edid) > 0 else None

for i in range(len(output_lines)):
    connector_match = connector_pattern.match(output_lines[i])
    if connector_match:
        edid_str = slurp_edid_string(i)
        if edid_str is None:
            return None
        return binascii.unhexlify(edid_str)

return None

if __name__ == '__main__':
    if len(sys.argv) != 2:
        sys.exit('Usage: {} <OUTPUT>'.format(basename(sys.argv[0])))

    connector_name = sys.argv[1]

    edid_bin = get_edid_for_connector(connector_name)
    if edid_bin is None:
        sys.exit('No EDID found for output {}'.format(connector_name))

    if sys.version_info >= (3, 0):
        sys.stdout.buffer.write(edid_bin)
    else:
        sys.stdout.write(edid_bin)

    sys.exit(0)

# vim: tabstop=8 expandtab shiftwidth=4 softtabstop=4

```

Speichern Sie dieses Skript, beispielsweise in Ihrem Home-Verzeichnis unter dem Namen *getedid.py* und rufen Sie es danach so auf:

Code-Ausschnitt 7.167: Auslesen der EDID-Daten

```
DISPLAY=:0.0 python ./getedid.py HDMI-1 > edid.bin
```

Die Datei *edid.bin* enthält danach alle erforderlichen Daten. Stellen Sie bitte sicher, dass die Datei nicht leer ist.



Vielleicht ist Ihnen der Zusatz DISPLAY=:0.0 aufgefallen. Er ist nur erforderlich, falls Sie „remote“, also beispielsweise per ssh auf Ihren Wohnzimmer-PC zugreifen. Hierdurch teilt man der Konsole mit, dass der erste (also eigentlich der nullte) angeschlossene Bildschirm (und damit der Fernseher) genutzt werden soll und nicht der Bildschirm des Remote-Rechners. Das Setzen der DISPLAY-Variable ist oft hilfreich, wenn man Programme remote starten möchte, deren Ausgabe dann aber auf dem Fernseher erfolgt.

Die erzeugte EDID-Firmware-Datei muss nun noch an die dafür vorgesehene Stelle kopiert werden,

Code-Ausschnitt 7.168: Kopieren der EDID-Daten

```
sudo mkdir /lib/firmware/edid
sudo cp edid.bin /lib/firmware/edid/edid.bin
```

Die hier beschriebene Vorgehensweise funktioniert immer. Auf manchen Systemen ist die Datei */sys/class/drm/card0-HDMI-A-1/edid* vorhanden und nicht leer. Falls das bei Ihnen der Fall ist, können Sie sich den Umweg über das Python-Skript sparen und stattdessen einfach den Aufruf

Code-Ausschnitt 7.169: Kopieren der EDID-Daten - 2

```
sudo mkdir /lib/firmware/edid
sudo cp /sys/class/drm/card0-HDMI-A-1/edid /lib/firmware/edid/edid.bin
```

verwenden.

Dem Kernel müssen nun noch entsprechende Boot-Parameter übergeben werden. Die Boot-Parameter werden in der Datei */etc/default/grub* hinterlegt und sehen für den *grub*-Bootloader wie folgt aus:

Code-Ausschnitt 7.170: Ändern der Grub-Kommando-Zeile

```
...
GRUB_CMDLINE_LINUX="video=HDMI-A-1:1920x1080@50D drm.edid_firmware=HDMI-A-1:edid/edid.bin"
...
```

Diese Zeile zwingt den HDMI-Anschluss zu einer Full-HD-Auflösung mit einer Bildwiederholrate von 50 Hz (1920x1080@50p). Der Buchstabe *D* ist dabei kein Tipp-Fehler. Vielmehr handelt es sich hier um einen digitalen Ausgang. Die Datei *edid.bin* muss ebenfalls im *initramfs* (initial ram filesystem) zur Verfügung stehen. Dieses enthält im Arbeitsspeicher die zum Systemstart erforderlichen Dateien. Bitte editieren Sie hierzu als *Root* die folgende Datei

Code-Ausschnitt 7.171: EDID-Daten zum *initramfs* hinzufügen – 1

```
sudo vi /etc/initramfs-tools/hooks/include-edid-data
```

und fügen Sie den folgenden Inhalt hinzu:

Code-Ausschnitt 7.172: EDID-Daten zum *initramfs* hinzufügen – 2

```
#!/bin/sh
PREREQ="udev"
prereqs()
{
    echo "$PREREQ"
}
case $1 in
prereqs)
    prereqs
    exit 0
;;
esac
. /usr/share/initramfs-tools/hook-functions
# Begin real processing below this line
if [ ! -e "${DESTDIR}/lib/firmware/edid" ]; then
    mkdir -p "${DESTDIR}/lib/firmware/edid"
fi
if [ -r "/lib/firmware/edid/edid.bin" ]; then
    cp "/lib/firmware/edid/edid.bin" "${DESTDIR}/lib/firmware/edid/"
fi
manual_add_modules i915 radeon
exit 0
```

Machen Sie die Datei im Anschluss ausführbar.

Code-Ausschnitt 7.173: EDID-Daten zum *initramfs* hinzufügen – 3

```
sudo chmod +x /etc/initramfs-tools/hooks/include-edid-data
```

Die neuen Konfigurationen werden wie folgt übernommen:

Code-Ausschnitt 7.174: Neue *initramfs* und Kernel-Optionen übernehmen

```
sudo update-initramfs -u
sudo update-grub
```

Abschließend können die EDID-Daten noch für den X-Server übernommen werden. Editieren Sie dazu die Datei */etc/X11/xorg.conf.d/20-intel.conf* wie folgt:

Code-Ausschnitt 7.175: X11 konfigurieren

```

Section "Device"
    Identifier "Device0"
    Driver     "intel"
    VendorName "Intel(R) Corporation"
    BoardName  "Intel HD Graphics 615"
    Option     "TearFree" "true"
    Option     "DRI"      "3"
    Option     "UseEDID"  "true"
    Option     "CustomEDID" "HDMI1:/lib/firmware/edid/edid.bin"
EndSection
Section "Screen"
    Identifier "Screen0"
    Device     "Device0"
    Monitor    "HDMI1"
    DefaultDepth 24
    SubSection "Display"
        Depth    24
        Modes    "1920x1080@50" "1920x1080@60" "1920x1080@24"
    EndSubSection
EndSection
Section "Monitor"
    Identifier "HDMI1"
    Modeline  "1920x1080@50" 141.50 1920 2032 2232 2544 1080 1083 1088 1114 -hsync +vsync
    Modeline  "1920x1080@60" 173.00 1920 2048 2248 2576 1080 1083 1088 1120 -hsync +vsync
    Modeline  "1920x1080@24"  63.00 1920 1976 2160 2400 1080 1083 1088 1098 -hsync +vsync
EndSection

```

Falls die Datei nicht existiert, können Sie diese neu anlegen. Nach einem Reboot mit ausgeschaltetem Fernseher (oder gezogenem HDMI-Kabel) sollten sich ähnliche Boot-einträge auf Ihrem System finden (*dmesg | grep drm*):

Code-Ausschnitt 7.176: Boot-Einträge beim Booten mit vorgegebenen EDID-Daten

```

[ 1.451717] [drm] Supports vblank timestamp caching Rev 2 (21.10.2013).
[ 1.451717] [drm] Driver supports precise vblank timestamp query.
[ 1.452867] [drm] Finished loading DMC firmware i915/kbl_dmc_ver1_04.bin (v1.4)
[ 2.539216] [drm] failed to retrieve link info, disabling eDP
[ 2.539293] [drm] forcing HDMI-A-1 connector on
[ 2.540955] [drm] Initialized i915 1.6.0 20190619 for 0000:00:02.0 on minor 0
[ 2.544114] [drm] Got external EDID base block and 1 extension from "edid/edid.bin" for ↵
connector: "HDMI-A-1"
[ 2.657957] fbcon: i915drmfb (fb0) is primary device
[ 2.658021] i915 0000:00:02.0: fb0: i915drmfb frame buffer device

```

Die Zeilen mit dem Schlüsselwort *Modeline* im Code-Ausschnitt 7.175 geben die Timings für die verschiedenen Auflösungen wieder. Für eine Intel-Grafikeinheit können sie wie folgt ermittelt werden:

Code-Ausschnitt 7.177: Ermitteln der unterstützten Auflösungen

```
DISPLAY=:0.0 xrandr -q
```

Das Programm *cvt* erlaubt dann die Berechnung der *Modeline*.

Code-Ausschnitt 7.178: Berechnung einer Modeline

```
cvt 1920 1080 50
```

Für eine Auflösung von 1920 × 1080 Pixeln bei einer Wiederholrate von 50 Hz sieht das Ergebnis so aus:

Code-Ausschnitt 7.179: Eintragen einer Modeline

```
# 1920x1080 49.93 Hz (CVT 2.07M9) hsync: 55.62 kHz; pclk: 141.50 MHz
Modeline "1920x1080_50.00" 141.50 1920 2032 2232 2544 1080 1083 1088 1114 -hsync +vsync
```

Die so ermittelten Werte können dann in der Sektion „Monitor“ in der Datei `20-intel.conf` eingetragen werden.

Bitte beachten Sie, dass die Modeline-Ausgaben mit einem Unterstrich „_“ erfolgen, aber mit einem „@“ eingetragen sind. Sowohl in der Datei `/etc/default/grub` als auch in der Datei `20-intel.conf` müssen diese Bezeichnungen identisch sein.



Sollte einmal ein Reset der Bildschirmausgabe erforderlich sein, kann das mit Hilfe des folgenden Skriptes erledigt werden:

Code-Ausschnitt 7.180: Reset der Bildschirmausgabe

```
#!/bin/bash
xrandr -display :0.0 --output HDMI1 --off
xrandr -display :0.0 --output HDMI1 --auto
```

Für meine Konfiguration mit zwei verschiedenen Screens (TV an DP1 und LCD an HDMI1) habe ich die Grub-Kommandozeile wie folgt angepasst:

Code-Ausschnitt 7.181: Grub Kommandozeile mit zwei Screens

```
GRUB_CMDLINE_LINUX="video=HDMI-A-1:1280x720_60.00D drm.edid_firmware=HDMI-A-1:edid/edid-↵
lcdhdm1.bin"
```

Die EDID-Datei des LCDs habe ich dabei unter dem Namen `edid-lcdhdm1.bin` gespeichert, die Modeline (1280x720_60.00D) findet sich in der Datei `20-intel.conf` im Code-Ausschnitt 7.140 unter den LCD-Einstellungen. Auf meinem Lattepanda Alpha ist es mir nicht gelungen, den DP1-Eingang auch bei ausgeschaltetem Fernseher dauerhaft einzuschalten. Das liegt vermutlich am USB-C-nach-HDMI-Adapter, den ich benutze. Dennoch kann ich mit der oben gezeigten Grub-Kommandozeile meinen Wohnzimmer-PC mit eingeschaltetem LCD und ausgeschaltetem Fernseher booten und diesen nachträglich einschalten. Nach kurzer Zeit wird dann das Bild auf dem Fernseher sichtbar.

Zusammenfassung und Ausblick 7 In diesem Kapitel haben wir einen digitalen Videorekorder gebaut, der nicht nur Fernsehprogramm empfangen und darstellen, sondern auch aufzeichnen kann. Den VDR haben wir mit einer Fernbedienung verbunden.

Verschiedene Plugins erweitern die Videorekorder-Funktionalität und hübschen das Aussehen des VDR auf und erlauben das Streaming des Fernsehprogramms auf ein mobiles Gerät im Heimnetz.

Weiterhin habe ich Ihnen gezeigt, wie Sie eine externe Festplatte an Ihren Receiver anschließen und nutzen können, um Fernsehprogramm aufzuzeichnen.

Im nächsten Kapitel werfen wir einen Blick auf *KODI*, den Nachfolger des Xbox-Media-Centers (*XBMC*). Wir werden *KODI* nicht nur aus den Sourcen installieren, sondern uns auch die *LibreELEC*-Distribution näher anschauen. ■

Literaturverzeichnis

- [54] Klaus Schmidinger. *VDR - The Video Disk Recorder*. 2019.
<http://www.tvdr.de>
besucht am 27. 12. 2019 (siehe Seite 137).
- [55] VDR Wiki. *Herzlich Willkommen auf den Seiten von VDR Wiki!* 2019.
<http://www.vdr-wiki.de>
besucht am 27. 12. 2019 (siehe Seiten 137, 138, 154, 161, 176, 180, 195).
- [56] Ludmila Rakova. *VDR Portal*. 2019.
<http://www.vdr-portal.de>
besucht am 27. 12. 2019 (siehe Seiten 137, 158, 169, 173, 199).
- [57] Dipl.-Ing. Claus Muus. *MiniDVBLinux Distribution (MLD)*. 2019.
<http://www.minidvblinux.de>
besucht am 27. 12. 2019 (siehe Seite 137).
- [58] Sundtek. *Netzwerk Installer*. 2019.
http://www.sundtek.de/media/sundtek_netinst.sh
besucht am 27. 12. 2019 (siehe Seite 138).
- [59] DVBSky. *Webseite*. 2019.
<http://dvbsky.net>
besucht am 27. 12. 2019 (siehe Seite 142).
- [60] LinuxTV. *Multimedia and Television Support on Linux*. 2019.
<http://www.linuxtv.org>
besucht am 27. 12. 2019 (siehe Seite 143).
- [61] Flirc. *Homepage*. 2019.
<https://flirc.tv/>
besucht am 27. 12. 2019 (siehe Seite 145).
- [62] Wiki. *Lirc*. 2019.
<https://wiki.ubuntuusers.de/Lirc/>
besucht am 27. 12. 2019 (siehe Seite 149).
- [63] Lirc homepage. *Lirc Infrared Linux Remote Control*. 2019.
<http://www.lirc.org/html/table.html>
besucht am 27. 12. 2019 (siehe Seite 150).
- [64] Sourceforge. *Lirc*. 2019.
<http://lirc.sourceforge.net/lirc.org/>
besucht am 27. 12. 2019 (siehe Seite 150).
- [65] Github. *A VA-API output device plugin for VDR*. 2019.
<https://github.com/pesintta/vdr-plugin-vaapidevice>
besucht am 27. 12. 2019 (siehe Seite 156).
- [66] ffmpeg. *Index of/releases*. 2019.
<https://www.ffmpeg.org/releases>
besucht am 27. 12. 2019 (siehe Seite 156).
- [67] Github. *A VA-API output device plugin for VDR*. 2019.
<https://github.com/rofafor/vdr-plugin-vaapidevice>
besucht am 27. 12. 2019 (siehe Seite 157).

- [68] Github. *A VA-API output device plugin for VDR*. 2019.
<https://github.com/jojo61/vdr-plugin-softhdcpuvid>
besucht am 27. 12. 2019 (siehe Seite 158).
- [69] vdr-developer.org. *vdradmin-am.git*. 2019.
<http://projects.vdr-developer.org/git/vdradmin-am.git/>
besucht am 27. 12. 2019 (siehe Seite 168).
- [70] vdr-developer.org. *VDR Skindesigner Plugin*. 2019.
<https://projects.vdr-developer.org/projects/plg-skindesigner/wiki>
besucht am 27. 12. 2019 (siehe Seite 169).
- [71] FrodoVDR. *channellogos*. 2019.
<https://github.com/FrodoVDR/channellogos>
besucht am 27. 12. 2019 (siehe Seite 172).
- [72] Webseite. *Shady-Skin*. 2019.
<http://anthra.de/shady-skin>
besucht am 27. 12. 2019 (siehe Seite 176).
- [73] QooTec. *3.5 inch HDMI LCD*. 2019.
https://www.amazon.de/QooTec-Touchscreen-Raspberry-Touchpen-Raspbian/dp/B07PLVQC78/ref=sr_1_8?__mk_de_DE=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=3.5+zoll+hdmilcd&qid=1577437846&s=computers&sr=1-8
besucht am 27. 12. 2019 (siehe Seite 180).
- [74] Wiki. *EasyVDR*. 2019.
<https://wiki.easy-vdr.de>
besucht am 27. 12. 2019 (siehe Seite 184).
- [75] warmcat. *libwebsockets releases*. 2019.
<https://github.com/warmcat/libwebsockets/releases>
besucht am 27. 12. 2019 (siehe Seite 184).
- [76] Ask Ubuntu. *Screen Corruption on Multi Monitor DisplayPort using Intel Graphics*. 2019.
<https://askubuntu.com/questions/995077/screen-corruption-on-multi-monitor-displayport-using-intel-graphics?rq=1>
besucht am 27. 12. 2019 (siehe Seite 190).
- [77] Christian Hennek. *Futaba MDM166A USB VFD Display*. 2019.
http://www.eisgold.de/Futaba-MDM166A-USB-VFD-Display_detail_2732.html?osCsId=4c0613e6151ebd8cb71ed9724ef357ac
besucht am 01. 01. 2020 (siehe Seite 193).
- [78] vlc. *VLC Media Player*. 2019.
<https://www.vlc.de/>
besucht am 27. 12. 2019 (siehe Seite 195).
- [79] Digital Devices. *Webseite*. 2019.
<https://digitaldevices.de>
besucht am 27. 12. 2019 (siehe Seite 196).
- [80] TBS technology. *DVB Produkte*. 2019.
<https://www.tbs-technology.de>
besucht am 27. 12. 2019 (siehe Seite 197).

- [81] Gary Hunt. *TBS Kernel Treiber*. 2019.
https://github.com/tbsdtv/linux_media/wiki
besucht am 27. 12. 2019 (siehe Seite 197).
- [82] Hauppauge. *WinTV USB-CI*. 2019.
<https://www.amazon.de/Hauppauge-WinTV-CI-Common-Interface-Karte/dp/B000Y19S12>
besucht am 27. 12. 2019 (siehe Seite 198).
- [83] Helmut Binder. *USB2CI Treiber*. 2019.
<https://github.com/siricco/usb2ci>
besucht am 27. 12. 2019 (siehe Seite 198).
- [84] Matt Vollrath. *get-edid.py*. 2019.
<https://gist.githubusercontent.com/mvollrath>
besucht am 27. 12. 2019 (siehe Seite 200).

8. KODI



8.1	KODI aus dem Repository	210
8.2	KODI aus dem Quelltext	210
8.3	LibreELEC	214
8.4	Mobile Helfer	229
8.5	AirPlay und Co.	230
8.6	MP3	233
8.7	Wichtige Hinweise	233

In diesem Kapitel schauen wir uns das Mediacenter KODI näher an. Dabei möchte ich Ihnen drei verschiedene Wege vorstellen, um KODI auf den Bildschirm zu zaubern. Wir werden es per apt-get installieren, aus dem Quelltext übersetzen und wir werfen einen Blick auf die LibreELEC-Distribution. Alle drei Wege haben ihre Vor- und Nachteile. Möchten Sie unter Ubuntu arbeiten, weil Sie beispielsweise den VDR benötigen, macht es sicherlich Sinn, KODI aus dem Repository zu installieren. LibreELEC hingegen ist eine kleine, leistungsfähige, eigene Distribution. Allerdings gibt es hier nicht die Möglichkeit, weitere Programme zu installieren außer denen, die für LibreELEC vorgesehen sind. Weiterhin werden wir einen Blick auf wichtige Erweiterungen werfen, wie beispielsweise einem Fernseh-Backend namens tvheadend oder den Airplay-Streaming-Service, der es ermöglicht, Musik drahtlos von einem Apple-Device (iPhone, iPad) zu einer Stereoanlage zu übertragen.



KODI

KODI ist aus dem XBox Media Center - kurz XBMC - hervorgegangen. XBMC wurde umbenannt, um rechtliche Probleme mit der XBox zu vermeiden. Inzwischen läuft KODI längst nicht mehr nur auf Microsoft's Spielekonsole, sondern auf vielen Embedded Boards und PCs. Dabei kann man auf eine Vielzahl von fertigen Distributionen oder sogar Bundles (Embedded Board mit installiertem KODI) zurückgreifen - der ideale Einstieg für Neulinge. Zu den bekanntesten Distributionen zählen OpenElec [85] (<https://openelec.tv> [86]) oder LibreELEC (<https://libreelec.tv/> [87]).

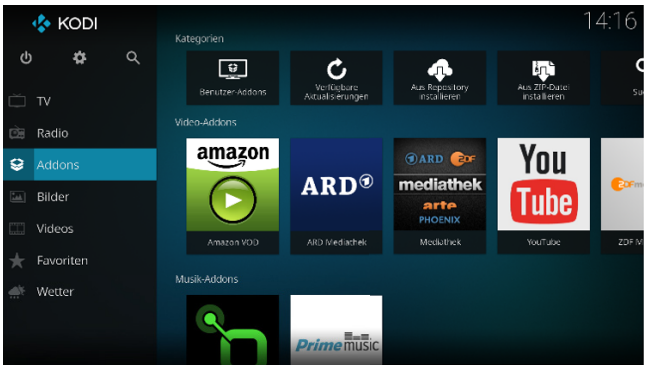


Abbildung 8.1: Das KODI Mediacenter spielt Filme, zeigt Bilder und vieles mehr.

Allerdings haben diese Distributionen alle eine gemeinsame Schwäche: Sie sind für Einsteiger gedacht und bringen keine Entwickler-Werkzeuge mit sich. Sie werden also Probleme haben, beispielsweise zwischen reinem VDR- und KODI-Betrieb umzuschalten oder KODI vom VDR aus aufzurufen. Auch für KODI gibt es eine Fülle an Plugins, um den VDR einzubinden, die aktuelle Wettervorhersage abzurufen oder um YouTube-Videos zu schauen.

8.1 KODI aus dem Repository

Beginnen wir damit, KODI aus einer Konsole heraus per *apt-get* zu installieren.

Code-Ausschnitt 8.1: Die Installation von KODI aus den Repositories

```
sudo apt-get install kodi
```

Nach erfolgreicher Installation können Sie KODI durch die Eingabe von *kodi* in einem Terminal starten. Das Programm steht ebenfalls unter den Anwendungen zur Verfügung und kann von dort aus mit einem Klick auf das Logo gestartet werden.

8.2 KODI aus dem Quelltext

Vielleicht werden Sie sich fragen, welchen Sinn es macht, KODI aus den Quellen zu installieren. Die Antwort ist ähnlich wie die bei der Installation von VDR. Die Pakete aus den Repositories hängen oft einige Versionsnummern hinterher. Möchte man also eine brandaktuelle KODI-Version haben, bietet sich das eigene Kompilieren an. KODI hängt von ganz vielen anderen Paketen ab, die wir alle installieren müssen, bevor das Programm übersetzt werden kann. Weil es sehr mühsam ist, alle Pakete zu suchen und sich die erforderliche Paketliste auch ändern kann, schlagen wir hier einen bequemer Weg ein: Wir lassen Ubuntu selbst alle erforderlichen Pakete suchen. Damit das funktioniert, müssen wir zunächst die Quellen für die Source-Codes in der Datei */etc/apt/sources.list* auskommentieren. Das erreichen wir durch Entfernen des #-Zeichens am jeweiligen Zeilenanfang. Da die o. g. Datei in einem geschützten Bereich steht, muss mit Hilfe von Root-Rechten editiert werden. Meine Datei *sources.list* sieht nach einem Update auf Ubuntu Dingo so aus:

Code-Ausschnitt 8.2: Freigabe der Ubuntu-Quelltexte

```
# deb cdrom:[Ubuntu 19.04 _Disco Dingo_ - Alpha amd64 (20190326.2)]/ disco main restricted

# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://de.archive.ubuntu.com/ubuntu/ disco main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://de.archive.ubuntu.com/ubuntu/ disco-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://de.archive.ubuntu.com/ubuntu/ disco universe
deb-src http://de.archive.ubuntu.com/ubuntu/ disco universe
deb http://de.archive.ubuntu.com/ubuntu/ disco-updates universe
deb-src http://de.archive.ubuntu.com/ubuntu/ disco-updates universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://de.archive.ubuntu.com/ubuntu/ disco multiverse
deb-src http://de.archive.ubuntu.com/ubuntu/ disco multiverse
deb http://de.archive.ubuntu.com/ubuntu/ disco-updates multiverse
deb-src http://de.archive.ubuntu.com/ubuntu/ disco-updates multiverse

## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
deb http://de.archive.ubuntu.com/ubuntu/ disco-backports main restricted universe multiverse

## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and the
## respective vendors as a service to Ubuntu users.
deb http://archive.canonical.com/ubuntu disco partner
deb-src http://archive.canonical.com/ubuntu disco partner

deb http://security.ubuntu.com/ubuntu disco-security main restricted
deb http://security.ubuntu.com/ubuntu disco-security universe
deb-src http://security.ubuntu.com/ubuntu disco-security universe
deb http://security.ubuntu.com/ubuntu disco-security multiverse
# deb-src http://security.ubuntu.com/ubuntu disco-security multiverse

# This system was installed using small removable media
# (e.g. netinst, live or single CD). The matching "deb cdrom"
# entries were disabled at the end of the installation process.
# For information about how to configure apt package sources,
# see the sources.list(5) manual.
# deb http://archive.canonical.com/ precise partner
# deb-src http://archive.canonical.com/ precise partner
```

Nach dem Editieren dieser Liste muss dem System mitgeteilt werden, dass es neue Einträge gibt. Das geschieht so:

Code-Ausschnitt 8.3: Aktualisieren des Ubuntu-Systems

```
sudo apt-get update
sudo apt-get upgrade
```

Alle Pakete, die zum Übersetzen von *KODI* erforderlich sind, können nun einfach so installiert werden:

Code-Ausschnitt 8.4: Installieren der Abhängigkeiten für *KODI*

```
sudo apt-get build-dep kodi
```

Würden Sie alle zum Übersetzen von *KODI* erforderlichen Pakete manuell installieren wollen, sähe die Liste so aus:

Code-Ausschnitt 8.5: Manuelles Installieren der Abhängigkeiten für *KODI*

```
sudo apt install debhelper autoconf automake autopoint gettext autotools-dev cmake curl default-
-jre doxygen gawk gcc gdc gperf libasound2-dev libass-dev libavahi-client-dev libavahi-
common-dev libbluetooth-dev libbluray-dev libbz2-dev libcdio-dev libp8-platform-dev
libcrossguid-dev libcurl4-openssl-dev libcwidi-dev libdbus-1-dev libegl1-mesa-dev libenca-
dev libflac-dev libfontconfig-dev libfreetype6-dev libfribidi-dev libfstcmp-dev libgcrypt-
dev libgif-dev libgles2-mesa-dev libgl1-mesa-dev libglu1-mesa-dev libgnutls28-dev libgpg-
error-dev libiso9660-dev libjpeg-dev liblcms2-dev libltdl-dev liblzo2-dev libmicrohttpd-dev
libmysqlclient-dev libnfs-dev libogg-dev libpcre3-dev libplist-dev libpng-dev libpulse-dev
libshairplay-dev libsmclient-dev libsqlite3-dev libssl-dev libtag1-dev libtiff5-dev
libtinyxml-dev libtool libudev-dev libva-dev libvdpau-dev libvorbis-dev libxmu-dev
libxrandr-dev libxslt1-dev libxt-dev lsb-release python-dev python-pil rapidjson-dev swig
unzip uuid-dev yasm zip zlib1g-dev
```

Da *KODI* in der Lage ist, DVDs abzuspielen, darf an dieser Stelle der Hinweis auf die *libdvdcss* nicht fehlen. Diese Bibliothek ist rechtlich umstritten, da sie in der Lage ist, die DVD-CSS-Verschlüsselung von Kauf-DVDs auszuhebeln. Ein finnisches Gericht hat 2007 entschieden, dass die Nutzung der *libdvdcss* legal ist. In Finnland könnte man die Bibliothek also einfach mittels

Code-Ausschnitt 8.6: Installieren der *libdvdcss*

```
sudo apt-get install libdvdcss2
```

installieren oder die neueste Version von *Videolan* mit den folgenden Kommandos herunterladen und installieren.

Code-Ausschnitt 8.7: Installieren der *libdvdcss* aus den Quelltexten

```
git clone git://git.videolan.org/libdvdcss
cd libdvdcss
make
sudo make install
```

In Deutschland ist die rechtliche Situation jedoch nicht klar (<http://de.wikipedia.org/wiki/Libdvdcss> [88]).

Zum Übersetzen von *KODI* werden einige weitere Pakete benötigt, die *build-dep* nicht alle installiert hat.

Code-Ausschnitt 8.8: Installieren weiterer Abhängigkeiten für *KODI*

```
sudo apt install libcec-dev libfmt-dev liblirc-dev doxygen libcap-dev libsndio-dev libmariadb-
dev
```

Nun ist es an der Zeit, die aktuelle Version von *KODI* auszuchecken. Wechseln Sie dazu in ein Verzeichnis Ihrer Wahl, beispielsweise das *Home*-Verzeichnis und laden Sie die Quelltexte von *KODI* so herunter:

Code-Ausschnitt 8.9: KODI-Quelltexte herunterladen

```
cd $HOME
git clone https://github.com/xbmc/xbmc kodi
```

Erstellen Sie danach ein Verzeichnis, in dem *KODI* kompiliert und gelinkt werden kann. Das Verzeichnis sollte sich nicht in dem oben angelegten *KODI*-Quelltext-Verzeichnis befinden.

Code-Ausschnitt 8.10: KODI-Build-Verzeichnis erstellen

```
mkdir $HOME/kodi-build
```

Wechseln Sie danach bitte in dieses Verzeichnis, konfigurieren Sie *KODI* und übersetzen Sie es so:

Code-Ausschnitt 8.11: KODI übersetzen

```
cd $HOME/kodi-build
cmake ../kodi -DCMAKE_INSTALL_PREFIX=/usr/local
cmake --build . -- VERBOSE=1 -j4
```

Als Prefix habe ich wieder */usr/local* gewählt. Nach einem *sudo make install* finden Sie die ausführbare *KODI*-Datei also unter */usr/local/bin*. Zum Übersetzen selbst habe ich alle vier Kerne meiner CPU ausgewählt (*-j 4*). Nachdem *KODI* erfolgreich übersetzt worden ist, können Sie es aufrufen und testen, bevor Sie es installieren:

Code-Ausschnitt 8.12: KODI aufrufen

```
./kodi-x11
```

Die Installation ist dann denkbar einfach:

Code-Ausschnitt 8.13: KODI installieren

```
sudo make install
```

Falls Sie *KODI* doch an einem anderen Ort als dem durch den Prefix vorgegebenen installieren möchten, können Sie das Installations-Verzeichnis so ändern:

Code-Ausschnitt 8.14: KODI in einem anderen Verzeichnis installieren

```
sudo make install DESTDIR=/verzeichnis
```

KODI sollte nun bereits vollständig mit Maus und Tastatur bedienbar sein. Eine Fernbedienung fügen wir später hinzu.

8.3 LibreELEC

LibreELEC – Just enough OS for KODI. Mit diesem Slogan wirbt die Webseite der *LibreELEC*-Macher (<https://libreelec.tv> [87]). Und der Slogan ist Programm. *LibreELEC* installiert lediglich soviel - oder besser gesagt so wenig - Betriebssystem, wie es zum Betrieb von *KODI* erforderlich ist. *LibreELEC* selbst ist aus dem Open-Source-Projekt *OpenELEC* hervorgegangen, nachdem verschiedene Entwickler unterschiedliche Meinungen hatten, wie die Software weiterentwickelt werden sollte. Etwas Ähnliches ist übrigens gerade bei *LibreELEC* geschehen: Einige Entwickler haben sich abgesetzt und bieten nun *CoreELEC* an (<http://www.coreelec.org> `url:coreelec`), eine *LibreELEC*-Version für Amlogic Embedded Boards (S905-, S912 und S922-Prozessoren).



Abbildung 8.2: LibreELEC – Just enough OS for KODI (Quelle: <https://www.kodinerds.net/> [89]).

In diesem Abschnitt zeige ich Ihnen, wie Sie *LibreELEC* auf einer Festplatte installieren, eine Fernbedienung anlernen können und *LibreELEC* für den Fernsehempfang vorbereiten. Am Ende werfen wir noch einen Blick auf einige Add-ons, wie beispielsweise die ARD-Mediathek oder Amazon Prime Video. Das Abspielen von Blu-rays darf natürlich auch nicht fehlen. *LibreELEC* ist also eine perfekte Ergänzung für einen Fernseher, der (noch) nicht Multimedia-tauglich ist.

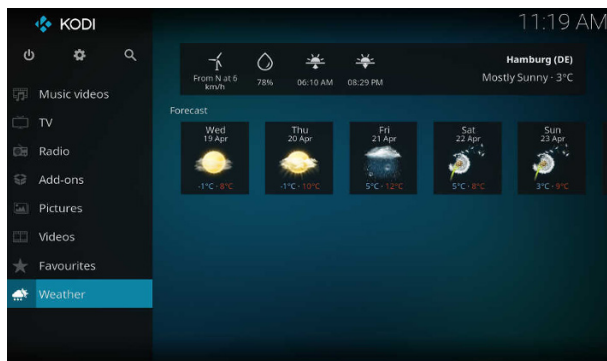


Abbildung 8.3: Estuary ist der Standard-Skin für KODI (Quelle: kodi.wiki).

LibreELEC nimmt gerne das komplette Device (z. B. USB-Stick) in Beschlag, auf dem es installiert wird. Es gibt keine Auswahlmöglichkeit, auf welcher Partition installiert werden soll. Darum zeige ich Ihnen hier einen Weg, der es Ihnen erlaubt, *LibreELEC* auf jeder beliebigen Partition zu installieren.

LibreELEC benötigt zwei Partitionen. Die erste Partition beinhaltet das gesamte Betriebssystem. Sie ist 512 MB groß und heißt „LIBREELEC“. Die zweite Partition beinhaltet die *LibreELEC*-Daten. Sie ist eine *ext4*-Partition und heißt „STORAGE“. Diese Partition kann beliebig groß werden. Nach dem ersten Start von *LibreELEC* vergrößert *LibreELEC* diese automatisch auf die volle Größe.

Um die Struktur unserer Festplatte nicht zu gefährden, müssen wir *LibreELEC* daher zuerst auf einem USB-Stick installieren, um es anschließend auf die Festplatte zu

übertragen. Als Stick-Größe sind dabei alle USB-Sticks ausreichend, die mindestens 1 GB Speicher bieten. Laden Sie sich zunächst zur Installation das aktuelle Image von <https://libreelec.tv> [87] unter der Rubrik *Downloads* herunter. Zum Zeitpunkt, als ich dieses Buch geschrieben habe, war das Image *[LibreELEC-Generic.x86_64-9.1.501.img.gz]* das aktuelle. Es ist identisch für alle Intel- oder AMD-CPU/GPU und ist ca. 243 MB groß. Unter Windows gibt es viele Programme (z. B. *win32disk imager*), die ein solches Image unter Verwendung einer grafischen Benutzeroberfläche auf einen USB-Stick schreiben können. Ich zeige Ihnen hier einen Weg, der mit der Konsole eines LINUX-Systems auskommt. Ich gehe davon aus, dass sich das heruntergeladene Image im Verzeichnis *~/Downloads* befindet. Wechseln Sie daher bitte zunächst in dieses Verzeichnis und entpacken Sie die Datei wie folgt:

Code-Ausschnitt 8.15: KODI-Image entpacken

```
cd ~/Downloads
gunzip -d LibreELEC-Generic.x86_64-9.1.501.img.gz
```

Stecken Sie als Nächstes einen USB-Stick in einen freien USB-Port.

Der folgende Vorgang löscht alle Daten auf Ihrem Stick. Bitte versichern Sie sich zuerst, dass Sie eventuell vorhandene Daten auf dem USB-Stick nicht mehr benötigen.



LINUX bindet USB-Sticks unter dem Device */dev/sdX* ein, wobei *X* ein Buchstabe mit *a* beginnend für das erste Device ist. Der Befehl *dmesg* hilft Ihnen dabei, das zuletzt eingesetzte USB-Device zu identifizieren. Beim Einschieben meines 16 GB USB-Sticks erscheint dort beispielsweise die Meldung

Code-Ausschnitt 8.16: Einschieben eines USB-Sticks

```
[23696.069596] sd 0:0:0:0: [sda] 30851072 512-byte logical blocks: (15.8 GB/14.7 GiB)
```

Bei mir steht der USB-Stick unter */dev/sda* zur Verfügung. Stellen Sie vor dem Beschreiben bitte sicher, dass der USB-Stick nicht gemounted ist. Das erreichen Sie mit dem folgenden Befehl:

Code-Ausschnitt 8.17: Unmounten eines USB-Sticks

```
umount /dev/sda
```

Schreiben Sie das Image so auf den USB-Stick:

Code-Ausschnitt 8.18: KODI-Image auf den USB-Stick schreiben

```
sudo dd if=LibreELEC-Generic.x86_64-7.0.0.img of=/dev/sdX bs=4M
```

Bitte stellen Sie noch einmal sicher, dass Sie nicht auf das falsche Device schreiben. Das Schreiben löscht unwiderruflich alle Daten auf dem Device.



Bevor Sie den USB-Stick entfernen, werden noch einmal alle Daten synchronisiert.

Code-Ausschnitt 8.19: Synchronisieren der Daten auf dem USB-Stick

```
sync
```

Booten Sie Ihr System anschließend neu und stellen Sie sicher, dass Ihr Wohnzimmer-PC vom USB-Stick bootet. Die dafür erforderlichen Einstellungen finden Sie im BIOS.

Achten Sie darauf, dass während des Boot-Vorganges eine USB-Tastatur an Ihrem Wohnzimmer-PC angeschlossen ist. Geben Sie mit dieser das Kommando

Code-Ausschnitt 8.20: Starten von *LibreELEC* vom USB-Stick aus

```
run
```

in der Eingabeaufforderung ein. Der USB-Stick wird jetzt als Start-Medium für *LibreELEC* verwendet. Beim ersten Start formatiert *LibreELEC* den USB-Stick so um, dass das System selbst von einer 512 MB großen Partition startet und der Rest des Sticks als „STORAGE“ zur Verfügung steht. Sie können *LibreELEC* nun ausprobieren oder booten Ihr LINUX-System erneut, um *LibreELEC* auf die Festplatte Ihres Wohnzimmer-PCs zu installieren. Auf meiner Festplatte habe ich dafür zwei Partitionen vorgesehen. Da meine Festplatte eine NVME-Festplatte ist, heißen die Partitionen dementsprechend:

Code-Ausschnitt 8.21: Meine vorbereitete Festplatte für *LibreELEC*

```
/dev/nvme0n1p5 1417883648 1418932223 1048576 512M Microsoft Basisdaten
/dev/nvme0n1p6 1418932224 1625284607 206352384 98,4G Linux-Dateisystem
```

Die Partition 5 meiner NVME-Platte wird später das *LibreELEC*-System beinhalten. Für die Daten habe ich die Partition 6 meiner NVME-Platte vorgesehen. Bitte ersetzen Sie die hier angegebenen Daten mit den Partitionen Ihrer Festplatte. Setzen Sie nach dem Booten Ihres LINUX-Systems den eben vorbereiteten *LibreElec*-USB-Stick wieder ein. In einem ersten Schritt übertragen wir nun die Systemdaten. Dazu nutzen wir wieder den Befehl *dd* (Disk Dump):

Code-Ausschnitt 8.22: *LibreELEC* von USB-Stick auf Festplatte umziehen

```
sudo dd if=/dev/sda1 of=/dev/nvme0n1p5
```



Seien Sie besonders beim Device-Namen hinter dem output-file (of) vorsichtig. Hier werden wieder alle Daten auf dem Ziel-Device unwiederbringlich gelöscht.

Nachdem das System nun erfolgreich vom USB-Stick übertragen wurde, müssen wir ihm noch mitteilen, wo die Daten gespeichert werden sollen. Diese sollen ja eben nicht mehr vom USB-Stick geladen werden, sondern ebenfalls von der Festplatte, in meinem Fall von der Partition Nr. 6. Mounten Sie zunächst die neu geschriebene Systempartition so:

Code-Ausschnitt 8.23: *LibreELEC*: Systempartition mounten

```
sudo mount /dev/nvme0n1p5 /mnt
```

Bitte ersetzen Sie die Partition 5 meiner NVME-Festplatte wieder durch die *LIBREElec*-Partition Ihrer Festplatte. Nach dem erfolgreichen Einbinden der Partition in das Verzeichnis */mnt* finden Sie dort eine Datei mit dem Namen *syslinux.cfg*. Diese Datei gibt unter Anderem an, wo sich die Daten für *LibreELEC* befinden. Ich habe die Datei wie folgt abgeändert:

Code-Ausschnitt 8.24: *syslinux.cfg*-Änderungen

```
DEFAULT run
TIMEOUT 1

LABEL run
  KERNEL /KERNEL
  APPEND boot=UUID=2102-3124 disk=UUID=6d70cf40-cdee-4cc0-b68a-ef9202904e4e tty portable quiet
```

Bitte beachten Sie, dass Sie alle Änderungen an dieser Datei als Administrator *Root* durchführen müssen. Das Timeout zum Laden des Systems habe ich auf eine Sekunde herabgesetzt. So bootet *LibreELEC* deutlich schneller und wartet nicht so lange. Die Storage-UUID (Universally Unique Identifier) der Festplatte (in meinem Fall Partition 6 der NVME-Platte) erhalten Sie durch Eingabe des Befehls

Code-Ausschnitt 8.25: UUID einer Festplatte ermitteln

```
sudo blkid /dev/nvme0n1p6
```

In meinem Fall lautet die Ausgabe

Code-Ausschnitt 8.26: UUID einer Festplatte ermitteln - Beispiel

```
/dev/nvme0n1p6: LABEL="STORAGE" UUID="6d70cf40-cdee-4cc0-b68a-ef9202904e4e" TYPE="ext4" ↔  
PARTLABEL="STORAGE" PARTUUID="827aa36d-6025-480b-9b48-03349137002b"
```

Die Boot-UUID erhalten Sie auf dieselbe Art und Weise, in dem Sie statt der System-Partition die Boot-Partition von *LibreELEC* angeben. In meinem Fall wäre das

Code-Ausschnitt 8.27: UUID einer Festplatte ermitteln

```
sudo blkid /dev/nvme0n1p5
```

mit der Ausgabe

Code-Ausschnitt 8.28: UUID einer Festplatte ermitteln - Beispiel

```
/dev/nvme0n1p5: SEC_TYPE="msdos" LABEL_FATBOOT="LIBREELEC" LABEL="LIBREELEC" UUID="2102-3124" ↔  
TYPE="vfat" PARTLABEL="LIBREELEC" PARTUUID="cbd6b178-46da-4e5d-af96-27cf7a62496f"
```

Jetzt müssen noch zwei Dinge erledigt werden. Zum Einen müssen wir die Daten des USB-Sticks auf die STORAGE-Partition unserer Festplatte kopieren. Danach werden wir *LibreELEC* noch in den Boot-Manager *CLOVER* unserer Festplatte einbinden. In den weiteren Erklärungen gehe ich davon aus, dass sich die Daten Ihres *LibreELEC* auf dem USB-Stick in der Partition */dev/sda2* befinden. Ersetzen Sie das Device bitte, falls sich Ihre Daten an einer anderen Stelle befinden sollten. Meine Zielpartition nennt sich */dev/nvme0n1p6*. Sicherlich wird Ihre Zielpartition auch ein anderes Device sein. Ersetzen Sie diese Stellen also bitte auch dementsprechend. Um die Daten vom USB-Stick auf die Festplatte zu kopieren, erstelle ich zunächst zwei Verzeichnisse, in die ich Quelle und Ziel mounten kann. Danach starte ich den Kopiervorgang.

Code-Ausschnitt 8.29: Kopieren der *LibreELEC*-Daten

```
cd ~  
mkdir quelle  
mkdir ziel  
sudo umount /mnt  
sudo mount /dev/sda2 quelle  
sudo mount /dev/nvme0n1p6 ziel  
sudo cp -R quelle/* ziel
```

Damit ist *LibreELEC* auf unserer Festplatte komplett einsatzbereit. Sie könnten also das BIOS Ihres Wohnzimmer-PCs so einstellen, dass *LibreELEC* gebootet wird. In meinem Fall müsste ich also die Partition *nvme0n1p5* booten.

Falls Sie weitere Betriebssysteme und, wie in diesem Buch beschrieben, vielleicht Mac-OS installiert habe, zeige ich Ihnen anschließend noch, wie Sie *LibreELEC* in *CLOVER* einbinden können. Die zentrale Steuerdatei für den *CLOVER*-Boot-Manager nennt sich */EFI/CLOVER/config.plist*. Sie kann entweder graphisch editiert werden (3) oder mit Hilfe eines Editors bearbeitet werden.



*Bitte fertigen Sie sich eine Sicherheitskopie der Datei */EFI/CLOVER/config.plist* an, bevor Sie diese Datei editieren. So können Sie im Falle von Fehleingaben jederzeit wieder den alten Status herstellen.*

Binden Sie zunächst die Partition ein, auf der sich der *CLOVER*-Boot-Manager befindet. In meinem Fall ist das *nvme0n1p1*.

Code-Ausschnitt 8.30: *CLOVER* Boot-Partition mounten

```
sudo mount /dev/nvme0n1p1 /mnt
```

Ergänzen Sie danach die Datei */mnt/EFI/CLOVER/config.plist* als Administrator (z. B. `sudo vi /mnt/EFI/CLOVER/config.plist`) so:

Code-Ausschnitt 8.31: Weiteren Boot-Eintrag zu *CLOVER* hinzufügen

```
...
<key>GUI</key>
...
<key>Entries</key>
...
<dict>
  <key>CustomLogo</key>
  <true/>
  <key>Disabled</key>
  <false/>
  <key>Ignore</key>
  <false/>
  <key>Title</key>
  <string>LIBREELEC</string>
  <key>Type</key>
  <string>Linux</string>
  <key>Volume</key>
  <string>CBD6B178-46DA-4E5D-AF96-27CF7A62496F</string>
</dict>
```

Der Eintrag *string* repräsentiert dabei wieder die UUID der *LibreELEC*-Boot-Partition.

Nachdem *LibreELEC* nun auf der Festplatte installiert worden ist, können später bedenkenlos alle Updates installiert werden. Sie werden aufgrund der durchgeführten Änderungen automatisch für die korrekten Partitionen angewendet. Nach einem Neustart (ohne USB-Stick) und der Auswahl von *LibreELEC* wird Ihnen das Logo aus der Abbildung 8.2 gezeigt. Anschließend werden sie mit einem Startbildschirm begrüßt. Um hier weiterzukommen, benötigen Sie auf jeden Fall eine USB-Tastatur und/oder eine USB-Maus. Falls Sie beides noch nicht angeschlossen haben, holen Sie das jetzt bitte nach.

Das System zeigt Ihnen zunächst einen Startbildschirm, den Sie mit einem Klick auf OK wegklicken können. Anschließend wird Ihnen entweder Ihre kabelgebundene Internetverbindung angezeigt oder Sie haben die Möglichkeit, eine Internetverbindung per WLAN festzulegen. Wählen Sie hierzu bitte das Netzwerk aus und geben Sie gegebenenfalls das Kennwort ein. In der nun folgenden Einstellung können Sie den *ssh*-Zugang und auch *samba* aktivieren. Bitte erlauben Sie beides, wir werden das noch im Laufe

dieses Kapitels benötigen. Nach diesen Vorbereitungen begrüßt Sie *LibreELEC* zuerst einmal mit dem Skin *Estuary*, das noch auf Englisch eingestellt ist. Wir stellen *LibreELEC* nun auf Deutsch um. Ich habe die besten Erfahrungen mit einer 50 Hz Bildwiederholrate gemacht, da diese Wiederholrate kompatibel mit der europäischen Fernsehnorm ist (50p, 50i). Die Änderung von 60 Hz auf 50 Hz können Sie unter *Einstellungen* (Settings) → *System* → *Video* vornehmen. Im selben Menü können Sie unter *Audio* die Option *Passthrough* wählen, falls Sie den HDMI-Ausgang Ihres Wohnzimmer-PCs oder Fernsehers mit einem HDMI-Receiver verbunden haben. Stellen Sie dann bitte unter *Einstellungen* (Settings) → *Regional Language* die Sprache auf „Deutsch“ um und wählen Sie im Punkt *Standardformat für Region* „Deutschland“ aus. Für eine deutsche Tastatur benötigen Sie die Einstellung *Tastaturbelegung* „German QWERTZ“. Für die *Zeitzone* wählen Sie bitte „Germany“.

Mit der linken Maustaste können Sie Einstellungen bestätigen. Ein Klick auf die rechte Maustaste bringt Sie eine Menüebene zurück. Falls Ihnen die Benutzung von Maus und Tastatur zu aufwendig ist, lernen Sie bald, wie man LibreELEC mit einer Fernbedienung steuern kann.



Falls Sie Bluetooth verwenden möchten, etwa um Ihr Mediacenter mit einem kabellosen Kopfhörer zu verbinden, aktivieren Sie bitte den Bluetooth-Dienst in der Rubrik *Einstellungen* → *LibreELEC* → *Dienste*.

Um sich per ssh mit LibreELEC zu verbinden, benutzen Sie bitte den Benutzernamen root und das Kennwort libreelec. Beides kann nicht verändert werden. Sollte Ihr LibreELEC-Rechner frei zugänglich im Internet sein, empfehle ich Ihnen, diesen Zugang nur zu aktivieren, wenn Sie Änderungen an den Einstellungen vornehmen möchten, etwa um eine Fernbedienung anzulernen.



Aktivieren Sie bitte ebenfalls den Dienst *lirc*, falls Sie einen *lirc*-Empfänger benutzen. Falls Sie *flirc* benutzen, lernen Sie gegebenenfalls fehlende Tastatur-Tasten an, um KODI komplett mit der Fernbedienung bedienen zu können. Im späteren Verlauf dieses Kapitels werden wir noch den Fernsehempfang per Satellit für *LibreELEC* nachrüsten. Hierzu verwende ich genau wie im vorangegangenen Kapitel die DVBSKY S960 USB-Karte. Diese wird unter *LibreELEC* sofort erkannt und die benötigte Firmware steht ebenfalls zur Verfügung. Sollten Sie den Sundtek-Stick benutzen, können Sie die Treiber hierfür aus einem der *LibreELEC* Repositories nachinstallieren.

8.3.1 Einrichten einer *lirc*-Fernbedienung für *LibreELEC* Version 8 und kleiner

Um sicherzustellen, dass Ihr Infrarot-Empfänger richtig angeschlossen ist, können Sie den folgenden Befehl in ein Terminal eingeben und danach Tasten auf Ihrer Fernbedienung drücken. Sie sollten dann im Terminal Ausgaben bei jedem Tastendruck sehen.

Code-Ausschnitt 8.32: Fernbedienung unter KODI testen

```
killall lircd && mode2 -d /dev/lirc0
```

Achten Sie auch hier auf die Verwendung des richtigen Devices (in meinem Fall */dev/lirc0*). Wechseln Sie nun bitte in das Verzeichnis */storage/.config*. Bitte beachten Sie den . vor *config*. Hierbei handelt es sich um ein verstecktes Verzeichnis, das unter anderem die Datei enthalten kann, die Codes für die jetzt anzulernende Fernbedienung enthält.

Danach starten wir das Anlernen der Fernbedienung. Dieses Prozedere kennen Sie bereits aus dem VDR-Kapitel.

Code-Ausschnitt 8.33: Fernbedienung unter KODI anlernen

```
cd /storage/.config
irrecord -d /dev/lirc0
```



Bitte lernen Sie mindestens die folgenden Tasten an, um KODI sinnvoll bedienen zu können:

- KEY_UP
- KEY_DOWN
- KEY_LEFT
- KEY_RIGHT
- KEY_PLAY (mit dieser Taste können Sie ein Video starten und auch wieder stoppen)
- KEY_PAUSE (diese Taste ist optional, da KEY_PLAY dieselbe Aufgabe erfüllen kann)
- KEY_STOP
- KEY_VOLUME_UP
- KEY_VOLUME_DOWN
- KEY_INFO
- KEY_OK
- KEY_EPG (mit dieser Taste können Sie später die elektronische Programmzeitschrift anzeigen)
- KEY_EXIT (das ist die „zurück“-Taste, mit der Sie eine Menüebene zurück kommen)
- KEY_MENU (mit dieser Taste können Sie verschiedene Kontext-Menüs öffnen)
- KEY_x (setzen Sie bitte für x die Zahlen 0...9 ein. Die Zahlen-Tasten ermöglichen es Ihnen später, ein TV-Programm direkt auszuwählen)

Nach dem Anlernen der Fernbedienung existiert im Verzeichnis `/storage/.config` eine Datei namens `name.lircd.conf`. `name` ist dabei der Name Ihrer Fernbedienung, den Sie eingegeben haben, bevor Sie die Tasten Ihrer Fernbedienung angelernt haben. Damit der Infrarot-Dienst die Daten Ihrer Fernbedienung findet, muss diese Datei nach `lircd.conf` umbenannt werden.

Code-Ausschnitt 8.34: Umbenennen der *lirc*-Datei

```
mv name.lircd.conf lircd.conf
```

Bitte denken Sie daran, `name` entsprechend zu ersetzen. Nach einem Neustart sollte Ihre Fernbedienung funktionieren und Sie können Tastatur und Maus entfernen.

8.3.2 Einrichten einer *lirc*-Fernbedienung für LibreELEC Version 9 und neuer

Für die LibreELEC-Versionen 9 und neuer ist der Weg, eine *lirc*-Fernbedienung einzurichten, etwas anders. Er wird in diesem Abschnitt beschrieben. Sollten Sie von einem LibreELEC 8 auf die Version 9 aktualisiert haben, benennen Sie im ersten Schritt die Konfigurations-Datei für *lirc* um und beenden Sie dann KODI und den *lirc*-Daemon. Falls das nicht der Fall war, sondern Sie mit der Installation eines LibreELEC 9 gestartet haben, überspringen Sie bitte diesen Schritt.

Code-Ausschnitt 8.35: *lirc*-Datei umbenennen und Daemon stoppen

```
mv /storage/.config/lircd.conf /storage/.config/lircd.conf.alt
systemctl stop kodi
systemctl stop eventlircd
```

Erzeugen Sie danach bitte eine leere Datei namens *my_remote* im Verzeichnis */storage/.config/rc_keymaps/*:

Code-Ausschnitt 8.36: Leere Fernbedienungsdatei erzeugen

```
vi /storage/.config/rc_keymaps/my_remote
systemctl stop kodi
systemctl stop eventlircd
```

Öffnen Sie bitte ein zweites Terminal, ohne das andere zu schließen. Wir müssen nun herausfinden, welches Protokoll die Fernbedienung verwendet, die wir benutzen wollen. Rufen Sie dazu

Code-Ausschnitt 8.37: Identifizieren des Fernbedienungs-Protokolls

```
ir-keytable
```

auf. Der Befehl listet alle verfügbaren Protokolle in der dritten Zeile auf.

Code-Ausschnitt 8.38: Identifizieren des Fernbedienungs-Protokolls (Ausgabebeispiel)

```
Found /sys/class/rc/rc0/ (/dev/input/event11) with:
  Driver dvb_usb_rtl28xxu, table rc-empty
  Supported protocols: NEC RC-5 RC-6 JVC SONY SANYO LIRC
  Enabled protocols: NEC RC-5 RC-6 JVC SONY SANYO LIRC
```

Um das passende Protokoll herauszufinden, wird der folgende Befehl abgesetzt:

Code-Ausschnitt 8.39: Testen des Fernbedienungs-Protokolls

```
ir-keytable -p rc-5 -t
```

Der Eintrag hinter *-p* entscheidet über das Protokoll. Drücken Sie einige Tasten Ihrer Fernbedienung und achten Sie auf mögliche Ausgaben. Stimmt das Protokoll nicht, werden Sie keine Ausgabe erhalten. Beenden Sie in diesem Fall das Programm *ir-keytable* mit der Tastenkombination *CTRL-C* und wählen Sie ein anderes Protokoll. Haben Sie das passende Protokoll erwischt, erscheinen z. B. folgende Ausgaben:

Code-Ausschnitt 8.40: Ausgabe bei richtigem Protokoll

```
Protocols changed to rc-5
Testing events. Please, press CTRL-C to abort.
1503592437.660155: event type EV_MSC(0x04): scancode = 0x101a
1503592437.660155: event type EV_SYN(0x00).
1503592437.774129: event type EV_MSC(0x04): scancode = 0x101a
1503592437.774129: event type EV_SYN(0x00).
1503592437.921009: event type EV_MSC(0x04): scancode = 0x101a
```

Füllen Sie (im anderen Terminal) die Datei */storage/.config/rc_keymaps/my_remote* mit der ersten Zeile:

Code-Ausschnitt 8.41: Eigene Fernbedienungsdatei erstellen

```
# table my_custom_remote, type: rc-5
```

Bitte denken Sie daran, Ihr gefundenes Protokoll einzusetzen, das nicht *rc-5* sein muss. Im nächsten Schritt müssen wir herausfinden, welche Scancodes mit welchen Fernbedienungstasten verbunden sind. Lassen Sie den Editor, den Sie gerade mit einer

Protokollzeile gefüllt haben, geöffnet und geben Sie in Ihrem zweiten Terminal den Befehl

Code-Ausschnitt 8.42: Ermitteln der Scancodes

```
ir-keytable -t
```

ein. Drücken Sie nun beliebige Tasten auf Ihrer Fernbedienung und tragen Sie den Scancode und eine passende *KEY*-Bezeichnung in den immer noch geöffneten Editor, beispielsweise so:

Code-Ausschnitt 8.43: Scancodes-Beispiel

```
# table my_custom_remote, type: rc-5
0x101a KEY_UP
0x101b KEY_DOWN
0x1013 KEY_LEFT
0x1014 KEY_RIGHT
0x1015 KEY_OK
```

Scancode und *KEY*-Name müssen dabei durch ein einzelnes Leerzeichen voneinander getrennt sein. Noch einmal zur Erinnerung: Eine Liste gültiger *KEY*-Bezeichnungen erhalten Sie durch Eingabe des Befehls

Code-Ausschnitt 8.44: Auflistung aller *lirc*-Keys

```
irrecord -l | grep KEY
```

Diese Keys werden in der Datei `/usr/share/kodi/system/Lircmap.xml` verwendet. Sollten Sie andere *KEY*-Namen verwenden, müssen Sie die entsprechende Rubrik `<remote device="devinput">` durch eine eigene *lircmap.xml*-Datei abändern.

Nachdem Sie alle Scancodes herausgefunden und eingetragen haben, können Sie die Datei `/storage/.config/rc_keymaps/my_remote` speichern. Abschließend erstellen Sie bitte eine Datei namens `/storage/.config/rc_maps.cfg` mit dem folgenden Inhalt:

Code-Ausschnitt 8.45: Einbinden der eigenen Konfigurationsdatei

```
* * my_remote
```

Nach einem Neustart sollte Ihre Fernbedienung dann wie gewohnt auch unter der *LibreELEC*-Version 9 oder höher funktionieren.

8.3.3 Der Wetterdienst

Lassen Sie uns noch schnell ein erstes Plugin installieren, bevor wir uns gleich um Fernsehen in *LibreELEC* kümmern. Keine Sorge, in *LibreELEC* ist dazu kein aufwendiges Kompilieren erforderlich. Add-ons können einfach per Fernbedienung installiert werden.

Navigieren Sie im *LibreELEC*-Hauptmenü zur Rubrik „Wetter“ und bestätigen Sie Ihre Auswahl mit *OK*. Da noch kein Wetterdienst installiert ist, können Sie sich einen aussuchen (→ *mehr*). Bei mir funktioniert der *Yahoo!*-Dienst einwandfrei. Nach der Installation des erforderlichen Plugins können Sie mit der Fernbedienung den Namen der nächsten Großstadt eingeben, beispielsweise „Düsseldorf“. Das Plugin sucht daraufhin nach diesem Namen in einer Datenbank und stellt Ihnen anschließend die Wettervorhersage für diese oder auch weitere Städte zur Verfügung.

8.3.4 Tvheadend zum Fernsehen

Was der VDR unter LINUX leistet, das kann *Tvheadend* unter KODI. *Tvheadend* ist ein TV-Klient, der beispielsweise mit DVB-S2 USB-Karten umgehen kann und gleichzeitig ein komfortables Webinterface zur Konfiguration oder zum Streaming zur Verfügung stellt.

Während die Installation von *Tvheadend* schnell erledigt ist, dauert das Einrichten deutlich länger. Zur Installation des HTSP-Klienten wählen Sie bitte *Add-on* → *Aus repository installieren* → *Alle Repositories* → *PVR clients* → *Tvheadend HTSP Client*. HTSP (*Home TV Streaming Protokoll*) ist ein TCP-basiertes Protokoll, welches darauf ausgelegt ist, Live-Streams (wie z. B. TV) zu übertragen. Installieren Sie bitte analog den *tvheadend*-Server, den Sie unter *Add-on* → *Aus repository installieren* → *Alle Repositories* → *Dienste* finden. Damit ist die Installation von *Tvheadend* auch schon erledigt und wir können uns um die Konfiguration kümmern. Diese erledigen wir bequem von einem Webbrowser Ihrer Wahl aus, der sich im selben Netzwerk befindet, in dem sich auch Ihr Wohnzimmer-PC befindet. Der HTSP-Server steht am Port 9981 zur Verfügung. Starten Sie also bitte Ihren Browser mit der URL `http://192.168.178.66:9981` und ersetzen Sie meine oben angegebene IP-Adresse mit der Ihres Wohnzimmer-PCs. *Tvheadend* begrüßt Sie daraufhin mit einem Konfigurationsbildschirm, der in Abbildung 8.4 gezeigt wird.

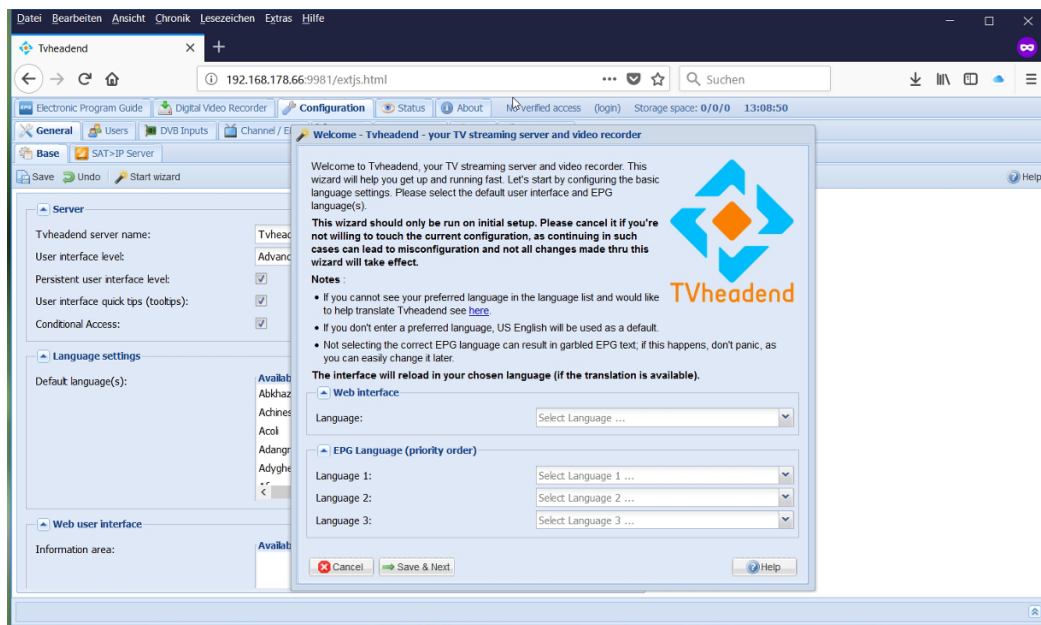


Abbildung 8.4: *tvheadend* wird komfortabel mit Hilfe eines Webinterfaces konfiguriert.

Ich gehe an dieser Stelle davon aus, dass Sie einen USB-Stick an Ihren Wohnzimmer-PC angeschlossen haben, der Ihnen den Empfang von Satellitenfernsehen ermöglicht. Für kabelgebundenen (DVB-C) oder terrestrischen Empfang gilt das hier Gesagte analog.

Die Sprache ist zunächst auf Englisch voreingestellt. Ändern Sie die Sprache, indem Sie unter *Language* „German“ auswählen. Dasselbe gilt für die Sprache des elektronischen Programmführers (*EPG-Language*). Klicken Sie auf *Save and exit*.

In der nächsten Rubrik werden Sie dazu aufgefordert, Netzwerkbeschränkungen sowie einen *Administrator-Namen* und ein *Administrator-Kennwort* festzulegen. Dasselbe gilt für einen Benutzer. Lassen Sie das Feld für die Netzwerkeinstellungen bitte frei. Damit gibt es keinerlei Restriktionen und das Risiko ist geringer, hier etwas Falsches einzustellen. Sie können diese Einstellungen jederzeit ändern, falls Ihr Wohnzimmer-PC im Internet hängt und Sie das Risiko unberechtigten Zugriffs verringern möchten. Für alle Benutzerkonten und die Kennwörter tragen Sie bitte einen * ein. Damit ist auch die Web-Oberfläche zur Konfiguration frei zugänglich. Speichern Sie diese Änderungen ebenfalls.

Im folgenden Schritt wählen Sie bitte unter *Netzwerktyp* das DVB-S Netzwerk aus und stellen Sie als vordefinierte Muxe 19.2E: Astra ein, falls Sie den Satelliten Astra auf 19.2° Ost nutzen möchten. Nach einem erneuten *Speichern und weiter* beginnt *tvheadend* damit, Sender zu scannen. Erfolgreiches Scannen können Sie daran erkennen, dass die Anzahl der gefundenen Muxe und die Anzahl der gefundenen Dienste (Services) steigt. Bitte bringen Sie Geduld mit: Ein vollständiger Scan des Astra-Satelliten kann gerne schon einmal 30 Minuten dauern.

Nach erfolgreichem Scan setzen Sie bitte alle drei Häkchen im nächsten Konfigurations-Bildschirm. Das ordnet alle Dienste (Services) zu und setzt sowohl Anbieter- als auch Netzwerk-Tags. Wenn Sie nun in *KODI* die Rubrik *TV* auswählen, werden Ihnen alle gefundenen Sender angeboten und Sie können das Fernsehprogramm mit einem Fernbedienungs-OK starten.



Bitte beachten Sie, dass für Sender mit DD5.1-Sound Audio-Passthrough in KODI aktiviert sein muss.

Während wir nun schon komfortabel TV schauen, Timer setzen und Sendungen aufzeichnen können, sieht das TV-Sendermenü noch ein wenig triste aus. Dies ändern wir durch Installation von Senderlogos. Die Webseite <https://www.picons.eu/> [90] bietet aktuelle Senderlogos zum kostenlosen Download an.

Wir benötigen zwei verschiedene Dateien aus der Download-Rubrik, nämlich *srp-full400x240-370x210.light.on.transparent...hardlink.tar.xz* und das entsprechende *snp-Pendant*.

Die heruntergeladenen Dateien müssen in den */storage/picons*-Ordner kopiert werden. Die *snp*-Datei muss in das Unterverzeichnis *tvh* kopiert werden, die *srp*-Datei in das Unterverzeichnis *vdr*. Sie können das z. B. bequem von einem Laptop aus unter der Verwendung von *samba* erledigen (Abbildung 8.5). Entpacken Sie die beiden Logo-Dateien in einem Terminal.

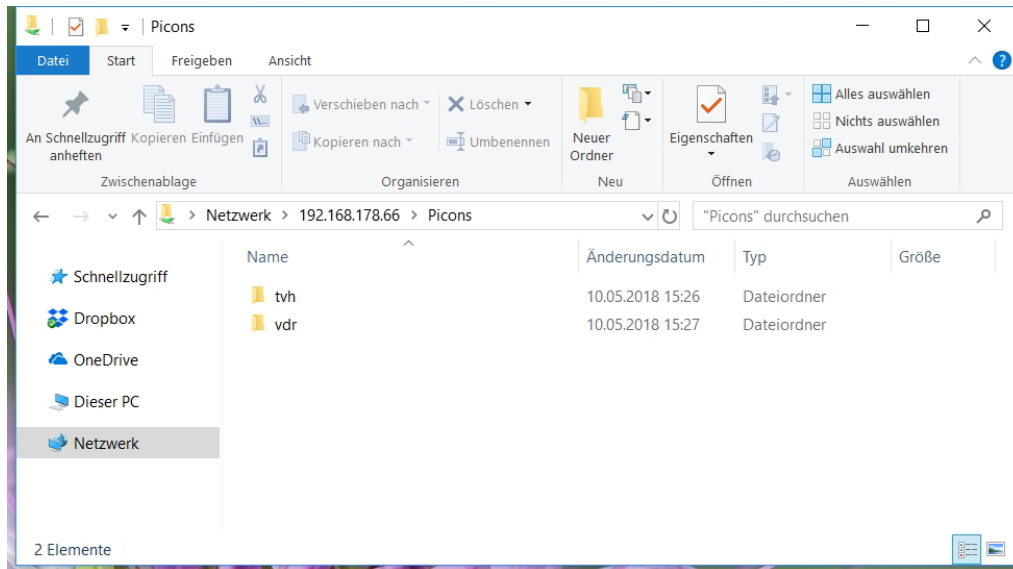


Abbildung 8.5: Mit Hilfe von *samba* können die Logo-Dateien schnell an ihren Bestimmungsort kopiert werden.

Code-Ausschnitt 8.46: Senderlogos unter *KODI* installieren

```
cd /storage/picons/tvh
xz -d snp<TAB>
tar xvf snp<TAB>
mv snp<TAB>/* .
rmdir snp<TAB>
rm snp<TAB>
cd /storage/picons/vdr
xz -d srp<TAB>
tar xvf srp<TAB>
mv srp<TAB>/* .
rmdir srp<TAB>
rm srp<TAB>
```

Das `xz -d`-Kommando entpackt die jeweilige Logo-Datei. Der `tar`-Befehl entpackt die einzelnen Logos, allerdings in einem weiteren Unterverzeichnis. Wir bewegen (`mv`) die Dateien aus dem Unterverzeichnis wieder in das übergeordnete Verzeichnis und löschen anschließend das leere Unterverzeichnis und die gepackte Logo-Datei. Die `TAB`-Taste ergänzt den jeweiligen Dateinamen automatisch und erspart so viel Tipparbeit. Die Verzeichnisse, in welche die jeweiligen Logos kopiert werden müssen, können Sie übrigens auch vom *tvheadend* Web-Frontend einstellen (Abb. 8.6).

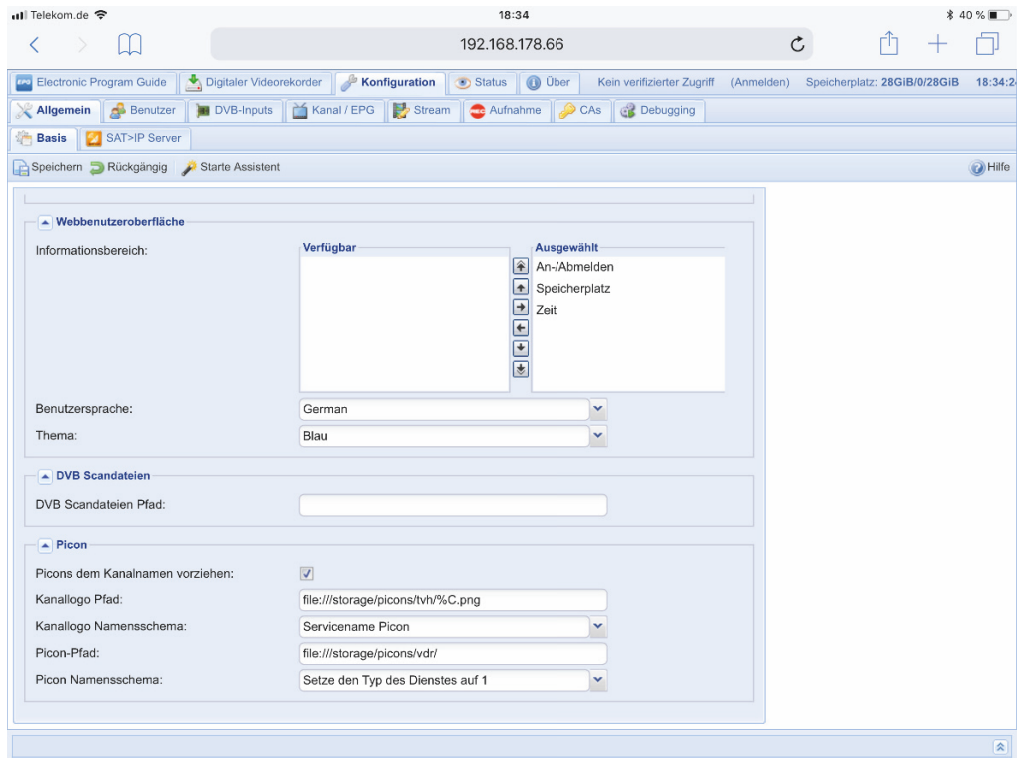


Abbildung 8.6: Die Verzeichnisse für die Senderlogos können im *tvheadend* Web-Frontend eingestellt werden.

Im letzten Schritt teilen wir *KODI* noch mit, dass wir in den o.g. Ordnern neue Kanallogos haben. Dazu wechseln wir vom *KODI*-Hauptmenü zu den Einstellungen und wählen *PVR & TV* → *Menü/OSD*. Als *Ordner mit Kanalsymbolen* geben wir das Verzeichnis */storage/picons/tvh* an und starten anschließend *Nach fehlenden Kanalsymbolen suchen*. Danach sind die Senderlogos in den Kanälen und in der elektronischen Programmzeitschrift (EPG) enthalten.

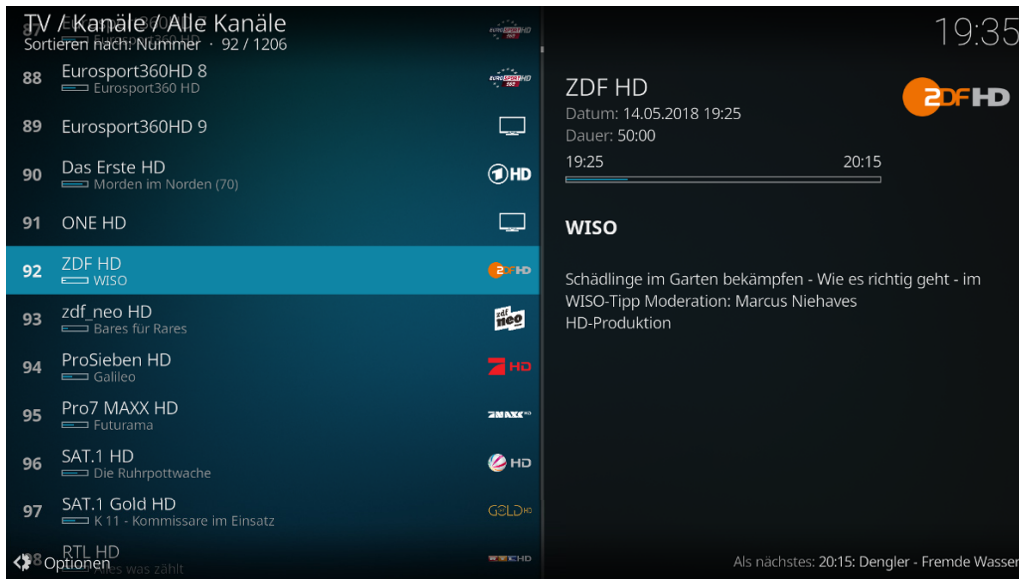


Abbildung 8.7: Mit Senderlogos sieht *tvheadend* gleich viel schöner aus.

8.3.5 DVD und Blu-ray



LibreELEC kann von Hause aus DVDs abspielen. Die *libdvd-css* ist also bereits eingebaut. Analog zur *libdvdcss* gibt es auch eine *libbluray*, für die ähnliche rechtliche Bedingungen gelten. Laut <https://wiki.libreelec.tv/blu-ray> [87] ist die Verwendung in den meisten Ländern legal.

Der Befehl

Code-Ausschnitt 8.47: KODI Blu-ray-Datenbank installieren

```
mkdir -p /storage/.config/aacs && curl https://web.archive.org/web/20170902140455if_/http://www.labdv.com/aacs/KEYDB.cfg -o /storage/.config/aacs/KEYDB.cfg
```

lädt eine Datenbank mit über 24.000 Blu-ray Discs auf Ihren Wohnzimmer-PC, die *KODI* danach problemlos abspielen kann.

Der oben genannte Befehl sollte von Zeit zu Zeit wiederholt werden, da ständig neue Blu-rays hinzugefügt werden.



Damit mein USB Blu-ray-Laufwerk richtig erkannt wurde, musste ich *LibreELEC* mit angestecktem Laufwerk neu starten. Danach konnte ich ohne Probleme und ruckelfrei eine Blu-ray Disc abspielen.

8.3.6 Amazon Prime

KODI bietet eine Fülle an Add-ons, wie verschiedene Mediatheken oder Internet-Radios. Ein sehr beliebtes ist das Amazon Prime VOD (*Video On Demand*) Add-on. Amazon selbst unterstützt das Abspielen von Videos mit Hilfe von KODI nicht. Aus diesem Grund geschieht es immer wieder, dass dieses Plugin zu bestimmten Zeiten funktioniert, nach einer Änderung von Amazon aber wieder nicht. Das Plugin selbst erfordert die *libwidevine*, eine Bibliothek, die DRM-geschützten (*Digital Rights Management*) Inhalt abspielen kann. Da Amazon keine Hardware-Beschleunigung beim Abspielen der angebotenen Inhalte unterstützt, können Filme auf schwachen PCs häufig nur in kleiner Auflösung abgespielt werden, damit sie ruckelfrei sind. Die Autoren von LibreELEC erlauben es nicht, diese Bibliothek auf Ihrem



Abbildung 8.8: Amazon VOD.

Server zu speichern oder zur Verfügung zu stellen. Das Amazon Prime Add-on, welches wir gleich installieren, setzt das *VideoPlayer Inputstream* Add-on voraus. Installieren Sie dieses bitte zuerst, indem Sie wie folgt navigieren: *Add-ons* → *Aus Repository installieren* → *Alle Repositories* → *VideoPlayer Inputstream Add-ons* → *Inputstream Adaptive*. Installieren Sie dieses Add-on bitte, nachdem Sie es ausgewählt haben. Das *Amazon VOD* Add-on liegt in einem Repository namens „Sandmann 79s Repository“. Dieses ist standardmäßig noch nicht aktiviert. Aktivieren Sie dieses Repository, indem Sie zu *Add-ons* → *Aus Repository installieren* → *Alle Repositories* → *Add-on Verzeichnis* → *Sandmann 79s Repository* navigieren und das Sandmann-Repository installieren. Anschließend steht Ihnen das *Amazon VOD* Add-on unter *Video Add-ons* → *Amazon VOD* zur Verfügung.

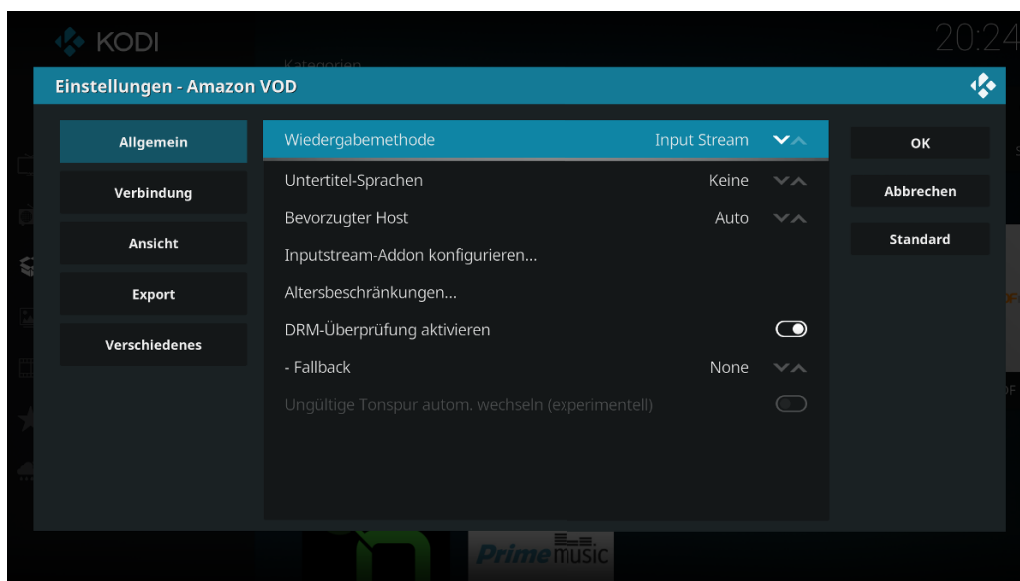


Abbildung 8.9: Als Wiedergabemethode im *Amazon VOD* Add-on muss *Input Stream* gewählt werden.

Konfigurieren Sie das *Amazon VOD* Add-on, indem Sie es selektieren und die Menü-Taste Ihrer Fernbedienung drücken. Dort können Sie *Einstellungen* auswählen. Wählen Sie als *Wiedergabemethode* *Input Stream* aus. Unter der Rubrik *Einstellungen Verbindung* können Sie unter *Anmelden* Ihren Amazon-Benutzernamen und Ihr Kennwort eingeben.

Jetzt ist der größte Teil der Arbeit geschafft. Es fehlt nur noch das Herunterladen der *libwdevine*. Das erledigt das *Amazon VOD Add-on* automatisch für uns bei der Auswahl des ersten Filmes. Starten Sie nun das Add-on durch Drücken der *OK*-Taste, nachdem Sie es selektiert haben. Navigieren Sie danach zu einem beliebigen Film und starten Sie ihn ebenfalls durch Drücken der *OK*-Taste Ihrer Fernbedienung. Das Add-on wird den Film nicht starten, sondern stattdessen die Meldung „Widewine CDM is required“ ausgeben. Bitte bestätigen Sie, dass Sie die Bibliothek installieren möchten und akzeptieren Sie die Lizenzbedingungen. Das Add-on lädt daraufhin ein Google Chrome Recovery Image aus dem Internet herunter, welches die *libwdevine* enthält. Dieses Recovery Image ist 1,9 GB groß, es kann also eine Weile dauern, bis es vollständig geladen wurde. Danach müssen Sie noch einmal das Entpacken des Images mit „OK“ bestätigen. Nach diesen Schritten sollte das Abspielen von Filmen funktionieren.

8.4 Mobile Helfer

In diesem Abschnitt möchte ich Ihnen noch einige mobile Helfer vorstellen, die den Umgang mit *LibreELEC* unter Verwendung eines Mobiltelefons oder eines Tablet-Computers deutlich vereinfachen können. Diese Apps erlauben es, *KODI* mit dem Telefon zu bedienen, Fernsehsendungen zum Tablet zu streamen oder TV-Aufnahmen von unterwegs zu programmieren.

8.4.1 TvheadClient

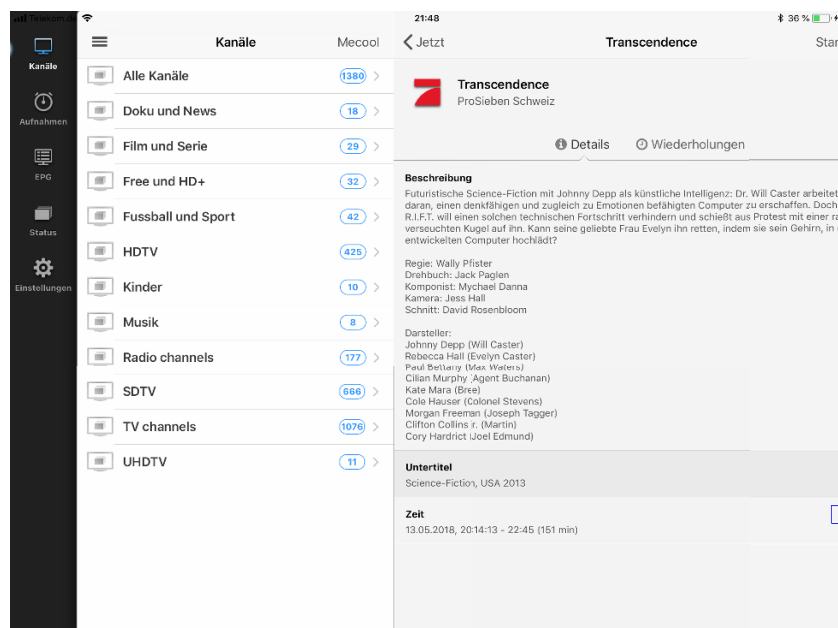


Abbildung 8.10: Die Verzeichnisse für die Senderlogos können im *tvheadend* Web-Frontend eingestellt werden.

Zu meinen Lieblings-Apps im Zusammenhang mit *KODI* und *tvheadend* zählt die iPad-APP *TvhClient*. *TvhClient* zeigt das EPG einzelner Sendungen oder das EPG über einen Zeitraum, es streamt das Fernsehprogramm oder Aufnahmen auf das iPad. Der Benutzer kann Timer erstellen, editieren und löschen. Für Android-Geräte empfehle ich Ihnen die kostenlose App *Kore*. Diese App ist eine perfekte Fernbedienung für *KODI*, welche Sie von Ihrem Mobilfunkgerät aus verwenden können. Dabei gibt es nicht nur die typischen Fernbedienungstasten, sondern auch noch weitere Informationen über aufgezeichnete Filme, wie eine Inhaltsangabe oder Schauspieler. *LibreELEC* stellt auch einen AirPlay-Server zur Verfügung, so dass Sie Musik von Ihrem Apple-Gerät zu *KODI* streamen können, um diese beispielsweise über Ihre Stereo-Anlage abspielen zu können. Der nächste Abschnitt zeigt Ihnen, wie Sie diesen Dienst auch ohne *LibreELEC* unter Ihrer Ubuntu-Distribution installieren können.

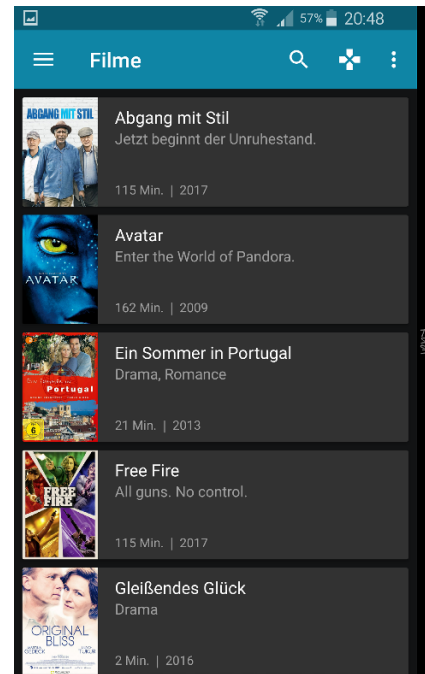


Abbildung 8.11: Die *Kore* App unter Android.

8.5 AirPlay und Co.

Ein großer Computerkonzern aus Cupertino wirbt auf der Webseite damit, dass seine Geräte *AirPlay* beherrschen, das Abspielen von Medieninhalten über die WLAN-Luft-Schnittstelle. Haben Sie das schon bei *KODI* versucht? Unter den Einstellungen können Sie *AirPlay* erlauben und so multimediale Inhalte von Ihrem Telefon oder Tablet auf dem Fernseher anzeigen - vorausgesetzt, *KODI* läuft¹. Das Mediacenter erkennt automatisch, dass ein *AirPlay*-Inhalt angefordert wird und stellt diesen dar - unabhängig davon, ob es sich um Urlaubsfotos und -videos oder um einen Film handelt. Eine ähnliche *AirPlay*-Schnittstelle gibt es für die Audio-Ausgabe. Allerdings macht es keinen Sinn, für ein paar Songs den Fernseher einzuschalten. Und wäre es nicht schön, wenn wir Webradio über unsere Stereoanlage hören könnten, vielleicht sogar noch in digitaler Qualität? Kein Problem! In diesem Abschnitt werden wir einen *AirPlay*-Server installieren und ihn mit unserer Stereoanlage verbinden. Eventuell besitzt Ihr Wohnzimmer-PC keinen digitalen Audioausgang. Daher zeige ich Ihnen weiterhin, wie Sie mit Hilfe von externer Zusatz-Hardware, etwa einer kleinen USB-Soundkarte für 12 € (Delock 61961), einen vollwertigen, digitalen Toslink-Anschluss realisieren können. Alternativ können Sie natürlich den Ton über HDMI abgreifen, wenn Ihr Audio-Receiver damit umgehen kann. Beginnen wir mit der Installation des *AirPlay*-Servers. Sicher kennen Sie bereits den ersten Schritt: die Installation weiterer Programmpakete.

¹Videos können nur bis iOS Version 8 übertragen werden. Audio-Übertragungen sind für alle iOS-Versionen möglich.

Code-Ausschnitt 8.48: Abhängigkeiten für *shairport-sync* installieren

```
sudo apt-get install autoconf automake avahi-daemon build-essential git libasound2-dev libavahi-<br>-client-dev libconfig-dev libdaemon-dev libpopt-dev libssl-dev libtool xsltoman
```

Die aktuelle Version des *AirPlay*-Servers nennt sich *shairport-sync* und hat ein eigenes GitHub-Repository. Bitte wechseln Sie in ein Verzeichnis Ihrer Wahl (z. B. */home*) und klonen Sie den *AirPlay*-Server so:

Code-Ausschnitt 8.49: Airplay-Server *shairport-sync* auschecken

```
git clone https://github.com/mikebrady/shairport-sync.git
```

Wechseln Sie anschließend in das *shairport-sync*-Verzeichnis

Code-Ausschnitt 8.50: In das *shairport-sync*-Verzeichnis wechseln

```
cd shairport-sync
```

und starten Sie den Übersetzungsvorgang für den Server.

Code-Ausschnitt 8.51: *shairport-sync* konfigurieren

```
autoreconf -i -f
./configure --with-alsa --with-avahi --with-ssl=openssl --with-systemd --with-metadata
```

In den beiden oberen Befehlen wird zunächst *autoreconf* ausgeführt. Dieser Befehl erstellt ein Konfigurationsskript, welches danach aufgerufen wird und das Makefile erstellt, das dann ausgeführt wird, um unseren *AirPlay*-Server zu übersetzen. Der Vorteil dieser Methode ist, dass die Konfiguration für verschiedene Systeme durchgeführt werden kann. *shairport-sync* wird mit Unterstützung für das ALSA-Soundsystem übersetzt. Weiterhin unterstützt unser Server Avahi und OpenSSL-Verschlüsselung.

Avahi ist eine freie Implementierung einer Technik zur Vernetzung von Geräten in einem lokalen Netzwerk, ohne dass diese manuell konfiguriert werden müssen. Der Parameter *--with-systemd* erlaubt den automatischen Start des *ShairPort*-Servers nach dem Bootvorgang. Der Parameter *--with-metadata* erlaubt es dem *ShairPort*-Server, Metadaten an kompatible Applikationen zu „pipen“. Das Übersetzen des Quelltextes und das Installieren der Applikation ist schnell erledigt:

Code-Ausschnitt 8.52: *shairport-sync* übersetzen und installieren

```
make
sudo make install
```

Damit *Shairport-Sync* nach dem Booten automatisch startet, ist ein letzter Befehl notwendig, der das Startup-Skript nach jedem Booten des Wohnzimmer-PCs automatisch ausführt.

Code-Ausschnitt 8.53: *shairport-sync* automatisch starten

```
sudo systemctl enable shairport-sync
```

Nach einem Neustart steht der *AirPlay*-Server zur Verfügung. Möchten Sie Ihren Wohnzimmer-PC nicht neu starten, können Sie auch mit dem Befehl

Code-Ausschnitt 8.54: *shairport-sync* manuell starten

```
sudo service shairport-sync start
```

dafür sorgen, dass der Server gestartet wird. Wenn Sie nun Musik mit Ihrem Apple-Device abspielen, taucht der *AirPlay*-Server unter dem Namen *Rudi* - also meinem Rechnernamen - auf.

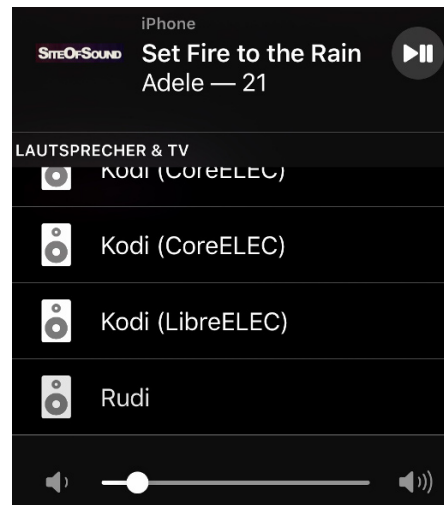


Abbildung 8.12: Auf einem Apple-Gerät taucht *shairport-sync* unter meinem Rechnernamen auf.

Während die digitale HDMI-Audioausgabe eine hervorragende Audio-Qualität liefert, kann man das für viele 3.5 mm analoge Klinkenausgänge von PCs nicht behaupten. Sollte Ihr Wohnzimmer-PC keinen digitalen Audio-Ausgang besitzen, können Sie diesen mit einer externen USB-Soundkarte, etwa der Delock 61961 sehr einfach nachrüsten. Der nur 12 € teure Stick stellt einen analogen und einen optischen Digital-Ausgang zur Verfügung. Das Einbinden des Sticks ist denkbar einfach. Fügen Sie in die Datei */etc/modules* folgende zwei Zeilen an beliebiger Stelle ein:



Abbildung 8.13: Delock 61961 ist eine preiswerte USB-Soundkarte (Quelle: Tragant).

Code-Ausschnitt 8.55: Delock USB-Soundstick einbinden

```
snd-usb-audio
snd-bcm2835
```

Nach einem Neustart des PCs lädt der Kernel zwei Treiber: den prinzipiellen USB-Audiotreiber und den *bcm2825* Audiotreiber. Die Datei */etc/modprobe.d/alsa-base.conf*, die ebenfalls angepasst werden muss, räumt dem USB-Treiber aber Vorrang vor anderen Treibern ein.

Code-Ausschnitt 8.56: Vorrang der Sound Devices regeln

```
options snd-usb-audio index=0
```

Nun sollte einem ungetrübten Musikgenuss nichts mehr im Wege stehen. Auf den *i*Geräten können Sie *PiPlay* als *AirPlay*-Plattform auswählen und der Ton wird über Ihren PC an die Stereoanlage oder ein paar Lautsprecherboxen weitergeleitet. Beide Geräte (also Tablet/Telefon und Wohnzimmer-PC) müssen sich dabei im selben Netzwerk befinden.

8.6 MP3

Eines darf in einem Multimedia-Kapitel sicher nicht fehlen: das Abspielen von mp3-Dateien. Unter Ubuntu ist dieses Thema schnell erledigt:

Code-Ausschnitt 8.57: MP3-Dateien unter Ubuntu abspielen

```
sudo apt-get install mpg123
```

Das Programm *mpg123* wird einfach mit dem Dateinamen des abzuspielenden Liedes als Parameter aufgerufen - fertig!

8.7 Wichtige Hinweise

Zum Abschluss dieses Kapitels möchte ich Ihnen noch einige wichtige Hinweise geben.



Abbildung 8.14: Der Delock 62988 ist ein USB-C auf HDMI-Adapter mit PD-Funktion (Quelle: Reichelt).

Falls Ihnen die Videoausgabe von UHD-Material wichtig ist, achten Sie darauf, dass Ihr Wohnzimmer-PC diese Auflösung unterstützt. Bei meinem Lattepanda Alpha PC gibt es einen HDMI-Ausgang, der UHD nur bis zu einer Wiederholrate von 30 Hz unterstützt. Für 50 Hz oder 60 Hz muss ich auf einen USB-C-nach-HDMI-Adapter zurückgreifen, der dann die volle Auflösung unterstützt. Da mein Lattepanda Alpha über diesen Adapter auch die Spannungsversorgung erhält, muss der Adapter selbst einen USB-C-Spannungsversorgungsanschluss mitbringen. Diese Adapter führen dann meist den Zusatz PD (Power Delivery) im Namen und kosten ca. 20 €. Die Adapter, die ich für meinen Wohnzimmer-PC gekauft habe, hatten aber beide ein Problem, wenn die Audioausgabe zwischen den Formaten 5.1, 2.0 oder normaler Stereoton

gewechselt hat. Oft war der Ton nach dem Umschalten weg und kam erst durch Abziehen und Wiedereinstecken des HDMI-Kabels zurück. Eine Lösung des Problems ergab sich durch ein Downgrade der Adapter-Firmware auf die ältere Version 0x46 (z. B. CAC-1070-Adapter). Die Firma Club3D war so nett und hat mir auf eine E-Mail-Anfrage hin sowohl das Flash-Tool als auch die entsprechende Firmware zugesendet. Nach dem Aufspielen dieser Firmware funktionierte dann auch das Wechseln zwischen verschiedenen Audioformaten einwandfrei.

LibreELEC-Updates können Sie sehr einfach installieren. Sie stehen als *tar*-Archive zur Verfügung, die lediglich in das Verzeichnis *Updates* bzw. *.update* kopiert werden müssen. Nach dem nächsten Start stellt *LibreELEC* dann fest, dass ein Update vorhanden ist und installiert dieses automatisch. Gefällt Ihnen ein Update nicht, können Sie jederzeit die letzte *tar*-Version wieder in das Verzeichnis kopieren, um so das Update rückgängig zu machen. Alle gespeicherten Daten bleiben beim Update (und auch beim Downgrade) erhalten.

Als Update-Quelle kann ich Ihnen die Milhouse Testbuilds empfehlen. Eine aktuelle Beschreibung aller Änderungen finden Sie unter <https://forum.kodi.tv/showthread.php?tid=343069> [91]. Diese Testbuilds sind immer aktuell. Um sie dauerhaft für *LibreELEC* zu aktivieren, gehen Sie bitte wie folgt vor:

- Öffnen Sie die Einstellungen von *KODI*
- Im Bereich Aktualisierungen stellen Sie bitte das Folgende ein:
 - Automatische Aktualisierungen → Manuell
 - Benutzerdefinierte Kanäle anzeigen → Ja
 - Benutzerdefinierter Kanal 1 → <http://milhouse.libreelec.tv/builds/master/generic> [91]
 - Update-Kanal → Milhouse-9.0
- Suchen Sie nun nach „Verfügbaren Versionen“ und installieren Sie die oberste. Das ist die neueste.

Das Update wird nach dem Ausführen dieser Schritte heruntergeladen und installiert. Der PC startet dazu mehrmals neu.

Zuletzt möchte ich Ihnen noch einen Tipp geben, wie Sie *KODI* so starten können, dass Audio Passthrough mit ALSA funktioniert. Der entsprechende Aufruf lautet:

Code-Ausschnitt 8.58: *KODI* mit ALSA Audio Passthrough starten

```
DISPLAY=:0.0 pasuspender -- env KODI_AE_SINK=ALSA kodi
```

Zusammenfassung und Ausblick 8 Das war das Projektkapitel *KODI*. Wohnzimmer PCs sind vielseitig. Sie können damit fernsehen, hochaufgelöste Videos schauen oder Medieninhalte streamen.

Probieren Sie einfach weitere *KODI*-Plugins aus. Machen Sie beispielsweise Telefonnummern von Anrufern mit dem Fritz!Box-Add-on auf Ihrem Fernsehgerät sichtbar oder installieren sie *radio.de*, ein Webradio. Schließen Sie ein externes USB-DVD-Laufwerk an und schauen Sie Ihre DVDs mit *KODI*.

Im nächsten Kapitel zeige ich Ihnen, wie Sie das multimediale Erlebnis mit Hilfe eines Ambilight weiter steigern können. ■

Literaturverzeichnis

- [85] lmd. „Multimedia-Zwerg“. In: *c't magazin für computer technik* 3 (Jan. 2014), Seiten 90–93 (siehe Seite 209).
- [86] OpenELEC. *OpenELEC*. 2019.
<https://openelec.tv>
besucht am 27. 12. 2019 (siehe Seite 209).
- [87] LibreELEC. *LibreELEC Blog*. 2019.
<https://libreelec.tv/>
besucht am 27. 12. 2019 (siehe Seiten 209, 214, 215, 227).
- [88] Wikipedia. *libdvdcss*. 2019.
<http://de.wikipedia.org/wiki/Libdvdcss>
besucht am 27. 12. 2019 (siehe Seite 212).
- [89] koderds.net. *Deutschsprachiges Forum zum Kodi Entertainment Center*. 2019.
<https://www.koderds.net/>
besucht am 27. 12. 2019 (siehe Seite 214).
- [90] picons. *Intro*. 2019.
<https://www.picons.eu/>
besucht am 27. 12. 2019 (siehe Seite 224).
- [91] Kodi. *Milhouse Test Builds*. 2019.
<https://forum.kodi.tv/showthread.php?tid=343069>
besucht am 27. 12. 2019 (siehe Seite 234).

9. Ambilight



9.1	Ambilight-Einkaufsliste für <i>LibreELEC</i>	237
9.2	Ambilight Hardware anschließen	238
9.3	Ambilight Software installieren	242
9.4	Ambilight Fernbedienung	252
9.5	Ambilight ein- und ausschalten	252

Der Begriff „Ambilight“ setzt sich zusammen aus „ambi“ (Umgebung) und „light“ (Licht). Ambilight bedeutet also so viel wie Umgebungslicht. Diese Technologie wurde von Philips für Fernseher entwickelt und patentiert. Ambilight vergrößert das TV-Gesichtsfeld in der Wahrnehmung des Zuschauers dadurch, dass der TV-Hintergrund in den Farben des aktuellen TV-Bildes angestrahlt wird. Was einst entwickelt wurde, um die Augen beim Fernsehen zu schonen, erfüllt nicht nur diesen Zweck, sondern sieht darüber hinaus auch noch gut aus. Ambilight ist Kontext-sensitiv: Ändern sich die Farben des TV-Bildes, ändern sich simultan die Farben dieser Hintergrundbeleuchtung. Dieses Kapitel stellt Ihnen eine preiswerte Möglichkeit vor, mit der Sie Ihren „normalen“ Fernseher in einen Ambilight-TV verwandeln können. Der minimalistische Weg nutzt das Mediacenter *KODI* und einen Micro-Controller für 5 €. Falls Sie einen Ambilight-TV von Philips Ihr Eigen nennen, können Sie dieses Kapitel getrost überspringen. Falls nicht, zeige ich Ihnen, wie sie Ihren Fernseher mit Ambilight nachrüsten können, so dass dieser dem Original in nichts nachsteht.

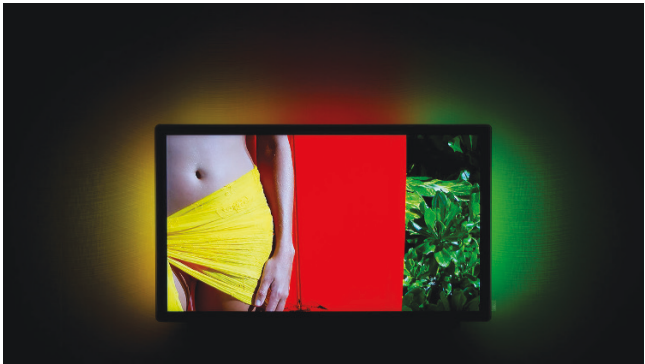


Abbildung 9.1: Ambilight verlängert das Fernsehbild über die Ränder des TVs hinaus (Quelle: Stephan Legachev, wikipedia.de [92]).



9.1 Ambilight-Einkaufsliste für *LibreELEC*

Der in diesem Abschnitt beschriebene Ambilight-Ansatz ist minimalistisch: Es werden nur sehr wenige Komponenten benötigt, um ein Ambilight zu verwirklichen:

- Ein Micro-Controller (z. B. Arduino Nano) für ca. 5 €
- Ein LED-Streifen (5 m für einen 55 Zoll Fernseher), beispielsweise APA102 mit 30 LEDs/m für ca. 50 €
- Ein Netzteil (5 V, ca. 10 A je nach Anzahl der LEDs) für 15 €
- Ein USB-nach-Mini-USB-Kabel zum Anschluss des Micro-Controllers für 1 €
- Litze bzw. Jumperkabel Buchse-Buchse für ca. 2 €
- Klebehaken zur Montage des LED-Streifens hinter dem Fernseher für ca. 2 €

In Summe macht das also ca. 75 €, um Ihren Fernseher bei der Verwendung von *LibreELEC* mit Ambilight auszustatten.

9.2 Ambilight Hardware anschließen

Dreh- und Angelpunkt des Ambilights sind die LED-Streifen, die hinter das Fernsehgerät montiert werden müssen. Hier gibt es verschiedene Hersteller und verschiedene Bestückungsdichten. Ich habe mit dem APA102-Streifen (ca. 30 LEDs pro Meter) sehr gute Erfahrungen gemacht. Für einen 55 Zoll Fernseher werden ca. 5 m LED-Streifen benötigt. Der APA102-Streifen ist beispielsweise bei Amazon oder Ebay erhältlich und



Abbildung 9.2: Der APA102(C) LED-Streifen eignet sich hervorragend, um ein Ambilight nachzubauen (Quelle: <https://amazon.de> [93]).

das sowohl auf weißem wie auf schwarzem PCB (*Printed Circuit Board*). Wie in Abbildung 9.2 gut zu erkennen ist, stellt der APA102 LED-Streifen vier Verbindungskabel zur Verfügung: Eine Spannungsversorgung (5 V, z. B. rot), eine Masseverbindung (GND,

z. B. schwarz), eine Datenverbindung (DO/I, z. B. gelb) in beliebiger Farbe sowie eine Clock-Verbindung (CO/I, z. B. blau) in beliebiger Farbe. Die Buchstaben „I“ und „O“ stehen dabei für „Input“ und „Output“. Alle LEDs sind seriell geschaltet: Der Ausgang der einen LED ist mit dem Eingang der nächsten LED verbunden.

Verbinden Sie bitte die 5 V-Leitung (rot) mit dem 5 V-Ausgang Ihres Netzteils. Dasselbe machen Sie mit der Masse-Verbindung.

Werden längere LED-Streifen nur von einer Seite mit einer Spannung gespeist (z. B. 5 V), leuchten die LEDs am Ende der Kette dunkler als die LEDs am Anfang der Kette. Daher empfehle ich Ihnen, sowohl den Anfang als auch das Ende der LED-Kette an die Spannungsversorgung anzuschließen (5 V und Masse).



Bei manchen vorkonfigurierten APA102-Streifen ist die Masse mit einem roten Kabel verbunden und die 5 V-Leitung mit einem schwarzen Kabel. Schauen Sie hier bitte ganz genau hin und stellen Sie sicher, dass Masse auch mit Masse verbunden wird und 5 V mit 5 V des Netzteils. Vertauschen Sie beide Anschlüsse, kann das sowohl das Netzteil, aber in erster Linie auch den LED-Streifen zerstören.



Den DATA-in-Anschluss des LED-Streifens und den Clock-Anschluss des LED-Streifens verbinden wir mit einem Micro-Controller. Als Micro-Controller empfehle ich Ihnen einen Arduino Nano oder einen entsprechenden Nachbau (Abbildung 9.3).

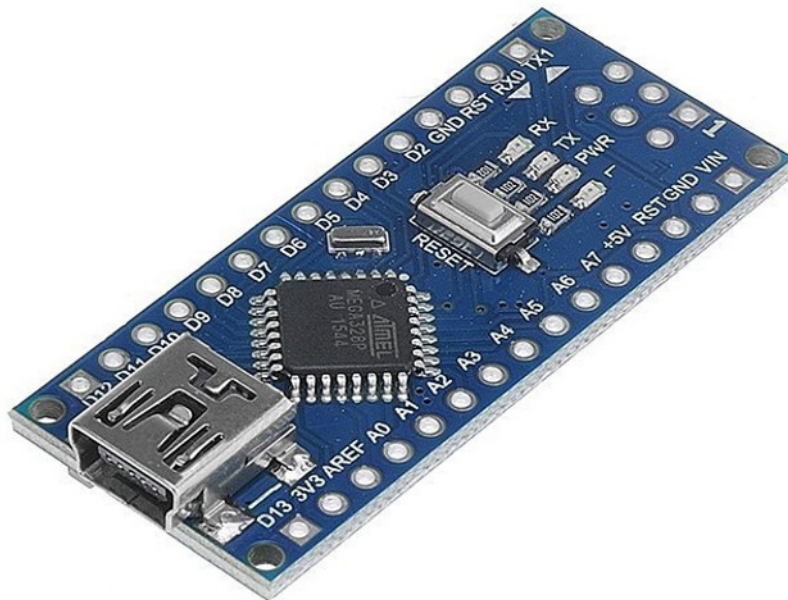


Abbildung 9.3: Einen Arduino Nano-Nachbau gibt es günstig im Dreierpack. Der Elegoo Nano V3 mit CH340 USB-Chip wird von LibreELEC ohne Probleme erkannt (Quelle: <https://www.amazon.de> [94]).



Einige Arduino (Nachbauten) werden als Set ausgeliefert, das man selbst zusammenlöten muss. Dies bezieht sich nur auf die Stiftleisten. Trauen Sie sich diese einfachen Lötarbeiten nicht zu, können Sie auch fertig konfektionierte Micro-Controller kaufen.

Der Masse-Anschluss des LED-Streifens muss mit einem Masse-Anschluss des Micro-Controllers verbunden werden.



Verbinden Sie auf keinen Fall die 5 V-Leitung des LED-Streifens oder des LED-Netzteils mit dem Arduino Micro-Controller.

Schließen Sie den LED-Streifen wie folgt an den Micro-Controller an:

Verbinden Sie bitte die Masse-Leitung des LED-Streifens mit einem GND-Pin des Arduino-Nano, z. B. dem 4. Pin von oben links (Abbildung 9.4). Verbinden Sie die Clock-Leitung (CLK) mit dem Pin D2. Verbinden Sie die Data-In-Leitung mit dem Pin D4.

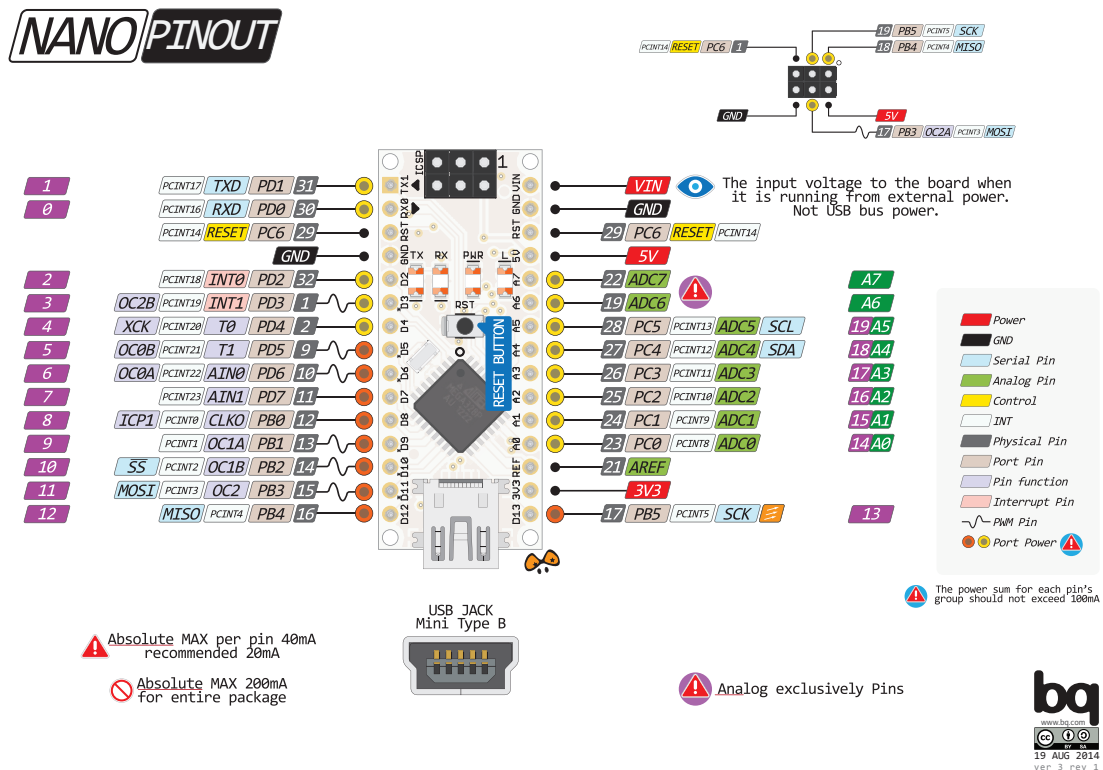


Abbildung 9.4: Die Pins des Arduino Nano (Quelle: Arduino Forum, user pighixx [95]).

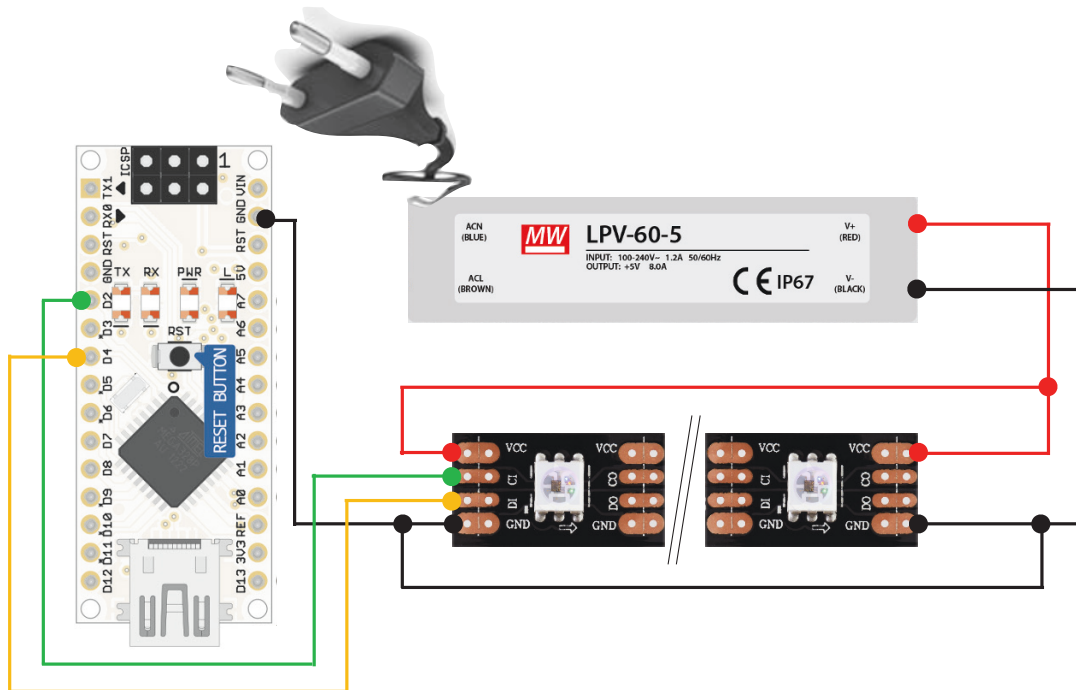


Abbildung 9.5: So verkabeln Sie Ihr Ambilight richtig. Bitte achten Sie auf die Pfeile des LED-Streifens. DATA- und CLK-Eingänge müssen sich am Anfang des LED-Streifens befinden.

Sollten Sie eine LED-Kette mit Pegelwandler (3.3 V auf 5 V) gekauft haben, können Sie die 3.3 V Spannungsversorgung des Pegelwandlers an den Pin 3V3 des Arduino Nano anschließen (2. Pin von unten rechts in Abbildung 9.4).



Verbinden Sie bitte den Micro-Controller mit Hilfe eines USB-nach-Mini-USB-Kabels mit Ihrem Wohnzimmer-PC.

Den LED-Streifen können Sie entweder direkt auf die Rückseite Ihres Fernsehers kleben oder mit Hilfe von selbstklebenden Haltern befestigen. Die LED-Streifen können Sie an jeder Stelle kürzen, indem Sie einfach mit einer Schere ein Stück zwischen den Kontakten entfernen. Wahrscheinlich wird Ihr LED-Streifen auch ein wenig zu lang sein. Das ist nicht weiter schlimm. Wir werden später die nicht benötigten LEDs per Software ausschalten.

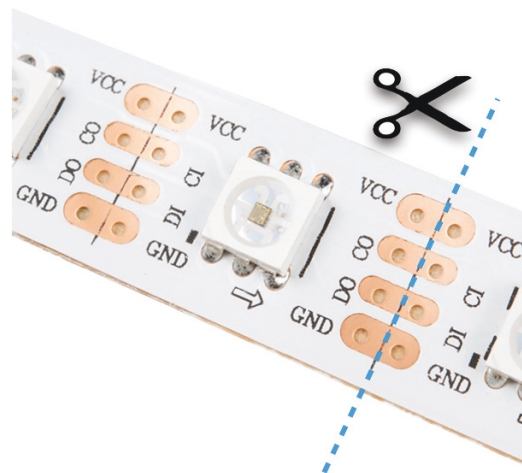


Abbildung 9.6: So schneiden Sie den LED-Streifen richtig.

Abbildung 9.7: Der LED-Streifen kann mit Hilfe von Klebehaken und Kabelbindern hinter dem Fernseher befestigt werden. Für einen 55 Zoll-Fernseher werden ca. 16 Klebehaken benötigt. Die LEDs strahlen auch seitlich. Kleben Sie den Streifen direkt auf den Fernseher, müssen Sie evtl. die Ecken schneiden und neu löten. Das Ambilight macht selbst bei curved-Fernsehern eine gute Figur.



9.3 Ambilight Software installieren

Damit unser Ambilight funktioniert, werden wir dreimal Software installieren. Zunächst einmal benötigt der Micro-Controller Software, einen sogenannten „Sketch“. Danach benötigen wir *Hyperion* für *LibreELEC* und zum Schluß müssen wir uns noch eine Konfigurationsdatei erzeugen, welche wir dann in ein Verzeichnis der *LibreELEC*-Distribution kopieren.

Beginnen wir mit dem Code für den Micro-Controller. Diesen benötigen wir, damit die LEDs unserer Lichterkette über USB mit Hilfe des Micro-Controllers angesprochen werden können. Die Entwicklungsumgebung für einen Arduino Nano und andere Derivate finden Sie unter <http://www.arduino.cc> [96]. Diese Internetseite stellt Software für MacOS, Windows und LINUX bereit, mit deren Hilfe Sie den Micro-Controller flashen können. Die Software selbst ist Open-Source-Software. Bei meinen Versuchen habe ich die Windows-Version genutzt, aber ich bin sicher, dass die anderen Versionen dieser Software genau so gut funktionieren.

Laden Sie also bitte die Version Ihrer Wahl herunter. Um den Sketch (also das Micro-Controller-Programm), das ich Ihnen gleich vorstellen werde, zu flashen, benötigen wir ebenfalls die Bibliothek namens *FastLED*. Diese können Sie von <http://fastled.io/> [97] herunterladen.

Die Arduino Software-IDE kann mit gepackten (ge-zippten) Bibliotheks-Dateien umgehen. Sie müssen also die Bibliothek FastLED nicht entpacken, sondern können der IDE den Speicherort der gepackten Bibliothek mitteilen (Sketch → Bibliothek einbinden).



Für den APA102 LED-Streifen habe ich den folgenden Sketch benutzt:

Code-Ausschnitt 9.1: Ambilight Sketch für den Arduino

```
#include "FastLED.h"

#define ANALOG_MODE_AVERAGE 0
#define ANALOG_MODE_LAST_LED 1

/*****
 S E T U P

    set following values to your needs
*****/

#define INITIAL_LED_TEST_ENABLED true
#define INITIAL_LED_TEST_BRIGHTNESS 32 // 0..255
#define INITIAL_LED_TEST_TIME_MS 2000 // 10..

// Number of leds in your strip. set to "1" and ANALOG_OUTPUT_ENABLED to "true" to activate ↔
// analog only
// As of 26/1/2017:
// 582 leaves ZERO bytes free and this
// 410 is ok
// tested with 500 leds and is fine (despite the warning)
#define MAX_LEDS 160

// type of your led controller, possible values, see below
#define LED_TYPE APA102

// 3 wire (pwm): NEOPIXEL BTM1829 TM1812 TM1809 TM1804 TM1803 UCS1903 UCS1903B UCS1904 UCS2903 ↔
// WS2812 WS2852
// S2812B SK6812 SK6822 APA106 PL9823 WS2811 WS2813 APA104 WS2811_40 GW6205 ↔
// GW6205_40 LPD1886 LPD1886_8BIT
// 4 wire (spi): LPD8806 WS2801 WS2803 SM16716 P9813 APA102 SK9822 DOTSTAR

// For 3 wire led stripes line Neopixel/WS2812, which have a data line, ground, and power, you ↔
// just need to define DATA_PIN.
// For led chipsets that are SPI based (four wires - data, clock, ground, and power), both ↔
// defines DATA_PIN and CLOCK_PIN are needed

// DATA_PIN, or DATA_PIN, CLOCK_PIN
// #define LED_PINS 6 // 3 wire leds
// #define LED_PINS 6, 8 // 4 wire leds
#define LED_PINS 4, 2 // 4 wire leds

#define COLOR_ORDER GRB // colororder of the stripe, set RGB in hyperion

#define OFF_TIMEOUT 15000 // ms to switch off after no data was received, set 0 to deactivate

// analog rgb uni color led stripe - using of hyperion smoothing is recommended
// ATTENTION this pin config is default for atmega328 based arduinos, others might work to
// if you have flickering analog leds this might be caused by unsynced pwm signals
// try other pins is more or less the only thing that helps
#define ANALOG_OUTPUT_ENABLED false
#define ANALOG_MODE ANALOG_MODE_LAST_LED // use ANALOG_MODE_AVERAGE or ↔
// ANALOG_MODE_LAST_LED
#define ANALOG_GROUND_PIN 8 // additional ground pin to make wiring a bit ↔
// easier
#define ANALOG_RED_PIN 9
#define ANALOG_GREEN_PIN 10
#define ANALOG_BLUE_PIN 11
```

```
// overall color adjustments
#define ANALOG_BRIGHTNESS_RED 255 // maximum brightness for analog 0-255
#define ANALOG_BRIGHTNESS_GREEN 255 // maximum brightness for analog 0-255
#define ANALOG_BRIGHTNESS_BLUE 255 // maximum brightness for analog 0-255

#define BRIGHTNESS 255 // maximum brightness 0-255
#define DITHER_MODE BINARY_DITHER // BINARY_DITHER or DISABLE_DITHER
#define COLOR_TEMPERATURE CRGB(255,255,255) // RGB value describing the color temperature
#define COLOR_CORRECTION TypicalLEDStrip // predefined fastled color correction
// #define COLOR_CORRECTION CRGB(255,255,255) // or RGB value describing the color correction
// Baudrate, higher rate allows faster refresh rate and more LEDs
// #define serialRate 460800 // use 115200 for ftdi based boards
// #define serialRate 115200 // use 115200 for ftdi based boards
#define serialRate 500000 // use 115200 for ftdi based boards

/*****
  A D A L I G H T C O D E

  no user changes needed
  *****/

// Adalight sends a "Magic Word" (defined in /etc/boblight.conf) before sending the pixel data
uint8_t prefix[] = {'A', 'd', 'a'}, hi, lo, chk, i;

unsigned long endTime;

// Define the array of leds
CRGB leds[MAX_LEDS];

// set rgb to analog led stripe
void showAnalogRGB(const CRGB& led) {
  if (ANALOG_OUTPUT_ENABLED) {
    byte r = map(led.r, 0,255,0,ANALOG_BRIGHTNESS_RED);
    byte g = map(led.g, 0,255,0,ANALOG_BRIGHTNESS_GREEN);
    byte b = map(led.b, 0,255,0,ANALOG_BRIGHTNESS_BLUE);
    analogWrite(ANALOG_RED_PIN, r);
    analogWrite(ANALOG_GREEN_PIN, g);
    analogWrite(ANALOG_BLUE_PIN, b);
  }
}

// set color to all leds
void showColor(const CRGB& led) {
  #if MAX_LEDS > 1 || ANALOG_OUTPUT_ENABLED == false
    LEDS.showColor(led);
  #endif
  showAnalogRGB(led);
}

// switch of digital and analog leds
void switchOff() {
  #if MAX_LEDS > 1 || ANALOG_OUTPUT_ENABLED == false
    memset(leds, 0, MAX_LEDS * sizeof(struct CRGB));
    FastLED.show();
  #endif
  showAnalogRGB(leds[0]);
}

// function to check if serial data is available
// if timeout occurred leds switch of, if configured
bool checkIncommingData() {
  boolean dataAvailable = true;
  while (!Serial.available()) {
    if (OFF_TIMEOUT > 0 && endTime < millis()) {
      switchOff();
      dataAvailable = false;
      endTime = millis() + OFF_TIMEOUT;
    }
  }
}
```

```

    return dataAvailable;
}

// main function that setups and runs the code
void setup() {
    Serial.begin(serialRate);

    // analog output
    if (ANALOG_OUTPUT_ENABLED) {
        // additional ground pin to make wiring a bit easier
        pinMode(ANALOG_GROUND_PIN, OUTPUT);
        digitalWrite(ANALOG_GROUND_PIN, LOW);
        pinMode(ANALOG_BLUE_PIN, OUTPUT);
        pinMode(ANALOG_RED_PIN, OUTPUT);
        pinMode(ANALOG_GREEN_PIN, OUTPUT);
    }

    int ledCount = MAX_LEDS;
    if (ANALOG_MODE == ANALOG_MODE_LAST_LED) {
        ledCount--;
    }

    #if MAX_LEDS > 1 || ANALOG_OUTPUT_ENABLED == false
        FastLED.addLeds<LED_TYPE, LED_PINS, COLOR_ORDER>(leds, ledCount);
    #endif

    // color adjustments
    FastLED.setBrightness ( BRIGHTNESS );
    FastLED.setTemperature( COLOR_TEMPERATURE );
    FastLED.setCorrection ( COLOR_CORRECTION );
    FastLED.setDither ( DITHER_MODE );

    // initial RGB flash
    #if INITIAL_LED_TEST_ENABLED == true
    for (int v=0;v<INITIAL_LED_TEST_BRIGHTNESS;v++)
    {
        showColor(CRGB(v,v,v));
        delay(INITIAL_LED_TEST_TIME_MS/2/INITIAL_LED_TEST_BRIGHTNESS);
    }

    for (int v=0;v<INITIAL_LED_TEST_BRIGHTNESS;v++)
    {
        showColor(CRGB(v,v,v));
        delay(INITIAL_LED_TEST_TIME_MS/2/INITIAL_LED_TEST_BRIGHTNESS);
    }
    #endif
    showColor(CRGB(0, 0, 0));

    Serial.print("Ada\n"); // Send "Magic Word" string to host

    boolean transmissionSuccess;
    unsigned long sum_r, sum_g, sum_b;

    // loop() is avoided as even that small bit of function overhead
    // has a measurable impact on this code's overall throughput.
    for(;;) {
        // wait for first byte of Magic Word
        for (i = 0; i < sizeof prefix; ++i) {
            // If next byte is not in Magic Word, the start over
            if (!checkIncommingData() || prefix[i] != Serial.read()) {
                i = 0;
            }
        }

        // Hi, Lo, Checksum
        if (!checkIncommingData()) continue;
        hi = Serial.read();
        if (!checkIncommingData()) continue;
        lo = Serial.read();
    }

```

```

    if (!checkIncommingData()) continue;
    chk = Serial.read();

    // if checksum does not match go back to wait
    if (chk != (hi ^ lo ^ 0x55)) continue;

    memset(leds, 0, MAX_LEDS * sizeof(struct CRGB));
    transmissionSuccess = true;
    sum_r = 0;
    sum_g = 0;
    sum_b = 0;

    int num_leds = min ( MAX_LEDS, (hi<<8) + lo + 1 );

    // read the transmission data and set LED values
    for (int idx = 0; idx < num_leds; idx++) {
        byte r, g, b;
        if (!checkIncommingData()) {
            transmissionSuccess = false;
            break;
        }
        r = Serial.read();
        if (!checkIncommingData()) {
            transmissionSuccess = false;
            break;
        }
        g = Serial.read();
        if (!checkIncommingData()) {
            transmissionSuccess = false;
            break;
        }
        b = Serial.read();
        leds[idx].r = r;
        leds[idx].g = g;
        leds[idx].b = b;
        #if ANALOG_OUTPUT_ENABLED == true && ANALOG_MODE == ANALOG_MODE_AVERAGE
            sum_r += r;
            sum_g += g;
            sum_b += b;
        #endif
    }

    // shows new values
    if (transmissionSuccess) {
        endTime = millis() + OFF_TIMEOUT;
        #if MAX_LEDS > 1 || ANALOG_OUTPUT_ENABLED == false
        FastLED.show();
        #endif

        #if ANALOG_OUTPUT_ENABLED == true
            #if ANALOG_MODE == ANALOG_MODE_LAST_LED
                showAnalogRGB(leds[MAX_LEDS-1]);
            #else
                showAnalogRGB(CRGB(sum_r/MAX_LEDS, sum_g/MAX_LEDS, sum_b/MAX_LEDS));
            #endif
        #endif
    }
}
} // end of setup

void loop() {
    // Not used. See note in setup() function.
}

```

Bitte ändern Sie die entsprechenden Zeilen am Anfang des Sketches, um evtl. andere LED-Streifen zu verwenden. Weiterhin sollten Sie den Wert von *MAX_LEDS* auf die Anzahl Ihrer LEDs anpassen. Dasselbe gilt für Zeile 36 des Sketches, den *DATA-Pin* und den *CLOCK-Pin*.

Für den von mir empfohlenen Arduino Nano-Nachbau habe ich die Arduino-IDE-Einstellung Arduino/Genuino Uno gewählt, um den Sketch auf den Micro-Controller hochzuladen. Kopieren Sie den Sketch in ein leeres Projekt und laden Sie ihn dann auf den Micro-Controller hoch. Nachdem der Arduino startet - der USB-Anschluss dient dabei gleichzeitig als Spannungsversorgung - sollten die LEDs bereits ein wenig leuchten.

Bitte starten Sie für den nächsten Schritt Ihre *LibreELEC*-Distribution und installieren Sie das *Hyperion*-Add-on. Dieses Add-on hat keinerlei Einstellungsmöglichkeiten. Nachdem es einmal geladen wurde, wird das Ambilight dauernd an sein. Ich stelle Ihnen im weiteren Verlauf dieses Kapitels noch ein eigenes Plugin vor, welches es Ihnen erlaubt, das Ambilight mit der Fernbedienung ein- und auszuschalten. Sie finden das Hyperion-Add-on unter *Alle Add-ons* → *Dienste*.

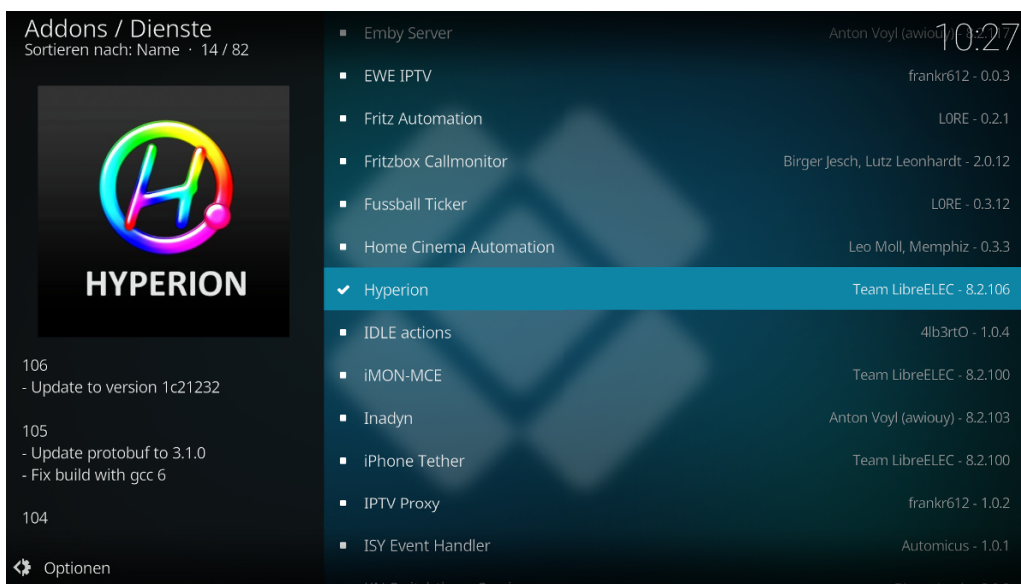


Abbildung 9.8: LibreELEC stellt das *Hyperion*-Add-on unter *Dienste* bereit.

Hyperion erwartet noch eine Konfigurationsdatei, damit das Ambilight funktioniert. Diese Datei heißt *hyperion.config.json* und muss in das *LibreELEC*-Verzeichnis *.kodi/user-data/addon_data/service.hyperion* kopiert werden. Es würde den Rahmen dieses Buches sprengen, meine komplette Konfigurationsdatei an dieser Stelle abzudrucken. Daher erwähne ich an dieser Stelle nur die wichtigsten Einstellungen.

Code-Ausschnitt 9.2: *Hyperion*-Konfigurationsdatei

```

{
    // DEVICE CONFIGURATION
    "device" :
    {
        "name"      : "MyHyperionConfig",
        "type"      : "adalight",
        "output"    : "/dev/ttyUSB0",
        "rate"      : 500000,
        "delayAfterConnect" : 0,
        "colorOrder" : "gbr"
    },
    // COLOR CALIBRATION CONFIG
    "color" :
    {
        "channelAdjustment" :
        [
            {
                "id" : "default",
                "leds" : "0-107",
                "pureRed" :
                {
                    "redChannel" : 255,
                    "greenChannel" : 0,
                    "blueChannel" : 0
                },
                "pureGreen" :
                {
                    "redChannel" : 0,
                    "greenChannel" : 255,
                    "blueChannel" : 0
                },
                "pureBlue" :
                {
                    "redChannel" : 0,
                    "greenChannel" : 0,
                    "blueChannel" : 255
                }
            },
            "temperature" :
            [
                {
                    "id" : "default",
                    "leds" : "0-107",
                    "correctionValues" :
                    {
                        "red" : 255,
                        "green" : 255,
                        "blue" : 255
                    }
                }
            ],
            "transform" :
            [
                {
                    "id" : "default",
                    "leds" : "0-107",
                    "hsl" :
                    {
                        "saturationGain" : 1.0500,
                    },
                    "luminanceGain" : 1.0500,
                    "luminanceMinimum" : 0.0000
                },
                "red" :
                {
                    "threshold" : 0.0800,
                },
                "gamma" : 1.4900
            ],
        ],
    },
    "gamma" : 1.4900
}

```



```

        "green" :
        {
            "threshold" : 0.0800,
            "gamma" : 1.4900
        },
        "blue" :
        {
            "threshold" : 0.0800,
            "gamma" : 1.4500
        }
    },
    // SMOOTHING CONFIG
    "smoothing" :
    {
        "type" : "linear",
        "time_ms" : 200,
        "updateFrequency" : 25.0000,
        "updateDelay" : 0
    },
    "framegrabber" :
    {
        "width" : 64,
        "height" : 64,
        "frequency_Hz" : 50.0,
        "priority" : 890
    },
    // BLACKBORDER CONFIG
    "blackborderdetector" :
    {
        "enable" : true,
        "threshold" : 0.05,
        "unknownFrameCnt" : 600,
        "borderFrameCnt" : 50,
        "maxInconsistentCnt" : 10,
        "blurRemoveCnt" : 1,
        "mode" : "default"
    },
    // KODI CHECK CONFIG
    "xbmcVideoChecker" :
    {
        "xbmcAddress" : "127.0.0.1",
        "xbmcTcpPort" : 9090,
        "grabVideo" : true,
        "grabPictures" : true,
        "grabAudio" : true,
        "grabMenu" : false,
        "grabPause" : false,
        "grabScreensaver" : true,
        "enable3DDetection" : true
    },
    // BOOTEFFECT CONFIG
    "bootsequence" :
    {
        "color" : [0,0,0],
        "effect" : "Rainbow swirl fast",
        "duration_ms" : 7000,
        "priority" : 700
    },
    // JSON SERVER CONFIG
    "jsonServer" :
    {
        "port" : 19444
    },
    // PROTO SERVER CONFIG
    "protoServer" :
    {
        "port" : 19445
    },

```

```
// WEBCONFIG SERVER
"webConfig" :
{
    "document_root" : "/storage/.kodi/addons/service.hyperion/webconfig",
    "port" : 8099
},
// EFFECT PATH
"effects" :
{
    "paths" :
    [
        "/storage/.kodi/addons/service.hyperion/effects",
        "/usr/share/hyperion/effects"
    ]
},
// LED CONFIGURATION
"leds" :
[
    {
        "index" : 0,
        "hscan" : { "minimum" : 0.0000, "maximum" : 0.0500 },
        "vscan" : { "minimum" : 0.9474, "maximum" : 1.0000 }
    },
    ...
],
"endOfJson" : "endOfJson"
}
```

Die genauen LED-Einstellungen ab Zeile 163 habe ich weggelassen.



Die komplette Datei `hyperion.config.json` für meine LED-Konfiguration steht im Download-Bereich dieses Buches auf zur Verfügung. Selbst, wenn meine LED-Konfiguration nicht für Ihre passt, sollte die LED-Anzeige eines Regenbogens bei Ihrem Ambilight auch mit meiner Konfigurationsdatei funktionieren.

Um die LEDs Ihres Ambilights zu konfigurieren, gibt es das kostenlose Tool *Hypercon* für *LibreELEC*. Das Tool steht zum Download unter <https://wiki.libreelec.tv/hypercon> [98] in einer LINUX- und in einer Windows-Version bereit. Da das Tool eine Java-Applikation ist, setzt es ein installiertes Java voraus. Unter LINUX können Sie Java so installieren

Code-Ausschnitt 9.3: Java installieren

```
sudo apt-get install openjdk-8-jre
```

und das Programm anschließend mit

Code-Ausschnitt 9.4: *Hypercon* starten

```
java -jar ./hypercon-LE.jar
```

aufzurufen. Unter Windows müssen Sie zunächst ein Java von <http://www.java.com/en/> [99] installieren, bevor Sie das Tool mit einem Doppelklick öffnen können.

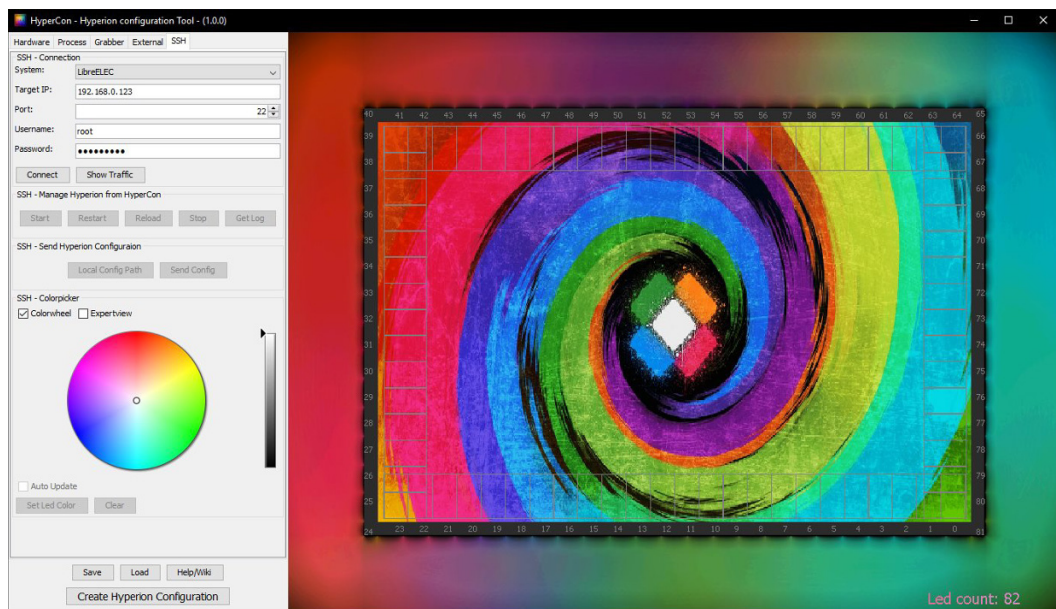


Abbildung 9.9: Mit dem Java-Tool *Hypercon* erstellt man schnell und komfortabel eine Konfigurationsdatei *hyperion.config.json* für *Hyperion* (Quelle: wiki.libreelec.tv [100]).



Abbildung 9.10: Ambilight sieht selbst hinter einem curved-Fernseher gut aus.

Tragen Sie unter TargetIP die IP-Adresse Ihres Wohnzimmer-PCs ein. Der *Username* lautet „root“ und das *Password* „libreelec“.

Die Daten für den Grabber können Sie aus meinem Listing (Quelltext 9.2) übernehmen. Die meisten Änderungen werden Sie im Abschnitt *Hardware* und in Bezug auf die Anordnung und Anzahl Ihrer LEDs vornehmen müssen.



Bei meinem Ambilight-Aufbau habe ich die erste LED unten links hinter dem Fernseher platziert und dann den LED-Streifen nach oben (nicht nach rechts) weiter befestigt. Sie können aber - falls Sie sich für eine andere Reihenfolge/Anordnung entschieden haben - Ihre eigenen Daten in Hypercon eintragen.

Ein Klick auf den Button *Create Hyperion Config* speichert die Datei *hyperion.config.json*, die Sie dann nur noch in das entsprechende *LibreELEC*-Verzeichnis kopieren müssen. Selbst auf meinem curved-TV macht das Ambilight „Marke Eigenbau“ eine gute Figur.

9.4 Ambilight Fernbedienung

Auch für das Ambilight gibt es eine Fernbedienung, die auf einem mobilen Telefon oder einem Tablet läuft. Die App nennt sich „Hyperion“. Es gibt sie sowohl für Apple-Geräte als auch für Android-Geräte kostenlos.

Mit Hilfe dieser App kann das Ambilight durch Antippen des Farbkreises auf eine bestimmte Farbe eingestellt werden. Weiterhin kann die Helligkeit des Ambilights eingestellt werden.

Darüber hinaus bietet die App verschiedene Sondereffekte wie beispielsweise den „Knight Rider-Effekt“ oder den „Regenbogen-Effekt“. Die App sucht automatisch nach einer *LibreELEC/Hyperion*-Installation im Heimnetz. Der benötigte Port ist dabei schon eingetragen.

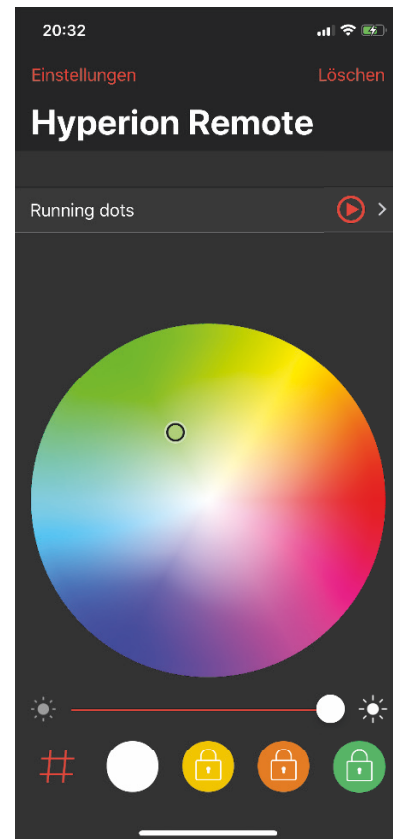


Abbildung 9.11: Hyperion remote.

9.5 Ambilight ein- und ausschalten

Wenn *Hyperion* einmal in *LibreELEC* aktiviert wurde, läuft es. Die einzige Möglichkeit, es abzuschalten, wäre, es wieder zu deinstallieren. Da das keine Option ist, zeige ich Ihnen in diesem Abschnitt, wie Sie ein einfaches Add-on für *LibreELEC* programmieren und installieren können, das genau diese Aufgabe übernimmt. Das Add-on erlaubt es, den *Hyperion*-Dienst per Fernbedienung an- bzw. auszuschalten. Wir können es sogar noch

erweitern und parallel das Ein- bzw. Ausschalten einer Funksteckdose ermöglichen, um das Netzteil für den LED-Streifen vom Netz zu trennen, um so Strom zu sparen. Doch dazu später mehr. Damit wir das Rad nicht komplett neu erfinden müssen, benutzen wir als Vorlage ein „Hello World“-Beispiel. Dieses Beispiel steht unter https://kodi.wiki/view/HOW-TO:HelloWorld_addon [101] zum Download bereit. Laden Sie bitte die Zip-Datei auf Ihren Rechner herunter (<https://github.com/zag2me/script.hello.world/archive/master.zip> [101]). Das Add-on könnte man installieren, indem man im Add-on-Browser den Eintrag *Aus Zip-Datei installieren* wählt. Bitte installieren Sie dieses Plugin aber noch nicht, da wir noch ein paar kleine Änderungen vornehmen werden.

Falls Sie ein KODI-Add-on installieren und es danach ändern, müssen Sie eventuell die KODI-Datenbank löschen, bevor beispielsweise geänderte Icons eines Plugins angezeigt werden. KODI speichert bei der Installation die Icons des Plugins in einer Datenbank und ruft sie danach aus der Datenbank ab. Änderungen werden dadurch erst nach Löschen der Datenbank sichtbar.

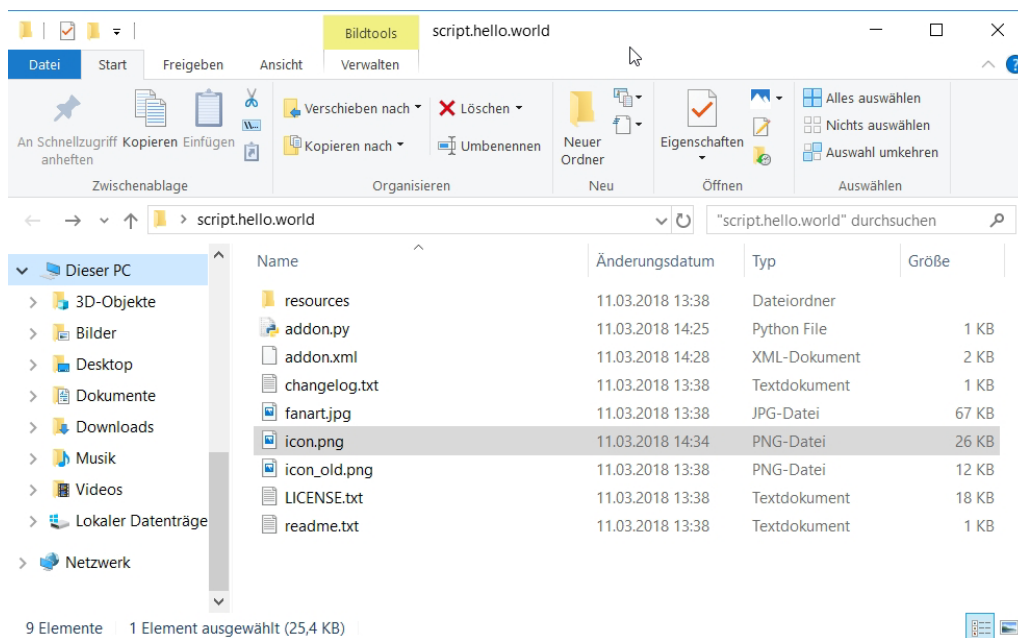


Abbildung 9.12: Die Struktur eines KODI-Add-ons.

Die wichtigsten Bestandteile des Add-ons sind:

- **addon.py:** KODI-Add-ons sind in Python programmiert. Diese Datei enthält den Python-Quelltext des Plugins.
- **addon.xml:** Diese Datei enthält die Metadaten des Add-ons. Die Daten liegen im XML-Format (*Extensible Markup Language*) vor und sind überwiegend selbsterklärend.
- **changelog.txt:** Diese Datei enthält Änderungsinformationen zum Add-on und wird üblicherweise von Update zu Update fortgeschrieben.
- **icon.png:** Diese Datei ist das Icon des Plugins. Es kann entweder 256×256 oder 512×512 Pixel groß sein.

- **LICENSE.txt:** Diese Datei ist eine Textdatei, welche die Lizenzinformationen des Add-ons beinhaltet.

Darüber hinaus kann es noch weitere Verzeichnisse geben (z. B. *ressources*), welche weitere Hintergrundbilder beinhalten können. Suchen Sie sich zunächst aus dem Internet ein schönes Icon für die Datei *icon.png* aus oder entwerfen Sie ein eigenes Icon. Eine Google-Suche nach „Ambilight Icon“ ist dabei sehr hilfreich.

Den eigentlichen Quelltext des Add-ons kopieren wir in die Datei *addon.py*, nachdem wir zuvor deren Inhalt gelöscht haben.

Code-Ausschnitt 9.5: *Hyperion*-Konfigurationsdatei

```
import xbmcaddon
import xbmcgui
import os
import subprocess

def get_process_id(name):
    child = subprocess.Popen(['pgrep', '-f', name], stdout=subprocess.PIPE, shell=False)
    response = child.communicate()[0]
    return [int(pid) for pid in response.split()]

addon = xbmcaddon.Addon()
addonname = addon.getAddonInfo('name')

try:
    pid = get_process_id("hyperion")[0]
    os.system('systemctl stop service.hyperion.service 2>/dev/null')
    os.system('echo "0" > /dev/ttyUSB1')
    line1 = "Hyperion was running. Stopping..."
except:
    os.system('systemctl start service.hyperion.service 2>/dev/null')
    os.system('echo "1" > /dev/ttyUSB1')
    line1 = "Hyperion was not running. Starting..."

xbmcgui.Dialog().ok(addonname, line1)
```

Zunächst werden weitere Bibliotheken importiert, welche vom Add-on benötigt werden. Nach der Definition der Funktion *get_process_id* und der Initialisierung des Add-ons stellt dieses fest, ob der *Hyperion*-Dienst läuft oder nicht. Läuft er, wird er abgeschaltet, läuft er nicht, wird er eingeschaltet. Das Add-on toggelt also zwischen „Ambilight an“ und „Ambilight aus“. Anschließend öffnet das Add-on ein Fenster in *KODI* und zeigt den aktuellen Status des Add-ons an.

Vielleicht sind Ihnen die beiden Zeilen aufgefallen, in denen eine „0“ bzw. „1“ auf das USB-Gerät */dev/ttyUSB1* geschrieben werden. Diese beiden Zeilen schalten eine Funksteckdose ein- bzw. aus, die bei mir von einem zweiten Arduino-Nano mit einem kleinen Funksender gesteuert wird. Diesen Aufbau erkläre ich Ihnen aber, nachdem wir das Add-on in *KODI* installiert haben.

Nachdem wir die Datei *addon.py* mit dem oben gezeigten Inhalt gefüllt haben, „zippen“ wir wieder alle Dateien unseres Plugins. Sollten Sie die Dateien unter Windows bearbeitet haben, können Sie alle Dateien markieren und mit Hilfe der rechten Maustaste zu einem „zip“-Archiv hinzufügen.



Da Windows andere Zeilenumbrüche nutzt als LINUX, empfehle ich Ihnen unter Windows den freien Editor Notepad++, der kompatibel mit LINUX ist.

Unter LINUX können Sie die Datei mit folgendem Befehl auspacken:

Code-Ausschnitt 9.6: ZIP-Datei auspacken

```
zip -r ambilight.zip *
```

Die „gezippte“ Datei heißt *ambilight.zip* und enthält alle Dateien (*) des aktuellen Ordners. Kopieren Sie diese Datei dann bitte mit Hilfe von Samba in das *storage*-Verzeichnis der *LibreELEC*-Distribution. Sie können dieses Plugin jetzt installieren, indem Sie unter Add-ons *Aus zip-Datei installieren* wählen. Das Add-on erscheint dann unter der Rubrik *Add-ons* im Hauptmenü.

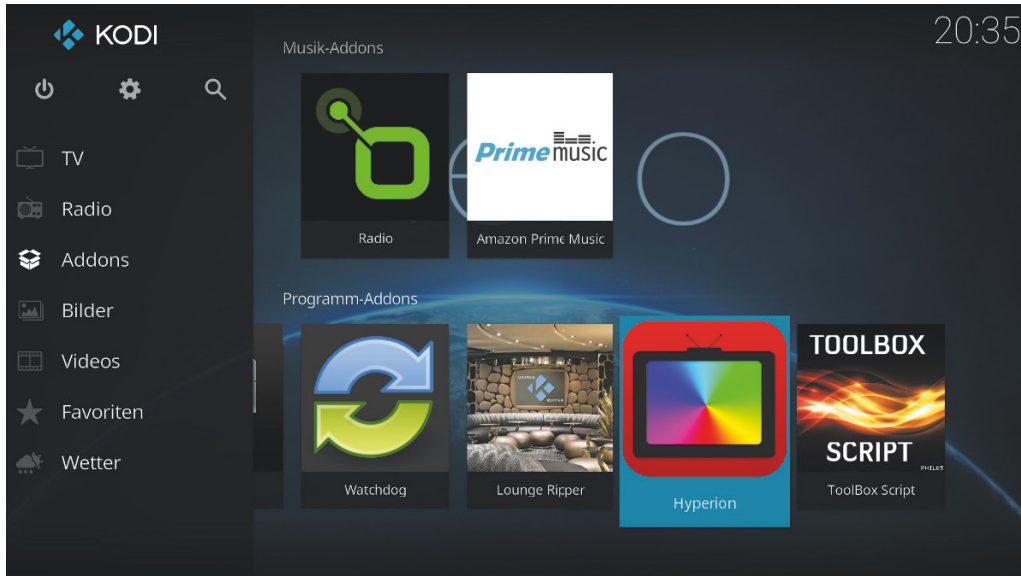


Abbildung 9.13: Das eigene *Hyperion*-Add-on. Im Hintergrund ist noch das Titelbild des *Hello*-Add-ons zu erkennen. Das Icon wurde getauscht.

Wenn Sie viele Add-ons installiert haben, kann das schon einmal unübersichtlich werden. In diesem Fall können Sie das *Ambilight*-Add-on unter *Favoriten* speichern, um schneller Zugriff darauf zu haben.

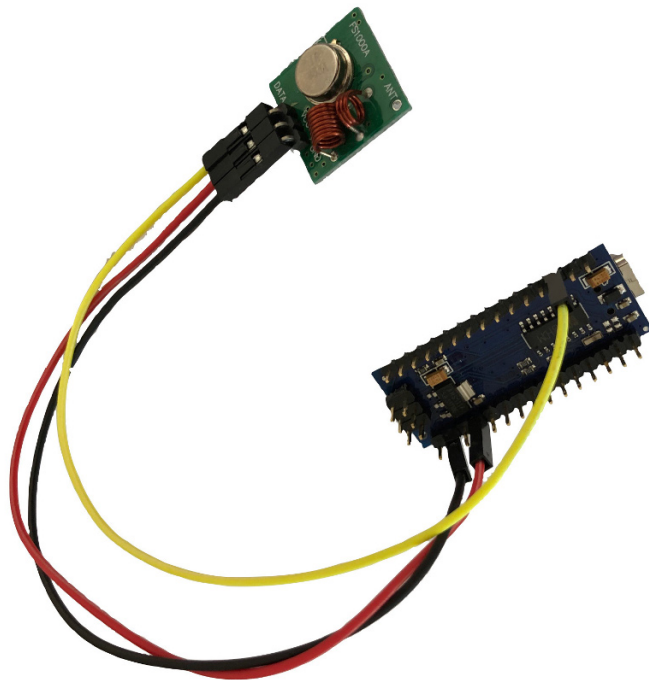


Das Add-on gibt Auskunft darüber, ob der *Hyperion*-Dienst läuft oder nicht und schaltet ihn dementsprechend ein oder aus. Falls Sie Strom sparen wollen, empfehle ich Ihnen die Verwendung einer Funksteckdose, die Sie zusammen mit dem Ambilight ein- bzw. ausschalten können. Auch das können Sie mit Hilfe eines Arduino Nano erledigen, an den Sie einen 433 MHz stecken. Verbinden Sie dafür bitte den Ground-Pin (GND) des Senders mit einer GND-Verbindung auf dem Arduino Nano, z. B. den 2. Pin von oben rechts (Abbildung 9.4). Den VCC-Pin verbinden Sie bitte mit einem 5 V-Pin des Arduino Nano. Der befindet sich zwei Pins unterhalb des oben genannten GND-Pins. Als DATA-Pin nutze ich den Pin 10. Das ist der dritte Pin von unten links. Für die Verbindungen von Sender und Arduino Nano bieten sich Jumperkabel an.



Abbildung 9.14: 433 MHz Sender (rechts) und Empfänger (links) (Quelle: <https://www.amazon.de> [102]).

Abbildung 9.15: Der 433 MHz-Sender kann mit Jumperkabeln schnell mit dem Arduino Nano verbunden werden. Bitte beachten Sie, dass der Arduino Nano hier auf dem Kopf liegt, also mit dem USB-Anschluss nach unten (im Gegensatz zur Abbildung 9.3).



Der Sketch, den ich zum Ausschalten meiner Funksteckdose nutze, sieht wie folgt aus:

Code-Ausschnitt 9.7: Sketch zum Ein- und Ausschalten einer Funksteckdose

```
#include <RCSwitch.h>
int incomingByte = 0; // for incoming serial data
RCSwitch mySwitch = RCSwitch();

void setup() {
  Serial.begin(9600);
  // Requires Libreelec screen /dev/ttyUSB1 9600
  // Transmitter is connected to Arduino Pin #10
  mySwitch.enableTransmit(10);
  // Optional set protocol (default is 1, will work for most outlets)
  mySwitch.setProtocol(1);
  // Optional set pulse length.
  mySwitch.setPulseLength(308);
  // Optional set number of transmission repetitions.
  mySwitch.setRepeatTransmit(15);
}

void loop() {
  // send data only when you receive data:
  if (Serial.available() > 0) {
    // read the incoming byte:
    incomingByte = Serial.read();

    if (incomingByte == 48)
    {
      mySwitch.sendTriState("0000FFFF0F");
      delay(1000);
    }
    if (incomingByte == 49)
    {
      mySwitch.sendTriState("0000FFFF0F");
      delay(1000);
    }
  }
}
```

Das Skript zum Ein- und Ausschalten des Ambilight geht davon aus, dass der Arduino Nano, der mit der LED-Kette verbunden ist, am USB-Anschluss 0 steckt und der Sender, der die Funksteckdose schaltet, am USB-Anschluss 1. Gegebenenfalls müssen Sie die beiden USB-Ports tauschen. Für die Kommunikation mit dem 433 MHz-Sender muss die Baud-Rate für den USB-Anschluss, an dem der Arduino Nano mit dem Funksender steckt, auf 9600 Baud eingestellt werden. Üblicherweise nutzt man hierzu das Programm *setserial*, das aber unglücklicherweise unter *LibreELEC* nicht zur Verfügung steht. Die einzige Möglichkeit, die ich gefunden habe, ist der Befehl *screen*, den man auch dazu missbrauchen kann, die Baud-Rate eines USB-Anschlusses einzustellen. Bitte ergänzen Sie die folgende Zeilen in der Datei *storage/.config/autostart.sh*.

Code-Ausschnitt 9.8: Baud-Rate unter *LibreELEC* setzen

```
systemctl stop service.hyperion.service
sleep 5
screen -dmS myscreen /dev/ttyUSB1 9600
```

Die erste Zeile sorgt dafür, dass der Ambilight-Dienst *Hyperion* nach dem Start von *LibreELEC* ausgeschaltet ist. Möchten Sie, dass er eingeschaltet ist, löschen Sie diese Zeile bitte. Ein *sleep 5* stellt sicher, dass das System auch hochgefahren wurde, bevor der *screen*-Befehl dann aufgerufen wird, um die Übertragungsgeschwindigkeit für den USB-Port mit Arduino Nano und Funksender zu setzen. Der Name des Screens ist dabei vollkommen willkürlich (*myscreen*).

Zusammenfassung und Ausblick 9 Dieses Kapitel hat Ihnen gezeigt, wie Sie ihr Mediacenter mit einem Ambilight ausstatten können, das die Farbe des Umgebungslichtes der Farbe des abgespielten Filmes anpasst.

Das nächste Kapitel gibt Ihnen einen Überblick über Server und Datenbanken. Lernen Sie, wie Sie jeden Drucker in Ihrem Haushalt nutzen können, um kabellos von mobilen Geräten aus zu drucken. Installieren Sie einen DLNA-Server, der Bilder, Musik und Filme in andere Räume Ihres Heimnetzes verteilt.

Weiterhin erklärt das nächste Kapitel, wie Sie eine Webseite mit Hilfe des Content Management Systems Wordpress servieren können oder per Skript Ihre Philips Hue-Lampen steuern können. ■

Literaturverzeichnis

- [92] Wikipedia. *Ambilight*. 2020.
<https://de.wikipedia.org/wiki/Ambilight>
besucht am 01. 01. 2020 (siehe Seite 237).
- [93] Amazon. *APA102(c)*. 2020.
https://www.amazon.de/APA102C-Adressierbares-Streifenlicht-wasserdicht-Schwarz/dp/B06WVHYMGK/ref=sr_1_5?__mk_de_DE=%C3%85M%C3%85C5BD%C3%95C3%91&keywords=apa102c+led+streifen&qid=1577966651&sr=8-5
besucht am 01. 01. 2020 (siehe Seite 238).
- [94] Amazon. *Arduino Nano*. 2020.
https://www.amazon.de/Moligh-doll-Atmega328-Kontroller-Arduino/dp/B07QMPPHPT/ref=sr_1_3?__mk_de_DE=%C3%85M%C3%85C5BD%C3%95C3%91&keywords=arduino+nano&qid=1577967107&s=diy&sr=1-3
besucht am 01. 01. 2020 (siehe Seite 239).
- [95] Arduino. *Arduino Forum*. 2020.
<https://forum.arduino.cc/index.php?topic=577916.0>
besucht am 01. 01. 2020 (siehe Seite 240).
- [96] Arduino. *Arduino*. 2019.
<http://www.arduino.cc>
besucht am 27. 12. 2019 (siehe Seite 242).
- [97] Daniel Garcia. *Fast, easy LED library for Arduino*. 2019.
<http://fastled.io/>
besucht am 27. 12. 2019 (siehe Seite 242).
- [98] LibreELEC. *Hypercon*. 2019.
<https://wiki.libreelec.tv/hypercon>
besucht am 27. 12. 2019 (siehe Seite 250).
- [99] Oracle. *JAVA Download*. 2019.
<http://www.java.com/en/>
besucht am 27. 12. 2019 (siehe Seite 250).
- [100] LibreELEC. *Hypercon*. 2020.
<https://libreelec.wiki/hypercon>
besucht am 01. 01. 2020 (siehe Seite 251).
- [101] KODI. *HOW-TO:HelloWorld addon*. 2019.
https://kodi.wiki/view/HOW-TO:HelloWorld_addon
besucht am 27. 12. 2019 (siehe Seite 253).
- [102] Amazon. *AZDelivery 433 MHz Funk - Sende und Empfänger Modul Set*. 2020.
https://www.amazon.de/AZDelivery-%E2%AD%90%E2%AD%90%E2%AD%90%E2%AD%90%E2%AD%90-433-MHz-Funk/dp/B01N5GV39I/ref=sr_1_7?__mk_de_DE=%C3%85M%C3%85C5BD%C3%95C3%91&keywords=433+mhZ+sender+empf%C3%A4nger&qid=1577969412&sr=8-7
besucht am 01. 01. 2020 (siehe Seite 256).

10. Server und Datenbanken

10.1	AirPrint	261
10.2	Servieren à la Python	267
10.3	Apache	268
10.4	CMS	273
10.5	DLNA	281
10.6	Philips Hue	285

Während sich die letzten Kapitel mit den multimedialen Fähigkeiten eines Wohnzimmer-PCs beschäftigt haben, wirft dieses Kapitel einen Blick auf Server und Datenbanken. Wir werden für beliebige Drucker einen AirPrint-Server einrichten, der dafür sorgt, dass Dokumente vom Mobilfunkgerät aus gedruckt werden können. Das ist besonders für ältere Drucker interessant, die diesen Service nicht an Bord haben. Weiterhin installieren wir einen DLNA-tauglichen Digital Media Server oder servieren Webseiten mit Hilfe von Wordpress.



10.1 AirPrint

Genau wie das weiter oben beschriebene *AirPlay*, ist auch *AirPrint* eine Erfindung eines im Silicon Valley ansässigen Computerkonzerns. Die Idee ist simpel und gut: kabellos drucken vom mobilen Gerät aus auf dem heimischen Drucker. *AirPrint* ist dabei nicht auf mobile Geräte beschränkt. Aber cool ist es schon, oder? Handy zücken, Email checken und den Anhang mal eben schnell ausdrucken. Dieser Abschnitt erklärt alle Schritte, die auf dem Wohnzimmer-PC erforderlich sind, damit dieser als *AirPrint*-Druckerserver arbeitet. Und das Beste dabei: Sie müssen nicht einmal einen *AirPrint*-fähigen Drucker besitzen, sondern benötigen lediglich einen von LINUX unterstützten Drucker; und das sind inzwischen die meisten. Wie so oft beginnen wir mit der Installation einiger erforderlicher Programmpakete. Das eine oder andere haben Sie sicher schon installiert.

Code-Ausschnitt 10.1: Installation des *AirPrint*-Servers

```
sudo apt-get install avahi-daemon cups cups-pdf python-cups foomatic-db foomatic-db-compressed-ppds samba-client
```




CUPS steht für *Common Unix Printing System*. *CUPS* ist ein frei nutzbares Drucksystem für UNIX-Systeme. Es wurde von Michael Sweet entwickelt, der inzwischen bei Apple angestellt ist. Damit hat Apple offiziell die Pflege und Weiterentwicklung von *CUPS* übernommen. *CUPS* selbst ist eine Server-Client-Struktur: Der Client sendet das zu druckende Dokument an den *CUPS*-Server. Dieser wiederum kümmert sich um den Ausdruck an dem Rechner, an dem der Drucker angeschlossen ist. Das bedeutet, Ihr Drucker muss selbst nicht am Wohnzimmer-PC angeschlossen sein, sondern kann sich an einem beliebigen Rechner in Ihrem Heimnetz (etwa an ein NAS angeschlossen) befinden. *CUPS* kann übrigens in einem Webbrowser konfiguriert werden. Nachdem wir *CUPS* installiert haben, müssen noch einige Änderungen an der Konfigurationsdatei vorgenommen werden, damit *CUPS* auch wirklich Drucker auf beliebigen Rechnern im Heimnetz adressieren kann. Öffnen Sie bitte die Datei `/etc/cups/cupsd.conf` in einem Editor Ihrer Wahl mit *root*-Rechten. Entfernen Sie ein mögliches #-Zeichen vor dem Port 631 und kommentieren sie *localhost* aus, so dass die entsprechende Zeile in der o. g. Datei so aussieht:

Code-Ausschnitt 10.2: Änderungen an der *CUPS*-Konfigurationsdatei

```
# Only listen for connections from the local machine.
# Listen localhost:631
Port 631
```

Ändern Sie die *Location*-Sektion wie folgt:

Code-Ausschnitt 10.3: Weitere Änderungen an der *CUPS*-Konfigurationsdatei

```
<Location />
  # Allow shared printing...
  Order allow,deny
  Allow @LOCAL
  Allow 10.*
</Location>
<Location /admin>
  Order allow,deny
  Allow @LOCAL
  Allow 10.*
</Location>
<Location /admin/conf>
  AuthType Default
  Require user @SYSTEM
  Order allow,deny
  Allow @LOCAL
  Allow 10.*
</Location>
```

Die Zeile *Allow Local* erlaubt dem lokalen Netzwerk den Zugriff auf den Drucker-Daemon. Die nächste Zeile *Allow 10.** erlaubt IP-Adressen, die mit 10 beginnen. Dieser Server vergibt dann IP-Adressen aus dem Adress-Raum 10.x.x.x. Durch den oben gezeigten Eintrag erlauben wir, die Heimdrucker aus der Ferne konfigurieren zu können, da der VPN-Server, den wir in Kapitel 5 installiert haben, diesen Adressraum nutzt. Da *CUPS* nach der Installation bereits gestartet wurde, muss der Dienst jetzt neu gestartet werden, damit unsere Änderungen an der Konfigurationsdatei Wirkung zeigen.

Code-Ausschnitt 10.4: Den *CUPS*-Dienst neu starten

```
sudo service cups restart
```

CUPS selbst läuft unter der Benutzergruppe *lpadmin*, was für *line printer administrator* steht, also Drucker-Administrator. Der Benutzer, unter dem Sie gerade arbeiten, gehört

dieser Gruppe nicht an. Damit der aktuelle Benutzer Einstellungen an den Druckern über *CUPS* ändern darf, muss er der Gruppe *lpadmin* angehören. Fügen wir ihn also zur Gruppe *lpadmin* hinzu:

Code-Ausschnitt 10.5: Benutzer in Gruppe der Drucker-Administratoren aufnehmen

```
sudo adduser name lpadmin
```

Ersetzen Sie in dem oben gezeigten Aufruf *name* bitte mit Ihrem Benutzernamen. Diesen finden Sie heraus, indem Sie den Befehl

Code-Ausschnitt 10.6: Eigenen Benutzernamen herausfinden

```
whoami
```

in einem Terminal eingeben.

*In manchen Ubuntu-Systemen ist der Benutzer, den Sie beim Installieren des Systems angelegt haben, bereits Mitglied der Gruppe *lpadmin*. In diesem Fall teilt Ihnen das System dieses mit.*



Wenn der Benutzer noch kein Mitglied der Gruppe *lpadmin* ist, bestätigt Ubuntu die erweiterte Gruppenzugehörigkeit mit

Code-Ausschnitt 10.7: Systemmeldung beim Hinzufügen neuer Benutzer zur Gruppe *lpadmin*

```
Füge Benutzer "name" der Gruppe "lpadmin" hinzu ...
Benutzer name wird zur Gruppe lpadmin hinzugefügt.
Fertig.
```

Der Befehl

Code-Ausschnitt 10.8: Auflisten der aktuellen *CUPS*-Einstellungen

```
cupscctl
```

ohne Parameter gibt die aktuellen *CUPS*-Einstellungen aus, die bei mir so aussehen:

Code-Ausschnitt 10.9: Aktuelle *CUPS*-Einstellungen

```
_debug_logging=0
_remote_admin=0
_remote_any=0
_share_printers=0
_user_cancel_any=0
BrowseLocalProtocols=dnssd
DefaultAuthType=Basic
JobPrivateAccess=default
JobPrivateValues=default
MaxLogSize=0
PageLogFormat=
SubscriptionPrivateAccess=default
SubscriptionPrivateValues=default
WebInterface=Yes
```

Unter Ubuntu sind also weder die lokalen Drucker im Netzwerk freigegeben, noch ist die Möglichkeit aktiviert, *CUPS* über das Netzwerk zu administrieren. Beides schalten wir nun mit dem Befehl

```
sudo cupscctl --share-printers --remote-admin
```

frei. Nach einem Neustart von *CUPS* steht hinter *_remote_admin* und hinter *_share_printers=* eine 1 und die entsprechenden Freigaben sind erteilt.

Die Konfiguration des Druckers nehmen wir über ein Webinterface vor. Dazu rufen wir im Browser die IP-Adresse des Wohnzimmer-PCs für den Port 631 auf. Auf diesem Port lauscht nämlich *CUPS*.

Code-Ausschnitt 10.10: Das Webinterface von *CUPS* öffnen

```
https://ip-adresse:631
```



Sollten Sie von einem anderen Rechner aus auf *CUPS* zugreifen, dessen IP-Adresse nicht in der Datei `cupsd.conf` erlaubt ist - etwa weil Sie für VPN einen anderen Adressraum eingetragen haben - erhalten Sie beim Aufruf der *CUPS*-Webseite lediglich ein „Verboten“.

Auf meinem Windows-PC unter Firefox sieht die *CUPS*-Oberfläche so aus wie in Abb. 10.1.

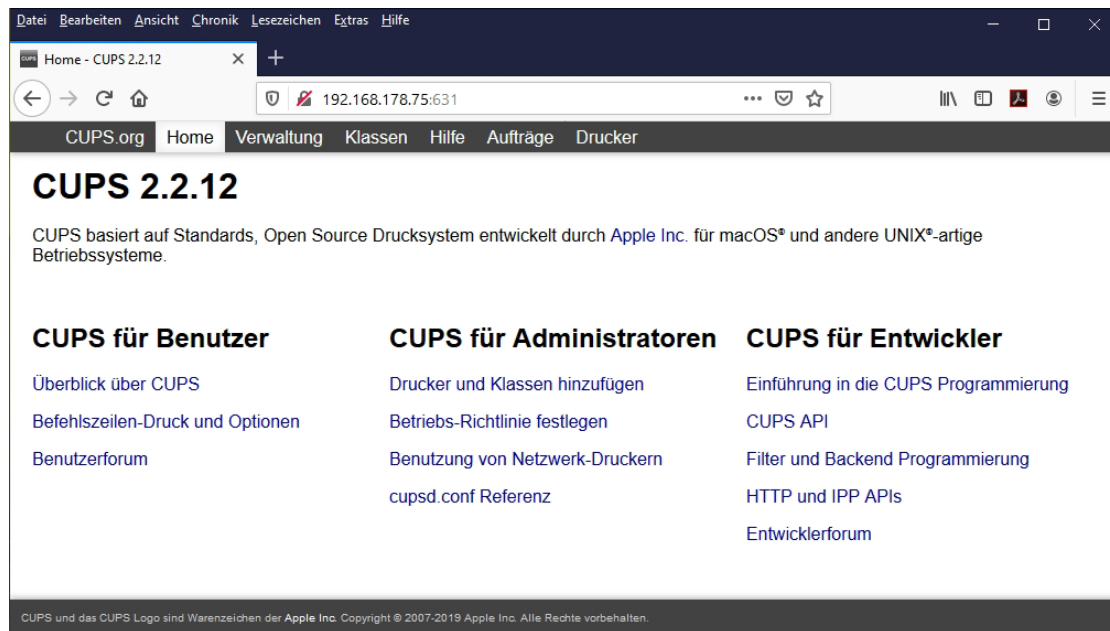


Abbildung 10.1: Der *CUPS*-Startbildschirm im Webbrowser.

Wählen Sie im Menü bitte den Punkt *Verwaltung* aus. Unter Server-Einstellungen wählen Sie bitte die folgenden Punkte an:

- Mit diesem System verbundene Drucker freigeben
- Fernwartung zulassen

Entscheiden Sie selbst, ob Sie das Drucken vom Internet aus erlauben möchten oder nicht. Wenn alle Einstellungen vorgenommen sind, klicken Sie bitte auf *Einstellungen ändern*. Der *CUPS*-Server wird neu gestartet.

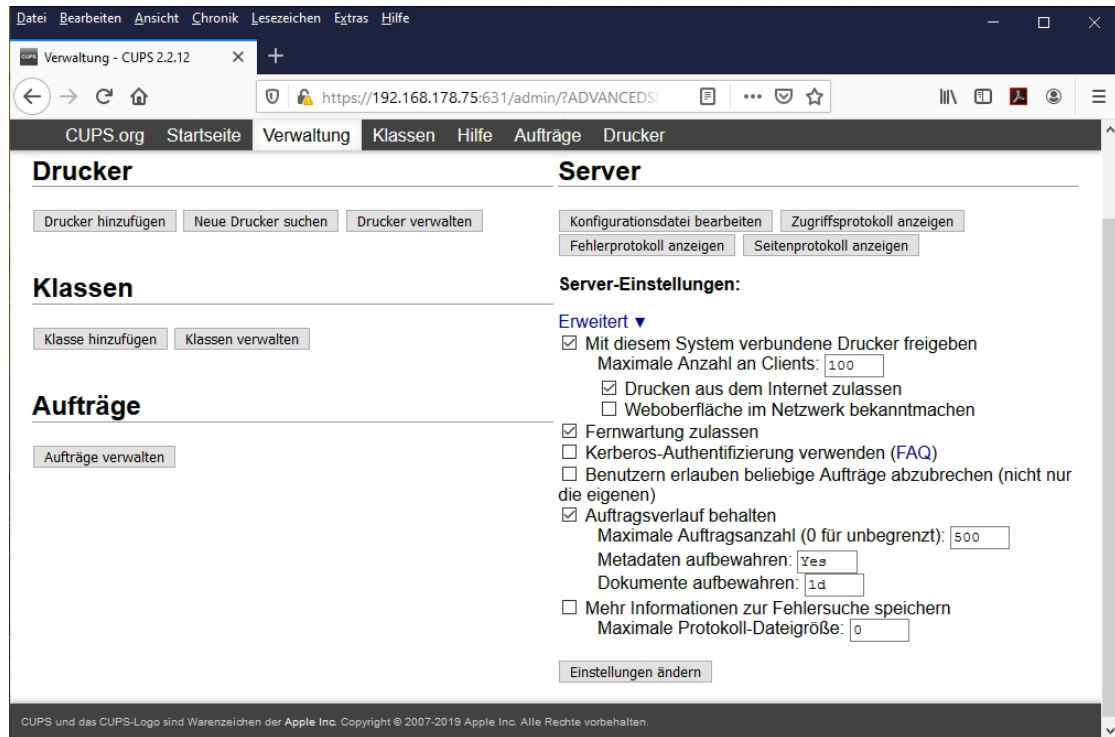


Abbildung 10.2: Die CUPS-Verwaltung im Webbrowser.

Sie haben nun zwei Möglichkeiten, fortzufahren:

1. Der Drucker steckt in einem USB-Anschluss des Wohnzimmer-PCs. Schalten Sie in diesem Fall bitte den Drucker ein.
2. Der Drucker befindet sich in Ihrem Netzwerk und ist an einen anderen Rechner angeschlossen. Schalten Sie in diesem Fall bitte den anderen Rechner und den Drucker ein.

Wählen Sie im Verwaltungs-Dialog (Abb. 10.1) *Verfügbare Drucker auflisten*. CPUS zeigt Ihnen nun alle Drucker, die gefunden werden. Klicken Sie bitte links neben dem Druckernamen auf *Diesen Drucker hinzufügen*, passen Sie die Beschreibung für Ort an und aktivieren Sie *Diesen Drucker freigeben*. Danach klicken Sie bitte auf *Weiter*. Daraufhin müssen Sie den Druckerhersteller und das Modell angeben. Der Drucker ist fertig eingerichtet. Das können Sie durch Drucken einer Testseite überprüfen.

Sollte der gewünschte Drucker nicht unter Verfügbare Drucker auftauchen, klicken Sie im Verwaltungs-Menü auf Drucker hinzufügen (Abb. 10.2). Sie können dann in einem weiteren Menü den Netzwerkdrucker und sein Protokoll selbst eintragen.



Nachdem alle Drucker im Drucker-Server CUPS hinzugefügt worden sind, kümmern wir uns jetzt um *AirPrint*. Wir installieren *AirPrint* im */opt*-Verzeichnis.

Code-Ausschnitt 10.11: Installation von *AirPrint*

```

sudo mkdir /opt/airprint
cd /opt/airprint
sudo wget -O airprint-generate.py --no-check-certificate https://raw.githubusercontent.com/tjfontaine/↵
    airprint-generate/master/airprint-generate.py
sudo chmod +x airprint-generate.py
sudo ./airprint-generate.py -d /etc/avahi/services
sudo service cups restart
sudo service avahi-daemon restart

```

Nach der Installation müssen *CUPS* und *Avahi* neu gestartet werden.



Mit Hilfe von *Avahi* können Geräte im lokalen Netzwerk miteinander vernetzt werden, ohne dass diese manuell konfiguriert werden müssen.

Laut Autor der *AirPrint*-Software kann es vorkommen, dass bei den oben angegebenen Befehlen eine Fehlermeldung auftaucht:

Code-Ausschnitt 10.12: Mögliche Fehlermeldung bei der Installation von *airprint*

```

image/urf is not in mime types, [PRINTER_NAME] may not be available on ios6 (see https://github.com/tjfontaine/airprint-generate/issues/5)

```

In diesem Fall gehen Sie bitte so vor: Erzeugen Sie als Benutzer *root* die Datei */usr/share/cups/mime/airprint.types* mit dem folgenden Inhalt:

Code-Ausschnitt 10.13: Die Datei *airprint.types* anlegen

```

#
# "$Id: $"
#
# AirPrint type
image/urf urf string(0,UNIRAST<00>)
#
# End of "$Id: $".
#

```

Erzeugen Sie weiterhin die Datei */usr/share/cups/mime/airprint.convs*, ebenfalls als Benutzer *root* - denken Sie also daran, dem Editor Ihrer Wahl ein *sudo* voran zu stellen. In diese Datei kopieren Sie bitte folgenden Inhalt:

Code-Ausschnitt 10.14: Die Datei *airprint.convs* anlegen

```

# "$Id: $"
#
# AirPrint
# Updated list with minimal set 25 Sept
image/urf application/pdf 100 pdftoraster
#
# End of "$Id: $".
#

```

Danach muss das *AirPrint*-Skript erneut ausgeführt werden und *CUPS* und *Avahi* müssen neu gestartet werden:

Code-Ausschnitt 10.15: *Avahi* und *CUPS* neu starten

```

sudo ./airprint-generate.py -d /etc/avahi/services
sudo service cups restart
sudo service avahi-daemon restart

```

10.2 Servieren à la Python

Darf ich Ihnen den kürzesten Webserver-Code der Welt vorstellen?

Code-Ausschnitt 10.16: Der kürzeste Webserver-Quelltext der Welt

```
import cherrypy

class HalloWelt(object):
    @cherrypy.expose
    def index(self):
        return "Hallo Welt!"

cherrypy.quickstart(HalloWelt())
```



(Quelle: Rolando Murillo)

Das glauben Sie nicht? Wir probieren es gleich aus. In diesem Abschnitt werden wir einen kleinen, minimalistischen Webserver mit Python aufsetzen. Danach schauen wir uns den großen Bruder an, *apache2*. Der oben gezeigte Webserver erwartet ein installiertes Python-*cherry3*-Modul, das sich so installieren lässt:

Code-Ausschnitt 10.17: Installation des *cherry3*-Moduls

```
sudo apt-get install python-cherry3
```

Erstellen Sie ein Verzeichnis */python*. Wechseln Sie in dieses Verzeichnis und editieren Sie eine Datei namens *server.py*, in die Sie den oben gezeigten minimalistischen Webserver-Quelltext hinein kopieren. Speichern Sie die Datei und starten Sie sie mit *python*.

Code-Ausschnitt 10.18: Python Webserver-Beispiel erstellen

```
cd ~
mkdir python
cd python
vi server.py
python ./server.py
```

Bei mir erscheint daraufhin die folgende Ausgabe:

Code-Ausschnitt 10.19: Ausgabe des minimalistischen Python-Webservers

```
[12/Dec/2019:11:54:59] ENGINE Listening for SIGHUP.
[12/Dec/2019:11:54:59] ENGINE Listening for SIGTERM.
[12/Dec/2019:11:54:59] ENGINE Listening for SIGUSR1.
[12/Dec/2019:11:54:59] ENGINE Bus STARTING
CherryPy Checker:
The Application mounted at '' has an empty config.

[12/Dec/2019:11:54:59] ENGINE Started monitor thread 'Autoreloader'.
[12/Dec/2019:11:54:59] ENGINE Started monitor thread '_TimeoutMonitor'.
[12/Dec/2019:11:54:59] ENGINE Serving on http://127.0.0.1:8080
[12/Dec/2019:11:54:59] ENGINE Bus STARTED
```

Rufen Sie einen Browser auf Ihrem Wohnzimmer-PC auf und geben Sie dort bitte die Adresse *http://localhost:8080* ein. 8080 ist der Port, auf dem unser Minimal-Server läuft. Mein Firefox liefert das Bild, das ich Ihnen in der nächsten Abb. zeige. Wenn Sie den Server von einem anderen Rechner im selben Netz aufrufen, wird dieser vorerst nicht funktionieren, da er an die Adresse 127.0.0.1, also den *localhost*, gebunden ist und nicht an die IP-Adresse des Wohnzimmer-PCs. Wenn Sie das ändern möchten, ergänzen Sie bitte die beiden vorletzten Zeilen in der Datei *server.py* so:



Abbildung 10.3: So sieht der minimalistische CherryPy-Server auf Firefox aus.

Code-Ausschnitt 10.20: Python-Webserver mit IP-Adresse und Port

```
...
cherry.py.server.socket_host = '192.168.178.75'
cherry.py.server.socket_port = 1111
cherry.py.quickstart(HalloWelt())
```

Ändern Sie meine IP-Adresse bitte in die Ihres Wohnzimmer-PCs um und setzen Sie einen beliebigen Port. Nach dem Start des Servers können Sie diesen dann von überall in Ihrem Heimnetz aus unter dem angegebenen Port erreichen.

10.3 Apache

„LINUX is like an Indian tent: No Windows, no Gates and an Apache inside“. Übersetzen könnte man das mit „LINUX ist wie ein Indianerzelt: keine Fenster, keine Tore und ein Apache sitzt darin“. Man kann diesen Satz aber auch auf ein Betriebssystem und den Firmengründer des Microsoft-Imperiums beziehen. Den „Apachen“ schauen wir uns in diesem Abschnitt genauer an. Die Installation ist schnell erledigt.



Code-Ausschnitt 10.21: Installation des Webservers *apache2*

```
sudo apt-get install apache2
```



Der Apache-Webserver ist ebenfalls ein Dienst, welcher unter /etc/init.d/apache2 zu Hause ist. Nach der Installation wird apache2 automatisch gestartet.

Es gibt einige Alternativen zu *apache2*, etwa *lighttpd* oder *LiteSpeed*. Die erste Alternative ist in Maik Schmidts Buch ausführlich beschrieben [103]. Ich habe mich für *Apache* entschieden, weil wir im späteren Verlauf noch einen Blick auf das Content Management System Wordpress werfen, das perfekt mit *apache* harmoniert. Aber dazu später mehr. Jetzt unterziehen wir den *Apachen* einem ersten Test. Rufen Sie in einem Browser, der Zugriff auf Ihr Netzwerk hat, die IP-Adresse des Wohnzimmer-PCs auf (<http://ip-adresse>). Der Browser sollte nun die *Apache* Standard-Seite anzeigen (Abb. 10.4).

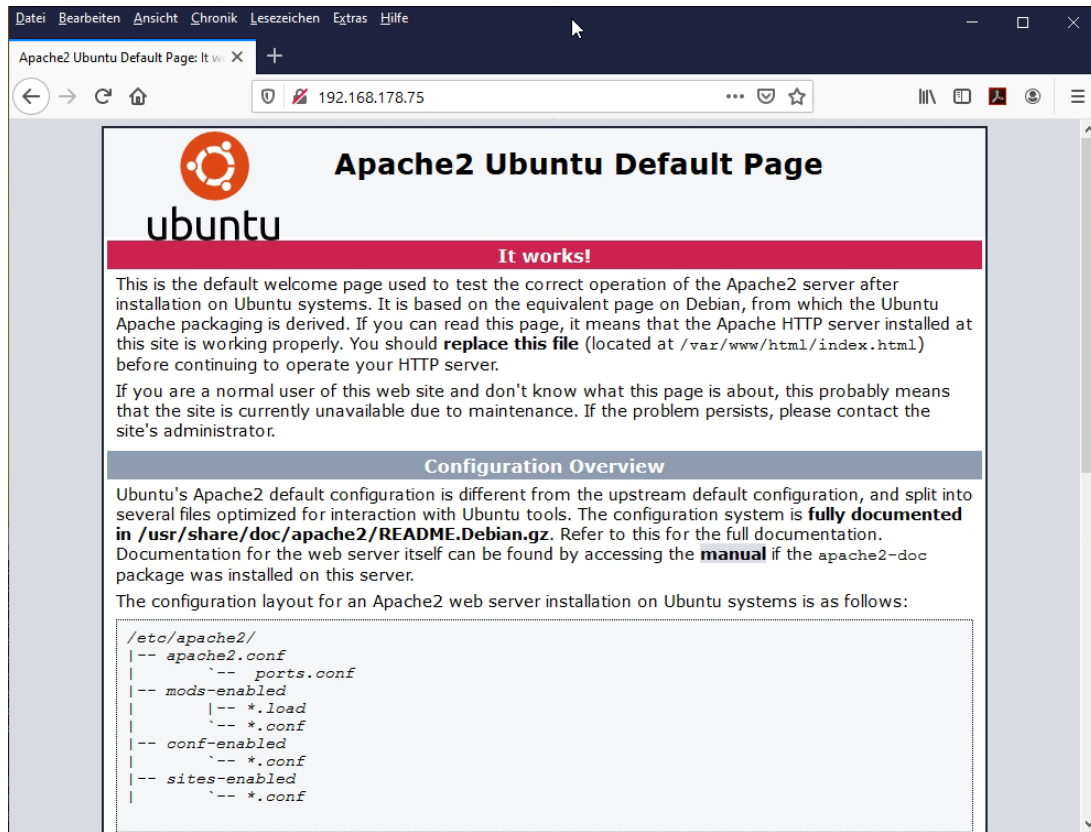


Abbildung 10.4: Die Apache Standard-Seite sagt „Hallo!“.

Wenn Sie den Browser auf dem Wohnzimmer-PC selbst starten, können Sie über localhost auf den eigenen Rechner zugreifen. Das ersetzt dann die lokale IP-Adresse.



Der Apache schaut im Verzeichnis `/var/www/html` nach vorhandenen Webseiten. Dort liegt nach der Installation auch die Datei `index.html`, welche den in Abb. 10.4 angezeigten Text liefert.

Die Endung `.html` bezeichnet eine HTML-Datei (HyperText Markup Language). Hierbei handelt es sich um eine Beschreibungssprache, die festlegt, wie ein im Internet-Browser dargestellter Inhalt aussieht.



10.3.1 PHP

Viele Webseiten werden mit Hilfe von *PHP* erstellt.



Die Endung *.php* bezeichnet eine *PHP*-Datei (*PHP Hypertext Preprocessor*). Hierbei handelt es sich um eine Skript-Programmiersprache, die sehr häufig beim Erstellen von Internetseiten verwendet wird. Die Sprache wird auf dem Server abgearbeitet und liefert dann z. B. die Darstellung eines Internet-Auftritts.

Da wir *PHP* im weiteren Verlauf noch benötigen, installieren wir es in der aktuellen Version 7.3.

Code-Ausschnitt 10.22: Installation von *php*

```
sudo apt-get install php7.3 libapache2-mod-php7.3
```

Kleines Beispiel gefällig? Editieren Sie bitte eine Datei */var/www/html/phpinfo.php* unter Verwendung eines vorangestellten *sudo* mit folgendem Inhalt:

Code-Ausschnitt 10.23: Minimales *php*-Skript

```
<?php
phpinfo();
?>
```

Rufen Sie dann in einem Browser die URL `http://ip-adresse/phpinfo.php` auf.

System	Linux rudi 5.3.0-24-generic #26-Ubuntu SMP Thu Nov 14 01:33:18 UTC 2019 x86_64
Build Date	Oct 24 2019 11:38:49
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/15-xml.ini, /etc/php/7.3/apache2/conf.d/20-bz2.ini, /etc/php/7.3/apache2/conf.d/20-calendar.ini, /etc/php/7.3/apache2/conf.d/20-curl.ini, /etc/php/7.3/apache2/conf.d/20-dom.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /etc/php/7.3/apache2/conf.d/20-fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-gd.ini, /etc/php/7.3/apache2/conf.d/20-gettext.ini, /etc/php/7.3/apache2/conf.d/20-iconv.ini, /etc/php/7.3/apache2/conf.d/20-json.ini, /etc/php/7.3/apache2/conf.d/20-mbstring.ini, /etc/php/7.3/apache2/conf.d/20-mysqli.ini, /etc/php/7.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.3/apache2/conf.d/20-phar.ini, /etc/php/7.3/apache2/conf.d/20-posix.ini, /etc/php/7.3/apache2/conf.d/20-readline.ini, /etc/php/7.3/apache2/conf.d/20-shmop.ini, /etc/php/7.3/apache2/conf.d/20-simplexml.ini, /etc/php/7.3/apache2/conf.d/20-sockets.ini, /etc/php/7.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.3/apache2/conf.d/20-sysvsem.ini, /etc/php/7.3/apache2/conf.d/20-sysvshm.ini, /etc/php/7.3/apache2/conf.d/20-tokenizer.ini, /etc/php/7.3/apache2/conf.d/20-wddx.ini, /etc/php/7.3/apache2/conf.d/20-xmlreader.ini, /etc/php/7.3/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.3/apache2/conf.d/20-xsl.ini, /etc/php/7.3/apache2/conf.d/20-zip.ini

Abbildung 10.5: Die *PHP*-Infoseite listet alle Standard-Einstellungen auf.

10.3.2 MySQL

Noch sind wir nicht am Ende unserer Installationen angelangt. Ein wichtiger Baustein für das Projekt fehlt noch: die Datenbank. Die Datenbanken werden wir im weiteren Verlauf dieses Kapitels noch benötigen, um Wordpress-Webseiten zu servieren. Viele bezeichnen die Open Source-Datenbank *MySQL* als die populärste der Welt; sie stellt die Grundlage unzähliger Webauftritte dar. Ähnlich wie LEGO beinhaltet der Name *MySQL* den Vornamen der Tochter eines Gründers (*My*) und die Abkürzung *SQL*, die für *Structured Query Language* (Strukturierte Anfrage Sprache) steht. Die Übersetzung trifft die Daseins-Berechtigung von *MySQL* genau. In der Datenbank stehen Einträge. Anfragen an die Datenbank werden mit entsprechenden Einträgen (oder auch keinen, wenn sie nicht vorhanden sind) beantwortet. Inzwischen ist *mysql* aber in einigen Bereichen von *mariadb* abgelöst worden. Das Internet-Lexikon Wikipedia wird mit Hilfe von *mariadb* serviert. *mariadb* war ursprünglich ein eigener Fork der Datenbank *mysql*, hat sich aber inzwischen stark weiterentwickelt. Bereits im VDR-Kapitel habe ich die Installation von *mariadb* empfohlen, weil diese Datenbank noch für *mysql* veraltete Syntax unterstützt, die beispielsweise noch von einigen VDR-Plugins verwendet wird. Daher empfehle ich auch hier die Installation von *mariadb*, die wie folgt vonstatten geht:



Abbildung 10.6:
(Quelle:
mysql.de).

Code-Ausschnitt 10.24: Installation der Datenbank *mariadb*

```
apt-get install mariadb-server libmariadbclient-dev php7.3-mysql
```

mariadb steht nach der Installation unter dem Aufrufnamen *mysql* zur Verfügung.

MySQL ist auch ein Dienst. Sie erinnern sich noch... `/etc/init.d/mysql`.



Abbildung 10.7:
(Quelle: phpmyadmin.net).

Das Programm *phpmyadmin* ist eine komfortable Web-Oberfläche zur Verwaltung der *MySQL*-Datenbank. Hier können Sie neue Datenbanken anlegen, Rechte vergeben oder Einträge erstellen und löschen. Ich habe es schon oft als Hilfsmittel gebraucht, wenn ich schnell in einer Datenbank etwas nachschauen wollte und keine Lust auf Konsolen-Befehle hatte.

Üblicherweise kann man *phpMyAdmin* mit allen erforderlichen Modulen so installieren:

Code-Ausschnitt 10.25: Installation von *phpMyAdmin*

```
sudo apt-get install libapache2-mod-auth-pgsql php7.3-mysql phpmyadmin
```

Die *phpMyAdmin*-Version für Ubuntu 19.10 zeigte allerdings noch einige Probleme und hat es nicht in diese Distribution geschafft. Für dieses System ist es daher erforderlich, *phpMyAdmin* auf einem anderen Repository so zu installieren:

Code-Ausschnitt 10.26: Die Installation von *phpMyAdmin* unter Ubuntu 19.10

```
sudo add-apt-repository ppa:phpmyadmin/ppa
sudo apt-get update
sudo apt-get install phpmyadmin
```

Diese Installation fügt zunächst einmal ein neues Repository hinzu *ppa:phpmyadmin/ppa*, indem es die Datei *phpmyadmin-ubuntu-ppa-eoan.list* im Verzeichnis */etc/apt/sources.list.d* anlegt mit dem folgenden Inhalt:

Code-Ausschnitt 10.27: *phpMyAdmin*-Repository unter Ubuntu 19.10

```
deb http://ppa.launchpad.net/phpmyadmin/ppa/ubuntu eoan main
# deb-src http://ppa.launchpad.net/phpmyadmin/ppa/ubuntu eoan main
```

Während der Installation werden Sie mit einigen Fragen konfrontiert. Wählen Sie als Frontend für *phpMyAdmin* *apache2* aus (und nicht *lighttpd*). Seit *MySQL* 5.7 hat sich das Sicherheitsmodell der Datenbank geändert: Der *MySQL*-Root-Zugang per Terminal erfordert jetzt ein vorangestelltes *sudo*. Das bedeutet, dass *phpMyAdmin* den Benutzer *root* nicht erlaubt. Das Problem kann einfach gelöst werden, indem man einen Benutzer-Zugang zur Datenbank erstellt und diesen mit den erforderlichen Privilegien ausstattet. Dazu verbinden wir uns zunächst in einem Terminal mit der Datenbank.

Code-Ausschnitt 10.28: Die Datenbank per Terminal öffnen

```
sudo mysql --user=root mysql
```

Nun erstellen wir einen Benutzer für *phpMyAdmin* mit dem Namen *phpMyAdmin* und einem sicheren Kennwort.

Code-Ausschnitt 10.29: Einen neuen Benutzer zur Datenbank hinzufügen

```
CREATE USER 'phpmyadmin'@'localhost' IDENTIFIED BY 'kennwort';
GRANT ALL PRIVILEGES ON *.* TO 'phpmyadmin'@'localhost' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

Bitte ersetzen Sie *kennwort* im oben gezeigten Quelltext durch ein sicheres Kennwort. Wenn Sie die Datenbank mit Hilfe von *phpMyAdmin* nur auf dem lokalen Rechner verbinden möchten, reichen diese Befehle aus. Möchten Sie darüber hinaus Remote-Verbindungen zulassen, rufen Sie bitte noch den folgenden Befehl auf:

Code-Ausschnitt 10.30: Zulassen von Remote-Verbindungen zur Datenbank

```
CREATE USER 'phpmyadmin'@'%' IDENTIFIED BY 'kennwort';
GRANT ALL PRIVILEGES ON *.* TO 'phpmyadmin'@'%' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

Denken Sie aber bitte daran, dass ein solcher Zugang von außen ein Sicherheitsrisiko darstellen könnten. Danach können Sie sich mit einem

Code-Ausschnitt 10.31: Schließen der Datenbank

```
quit
```

von der Datenbank abmelden.

Im letzten Schritt editieren Sie bitte die Datei */etc/dbconfig-common/phpmyadmin.conf* und fügen Sie den Datenbankbenutzer *phpMyAdmin* und das oben vergebene Kennwort (*kennwort*) hinzu, sofern diese Einträge noch nicht vorhanden sind.

Code-Ausschnitt 10.32: Datenbankbenutzer für *phpMyAdmin* hinzufügen

```
# dbc_dbuser: database user
#     the name of the user who we will use to connect to the database.
dbc_dbuser='phpmyadmin'
# dbc_dbpass: database user password
#     the password to use with the above username when connecting
#     to a database, if one is required
dbc_dbpass='kennwort'
```

Um die fertige Installation zu testen, rufen Sie bitte `http://ip-adresse/phpmyadmin` in Ihrem Lieblings-Browser auf. Wählen Sie sich mit dem Benutzernamen *phpMyAdmin* und dem eben vergebenen Kennwort ein. Daraufhin sollten Sie die *phpMyAdmin*-Oberfläche sehen (Abb. 10.8).

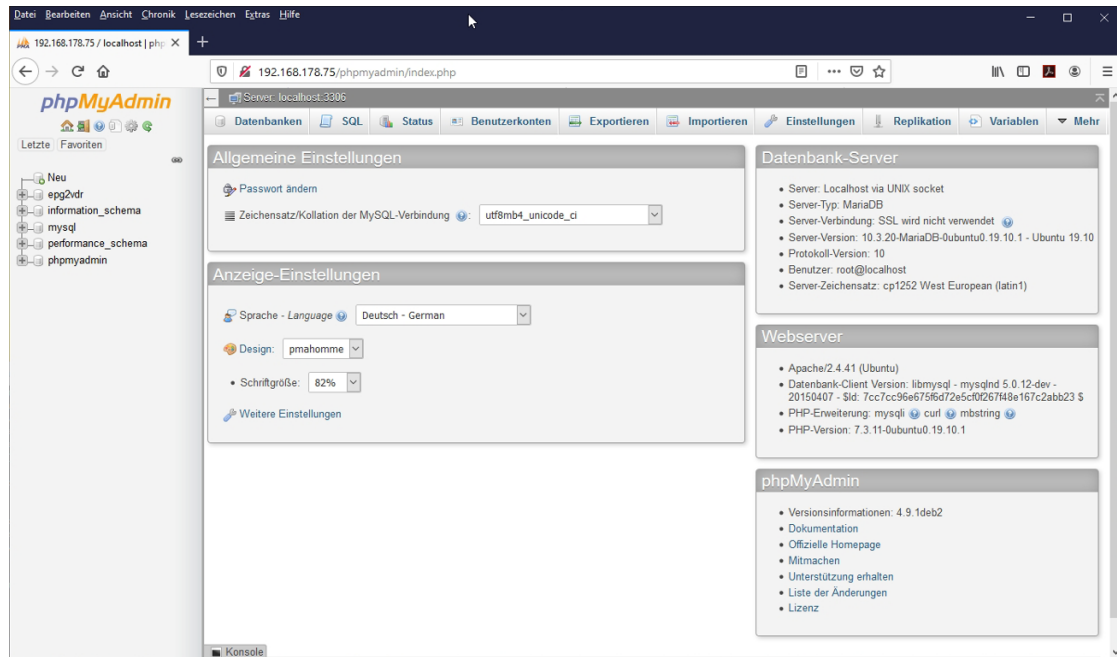


Abbildung 10.8: Die *phpMyAdmin*-Oberfläche verwaltet die *MySQL*-Datenbank.

Sollte Ihr Browser die *phpMyAdmin*-Oberfläche nicht anzeigen, legen Sie bitte den folgenden symbolischen Link an:



```
sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-enabled/phpmyadmin.conf
```

Danach muss der Apache-Dienst neu gestartet werden.

In meinem Beispiel (Abb. 10.8) sehen Sie unter anderem auch die im *VDR*-Kapitel angelegte Datenbank *epg2vdr*.

10.4 CMS

Content Management Systeme (CMS) generieren Internetseiten aus einer Datenbank heraus. Möchte man den Inhalt einer Webseite ändern, ändert man den Inhalt der Datenbank und damit ist der Inhalt der Webseite automatisch geändert. In diesem Abschnitt zeige ich Ihnen, wie Sie das CMS *WordPress* installieren können. Dabei gehe ich davon aus, dass Sie *mariadb* bereits wie im Kapitel *VDR* (7) beschrieben, installiert haben und dass *php* ebenfalls läuft. Falls Sie eine neuere oder auch ältere Version von *php* installiert haben, ändern Sie bitte in den folgenden Aufrufen die Versionsnummer 7.3 zu der bei Ihnen installierten Version um.

Editieren Sie in einem ersten Schritt die Datei `/etc/php/7.3/apache2/php.ini` und stellen Sie sicher, dass die folgenden Einträge vorhanden sind:

Code-Ausschnitt 10.33: Einstellungen für die Datei *php.ini*

```
file_uploads = On
allow_url_fopen = On
memory_limit = 256M
upload_max_filesize = 100M
max_execution_time = 360
```

WordPress benötigt eine Datenbank, damit es arbeitet. Diese Datenbank legen wir jetzt zusammen mit einem neuen Datenbank-Benutzer an. Die Datenbank heißt *wpdb*, was für *WordPress*-Datenbank steht. Der Benutzer heißt *wpdbuser*. Das steht für *WordPress*-Datenbank-Nutzer. Loggen Sie sich zunächst so in die Datenbank ein:

Code-Ausschnitt 10.34: In die Datenbank einwählen

```
sudo mysql -u root -p
```

Danach setzen Sie bitte die folgenden Befehle ab:

Code-Ausschnitt 10.35: Neue Datenbank und neuen Benutzer erstellen

```
CREATE DATABASE wpdb;
CREATE USER 'wpdbuser'@'localhost' IDENTIFIED BY 'kennwort';
GRANT ALL ON wpdb.* TO 'wpdbuser'@'localhost' IDENTIFIED BY 'kennwort' WITH GRANT OPTION;
FLUSH PRIVILEGES;
EXIT;
```

Ersetzen Sie bitte in den oben gezeigten Aufrufen *kennwort* durch ein Kennwort Ihrer Wahl. Ubuntu bietet *WordPress* zwar in seinem Repository an, wir werden aber dennoch die neueste Version vom *WordPress*-Server herunterladen und installieren. Nach dem Herunterladen packen wir *WordPress* aus, erstellen ein Verzeichnis *beispiel.de*, in dem wir unsere Seite hosten, und kopieren *WordPress* dort hinein.

Code-Ausschnitt 10.36: *WordPress* in der neuesten Version installieren

```
mkdir ~/tmp
cd ~/tmp
wget https://wordpress.org/latest.tar.gz
tar -zxvf latest.tar.gz
cd wordpress
sudo mkdir /var/www/html/beispiel.de
sudo cp -R * /var/www/html/beispiel.de/
```

Nach der erfolgreichen Installation von *WordPress* in dem Verzeichnis, in dem die neue Webseite angelegt werden soll, teilen wir das nun dem *Apache2*-Server. Editieren Sie dafür bitte eine Datei mit dem Namen *wordpress.conf* als Benutzer *root* (vorangestelltes *sudo* im Verzeichnis */etc/apache2/sites-available/wordpress.conf*) mit dem folgenden Inhalt:

Code-Ausschnitt 10.37: Apache2 die neue *WordPress*-Seite mitteilen

```
<VirtualHost *:80>
    ServerAdmin admin@example.com
    DocumentRoot /var/www/html/beispiel.de/
    ServerName beispiel.de
    ServerAlias www.beispiel.de

    <Directory /var/www/html/beispiel.de/>
        Options +FollowSymlinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Nach Anlegen der oben gezeigten Konfiguration teilen wir diese dem Apache nun mit:

Code-Ausschnitt 10.38: Konfiguration der neuen Webseite für Apache übernehmen

```
sudo a2ensite wordpress.conf
sudo a2enmod rewrite
```

Anschließend muss Apache2 neu gestartet werden:

Code-Ausschnitt 10.39: Apache2 neu starten

```
sudo systemctl restart apache2.service
```

Bevor *WordPress* ein erstes Mal gestartet werden kann, müssen der *WordPress*-Konfigurationsdatei noch die neuen Datenbank-Einträge mitgeteilt werden. Das geschieht in der Datei *wp-config.php*, die Sie unter */var/www/html/beispiel.de* finden. In einem ersten Schritt kopieren wir eine Beispieldatei mit diesem Namen:

Code-Ausschnitt 10.40: Neue *WordPress*-Konfigurationsdatei anlegen

```
sudo mv /var/www/html/beispiel.de/wp-config-sample.php /var/www/html/beispiel.de/wp-config.php
```

Bei mir lag die Datei, die ich mir aus der Beispielkonfiguration kopiert habe, im DOS-Format vor. Das kann man an den Sonderzeichen am Ende jeder Zeile erkennen. Das Programm dos2unix kann dazu verwendet werden, das Format einer DOS-Datei in das Format einer UNIX-Datei zu überführen. Installieren Sie dieses Programm so:

Code-Ausschnitt 10.41: Das Programm *dos2unix* installieren

```
sudo apt install dos2unix
```

und rufen Sie es anschließend so auf:

Code-Ausschnitt 10.42: Datei vom DOS- in das UNIX-Format umwandeln

```
sudo dos2unix /var/www/html/beispiel.de/wp-config.php
```

Editieren Sie die eben kopierte Datei ebenfalls wieder mit Root-Rechten und ergänzen Sie den folgenden Inhalt:

Code-Ausschnitt 10.43: Datenbankdetails in die *WordPress*-Konfiguration eintragen

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wpdb');

/** MySQL database username */
define('DB_USER', 'wpdbuser');

/** MySQL database password */
define('DB_PASSWORD', 'kennwort');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
```

Ersetzen Sie wieder *kennwort* mit dem Kennwort, das Sie beim Anlegen der Datenbank vergeben haben.

Nach dieser Vorarbeit erfolgen die restlichen Konfigurations-Schritte bequem im Browser. Rufen Sie dazu bitte die Beispielseite wie folgt in einem Browser Ihrer Wahl auf und verwenden Sie dabei die IP-Adresse Ihres Wohnzimmer-PCs anstelle von meiner IP-Adresse:

Code-Ausschnitt 10.44: Konfiguration von *WordPress* im Browser abschließen

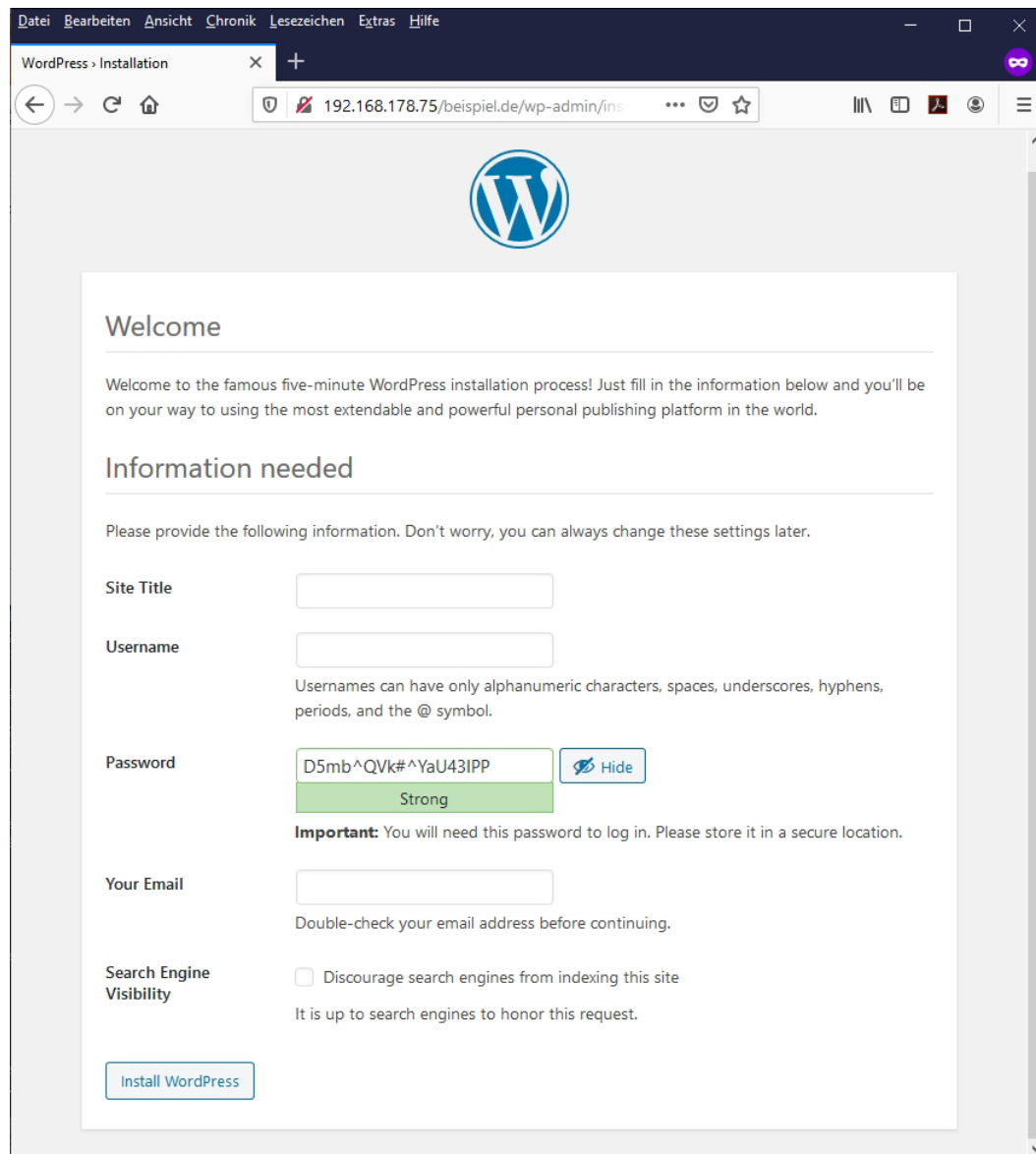
<http://192.168.178.75/beispiel.de/>

Unter dem Feld *Site Title* tragen Sie den Namen ein, den Ihre Webseite später haben soll. Keine Sorge, falls Sie sich später umentscheiden möchten, ist das kein Problem. Der Name der Webseite kann jederzeit geändert werden. Vergeben Sie anschließend einen Benutzernamen (*Username*). Dieser Benutzer mit dem unter *Password* angegebenen Kennwort hat die Möglichkeit, sich in das *WordPress*-Backend einzuloggen und Inhalte zu ändern. Es kann entweder das vorgegebene Kennwort verwendet werden oder ein selbst vergebenes.



Falls Sie Ihre Seite später im Internet veröffentlichen möchten, vergeben Sie bitte ein schwer zu erratendes Kennwort.

Geben Sie weiterhin Ihre E-Mail-Adresse an und klicken Sie dann auf *Install WordPress*. Sollten während der Installation keine Fehler aufgetreten sein, teilt *WordPress* das unter *Sucess!* mit.



The screenshot shows a web browser window with the title "WordPress > Installation". The address bar displays "192.168.178.75/beispiel.de/wp-admin/ins". The page features the WordPress logo at the top. Below it, a "Welcome" section states: "Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world." The "Information needed" section follows, with the instruction: "Please provide the following information. Don't worry, you can always change these settings later." The form includes fields for "Site Title", "Username", "Password", and "Your Email". The "Password" field shows a generated password "D5mb^QV#^YaU43IPP" with a "Hide" button and a "Strong" strength indicator. Below the password field, an "Important" note reads: "You will need this password to log in. Please store it in a secure location." The "Your Email" field has a note: "Double-check your email address before continuing." At the bottom, there is a checkbox for "Search Engine Visibility" with the text "Discourage search engines from indexing this site" and "It is up to search engines to honor this request." An "Install WordPress" button is located at the bottom left of the form.

WordPress > Installation

192.168.178.75/beispiel.de/wp-admin/ins

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username

Username can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password

Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email

Double-check your email address before continuing.

Search Engine Visibility

☐ Discourage search engines from indexing this site
It is up to search engines to honor this request.

Abbildung 10.9: Die weitere Konfiguration von *WordPress* erfolgt aus dem Browser heraus.

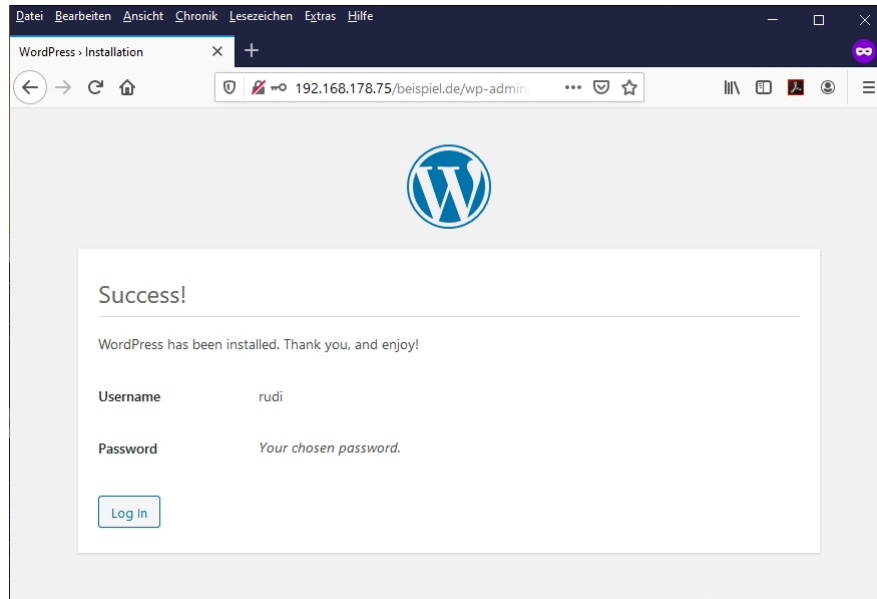


Abbildung 10.10: Nach erfolgreicher Installation teilt *WordPress* das mit.

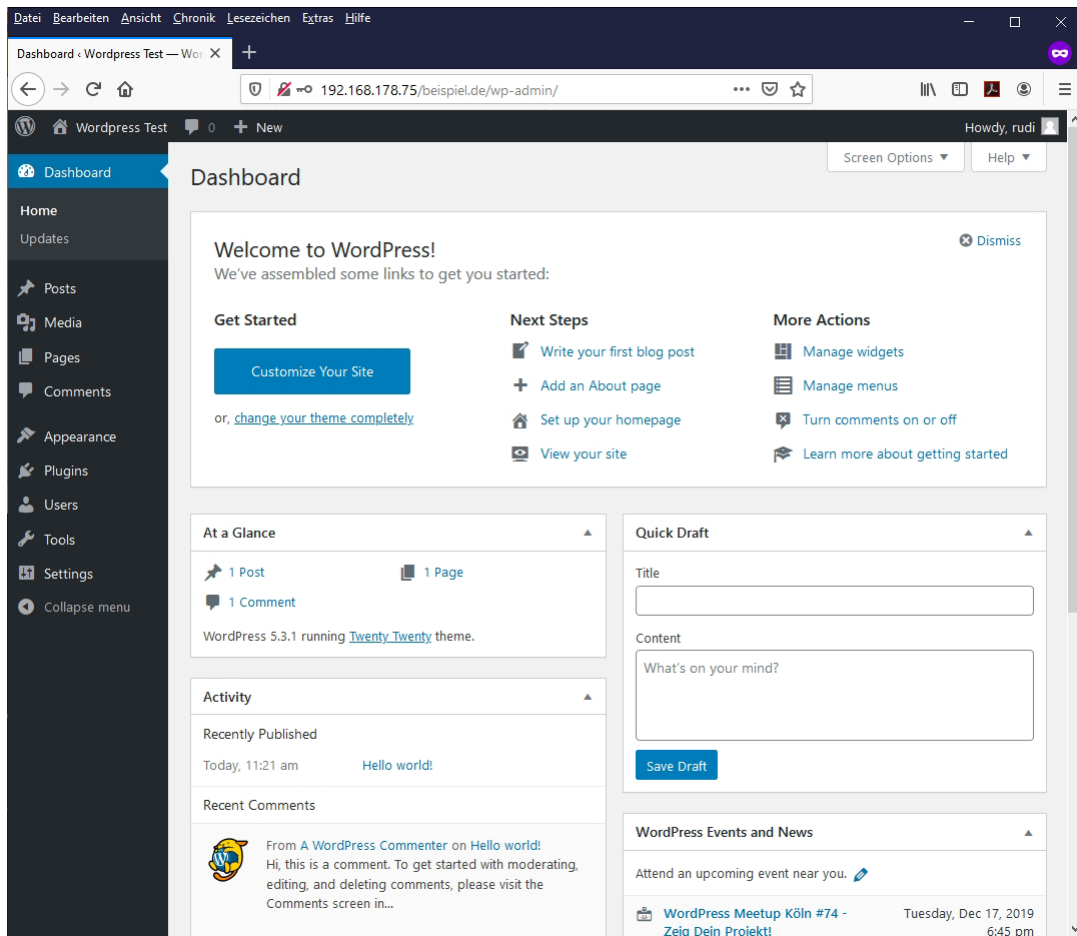


Abbildung 10.11: Das Dashboard von *WordPress* erlaubt weitere Einstellungen.

Danach können Sie ins sogenannte Dashboard wechseln und weitere Einstellungen für Ihre Webseite vornehmen oder mit dem Programmieren der Seite beginnen.

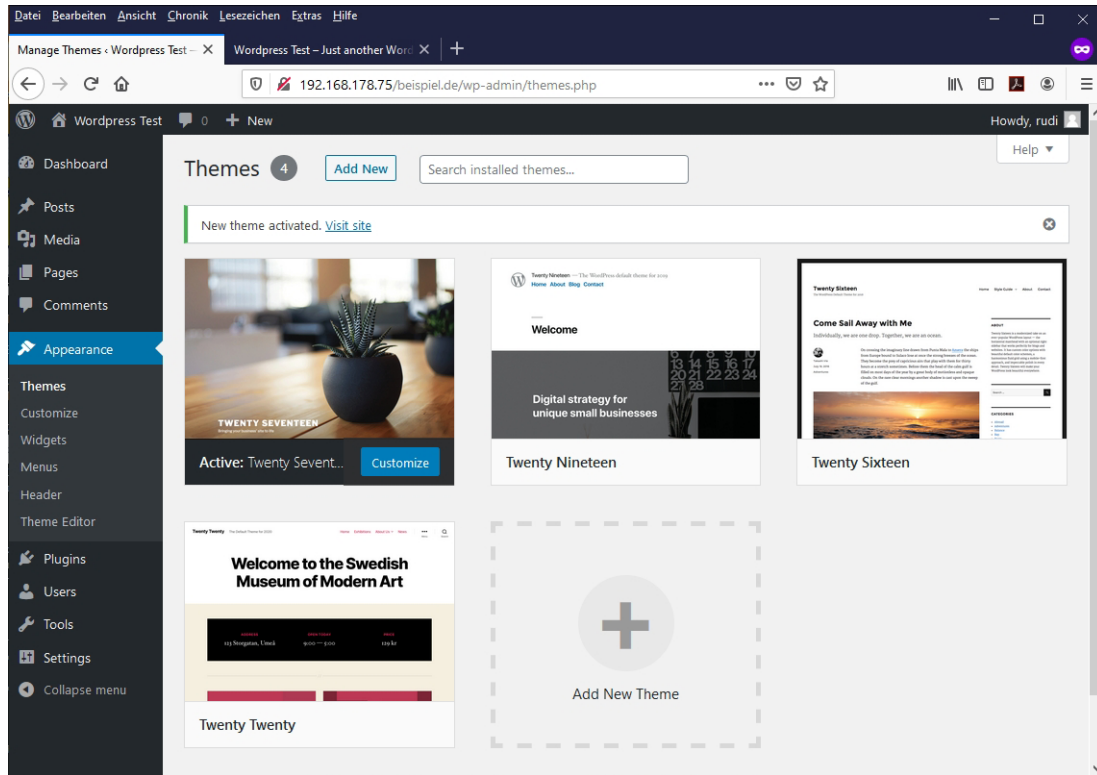


Abbildung 10.12: Unter der Rubrik *Appearance* können verschiedene Themen für die eigene Webseite gewählt werden.

Falls Sie sich aus Versehen ausloggen und sich fragen, wie Sie sich wieder in das Backend von *WordPress* einloggen können, verwenden Sie die folgende URL. Ersetzen Sie dabei bitte wieder die IP-Adresse meines Wohnzimmer-PCs mit Ihrer eigenen.

Code-Ausschnitt 10.45: Einwahl in des Backend von *WordPress*

`http://192.168.178.75/beispiel.de/wp-login.php`

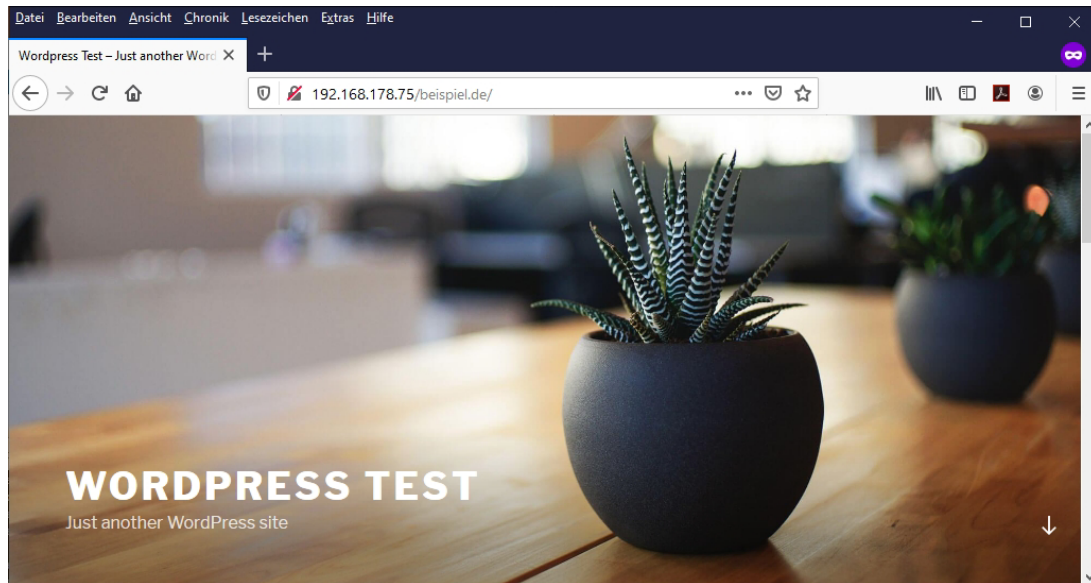


Abbildung 10.13: Die Standard Themen von *WordPress* hinterlassen einen guten Eindruck.

Unter der Rubrik *Appearance* im Dashboard sind bereits einige Themen vorinstalliert. Suchen Sie sich eines aus und aktivieren Sie es, um Ihrer Webseite ein anderes Aussehen zu verleihen. Das Internet bietet unzählige kostenlose Themen für *WordPress*. Damit diese Themen funktionieren, müssen noch ein paar Kleinigkeiten an der Installation der Beispielseite geändert werden. Da das Verzeichnis *beispiel.de* bisher dem Benutzer *root* gehört, darf *WordPress* es nicht beschreiben. Beim Versuch, ein externes Thema zu installieren, erscheint daher die folgende Fehlermeldung:

Code-Ausschnitt 10.46: Fehlermeldung von *WordPress* beim Schreiben

```
Unable to create directory wp-content/uploads/2019/12. Is its parent directory writable by the server?
```

Das kann man ändern, indem man den Benutzer und Besitzer des Beispiel-Verzeichnisses nach *www-data* ändert:

Code-Ausschnitt 10.47: Ändern des Benutzers/Besitzers von Dateien

```
sudo chown -R www-data:www-data beispiel.de/
```

Sollten Sie weiterhin die Fehlermeldung

Code-Ausschnitt 10.48: Fehlermeldung bei zu kleiner Dateigröße zum Upload

```
The uploaded file exceeds the upload_max_filesize directive in php.ini.
```

erhalten, müssen Sie die Variable *upload_max_filesize* in der Datei *php.ini* auf einen höheren Wert setzen. Anschließend können Sie sich ein anderes Thema herunterladen, beispielsweise das von der URL <https://athemes.com/theme/alizee/> [104]. Speichern Sie die Ihnen angebotene zip-Datei auf dem Rechner, auf dem Sie auch den Browser nutzen, um sich die Beispielseite anzuzeigen. Das Dashboard bietet Ihnen die Möglichkeit, diese zip-Datei als externes Thema zu installieren und zu aktivieren. Externe Themen können ebenfalls automatisch auf eine neuere Version upgedated werden.

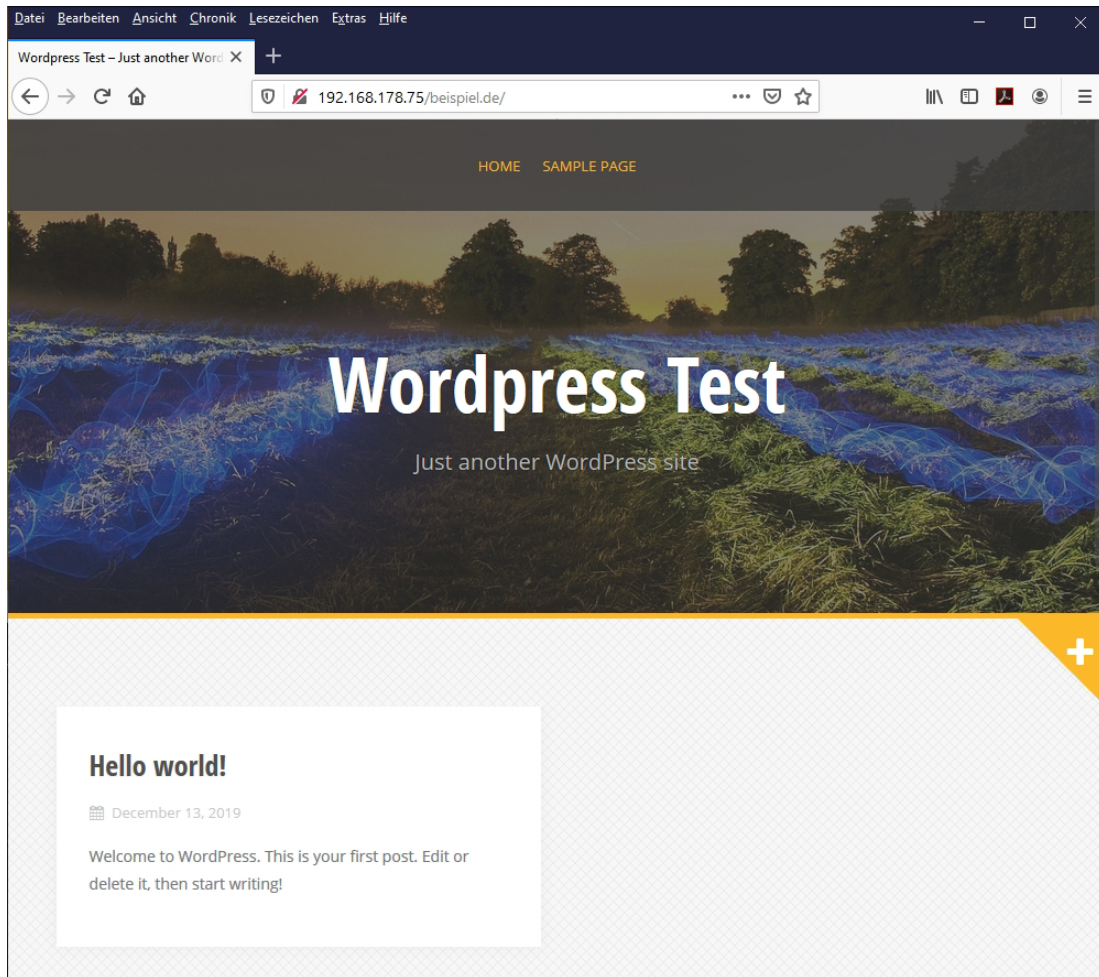


Abbildung 10.14: Es gibt unzählige Themen für *WordPress*, die im Internet kostenlos zur Verfügung gestellt werden. Das Thema *Alizee* ist eines davon.

Es würde den Rahmen dieses Buches sprengen, eine komplette Webseite in *WordPress* zu erstellen. Jetzt, wo die Voraussetzungen geschaffen worden sind, empfehle ich Ihnen zur weiteren Lektüre das Buch *Einstieg in WordPress 5* von Peter Müller [105]. In diesem Buch erklärt Peter Müller auf über 500 Seiten auf eine sehr anschauliche Art und Weise, wie Sie eine eigene Webseite mit Hilfe von *WordPress* erstellen können.

10.5 DLNA

DLNA steht für Digital Living Network Alliance und bezeichnet einen Zusammenschluss von Herstellern, die durch Vergabe eines Zertifikats (*DLNA*) erreichen wollten, dass Geräte verschiedener Hersteller in einem Heimnetzwerk miteinander kommunizieren können. Die Vereinigung selbst löste sich im Januar 2017 auf. Alle verbleibenden kommerziellen Aspekte der Zertifizierung werden von der Firma SpireSpark fortgeführt.

In diesem Abschnitt werden wir einen *DLNA*-Server aufsetzen, der kompatible Geräte im Heimnetz mit Filmen, Hörbüchern und Musik versorgen kann. Unter Ubuntu wird der *DLNA*-Server von der Firma ReadyMedia zur Verfügung gestellt. *DLNA/UPnP*-

Geräte (Universal Plug and Play) erkennen den Server, den wir gleich installieren werden, automatisch und zeigen die multimedialen Inhalte des Servers an. ReadyMedia beherrscht selbst aber keine Transkodierung. Das bedeutet, dass das Gerät, welches beispielsweise einen Film mit einer H.265-Kodierung abspielen soll, diesen Codec selbst können muss. Das trifft auf viele Geräte zu, aber nicht auf alle. Manche Hersteller scheuen die Kosten für verschiedene Codecs. Da ReadyMedia früher einmal MiniDLNA hieß, kann der *DLNA*-Server so installiert werden:

Code-Ausschnitt 10.49: Installation von *DLNA*

```
sudo apt install minidlna
```

Wer den *DLNA*-Server aus den Quelltexten übersetzen möchte, benötigt die folgenden Abhängigkeiten:

Code-Ausschnitt 10.50: Erforderliche Abhängigkeiten zum Übersetzen des *DLNA*-Servers

```
sudo apt-get install libexif12 libjpeg62 libid3tag0 libflac8 libvorbisfile3 sqlite3 ↵  
libavformat53 libuuid1
```

Den Quelltext findet man unter <https://sourceforge.net/projects/minidlna/> [106].

Damit ist die Installation auch schon erledigt und der *DLNA*-Server muss nur noch konfiguriert werden. Dies geschieht in der Datei */etc/minidlna.conf*. Bevor wir uns die Konfiguration im Detail anschauen, rufen wir MiniDLNA im Webbrowser auf. Öffnen Sie dazu bitte die URL `http://192.168.178.75:8200` in einem Webbrowser Ihrer Wahl und ersetzen Sie die IP-Adresse meines Wohnzimmer-PCs wieder mit Ihrer.

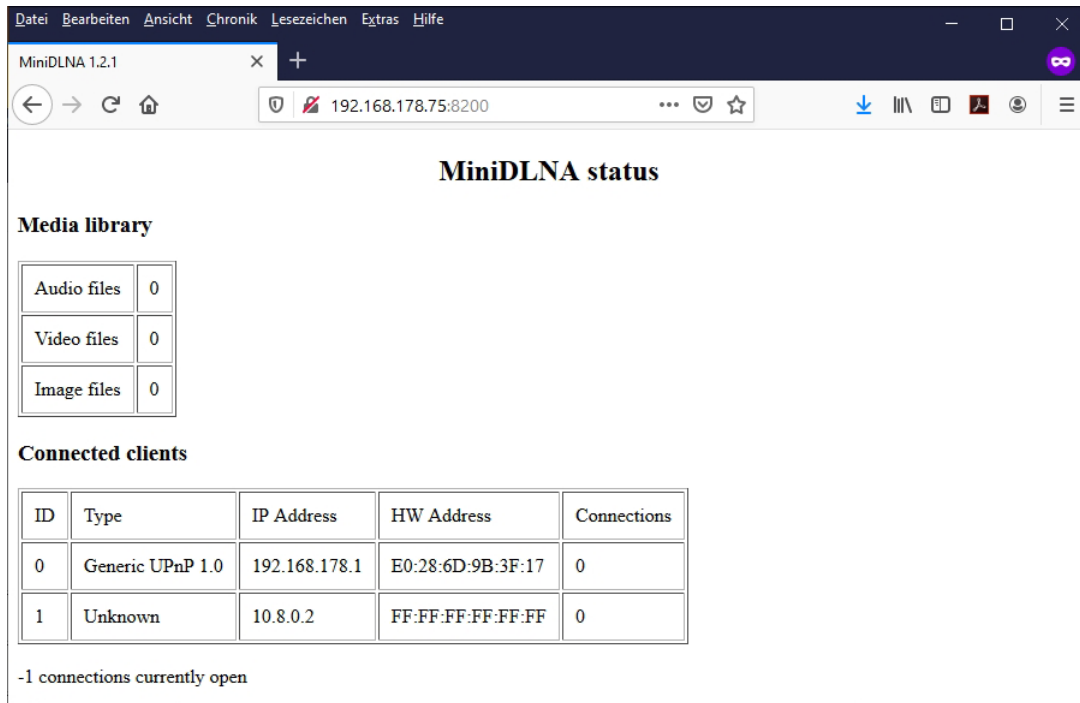


Abbildung 10.15: Der Status von MiniDLNA kann in einem Browser über den Port 8200 abgefragt werden.

Meine Medien-Bibliothek ist noch leer. Ich habe keine Audio-Dateien, Files oder Bilder freigegeben. Als verbundene Klienten werden lediglich meine VPN-Verbindung (10.8.0.2) und mein Gateway (192.168.178.1) erkannt. Schauen wir uns nun die Konfiguration näher an, bei der folgende Eintragungen geändert und angepasst werden können.

- **network_interface=eth0:**

An dieser Stelle wird die Netzwerkkarte angegeben, auf der MiniDLNA seine Dienste anbietet. Auf neueren Ubuntu-Versionen heißt das Ethernet-Interface *enp2s0*. Sie können die zur Verfügung stehenden Interfaces mit dem Befehl *ifconfig* auflisten.

- **media_dir=/var/lib/minidlna:**

In diesem Verzeichnis sucht MiniDLNA nach neuen Multimedia-Dateien, die Audio-Dateien (A), Videos (V) oder Bilder (P für Picture) sein können. Es können auch separate Verzeichnisse wie folgt angegeben werden:

```
# Musik (Audio)
media_dir=A,/opt/music
# Videos
media_dir=V,/opt/videos
# Bilder (Pictures)
media_dir=P,/opt/pics
```

- **friendly_name=DLNA-Server Name:**

Mit diesem Eintrag können Sie einen Namen für den DLNA-Server vergeben. Der Name wird dann von den kompatiblen Abspielgeräten angezeigt.

- **db_dir=/var/cache/minidlna:**
Unter `db_dir` kann man ein Datenbank-Verzeichnis angeben, das MiniDLNA zum Puffern der eigenen Daten verwendet.
- **inotify=yes:**
Wird `inotify` auf `yes` gesetzt, sucht MiniDLNA automatisch nach neuen Dateien in den `media_dir`-Verzeichnissen.

Nach Änderungen an der Konfigurationsdatei muss der *DLNA*-Server neu gestartet werden.

Code-Ausschnitt 10.51: DLNA-Server neu starten

```
sudo service minidlna restart
```

Zum Testen habe ich meine Konfiguration so belassen, wie sie war und eine mp3-Datei in das Verzeichnis `/var/lib/minidlna` kopiert (*sudo* nicht vergessen). Nach einem Neustart des *MiniDLNA*-Servers wird die Audio-Datei korrekt erkannt und im Webbrowser wird unter *Audio* korrekterweise eine 1 angezeigt. Ab sofort steht die Audiodatei für DLNA-fähige Geräte im Heimnetz zum Abspielen bereit.

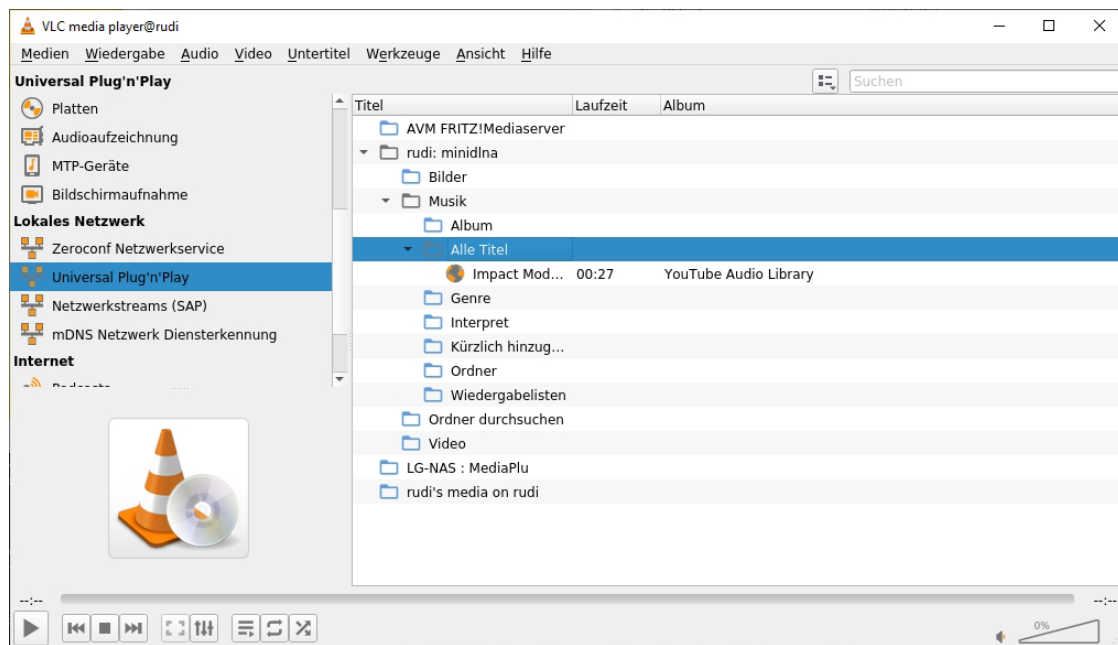


Abbildung 10.16: Unter VLC sind die Einstellungen zum Abspielen von DLNA-Inhalt unter der Rubrik *Ansicht* versteckt.

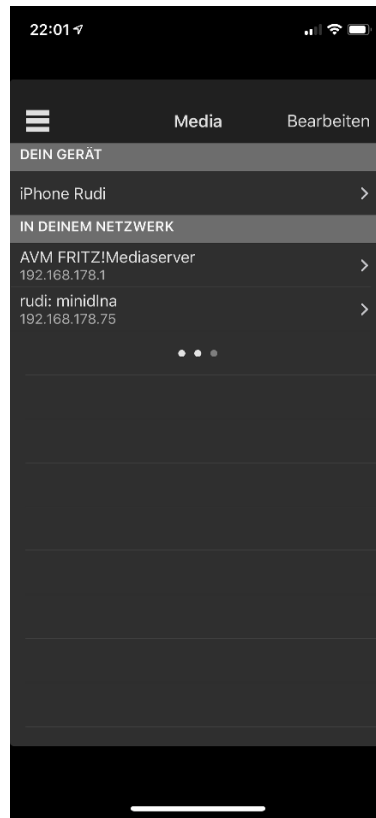


Abbildung 10.17: Auch die iPhone-App JuP&P erkennt meinen DNLA-Server sofort.

10.6 Philips Hue

In diesem Abschnitt zeige ich Ihnen, wie Sie mit einfachen Skripten auf den Philips Hue-Server zugreifen können. Philips Hue ist ein System, mit dem Sie Lampen per App an- und ausschalten können. Darüber hinaus können Sie mit passenden Leuchtmitteln die Helligkeit oder die Farbe einer Lampe ändern. Zeitgesteuertes Ein- und Ausschalten ist ebenfalls möglich. Die Befehle, die gleich vorgestellt werden, können eins zu eins in die Datei *commands.conf* (Kapitel 7) übernommen werden. Damit kann dann z. B. per Fernbedienung aus dem VDR heraus das Licht gedimmt werden.

Um Kontakt mit dem Philips Hue-Server aufzunehmen, bietet sich die in Python programmierte Bibliothek *phue* an. Den Quelltext dieser Bibliothek findet man unter <https://github.com/studioimaginaire/phue> [107]. Die eigentliche Bibliothek benötigt dabei nur eine einzige Datei, die *phue.py* heißt. Laden Sie den Quelltext zunächst so herunter:

Code-Ausschnitt 10.52: Herunterladen der Python-Bibliothek *phue*

```
cd ~  
git clone https://github.com/studioimaginaire/phue  
cd phue
```

Die Bibliothek ist mit allen Python-Versionen ab Version 2.6 kompatibel. Wenn Sie sie nutzen möchten reicht es, die Datei *phue.py* in das Verzeichnis zu kopieren, in dem sich Ihre Skripte zur Lampensteuerung befinden.

Damit alle Skripte funktionieren, muss der Wohnzimmer-PC dem Philips Hue-Server bekanntgemacht werden. Diese Anlernmethode kennen Sie bereits, wenn Sie die Philips Hue-App installiert haben. Diese erwartet, dass Sie zu einem bestimmten Zeitpunkt den großen runden Knopf auf der Philips Hue-Bridge drücken. Diese Anlernmethode verwenden wir auch hier. Nachdem Sie in das Verzeichnis *phue* gewechselt haben, rufen Sie bitte *python* auf. Drücken Sie dann den Anmeldeknopf Ihrer Philips Hue-Bridge und geben Sie die folgenden Befehle ein:

Code-Ausschnitt 10.53: Anlernen der Philips Hue-Bridge für *phue*

```
python
Python 2.7.17 (default, Nov 7 2019, 10:07:09)
[GCC 9.2.1 20191008] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from phue import Bridge
>>> b = Bridge('ip_hue')
>>> b.connect()
```

Bitte ersetzen Sie im oben gezeigten Aufruf *ip_hue* mit der IP-Adresse Ihrer Philips Hue-Bridge. Falls der letzte Befehl keine Fehlermeldung ausgibt, hat das Anlernen problemlos funktioniert. Das Anlernen selbst muss nur einmal durchgeführt werden.



Der Befehl *b.connect()* muss nur einmal ausgeführt werden, um den Wohnzimmer-PC an der Philips Hue-Bridge anzumelden.

Die Lampen, die Sie in Ihr Hue-System eingebunden haben, sind durchnummeriert. Im Anschluss an die oben abgesetzten Befehle können Sie (immer noch in der Python-Umgebung) folgende Befehle eingeben, um mit den Lichtern zu spielen:

Code-Ausschnitt 10.54: Ausprobieren verschiedener *phue*-Befehle

```
# Der folgende Befehle gibt aus, ob die Lampe 1 eingeschaltet ist oder nicht
b.get_light(1, 'on')

# Der folgende Befehl setzt die Helligkeit der Lampe 1 auf das Maximum
b.set_light(1, 'bri', 254)

# Der folgende Befehl schaltet die Lampe 2 an.
# Ersetzen Sie True mit False, um die Lampe 2 auszuschalten
b.set_light(2, 'on', True)

# Mehrere Lampen können gleichzeitig so angesprochen werden
b.set_light([1,2], 'on', True)

# Der nächste Befehl gibt den Namen der Lampe 1 zurück
b.get_light(1, 'name')

# Der Name kann dann so verwendet werden
b.get_light('Wohnzimmer')
b.set_light('Wohnzimmer', 'bri', 254)
```

Alle weiteren Befehle und zusätzliche Beispiele finden Sie in den Verzeichnissen *examples* und *tests*. Hier gibt es auch Beispiele, die Ihnen zeigen, wie die Farben der Lampen eingestellt werden können. Die für Sie interessantesten Befehle können Sie in einem Skript zusammenfassen, das Sie dann mit *python skriptname* aufrufen können. Um die Lampe 2 einzuschalten, sähe das Skript wie folgt aus:

Code-Ausschnitt 10.55: Skript, um die Hue-Lampe 2 einzuschalten

```
#!/usr/bin/env python
from phue import Bridge
b = Bridge('ip_hue')
b.set_light(2, 'on', True)
```

Bitte ersetzen Sie wieder das Wort *ip_hue* mit der IP-Adresse Ihrer Hue-Bridge.

Zusammenfassung und Ausblick 10 Nach dem Ausflug in die Themen Server und Datenbanken schauen wir uns im letzten Kapitel die Möglichkeiten an, mit dem Wohnzimmer-PC Schriftstücke zu verfassen. Dabei zeige ich Ihnen drei Möglichkeiten: Wir nutzen mit *LibreOffice* ein freies und mächtiges Tool, das sich nicht hinter Microsofts OfficeSuite verstecken muss. Ich zeige Ihnen die Fähigkeiten von \LaTeX am Beispiel von verschiedenen Bewerbungen und einem Buch. Sollte es dennoch ein Microsoft Office sein, zeige ich Ihnen, wie Sie die Windows-Version dieser Programme unter LINUX laufen lassen können. ■

Literaturverzeichnis

- [103] Maik Schmidt. *Raspberry Pi: Einstieg - Optimierung - Projekte*. 1. Auflage. Band 1. Heidelberg: dpunkt-Verlag, 2013 (siehe Seite 268).
- [104] Athemes. *ALIZEE*. 2019.
<https://athemes.com/theme/alizee/>
besucht am 27. 12. 2019 (siehe Seite 280).
- [105] Peter Müller. *Einstieg in WordPress 5*. Rheinwerk Computing, 2019. ISBN: ISBN 978-3-8362-6155-5 (siehe Seite 281).
- [106] Sourceforge. *ReadyMedia*. 2019.
<https://sourceforge.net/projects/minidlna/>
besucht am 27. 12. 2019 (siehe Seite 282).
- [107] Github. *phue*. 2019.
<https://github.com/studioimaginaire/phue>
besucht am 27. 12. 2019 (siehe Seite 285).

11. Schreiben mit dem Wohnzimmer-PC

11.1	LibreOffice	289
11.2	L ^A T _E X	292
11.3	PlayOnLinux	304

Jeder PC-Benutzer hat sicher schon einmal ein Schriftstück mit Hilfe des Computers verfasst. Die bekannteste Office-Anwendung ist sicherlich das Office-Paket von Microsoft. In diesem Kapitel werden wir zunächst einen Blick auf die freien Alternativen werfen: LibreOffice und L^AT_EX. LibreOffice braucht sich inzwischen nicht mehr hinter dem großen Bruder aus Redmond zu verstecken und für viele ist L^AT_EX die erste Wahl, wenn es um das Verfassen großer oder schöner Dokumente wie Bücher oder Bewerbungen geht. Aber auch die Microsoft Office-Tools laufen unter LINUX. In einer Schritt-für-Schritt Anleitung zeige ich Ihnen dann, wie Sie Windows-Programme, etwa Microsoft WORD, unter LINUX installieren können.



11.1 LibreOffice

LibreOffice umfasst ein Programm zur Textverarbeitung, eine Tabellenkalkulation, ein Präsentationsprogramm und ein Programm zum Erstellen von Zeichnungen. Ein Formel-Editor ist genau so vorhanden wie ein Datenbank-Managementsystem. LibreOffice ist aus dem OpenOffice-Projekt entstanden. Grund für die Abspaltung war die Unzufriedenheit über die Firma Oracle, die das OpenOffice-Paket verantwortlich übernommen hat. Inzwischen hat Oracle diese Verantwortung wieder an die Apache Software Foundation abgegeben. LibreOffice ist ein aus zwei Wörtern zusammengesetztes Wort. Im Spanischen bedeutet *libre* „frei“, im Englischen bedeutet *office* soviel wie „Büro“. Zweck von LibreOffice ist es also, eine freie Büro-Software zur Verfügung zu stellen. In der Ubuntu-Distribution liegt LibreOffice als installierbares Programmpaket bereit. Falls es noch nicht installiert ist, installieren Sie es mit



Abbildung 11.1: LibreOffice stellt ein komplettes Office-Paket kostenlos bereit (Quelle: libreoffice.org).

Code-Ausschnitt 11.1: Installation von LibreOffice

```
sudo aptitude install libreoffice
```

Wer möchte, kann direkt noch das deutsche Sprachpaket und die deutsche Hilfe installieren.

Code-Ausschnitt 11.2: Deutsches Sprachpaket und Hilfe für *LibreOffice* installieren

```
sudo apt-get install libreoffice-help-de libreoffice-l10n-de
```

Es gibt noch viele andere Pakete für *LibreOffice*, wie z. B. eine Grammatik-Überprüfung oder Clip-Art-Bibliotheken. Ein

Code-Ausschnitt 11.3: Alle installierbaren *LibreOffice*-Pakete auflisten

```
sudo aptitude search libreoffice
```

zeigt Ihnen alle installierbaren *LibreOffice*-Pakete an. *LibreOffice* wird nach der Installation von der X-Oberfläche aus gestartet. Beim Start des Programms wird Ihnen ein Auswahl-Menü präsentiert, aus dem Sie die Büro-Applikation auswählen können, mit der Sie arbeiten wollen.

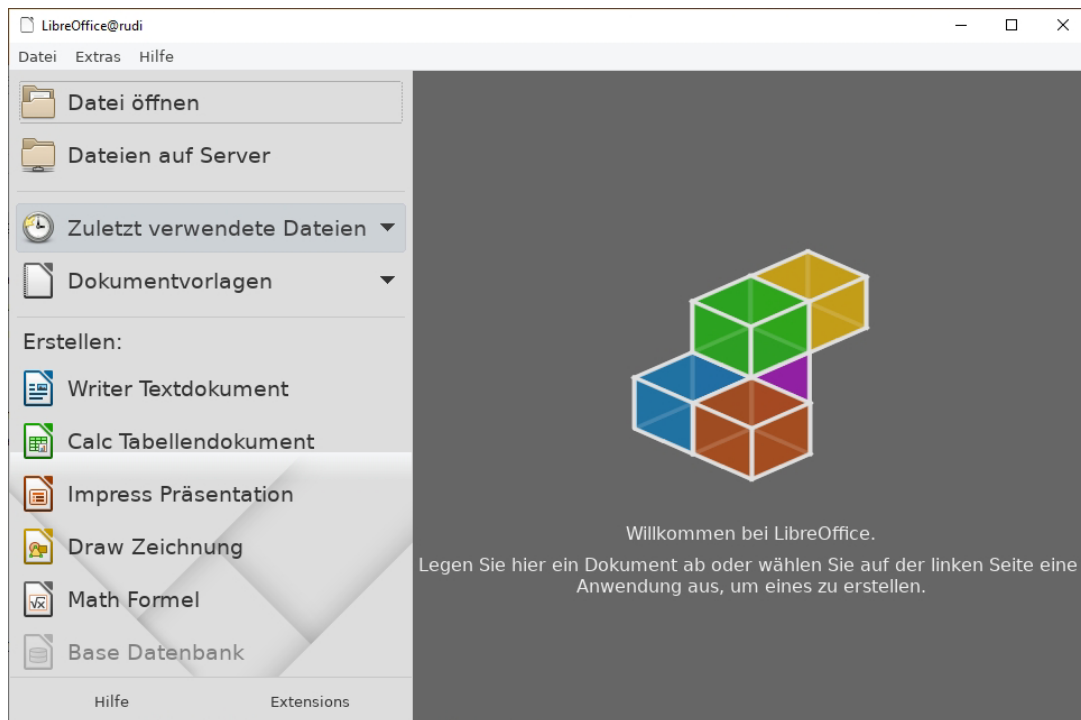


Abbildung 11.2: Das Startmenü von *LibreOffice* erlaubt die einfache Auswahl der zu startenden Büro-Applikation.

11.1.1 LibreWriter

Wir werfen an dieser Stelle einen kurzen Blick auf das *WORD*-Pendant namens *LibreWriter*, das Sie durch einen Klick auf *Writer Textdokument* starten können.

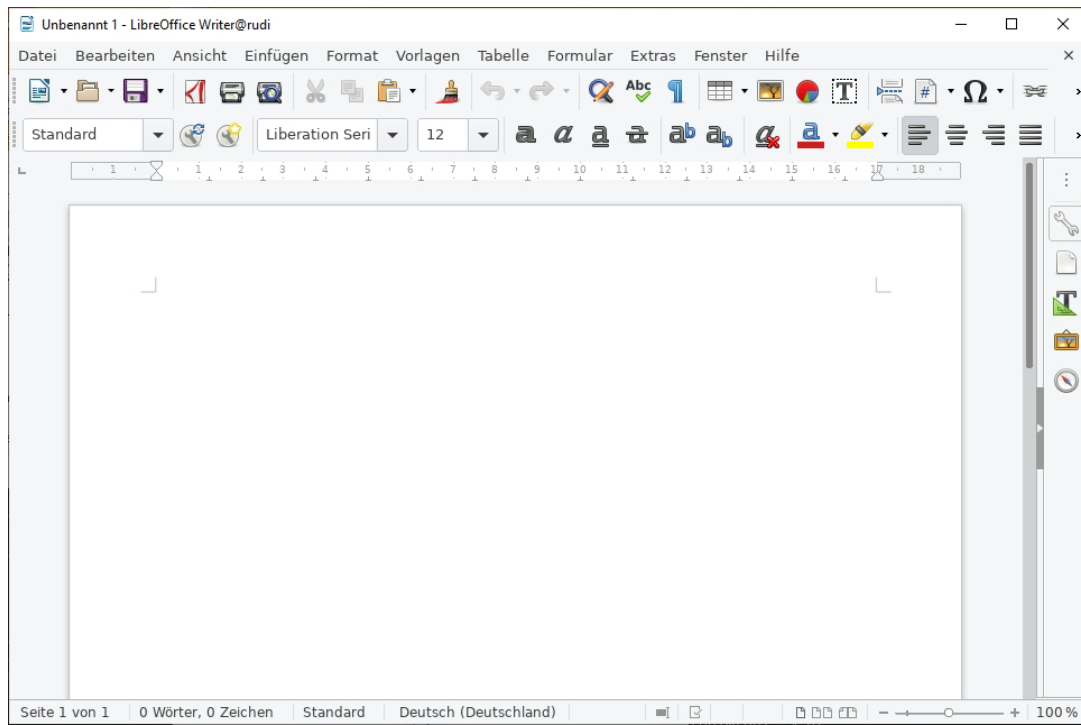


Abbildung 11.3: *LibreWriter* erlaubt das Erstellen von Texten ähnlich wie *WORD*.

Wenn Sie ein geschriebenes Dokument ausdrucken möchten, können Sie auf Drucker zurückgreifen, die Sie, wie in Kapitel 10.1 angegeben, mit Hilfe von AirPrint und/oder CUPS installiert haben.



LibreWriter erlaubt das Einlesen und Speichern von Microsoft *WORD*-Dokumenten in den Formaten *doc* und *docx*.


```

Alle Formate
ODF-Textdokument (.odt)
ODF Textdokument Dokumentvorlage (.ott)
Flat XML ODF-Textdokument (.fodt)
Unified Office Format Textdokument (.uot)
Word 2007-365 (.docx)
Word 2007-365 Dokumentvorlage (.dotx)
Word 2003 XML (.xml)
Word 97-2003 (.doc)
Word 97-2003 Dokumentvorlage (.dot)
DocBook (.xml)
HTML-Dokument (Writer) (.html)
Rich Text (.rtf)
Text (.txt)
Text - Textkodierung wählen (.txt)
Office Open XML Text (.docx)

```

Abbildung 11.4: *LibreWriter* kann *WORD*-Dateien lesen und speichern.

Bei aufwendig formatierten Texten ist es mir allerdings häufiger passiert, dass die Formatierungen in *LibreOffice* anders aussahen als in *Microsoft Office*. Arbeitet man mit mehreren Kollegen und verschiedenen *Office*-Programmen am selben Dokument, kann das schon einmal zu Stress führen. Weitere Informationen über *LibreOffice* erhalten Sie unter <http://de.libreoffice.org/> [108]. Wer größere Schreib-Projekte umsetzen möchte, dem empfehle ich einen Blick in folgenden Abschnitt (11.2).

11.2 L^AT_EX

L^AT_EX ist ein Programmpaket, welches das Setzen von Text mit Hilfe des Textsatzsystems T_EX unter Verwendung von Makros vereinfacht.



Ein Makro in der Computersprache bedeutet die Zusammenfassung mehrerer Anweisungen zu einer kürzeren Anweisung.

L^AT_EX ist kein WYSIWYG (*What you see is what you get*)-System. Texte werden mit Editoren wie *vi* oder *joe* oder mit speziellen IDEs (Integrated Development Environment) geschrieben. Inhalt und Form sind getrennt und können sogar in getrennten Dateien verwaltet werden. Erst nach einem Übersetzungslauf werden beide zusammengeführt. Formatierungen werden mit einfachem Text gekennzeichnet. So weist die Anweisung

Code-Ausschnitt 11.4: Ein neues Kapitel in L^AT_EX beginnen

```
\chapter{Kapitelname}
```

den L^AT_EX-Übersetzer an, ein neues Kapitel mit dem Namen *Kapitelname* zu erstellen. Das sieht im ersten Moment zwar nach viel Arbeit aus, da viele sicher gewohnt sind,

mit der Maus den geschriebenen Text zu markieren und dann auf *Überschrift 1* zu klicken. \LaTeX nimmt dem Benutzer dafür an anderer Stelle sehr viel Arbeit ab: Kapitel eines Buches beginnen automatisch auf der rechten Seite, entsprechende linke Leerseiten werden automatisch eingefügt. Kapitel werden automatisch nummeriert und der definierte Absatz wird verwendet. Diese Vorgehensweise ist auch bei der Eingabe von Formeln vorteilhaft: Diese werden nicht nur in der korrekten Größe dargestellt und richtig eingerückt, sondern auch gleich noch passend nummeriert. \LaTeX läuft auf einer Vielzahl von Betriebssystemen, darunter *LINUX*, *Windows* oder *MacOS*. Dadurch, dass Texte nicht in einem binären Format gespeichert werden (wie z. B. *docx*), sind sie für alle Rechner- und Betriebssystem-Typen kompatibel: Ein Buch, welches auf einem Wohnzimmer-PC geschrieben wurde, kann in Sekunden auf einem Mac-Computer sichtbar gemacht werden - vorausgesetzt, \LaTeX ist auf diesem Rechner installiert. Apropos: Wer die Minimal-Version installieren will, geht so vor:

Code-Ausschnitt 11.5: Minimale \LaTeX -Version installieren

```
sudo apt install texlive texlive-lang-german
```

Die komplette Installation erhält man nach Aufruf von

Code-Ausschnitt 11.6: Komplette \LaTeX -Version installieren

```
sudo apt-get install texlive-full
```

\LaTeX bietet eine Reihe nützlicher Zusatzprogramme an:

- *BibTeX* automatisiert das Erstellen von Literaturverzeichnissen.
- *pdfLaTeX* erzeugt aus \TeX -Dateien *PDF*-Dateien.
- *MakeIndex* erzeugt Stichwortverzeichnisse und
- *MusiXTeX* kann zum Setzen von Noten verwendet werden.

Die entsprechenden Pakete installieren Sie so:

Code-Ausschnitt 11.7: Zusätzliche \LaTeX -Pakete installieren

```
sudo apt-get install biber texlive-music
```

11.2.1 \LaTeX -Beispiel

Die Wikipedia-Seite <http://de.wikipedia.org/wiki/LaTeX> [109] stellt ein schönes \LaTeX -Beispiel zur Verfügung.

Code-Ausschnitt 11.8: Ein einfaches L^AT_EX-Beispiel

```

%% Erläuterungen zu den Befehlen erfolgen unter
%% diesem Beispiel.
\documentclass{scrartcl}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage[ngerman]{babel}
\usepackage{amsmath}

\title{Ein Testdokument}
\author{Otto Normalverbraucher}
\date{05. Januar 2004}
\begin{document}

\maketitle
\tableofcontents
\section{Einleitung}

Hier kommt die Einleitung. Ihre Überschrift kommt
automatisch in das Inhaltsverzeichnis.

\subsection{Formeln}

\LaTeX{} ist auch ohne Formeln sehr nützlich und
einfach zu verwenden. Grafiken, Tabellen,
Querverweise aller Art, Literatur- und
Stichwortverzeichnis sind kein Problem.

Formeln sind etwas schwieriger, dennoch hier ein
einfaches Beispiel. Zwei von Einsteins
berühmtesten Formeln lauten:
\begin{align}
E &= mc^2 && \\\
m &= \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}} && \\
\end{align}
Aber wer keine Formeln schreibt, braucht sich
damit auch nicht zu beschäftigen.
\end{document}

```

Speichern Sie diesen Text für eine erste Begegnung mit L^AT_EX unter */wiki.tex* und rufen Sie danach L^AT_EX so auf:

Code-Ausschnitt 11.9: Den L^AT_EX-Übersetzer aufrufen

```
pdflatex wiki.tex
```

Der Aufruf erstellt eine Datei mit dem Namen *wiki.pdf*. Das Ergebnis ist in Abb. 11.5 zu sehen. Da es sehr viele und gute L^AT_EX-Referenzen gibt, verzichte ich an dieser Stelle darauf, dieses Satzsystem zu erklären. Stattdessen stelle ich Ihnen im nächsten Abschnitt eine schöne Möglichkeit vor, Bewerbungen mit dem Wohnzimmer-PC zu schreiben.

Ein Testdokument

Otto Normalverbraucher

05. Januar 2004

Inhaltsverzeichnis

1 Einleitung	1
1.1 Formeln	1

1 Einleitung

Hier kommt die Einleitung. Ihre Überschrift kommt automatisch in das Inhaltsverzeichnis.

1.1 Formeln

\LaTeX ist auch ohne Formeln sehr nützlich und einfach zu verwenden. Grafiken, Tabellen, Querverweise aller Art, Literatur- und Stichwortverzeichnis sind kein Problem.

Formeln sind etwas schwieriger, dennoch hier ein einfaches Beispiel. Zwei von Einsteins berühmtesten Formeln lauten:

$$E = mc^2 \tag{1}$$

$$m = \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}} \tag{2}$$

Aber wer keine Formeln schreibt, braucht sich damit auch nicht zu beschäftigen.

11.2.2 Bewerbungen schreiben

Mit L^AT_EX lassen sich hervorragend Bewerbungen verfassen. Dabei kann man sehr einfach das Anschreiben, den Lebenslauf und die Zeugnisse zu einem elektronischen Dokument zusammenfassen, welches man dann per E-Mail an die Personalabteilung versenden kann. Im Internet finden sich viele Vorlagen für L^AT_EX-Bewerbungen. Allerdings hat mir persönlich keine so gut gefallen wie die von *stefanqn*. Er stellt seine Vorlage freundlicherweise unter <https://github.com/Stefanqn/Bewerbung> zur Verfügung. Erstellen Sie sich ein Verzeichnis `/home/pi/bewerbung` und laden Sie dann die Vorlage herunter.

Code-Ausschnitt 11.10: Vorlage von <https://github.com/Stefanqn/Bewerbung> [110]

```
cd ~
svn co https://github.com/Stefanqn/Bewerbung.git bewerbung
cd bewerbung
```

Sie erhalten dann ein Verzeichnis namens *Bewerbung.git*, in dem alle benötigten Dateien enthalten sind. Die Vorlage selbst ist DIN5008-konform (http://de.wikipedia.org/wiki/DIN_5008 [111]).



Ich habe ein paar Änderungen an den Vorlagen durchgeführt. Sie können meine geänderten Vorlagen mit einem kompletten Beispiel im Download-Bereich zu diesem Buch herunterladen.

Nach dem Download befinden sich die folgenden Unterverzeichnisse im Verzeichnis *bewerbung*:

- *Bsp-Bin*,
- *Bsp-Latex*,
- *DIN5008*,
- *Fertiges-PDF* und
- *Vorlage*.

Bsp-Bin

Das Verzeichnis *Bsp-Bin* enthält alle Dateien, die im Binär-Format vorliegen. Dazu zählen z. B. gescannte Zeugnisse im *PDF*-Format oder eine eingescannte Unterschrift im *png*-Format.

Bsp-Latex

Das Verzeichnis *Bsp-Latex* enthält drei wichtige Dateien:

1. *Inhalt-Anlagen.tex*: Diese Datei enthält alle *PDF*-Anlagen, die der Bewerbung beigefügt werden sollen.
2. *Inhalt-Anschreiben.tex*: Diese Datei enthält das Anschreiben der Bewerbung.

3. *Inhalt-CV.tex*: Diese Datei enthält den kompletten Lebenslauf.
4. *mm.tex*: Diese Datei enthält gemeinsame Daten, die sowohl im Anschreiben als auch im Lebenslauf verwendet werden. Dazu zählen z. B. der Name des Bewerbers oder der Wohnort.

Die Datei *Inhalt-Anlagen.tex* hat den folgenden Inhalt (Beispiel):

Code-Ausschnitt 11.11: Inhalt der Datei *Inhalt-Anlagen.tex*

```
% Weitere Anlagen
\newcommand* \pics {\priv Anhang/}

\bookmark[page=\thepage,level=0]{Arbeitszeugnisse}
\bookmark[page=\thepage,level=1]{Zeugnis 1}
\includepdf[pages=-] {\pub zeugnis1.pdf}
\bookmark[page=\thepage,level=1]{Zeugnis 2}
\includepdf[pages=-] {\pub zeugnis2.pdf}
...
```

Unter der Überschrift *Arbeitszeugnisse* werden alle gescannten Zeugnisse (*PDF*-Dateien) an die Bewerbung angehängt. Gleichzeitig wird ein Inhaltsverzeichnis-Eintrag für das Zeugnis angelegt. Die Datei *Inhalt-Anschreiben.tex* stellt das komplette Anschreiben der Bewerbung im Brief-Format dar. Ein Beispiel sieht so aus:

Code-Ausschnitt 11.12: Typischer Inhalt der Datei *Inhalt-Anschreiben.tex*

```
\setkomavar{date}{1. Januar 2014}
\emergencystretch=20pt\tolerance=1200\hyphenpenalty=1000% Gewichte für Trennung
\begin{letter}{%
Firma\\
z. Hd. Frau/Herrn XYZ\\
Stra{\ss}e 123\\
12345 Ort\\
}
\setkomavar{subject}[Betreff]{Bewerbung}
\opening{Sehr geehrte(r) Herr/Frau XYZ,}
hier steht der Bewerbungstext.
\newline
Mit freundlichen Grüßen\\sig%
\encl{}
\end{letter}
```

Die Datei *Inhalt-CV.tex* enthält den Lebenslauf. Eine beispielhafte Datei sieht so aus:

Code-Ausschnitt 11.13: Typischer Inhalt der Datei *Inhalt-CV.tex*

```

\laengsteStadt{Hamburg}
%\toggletrue{keinFoto}
%\toggletrue{lebenslaufTitelZentriert}
%\chead{\anschriftKopfZeile}

\emergencystretch=25pt\tolerance=800\hyphenpenalty=1000\hbadness=10000\linepenalty=5000%\↔
linepenalty
=40\hyphenpenalty=80%\exhyphenpenalty=3000%
\titel{\vspace*{5mm}\emph{Lebenslauf}\vspace*{2mm}}

\section{Persönliche Daten}
\eintragK{Name} {\nachname}
\eintragK{Vorname} {\vorname}
\eintragK{Geburtsdatum} {1. Januar 2014}
\eintragK{Geburtsort} {Ort}
\eintragK{Nationalität} {deutsch}
\eintragK{Familienstand} {ledig}
\eintragK{Anschrift} {\strasse}
\eintragK{} {\PLZ \ wohnort}
\eintragK{Telefon} {\mobilNr}
\eintragK{E-Mail} {\myEmail}

\section{Berufserfahrung}
\subsection{\href{http://url.de}{Firmenname}}

\eintragL{\von[01.2014]bis{\small heute}}{Position}{}{Ort}{%
  \begin{itemize}
    \item Stichwort 1
    \item Stichwort 2
  \end{itemize}
}

\eintragL{\zeit[01.2013-01.2014]}{Position}{Abteilung}{Ort}{\vskip 3mm%
  \begin{itemize}
    \item Stichwort 1
  \end{itemize}
}

\section{Referenzen}
\eintragKb{Firma}{Name (email@firma.de)}

\unterschriftLL

```

In der Datei *mm.tex* stehen folgende Angaben:

Code-Ausschnitt 11.14: Inhalt der Datei *mm.tex*

```

\RequirePackage{GemeinsameDaten}
\newcommand* \buildDir {../Vorlage/build/}
\meinVorname{Vorname}
\meinNachname{Nachname}
\meineStrasse{Straße Nr.}
\meinePLZ{PLZ}
\meinWohnort{Ort}
\meineMobilNr{Mobilnummer}
\meineFestnetzNr{Festnetznummer}
%\meineFaxNr{Faxnummer}
\meineEmail{email@privat.de}
\unterschrift[0.65][\hspace*{-30mm}]{../Bsp-Bin/Unterschrift.png}
\foto[4cm][0.4pt]{../Bsp-Bin/picture}

```

DIN5008

Das Verzeichnis *DIN5008* enthält lediglich die Datei *din5008frame.tex*, die sich um die DIN-gemäße Formatierung der Bewerbung kümmert.

Fertiges-PDF

Dieses Verzeichnis enthält nach Fertigstellung die kompletten Bewerbungsunterlagen im *PDF*-Format.

Vorlagen

Das Verzeichnis *Vorlagen* enthält alle \TeX -Steuerungs-Dateien. Die Datei *anschreiben.tex* bindet den Inhalt des Anschreibens im Brief-Format ein.

Code-Ausschnitt 11.15: Inhalt der Datei *anschreiben.tex*

```

\documentclass[DINmtext,draft=false]{scr11tr2}
\usepackage[utf8x]{inputenc}
\usepackage{Vorlage_Anschreiben}
\input{../Bsp-Latex/mm.tex}
\input{../DIN5008/din5008frame.tex}
\farbe{blue}
\begin{document}
\input{../Bsp-Latex/Inhalt-Anschreiben.tex}
\end{document}

```

Die Datei *cv.tex* bindet den Lebenslauf analog ein:

```

\PassOptionsToPackage{dvipsnames}{xcolor}
\documentclass[12pt,a4paper,sans]{moderncv} % ***test letter*** % font size ('10pt', '11pt' and↵
'12p
t'), paper size ('a4paper', 'letterpaper', 'a5paper', 'legalpaper', 'executivepaper' and '↵
landscape')
and font family ('sans' and 'roman')
\usepackage{Vorlage_Lebenslauf}
\usepackage[utf8x]{inputenc}
\PrerenderUnicode{ßÄäÜüÖö}
\input{../Bsp-Latex/mm.tex}
\begin{document}
\input{../Bsp-Latex/Inhalt-CV.tex}
\end{document}

```

Die entsprechenden Dateien mit dem Zusatz *Gen* generieren die dazugehörigen *PDF*-Dateien. Die Dateien mit dem Präfix *Vorlage_*.sty* bestimmen das Aussehen von Anschreiben, Lebenslauf und der gesamten Bewerbung. Das Erstellen der kompletten Bewerbungsunterlagen funktioniert nun wie folgt:

1. Füllen Sie die Datei *mm.tex* mit Ihren persönlichen Daten.
2. Schreiben Sie das Anschreiben und den Lebenslauf.
3. Passen Sie das jeweilige Datum an.
4. Rufen Sie L^AT_EX so auf:

Code-Ausschnitt 11.16: L^AT_EX-Befehle zum Erstellen der Bewerbung

```
cd Vorlage
pdflatex anschreibenGen.tex
pdflatex cvGen.tex
cp *Gen.pdf build
```

Die *PDF*-Dateien für das Anschreiben und den Lebenslauf müssen in das *build*-Verzeichnis kopiert werden.

5. Erstellen Sie die komplette Bewerbungsmappe mit

Code-Ausschnitt 11.17: Erstellen der Bewerbungsmappe

```
pdflatex Bewerbung_Komplett.tex
```

6. Die fertigen Bewerbungs-Unterlagen stehen danach als *Bewerbung_Komplett.pdf* im Verzeichnis *Vorlage* zur Verfügung und können in das Verzeichnis *Fertiges-PDF* zur Archivierung kopiert werden.

Ein Bewerbungs-Beispiel gibt es unter https://github.com/Stefanqn/Bewerbung/blob/master/Fertiges-PDF/Bewerbung_Komplett.pdf [110] oder im Download für dieses Kapitel.

11.2.3 Rechtschreib-Prüfung

Mit ein wenig Übung kann man L^AT_EX hervorragend in einem kleinen Texteditor schreiben. Allerdings fehlt hier die rote Unterstreichung von Wörtern, die entweder falsch geschrieben sind oder nicht im Lexikon auftauchen. Abhilfe schafft hier das Programm *aspell*. Installieren Sie das Programm zusammen mit dem deutschen Sprachpaket so:

Code-Ausschnitt 11.18: Installation einer Rechtschreibprüfung

```
sudo apt-get install aspell aspell-de
```

Nach der Installation können Sie die Rechtschreibung im Terminal durch Aufruf des Befehls

Code-Ausschnitt 11.19: Datei auf Rechtschreibung überprüfen

```
aspell check dateiname
```

überprüfen. Bitte ersetzen Sie dabei *dateiname* mit dem Namen der Datei, die Sie bezüglich der Rechtschreibung überprüfen möchten.

```

rudi@rudi: ~/latex
% If you don't need one or more of the below, just remove the content leaving \
the command, e.g. \cvnumberphone{}

\profilepic{alice.jpeg} % Profile picture

\cvname{Alice} % Your name
\cvjobtitle{Adventurer} % Job title/career

\cvdate{26 November 1865} % Date of birth
\cvaddress{United Kingdom} % Short address/location, use \newline if more than \
1 line is required
\cvnumberphone{+39 0325658974} % Phone number

1) Alice                                2) Alices

i) Ignorieren                          I) Alle ignorieren
r) Ersetzen                            R) Alle ersetzen
a) Hinzufügen                          l) In Kleinbuchstaben hinzufügen
b) Abbrechen                           x) Beenden

?

```

Abbildung 11.6: Mit *aspell* gibt es die Möglichkeit, die Rechtschreibung einer Datei zu überprüfen.

Während der Überprüfung können Sie entscheiden, ob ein nicht erkanntes Wort mit einem bekannten ersetzt werden soll, ob der Fehler einmal oder immer ignoriert werden soll oder ob ein nicht erkanntes Wort ins Lexikon übernommen werden soll.

11.2.4 Eine einseitige Bewerbung

Wem die oben gezeigte Methode, eine Bewerbung zu schreiben, zu kompliziert ist, findet auf der Webseite <https://www.latextemplates.com> [112] auch einseitige Bewerbungen. Ein gelungenes Beispiel ist der sogenannte 20 Sekunden-Lebenslauf. Er ist ein schönes Beispiel, wie kompakt L^AT_EX sein kann und wie Form und Inhalt getrennt werden. Die Form wird in der Datei *twentysecondcv.cls* festgelegt, der Inhalt steht in der Datei *template.tex*. Weiterhin wird nur noch ein Foto benötigt. Im Beispiel heißt es *alice.jpeg*. Laden Sie das Beispiel so herunter und übersetzen Sie es anschließend:

Code-Ausschnitt 11.20: Beispiel für einen einseitigen L^AT_EX-Lebenslauf

```

cd ~
mkdir latex
cd latex
wget http://www.latextemplates.com/templates/curriculum_vitae/16/cv_16.zip
unzip cv_16.zip
pdflatex template.tex

```



Alice

Adventurer

i

26 November 1865

e

United Kingdom

t

+39 0325658974

g

http://en.wikipedia.org

@

alice@wonderland.com

About me

Alice is a sensible prepubescent girl from a wealthy English family who finds herself in a strange world ruled by imagination and fantasy. Alice feels comfortable with her identity and has a strong sense that her environment is comprised of clear, logical, and consistent rules and features. Alice's familiarity with the world has led one critic to describe her as a "disembodied intellect". Alice displays great curiosity and attempts to fit her diverse experiences into a clear understanding of the world.

Skills

Java

polite

outgoing

good manners

pursuer of rabbits

lovely*4 narcissistic*3

(*)[The skill scale is from 0 (Fundamental Awareness) to 6 (Expert).]

Interests

The heroine and the dreamer of Wonderland; Alice is the principal character.

since 1865

Ph.D. candidate in Computer Science
A Quantified Theory of Social Cohesion.

Education

1863-1865

M.Sc. magna cum laude
Majoring in Computer Science

1861-1863

B.Sc. magna cum laude
Majoring in Computer Science

1856-1861

High school
Specializing in mathematics and physics.

Publications

1865

Chapter One, Down the Rabbit Hole.

1865

Chapter Two, The Pool of Tears.

1865

Chapter Three, The Caucus Race and a Long Tale.

1865

Chapter Four, The Rabbit Sends a Little Bill.

1865

Chapter Five, Advice from a Caterpillar.

Awards

1987

All-Time Best Fantasy Novel.

1998

All-Time Best Fantasy Novel before 1990.

Experience

1900

Alice in Wonderland-The Circra (1900's) Silent Film.
The first Alice on film was over a hundred years ago.

Film

1933

Alice in Wonderland 1933 version.
This film stars Ethel griffies and Charlotte Henry. It was a box office flop when it was released.

Film

1951

Disney Film.
Walt Disney brings Lewis Carroll's fantasy story to life in this well done animated classic. Even though many elements from the book were dropped, such as the duchess with the baby pig and mock turtle, this version is without a doubt the most famous Alice adaption made.

Film

Other information

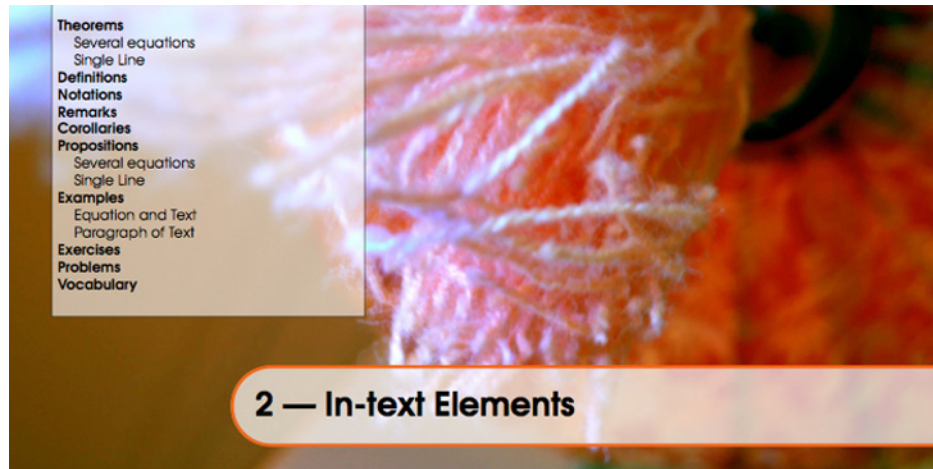
Review

Alice approaches Wonderland as an anthropologist, but maintains a strong sense of noblesse oblige that comes with her class status. She has confidence in her social position, education, and the Victorian virtue of good manners. Alice has a feeling of entitlement, particularly when comparing herself to Mabel, whom she declares has a "poky little house," and no toys. Additionally, she flaunts her limited information base with anyone who will listen and becomes increasingly obsessed with the importance of good manners as she deals with the rude creatures of Wonderland. Alice maintains a superior attitude and behaves with solicitous indulgence toward those she believes are less privileged.

Abbildung 11.7: Ein einseitiger Lebenslauf mit L^AT_EX.

11.2.5 Ein Buch schreiben

\LaTeX eignet sich ebenfalls hervorragend, um ein Buch zu schreiben. Dass das nicht langweilig aussehen muss, beweisen die kostenlosen *Templates* von <http://www.latextemplates.com> [112]. Unter <http://www.latextemplates.com/cat/books> [112] stehen mehrere Buchvorlagen zum Download bereit. Das *The Legrand Orange Book-Template* ist ein Beispiel für professionelles Aussehen. Alle Templates, die auf <http://www.latextemplates.com> [112] verfügbar sind, dürfen sogar für kommerzielle Zwecke verwendet werden, vorausgesetzt, das Projekt gibt einen Verweis auf die o. g. Webseite.



2.1 Theorems

This is an example of theorems.

2.1.1 Several equations

Theorem 2.1 In $E = \mathbb{R}^n$ all norms are equivalent. It has the properties:

$$||\mathbf{x}|| - ||\mathbf{y}|| \leq ||\mathbf{x} - \mathbf{y}|| \quad (2.1)$$

$$||\sum_{i=1}^n \mathbf{x}_i|| \leq \sum_{i=1}^n ||\mathbf{x}_i|| \quad \text{where } n \text{ is a finite integer} \quad (2.2)$$

2.1.2 Single Line

Theorem 2.2 A set $\mathcal{D}(G)$ is dense in $L^2(G)$, $|\cdot|_0$.

2.2 Definitions

This is an example of a definition. A definition could be mathematical or it could define a concept.

Definition 2.1 — Definition name. Given a vector space E , a norm on E is an application, denoted $||\cdot||$, E in $\mathbb{R}^+ = [0, +\infty[$ such that:

$$||\mathbf{x}|| = 0 \Rightarrow \mathbf{x} = \mathbf{0} \quad (2.3)$$

$$||\lambda \mathbf{x}|| = |\lambda| \cdot ||\mathbf{x}|| \quad (2.4)$$

$$||\mathbf{x} + \mathbf{y}|| \leq ||\mathbf{x}|| + ||\mathbf{y}|| \quad (2.5)$$

Abbildung 11.8: Das *The Legrand Orange Book-Template* ist ein professionelles \LaTeX -Template zum Schreiben von Büchern. Quelle: www.latextemplates.com [112].



Das Buch-Template, welches <http://www.latextemplates.com> [112] zur Verfügung stellt, basiert auf englischen Sprachregeln. Ich habe aus dieser Version eine Version erstellt, welche auf deutschen Regeln basiert. Die Änderungen stehen im Download-Bereich zu diesem Buch zur Verfügung.

11.3 PlayOnLinux



Für alle, die sich nicht mit *LibreOffice* oder \LaTeX anfreunden können, gibt es eine weitere Möglichkeit, Dokumente auf dem Wohnzimmer-PC zu verfassen. Um Microsoft Office zu verwenden, muss nicht einmal das Betriebssystem gewechselt werden. *PlayOnLinux* stellt eine schöne Oberfläche zur Verfügung, die es erlaubt, verschiedene *wine*-Versionen zu starten, um damit beispielsweise das Office-Paket von Microsoft unter LINUX zu installieren und zu nutzen. In diesem Abschnitt zeigen ich Ihnen, wie Sie Office-Programme mit Hilfe von *PlayOnLinux* installieren und benutzen können. Die Installation ist denkbar einfach:

Code-Ausschnitt 11.21: Installation von *PlayOnLinux*

```
sudo apt install playonlinux winbind
```

Allerdings werden dabei so viele weitere Pakete mitinstalliert, dass der gesamte Speicherbedarf der Installation zwischen 700 MB und 800 MB beträgt. Beim ersten Start sieht *PlayOnLinux* noch sehr leer aus. Das ist verständlich, da ja noch kein Windows-Programm installiert wurde.

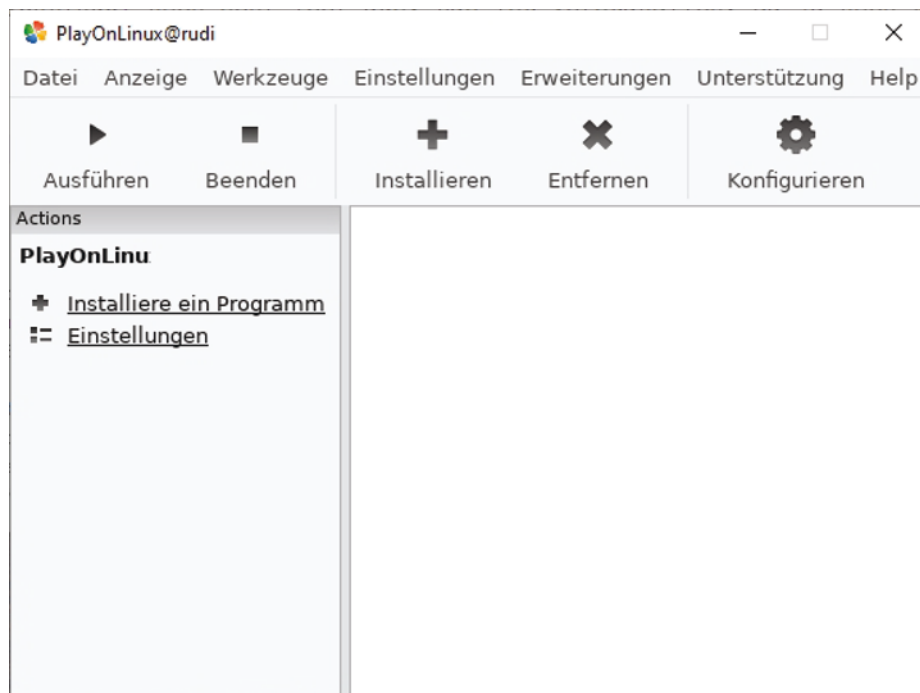


Abbildung 11.9: Beim ersten Start sieht *PlayOnLinux* noch sehr leer aus.

Die weitere Installation habe ich mit einem Microsoft Office 2010-Paket vorgenommen. Auch zum Betrieb unter LINUX benötigen Sie eine gültige Lizenz für ein Office-Paket. Ansonsten laufen Word & Co. nur im Test-Modus. Stellen Sie daher bitte zur weiteren

Installation sicher, dass sich Ihre Office-Installationsdateien in einem Verzeichnis oder auf einer DVD befinden, auf die Ihr LINUX-System zugreifen kann.

Wählen Sie nun die Option *Installiere ein Programm* und klicken Sie auf das *Büro*-Icon. Scrollen Sie dann zur Rubrik *Microsoft Office* und wählen Sie die Version, die Sie installieren möchten. In meinem Fall (Abb. 11.10) ist das *Microsoft Office 2010*. Klicken Sie dann auf *Installieren*.

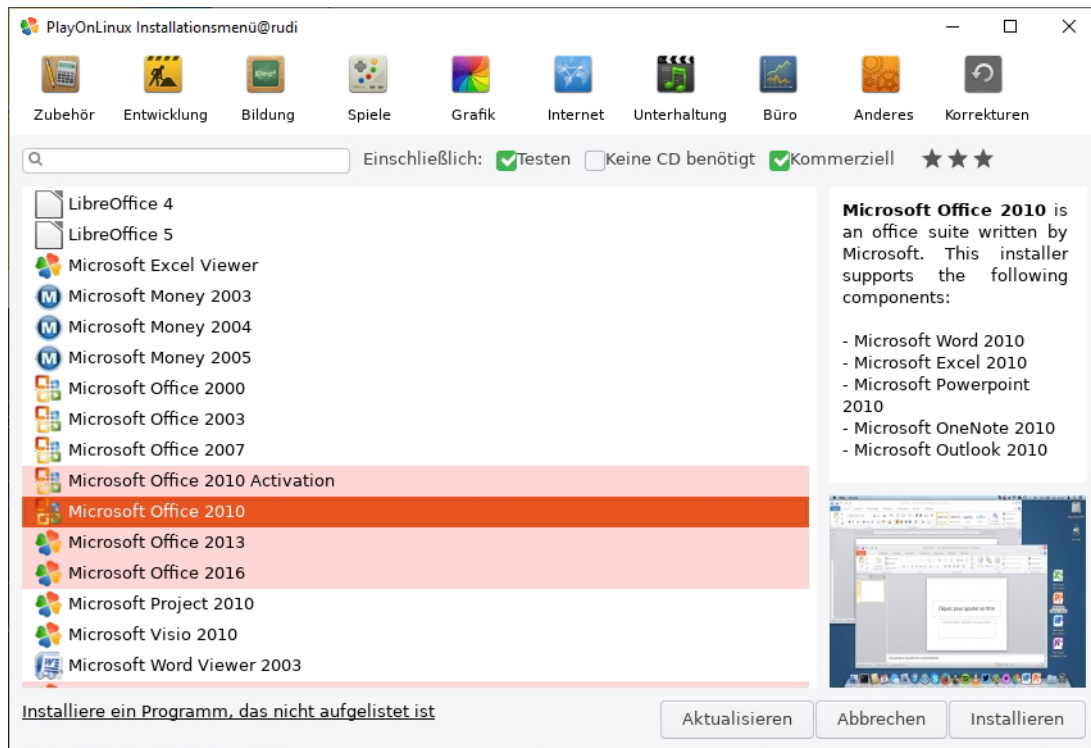


Abbildung 11.10: Unter *PlayOnLinux* sind viele Programme zur Installation bereits vorkonfiguriert.

Wenn Sie zum ersten Mal ein Windows-Programm mit *PlayOnLinux* installieren, erscheint ein Fenster mit verschiedenen Warnungen:

- Bitte verändern Sie für verschiedene Installationen nicht das Default-Installations-Verzeichnis. Nutzen Sie bitte das Standard-Verzeichnis, welches Ihnen unter dem C:-Laufwerk zur Installation angeboten wird.
- Wenn Sie gefragt werden, ob der Installer das frisch installierte Programm für Sie starten soll, antworten Sie bitte mit *nein*.
- Wenn Sie aufgefordert werden, Windows neu zu starten, können Sie mit *ja* antworten. Starten Sie aber auf keinen Fall Ihr LINUX-Betriebssystem zu diesem Zeitpunkt neu.

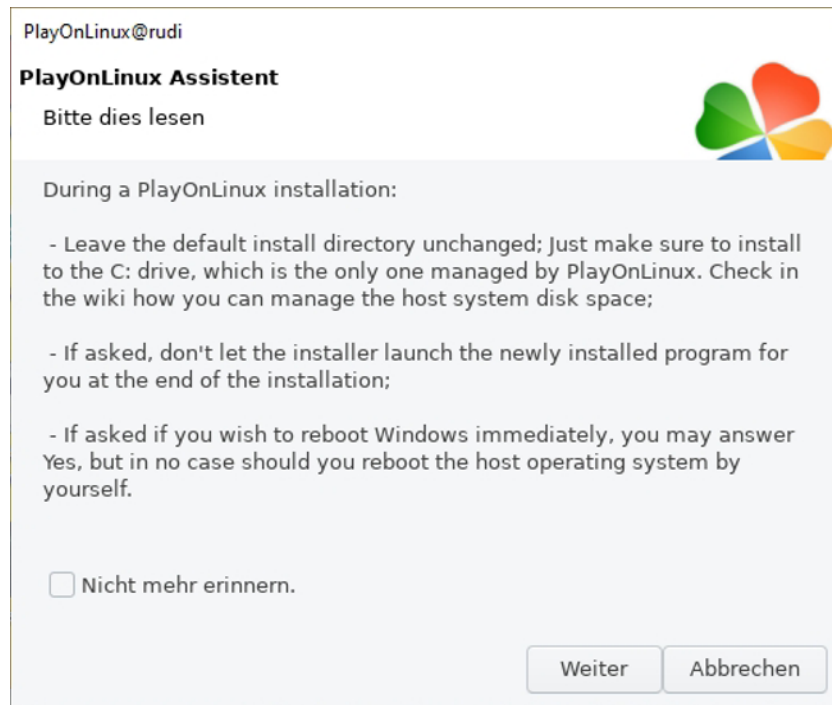


Abbildung 11.11: Bei der Installation von Windows-Programmen gibt der *PlayOnLinux*-Assistent einige Warnungen aus.



Abbildung 11.12: Vor der Installation gibt der *PlayOnLinux*-Installer einige letzte Informationen aus.

Nach einem Klick auf *Weiter* erscheint eine weitere Warnung, die sagt, dass bestimmte Windows-Programme mit bestimmten *wine*-Versionen der Installer zusammenarbeiten. Die verschiedenen Programme sind jeweils nur mit den speziellen *wine*-Versionen getestet. Das bedeutet nicht, dass neuere *wine*-Versionen nicht mit dem jeweiligen Programm funktionieren. Allerdings gibt es hierfür keine Garantie.

Sollten Sie vergessen haben, zusammen mit PlayOnLinux das Programm winbind zu installieren, gibt PlayOnLinux eine Fehlermeldung aus und unterbricht die Installation. Holen Sie in diesem Falle die Installation von winbind nach und wiederholen Sie die Installation des Windows-Programms.



Im Anschluss haben Sie die Wahl, ob Sie die Installation von einer DVD oder aus von einem Verzeichnis heraus auf dem Computer starten möchten.

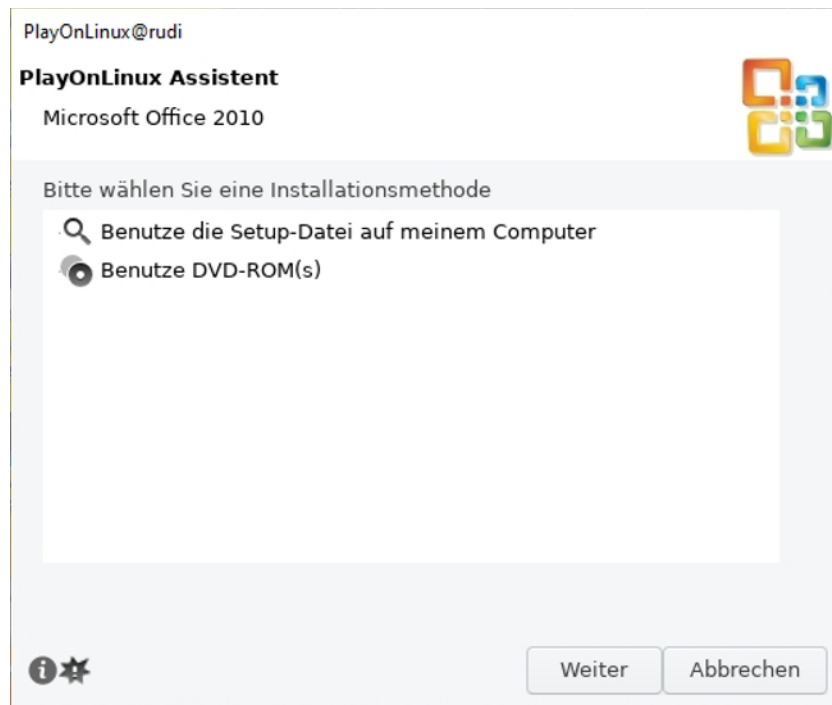


Abbildung 11.13: *PlayOnLinux* erlaubt die Installation von DVD oder aus einem Verzeichnis heraus.

Stellen Sie auf jeden Fall sicher, dass der Installer die Datei *setup.exe* finden kann und wählen Sie diese aus. Der Rest der Installation läuft vollautomatisch ab.

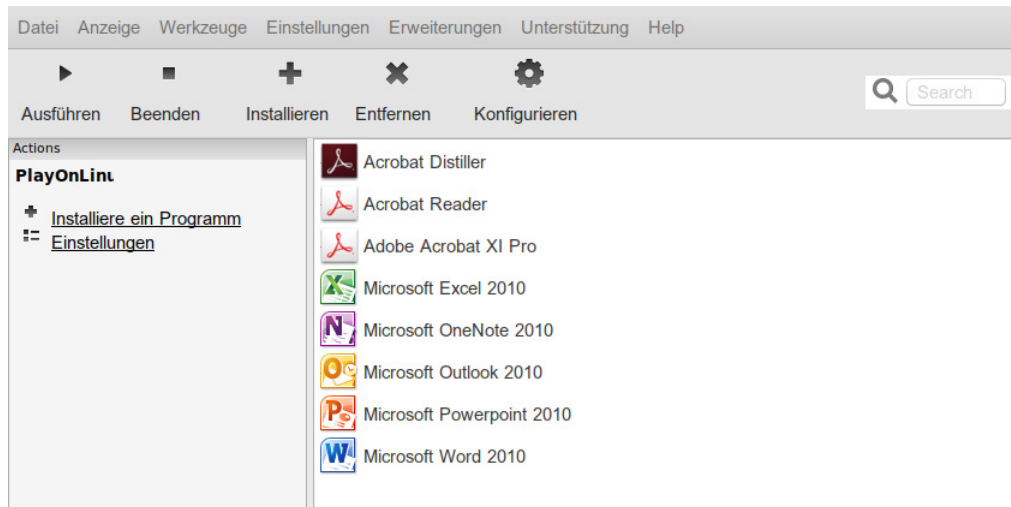


Abbildung 11.14: Nach der Installation zeigt *PlayOnLinux* alle installierten Windows-Programme an.

Ein Doppelklick auf das installierte Programm startet dieses. Auf diese Art können Sie unter Ihrem Ubuntu-Desktop mit Programmen aus dem Microsoft Office-Paket arbeiten.

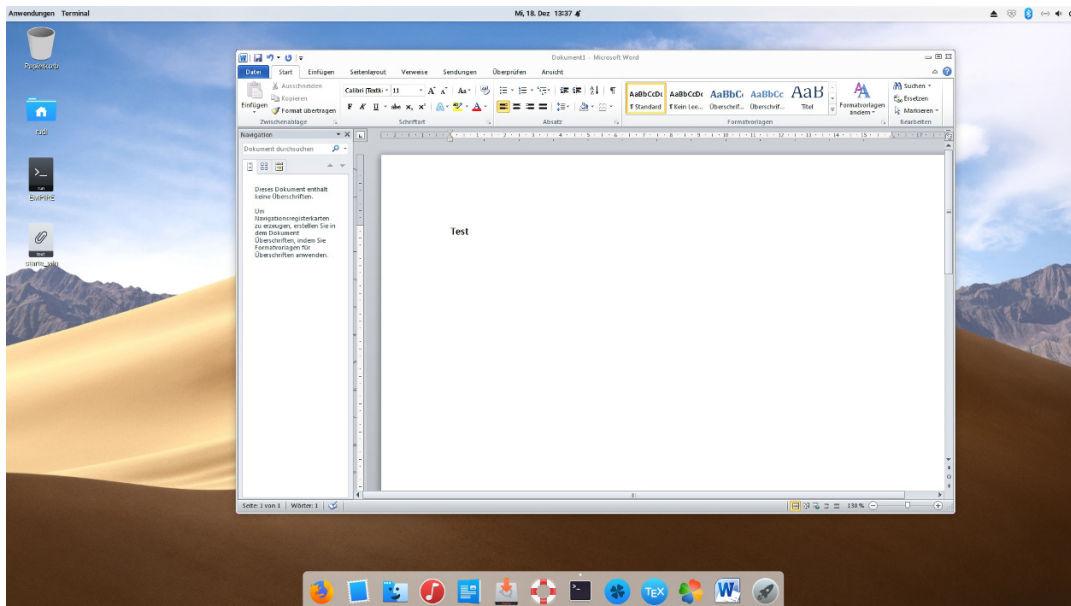


Abbildung 11.15: Microsoft WORD fügt sich nahtlos in den LINUX-Desktop ein.

Zusammenfassung und Ausblick 11 Dieses Kapitel hat Ihnen gezeigt, wie Sie mit Ihrem Wohnzimmer-PC Schriftstücke verfassen können. Während *LibreOffice* sicherlich gut geeignet ist, Briefe zu schreiben, empfehle ich Ihnen \LaTeX für umfangreichere Schriftstücke.

Sicherlich erfordert \LaTeX ein wenig Einarbeitungszeit. Wenn Sie aber vorhaben, Dokumente von mehreren Hundert Seiten zu verfassen, ist dieses Satzsystem die erste Wahl.

Wer dennoch nicht ohne Word & Co. auskommt, kann mit Hilfe von *PlayOnLinux* sogar Windows-Programme unter LINUX nutzen. ■

Literaturverzeichnis

- [108] LibreOffice. *Webseite*. 2019.
<http://de.libreoffice.org/>
besucht am 27. 12. 2019 (siehe Seite 292).
- [109] Wikipedia. *L^AT_EX*. 2019.
<http://de.wikipedia.org/wiki/LaTeX>
besucht am 27. 12. 2019 (siehe Seite 293).
- [110] Github. *Bewerbung*. 2019.
<https://github.com/Stefanqn/Bewerbung>
besucht am 27. 12. 2019 (siehe Seiten 296, 300).
- [111] Wikipedia. *DIN 5008*. 2019.
http://de.wikipedia.org/wiki/DIN_5008
besucht am 27. 12. 2019 (siehe Seite 296).
- [112] Latex Templates. *The best source of free quality LaTeX templates*. 2019.
<https://www.latextemplates.com>
besucht am 27. 12. 2019 (siehe Seiten 301, 303, 304).



Fehlersuche

Die meisten Programme, die Sie übersetzen werden, sind sicherlich C/C++-Programme. Die ersten Fehler im Übersetzungsprozess zeigen sich schon beim Aufruf der Konfiguration mit

Code-Ausschnitt 11.22: Konfiguration vieler C/C++-Programme

```
./configure
```

Meist wird ein Paket nicht gefunden, welches für das Übersetzen des Programmes erforderlich ist. In der Regel wird die Entwicklungsversion, das sogenannte *Developer*-Paket benötigt. Die erkennt man im Repository meist am Zusatz *-dev*. Für die ALSA-Sound-Bibliothek wäre der Name also

Code-Ausschnitt 11.23: Name eines Beispiel-Paketes

```
libasound2-dev
```

./configure gibt oftmals die Namen der fehlenden Pakete an. Sollte dies nicht der Fall sein, empfiehlt sich eine Internetsuche nach dem Namen des fehlenden Paketes. Wenn ein Compiler beim Übersetzen eines Programmes einen Fehler meldet, kann dies viele Ursachen haben:

- Es fehlt eine Header-Datei (*include*). Entweder ist das entsprechende Developer-Paket dann nicht installiert, oder der Pfad zur Header-Datei ist nicht richtig angegeben. Suchen Sie in diesem Fall nach der Header-Datei und schauen Sie, wo sich diese befindet. Header-Dateien liegen meist in den Verzeichnissen */usr/include* oder */usr/include*. Möchten Sie dort nach einer Datei namens *header.h* suchen, können Sie das so machen:

Code-Ausschnitt 11.24: Nach einem Dateinamen suchen

```
cd /usr/include  
find . -name "header.h" -print
```




Der `find`-Befehl findet Dateien, im oberen Fall ab dem aktuellen Verzeichnis (`.`), die `header.h` heißen, und gibt den Namen der Datei inklusive des Pfades an. Diesen Pfad können Sie dann im Fehlerfall korrekt in das C/C++-Programm eintragen.

- Steigt der C-Compiler mit einer anderen Fehlermeldung aus, hilft es oft, den Fehler im Internet zu suchen. Eventuell haben andere schon eine Lösung für den Fehler gefunden und stellen diese bereit.
- Manchmal meckert der Compiler ein *permissive* an. Damit weist er eigentlich auf unsaubere Programmierung hin, die manche Compiler akzeptieren. Der Compiler-Schalter `-fpermissive` erlaubt nicht-konformen Code und das Programm lässt sich eventuell damit übersetzen. Bei meinen Übersetzungsversuchen war dieser Schalter z. B. beim *vomplient* erforderlich.
- Beim Linken eines Programmes fehlt oftmals eine Bibliothek. Diese kann mit der Option `-bibname` beim Linken (z. B. im Makefile) angegeben werden. Ersetzen Sie bitte *bibname* durch den Namen der Bibliothek.
- Oftmals verschwindet der eigentliche Fehler in den Makefile-Ausgaben. Leiten Sie in diesem Fall einfach die Ausgabe in eine Datei um, die Sie dann in aller Ruhe inspizieren können.

Code-Ausschnitt 11.25: Umleitung der Terminal-Ausgabe in eine Datei

```
make >log 2>&1
```

Die Datei *log* enthält dann mögliche Fehler.

- Ein `make install` scheitert oft an den Root-Rechten. Ändern Sie das dann so: `sudo make install`.
- Oftmals sucht man Text in Dateien, gerade bei der Fehleranalyse. Hier ist der `grep`-Befehl eine große Hilfe.

Code-Ausschnitt 11.26: Der `grep`-Befehl zum Suchen in Dateien

```
grep wort *
```

sucht das Wort *wort* in allen (*) Dateien im aktuellen Verzeichnis.



Das *-Zeichen steht für alle Dateien im aktuellen Verzeichnis. Möchten Sie beispielsweise alle Backup-Dateien des Editors *joe* löschen, können Sie das mit dem Befehl

Code-Ausschnitt 11.27: Löschen von Backup-Dateien

```
rm *~
```

- Oftmals ändern sich Funktionsaufrufe, weil sich Parameter ändern oder neue Parameter dazugekommen sind. In diesem Fall müssen die Aufrufe mit den neuen Parametern angepasst werden.
- Viele Repositories ändern sich täglich. Hier hilft es, eine bestimmte Version auszuchecken. Mit `git` kann eine bestimmte Version so ausgecheckt werden:

Code-Ausschnitt 11.28: Auschecken einer bestimmten Version eines Programmes

```
git clone git://repository/project.git  
cd project  
git checkout version
```

Die wichtigsten LINUX-Befehle

Tabelle 11.1 fasst die wichtigsten LINUX-Befehle zusammen.

LINUX-Befehl	Bedeutung
<i>bzip2 datei</i>	Komprimiert eine Datei
<i>cat /proc/cpuinfo</i>	Zeigt Informationen über die CPU an
<i>cd ~</i>	Wechselt in das Benutzer-Home-Verzeichnis
<i>cd verzeichnis</i>	Wechselt in ein Verzeichnis
<i>CTRL-c</i>	Beendet den aktuellen Vorgang im Terminal
<i>cp name1 name2</i>	Kopiert eine Datei
<i>date</i>	Zeigt Datum und Zeit an oder setzt diese
<i>df -hk</i>	Zeigt den freien Speicherplatz an
<i>dmesg</i>	Zeigt die Kernel-Meldungen an
<i>exit</i>	Beendet ein Terminal oder meldet Administrator ab
<i>fdisk</i>	Partitioniert einen Datenträger
<i>find wo name</i>	Sucht nach Dateinamen
<i>free</i>	Zeigt freien Arbeitsspeicher an
<i>grep was datei</i>	Sucht in einer Datei
<i>ifconfig</i>	Zeigt Netzwerk-Informationen an
<i>joe name</i>	Editiert eine Datei
<i>kill -9 pid</i>	Beendet einen Prozess
<i>ls -la</i>	Zeigt alle Dateien und Verzeichnisse im aktuellen Verzeichnis an
<i>man name</i>	Ruft die Hilfeseite zu einem Programm auf
<i>mkdir name</i>	Erzeugt ein neues Verzeichnis
<i>modprobe</i>	Lädt ein Kernel-Modul
<i>more name</i>	Zeigt den Inhalt einer Datei seitenweise an
<i>mount</i>	Bindet Laufwerke in Verzeichnisse ein
<i>mv name1 name2</i>	Benennt eine Datei um oder verschiebt sie
<i>ping ip-adresse</i>	Testet die Verbindung zu einem Rechner
<i>patch j datei</i>	Patched ein Programm mit einer Patch-Datei
<i>ps -elf</i>	Zeigt alle laufenden Prozesse an
<i>ps -elf grep name</i>	Zeigt Prozess-ID eines Prozesses an
<i>pwd</i>	Zeigt das aktuelle Verzeichnis an
<i>rm name</i>	Löscht eine Datei
<i>rmdir name</i>	Löscht ein Verzeichnis
<i>rmmod</i>	Entlädt ein Kernel-Modul
<i>startx</i>	Startet die X-Oberfläche
<i>sudo halt -p</i>	Führt das System herunter
<i>sudo su</i>	Meldet den Systemadministrator an
<i>tail -f /var/log/syslog</i>	Zeigt Systemlog permanent an
<i>tar cvf datei.tar *</i>	Packt alles aus dem aktuellen Verzeichnis in ein Archiv
<i>tar xvfj datei</i>	Packt eine <i>bzip2</i> -komprimierte Datei aus
<i>top</i>	Zeigt alle laufenden Prozesse und die Auslastung an
<i>uname -a</i>	Zeigt die aktuelle Kernel-Version an
<i>vi name</i>	Editiert eine Datei
<i>which name</i>	Zeigt den Pfad zu einem Programm an
<i>who</i>	Zeigt, welche Benutzer im System eingewählt sind
<i>whoami</i>	Zeigt den aktuell eingewählten Benutzer an

Tabelle 11.1: Die wichtigsten LINUX-Befehle.



- [1] Rüdiger Follmann. *Das Raspberry Pi Kompendium*, 2. Auflage. Springer Verlag, Heidelberg, 2018. ISBN: 978-3-662-58143-8 (siehe Seiten XIII, 4).
- [2] DFRobot. *Webseite*. 2019.
<https://www.dfrobot.com>
besucht am 23. 12. 2019 (siehe Seite XV).
- [3] Zotac. *Webseite*. 2019.
<https://www.zotac.com/>
besucht am 23. 12. 2019 (siehe Seite 1).
- [4] tonymacx86. *Building a CustoMac Hackintosh: The Ultimate Buyer's Guide*. 2019.
<https://www.tonymacx86.com/buyersguide/building-a-customac-hackintosh-the-ultimate-buyers-guide/>
besucht am 23. 12. 2019 (siehe Seiten 2, 12, 16).
- [5] tonymacx86. *[Guide] Intel NUC7/NUC8 using Clover UEFI (NUC7i7Bxx,etc)*. 2019.
<https://www.tonymacx86.com/threads/guide-intel-nuc7-nuc8-using-clover-uefi-nuc7i7bxx-nuc8i7bxx-etc.261711/>
besucht am 23. 12. 2019 (siehe Seiten 3, 16).
- [6] silentmaxx. *Webseite*. 2019.
<https://www.silentmaxx.de>
besucht am 23. 12. 2019 (siehe Seiten 3, 4).
- [7] Hardkernel. *ODROID-H2*. 2019.
<https://www.hardkernel.com/shop/odroid-h2/>
besucht am 23. 12. 2019 (siehe Seite 4).
- [8] lattepanda. *Webseite*. 2019.
<https://www.lattepanda.com>
besucht am 23. 12. 2019 (siehe Seite 5).
- [9] dfrobot. *Webseite*. 2019.
<https://www.dfrobot.com/product-1728.html>
besucht am 23. 12. 2019 (siehe Seite 6).

- [10] Intel. *Intel® Core™ m3-7Y30 Prozessor*. 2019.
<https://ark.intel.com/de/products/95449/Intel-Core-m3-7Y30-Processor-4M-Cache-2-60-GHz->
besucht am 23. 12. 2019 (siehe Seite 10).
- [11] NVIDIA. *Unix-Treiber*. 2019.
<https://www.nvidia.de/object/unix-de.html>
besucht am 23. 12. 2019 (siehe Seiten 11, 63).
- [12] AMD. *AMD Drivers and Support*. 2019.
<https://www.amd.com/en/support>
besucht am 23. 12. 2019 (siehe Seite 11).
- [13] Apple. *Webseite*. 2019.
<https://www.apple.de>
besucht am 12. 12. 2018 (siehe Seite 11).
- [14] Samsung. *970 EVO NVMe M.2 SSD*. 2019.
<https://www.samsung.com/de/memory-storage/970-evo-nvme-m-2-ssd/MZ-V7E1T0BW/>
besucht am 23. 12. 2019 (siehe Seite 17).
- [15] gparted. *GNOME Partition Editor*. 2019.
<https://gparted.org/download.php>
besucht am 23. 12. 2019 (siehe Seite 18).
- [16] Sourceforge. *Win32 Disk Imager*. 2019.
<https://sourceforge.net/projects/win32diskimager/>
besucht am 23. 12. 2019 (siehe Seite 18).
- [17] Jason Harder. *So ein HackMac – Projekt Hackintosh*. 2016.
<http://printpraxis.net/2016/10/07/so-ein-hackmac-projekt-hackintosh-die-installation>
besucht am 20. 12. 2018 (siehe Seite 20).
- [18] Tu Nguyen. *Create macOS Mojave bootable USB from Windows with Clover Boot Disk*. 2019.
<https://sourceforge.net/projects/win32diskimager/>
besucht am 23. 12. 2019 (siehe Seiten 20, 22).
- [19] Rehabman. *Repositories*. 2019.
<https://bitbucket.org/RehabMan/>
besucht am 23. 12. 2019 (siehe Seite 21).
- [20] tonymacx86. *Kexts*. 2019.
<https://www.tonymacx86.com/resources/categories/kexts.11/>
besucht am 23. 12. 2019 (siehe Seite 21).
- [21] novaspirit. *macpanda*. 2019.
<https://github.com/novaspirit/macpanda>
besucht am 23. 12. 2019 (siehe Seiten 21, 35).
- [22] Github. *CloverBootloader*. 2019.
<https://github.com/CloverHackyColor/CloverBootloader/releases>
besucht am 23. 12. 2019 (siehe Seite 25).

- [23] mackie Projects. *Download Clover Configurator*. 2019.
<https://mackie100projects.altervista.org/download-clover-configurator/>
besucht am 23. 12. 2019 (siehe Seite 27).
- [24] tonymacx86.com. *iTunes Protected Video Black Screen Solution (HDCP)*. 2019.
<https://www.tonymacx86.com/threads/itunes-protected-video-black-screen-solution-hdcp.195083/>
besucht am 23. 12. 2019 (siehe Seite 35).
- [25] Cyberlink. *Webseite*. 2019.
<https://de.cyberlink.com/stat/bd-support/deu/index.jsp>
besucht am 23. 12. 2019 (siehe Seite 35).
- [26] 4kmedia. *Webseite*. 2019.
<https://4kmedia.org/big-buck-bunny-4k-demo/>
besucht am 23. 12. 2019 (siehe Seite 39).
- [27] Epicgames. *Fortnite*. 2019.
<https://www.epicgames.com/fortnite/download>
besucht am 23. 12. 2019 (siehe Seite 40).
- [28] ubuntu releases. *These releases of Ubuntu are available*. 2019.
<http://releases.ubuntu.com>
besucht am 27. 12. 2019 (siehe Seite 48).
- [29] 4ksamples. *Free Downloads 4k Sample Content*. 2019.
<http://4ksamples.com>
besucht am 27. 12. 2019 (siehe Seite 64).
- [30] putty. *Download PuTTY*. 2019.
<https://www.putty.org>
besucht am 27. 12. 2019 (siehe Seite 80).
- [31] Wikipedia. *Xming*. 2019.
<https://de.wikipedia.org/wiki/Xming>
besucht am 27. 12. 2019 (siehe Seite 80).
- [32] MobaXTerm. *Home Edition*. 2019.
<https://mobaxterm.mobatek.net/download.html>
besucht am 27. 12. 2019 (siehe Seite 80).
- [33] RealVNC. *VNC Connect*. 2019.
<https://www.realvnc.com/de/connect/download/viewer/>
besucht am 27. 12. 2019 (siehe Seite 82).
- [34] OpenVPN. *OpenVPN Connect for Windows*. 2019.
<https://openvpn.net/client-connect-vpn-for-windows/>
besucht am 27. 12. 2019 (siehe Seite 93).
- [35] WinSCP. *Free Award-Winning File Manager*. 2019.
<https://winscp.net/eng/index.php>
besucht am 27. 12. 2019 (siehe Seiten 94, 95).
- [36] Python. *Webseite*. 2019.
<http://www.python.org>
besucht am 27. 12. 2019 (siehe Seite 97).

- [37] wxWidgets. *wxWidgets 3.1.2*. 2019.
<https://github.com/wxWidgets/wxWidgets/releases/tag/v3.1.2>
besucht am 27. 12. 2019 (siehe Seite 101).
- [38] VeraCrypt. *Open source disk encryption with strong security for the Paranoid*. 2019.
<https://github.com/wxWidgets/wxWidgets/releases/tag/v3.1.2>
besucht am 27. 12. 2019 (siehe Seite 102).
- [39] gnome-look.org. *Eyecandy for your Gnome Desktop*. 2019.
<https://www.gnome-look.org/>
besucht am 27. 12. 2019 (siehe Seite 106).
- [40] McMojave. *GTK3 Themes*. 2019.
<https://www.gnome-look.org/p/1275087/>
besucht am 27. 12. 2019 (siehe Seite 106).
- [41] trongthanh. *mojave.xml*. 2019.
<https://gist.github.com/trongthanh/7d632e90687e1bc219e1f3262d337702>
besucht am 27. 12. 2019 (siehe Seite 109).
- [42] Nintendo. *Super Mario Run*. 2019.
<https://supermariorun.com/de/index.html>
besucht am 27. 12. 2019 (siehe Seite 111).
- [43] Cemu - Wii U Emulator. *Experimental software to emulate Wii U applications on PC*. 2019.
<http://cemu.info>
besucht am 27. 12. 2019 (siehe Seite 111).
- [44] Woo. *Use your PS4 Controller on your PC to its full potential*. 2019.
<http://ds4windows>
besucht am 27. 12. 2019 (siehe Seite 113).
- [45] WineHQ. *Wine Download Server*. 2019.
<https://dl.winehq.org/>
besucht am 27. 12. 2019 (siehe Seite 116).
- [46] Cemu hook. *Current versions*. 2019.
<https://cemuhook.sshnuke.net/#Downloads>
besucht am 27. 12. 2019 (siehe Seite 119).
- [47] Cemu Graphic Packs. *Official website of the Community Graphic Packs for Cemu, used to upscale and change your Wii U games*. 2019.
https://slashiee.github.io/cemu_graphic_packs/
besucht am 27. 12. 2019 (siehe Seite 119).
- [48] yuzu emulator team. *Nintendo Switch Emulator*. 2019.
<https://yuzu-emu.org/>
besucht am 27. 12. 2019 (siehe Seite 123).
- [49] Dolphin Emulator Project. *Dolphin Emulator*. 2019.
<http://de.dolphin-emu.org>
besucht am 27. 12. 2019 (siehe Seite 126).
- [50] Github. *Dolphin Quelltext*. 2019.
<https://github.com/dolphin-emu/dolphin>
besucht am 27. 12. 2019 (siehe Seite 126).

- [51] Sourceforge. *A frontend for Mupen64Plus*. 2019.
<http://m64py.sourceforge.net/>
besucht am 27. 12. 2019 (siehe Seite 130).
- [52] Lutris. *Play all your games on Linux*. 2019.
<https://lutris.net/>
besucht am 27. 12. 2019 (siehe Seiten 132, 133).
- [53] Flatpak. *The Future of Apps On Linux*. 2019.
<https://flatpak.org/>
besucht am 27. 12. 2019 (siehe Seite 133).
- [54] Klaus Schmidinger. *VDR - The Video Disk Recorder*. 2019.
<http://www.tvdr.de>
besucht am 27. 12. 2019 (siehe Seite 137).
- [55] VDR Wiki. *Herzlich Willkommen auf den Seiten von VDR Wiki!* 2019.
<http://www.vdr-wiki.de>
besucht am 27. 12. 2019 (siehe Seiten 137, 138, 154, 161, 176, 180, 195).
- [56] Ludmila Rakova. *VDR Portal*. 2019.
<http://www.vdr-portal.de>
besucht am 27. 12. 2019 (siehe Seiten 137, 158, 169, 173, 199).
- [57] Dipl.-Ing. Claus Muus. *MiniDVBLinux Distribution (MLD)*. 2019.
<http://www.minidvblinux.de>
besucht am 27. 12. 2019 (siehe Seite 137).
- [58] Sundtek. *Netzwerk Installer*. 2019.
http://www.sundtek.de/media/sundtek_netinst.sh
besucht am 27. 12. 2019 (siehe Seite 138).
- [59] DVBSky. *Webseite*. 2019.
<http://dvbsky.net>
besucht am 27. 12. 2019 (siehe Seite 142).
- [60] LinuxTV. *Multimedia and Television Support on Linux*. 2019.
<http://www.linuxtv.org>
besucht am 27. 12. 2019 (siehe Seite 143).
- [61] Flirc. *Homepage*. 2019.
<https://flirc.tv/>
besucht am 27. 12. 2019 (siehe Seite 145).
- [62] Wiki. *Lirc*. 2019.
<https://wiki.ubuntuusers.de/Lirc/>
besucht am 27. 12. 2019 (siehe Seite 149).
- [63] Lirc homepage. *Lirc Infrared Linux Remote Control*. 2019.
<http://www.lirc.org/html/table.html>
besucht am 27. 12. 2019 (siehe Seite 150).
- [64] Sourceforge. *Lirc*. 2019.
<http://lirc.sourceforge.net/lirc.org/>
besucht am 27. 12. 2019 (siehe Seite 150).
- [65] Github. *A VA-API output device plugin for VDR*. 2019.
<https://github.com/pesintta/vdr-plugin-vaapidevice>
besucht am 27. 12. 2019 (siehe Seite 156).

- [66] ffmpeg. *Index of /releases*. 2019.
<https://www.ffmpeg.org/releases>
besucht am 27. 12. 2019 (siehe Seite 156).
- [67] Github. *A VA-API output device plugin for VDR*. 2019.
<https://github.com/rofafor/vdr-plugin-vaapidevice>
besucht am 27. 12. 2019 (siehe Seite 157).
- [68] Github. *A VA-API output device plugin for VDR*. 2019.
<https://github.com/jojo61/vdr-plugin-softhdcuvid>
besucht am 27. 12. 2019 (siehe Seite 158).
- [69] vdr-developer.org. *vdradmin-am.git*. 2019.
<http://projects.vdr-developer.org/git/vdradmin-am.git/>
besucht am 27. 12. 2019 (siehe Seite 168).
- [70] vdr-developer.org. *VDR Skindesigner Plugin*. 2019.
<https://projects.vdr-developer.org/projects/plg-skindesigner/wiki>
besucht am 27. 12. 2019 (siehe Seite 169).
- [71] FrodoVDR. *channellogos*. 2019.
<https://github.com/FrodoVDR/channellogos>
besucht am 27. 12. 2019 (siehe Seite 172).
- [72] Webseite. *Shady-Skin*. 2019.
<http://anthra.de/shady-skin>
besucht am 27. 12. 2019 (siehe Seite 176).
- [73] QooTec. *3.5 inch HDMI LCD*. 2019.
https://www.amazon.de/QooTec-Touchscreen-Raspberry-Touchpen-Raspbian/dp/B07PLVQC78/ref=sr_1_8?__mk_de_DE=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=3.5+zoll+hdmilcd&qid=1577437846&s=computers&sr=1-8
besucht am 27. 12. 2019 (siehe Seite 180).
- [74] Wiki. *EasyVDR*. 2019.
<https://wiki.easy-vdr.de>
besucht am 27. 12. 2019 (siehe Seite 184).
- [75] warmcat. *libwebsockets releases*. 2019.
<https://github.com/warmcat/libwebsockets/releases>
besucht am 27. 12. 2019 (siehe Seite 184).
- [76] Ask Ubuntu. *Screen Corruption on Multi Monitor DisplayPort using Intel Graphics*. 2019.
<https://askubuntu.com/questions/995077/screen-corruption-on-multi-monitor-displayport-using-intel-graphics?rq=1>
besucht am 27. 12. 2019 (siehe Seite 190).
- [77] Christian Hennek. *Futaba MDM166A USB VFD Display*. 2019.
http://www.eisgold.de/Futaba-MDM166A-USB-VFD-Display_detail_2732.html?osCsid=4c0613e6151ebd8cb71ed9724ef357ac
besucht am 01. 01. 2020 (siehe Seite 193).
- [78] vlc. *VLC Media Player*. 2019.
<https://www.vlc.de/>
besucht am 27. 12. 2019 (siehe Seite 195).

- [79] Digital Devices. *Webseite*. 2019.
<https://digitaldevices.de>
besucht am 27. 12. 2019 (siehe Seite 196).
- [80] TBS technology. *DVB Produkte*. 2019.
<https://www.tbs-technology.de>
besucht am 27. 12. 2019 (siehe Seite 197).
- [81] Gary Hunt. *TBS Kernel Treiber*. 2019.
https://github.com/tbsdtv/linux_media/wiki
besucht am 27. 12. 2019 (siehe Seite 197).
- [82] Hauppauge. *WinTV USB-CI*. 2019.
<https://www.amazon.de/Hauppauge-WinTV-CI-Common-Interface-Karte/dp/B000Y19S12>
besucht am 27. 12. 2019 (siehe Seite 198).
- [83] Helmut Binder. *USB2CI Treiber*. 2019.
<https://github.com/siricco/usb2ci>
besucht am 27. 12. 2019 (siehe Seite 198).
- [84] Matt Vollrath. *get-edid.py*. 2019.
<https://gist.githubusercontent.com/mvollrath>
besucht am 27. 12. 2019 (siehe Seite 200).
- [85] lmd. „Multimedia-Zwerg“. In: *c't magazin für computer technik* 3 (Jan. 2014),
Seiten 90–93 (siehe Seite 209).
- [86] OpenELEC. *OpenELEC*. 2019.
<https://openelec.tv>
besucht am 27. 12. 2019 (siehe Seite 209).
- [87] LibreELEC. *LibreELEC Blog*. 2019.
<https://libreelec.tv/>
besucht am 27. 12. 2019 (siehe Seiten 209, 214, 215, 227).
- [88] Wikipedia. *libdvdcss*. 2019.
<http://de.wikipedia.org/wiki/Libdvdcss>
besucht am 27. 12. 2019 (siehe Seite 212).
- [89] kodinerds.net. *Deutschsprachiges Forum zum Kodi Entertainment Center*. 2019.
<https://www.kodinerds.net/>
besucht am 27. 12. 2019 (siehe Seite 214).
- [90] picons. *Intro*. 2019.
<https://www.picons.eu/>
besucht am 27. 12. 2019 (siehe Seite 224).
- [91] Kodi. *Milhouse Test Builds*. 2019.
<https://forum.kodi.tv/showthread.php?tid=343069>
besucht am 27. 12. 2019 (siehe Seite 234).
- [92] Wikipedia. *Ambilight*. 2020.
<https://de.wikipedia.org/wiki/Ambilight>
besucht am 01. 01. 2020 (siehe Seite 237).

- [93] Amazon. *APA102(c)*. 2020.
https://www.amazon.de/APA102C-Adressierbares-Streifenlicht-wasserdicht-Schwarz/dp/B06WVHYMGK/ref=sr_1_5?__mk_de_DE=%C3%85M%C3%85%C5BD%C3%95%C3%91&keywords=apa102c+led+streifen&qid=1577966651&sr=8-5
 besucht am 01. 01. 2020 (siehe Seite 238).
- [94] Amazon. *Arduino Nano*. 2020.
https://www.amazon.de/Moligh-doll-Atmega328-Kontroller-Arduino/dp/B07QMPPHPT/ref=sr_1_3?__mk_de_DE=%C3%85M%C3%85%C5BD%C3%95%C3%91&keywords=arduino+nano&qid=1577967107&s=diy&sr=1-3
 besucht am 01. 01. 2020 (siehe Seite 239).
- [95] Arduino. *Arduino Forum*. 2020.
<https://forum.arduino.cc/index.php?topic=577916.0>
 besucht am 01. 01. 2020 (siehe Seite 240).
- [96] Arduino. *Arduino*. 2019.
<http://www.arduino.cc>
 besucht am 27. 12. 2019 (siehe Seite 242).
- [97] Daniel Garcia. *Fast, easy LED library for Arduino*. 2019.
<http://fastled.io/>
 besucht am 27. 12. 2019 (siehe Seite 242).
- [98] LibreELEC. *Hypercon*. 2019.
<https://wiki.libreelec.tv/hypercon>
 besucht am 27. 12. 2019 (siehe Seite 250).
- [99] Oracle. *JAVA Download*. 2019.
<http://www.java.com/en/>
 besucht am 27. 12. 2019 (siehe Seite 250).
- [100] LibreELEC. *Hypercon*. 2020.
<https://libreelec.wiki/hypercon>
 besucht am 01. 01. 2020 (siehe Seite 251).
- [101] KODI. *HOW-TO:HelloWorld addon*. 2019.
https://kodi.wiki/view/HOW-TO:HelloWorld_addon
 besucht am 27. 12. 2019 (siehe Seite 253).
- [102] Amazon. *AZDelivery 433 MHz Funk - Sende und Empfänger Modul Set*. 2020.
https://www.amazon.de/AZDelivery-%E2%AD%90%E2%AD%90%E2%AD%90%E2%AD%90%E2%AD%90-433-MHz-Funk/dp/B01N5GV39I/ref=sr_1_7?__mk_de_DE=%C3%85M%C3%85%C5BD%C3%95%C3%91&keywords=433+mh+sender+empf%C3%A4nger&qid=1577969412&sr=8-7
 besucht am 01. 01. 2020 (siehe Seite 256).
- [103] Maik Schmidt. *Raspberry Pi: Einstieg - Optimierung - Projekte*. 1. Auflage. Band 1. Heidelberg: dpunkt-Verlag, 2013 (siehe Seite 268).
- [104] Athemes. *ALIZEE*. 2019.
<https://athemes.com/theme/alizee/>
 besucht am 27. 12. 2019 (siehe Seite 280).
- [105] Peter Müller. *Einstieg in WordPress 5*. Rheinwerk Computing, 2019. ISBN: ISBN 978-3-8362-6155-5 (siehe Seite 281).

- [106] Sourceforge. *ReadyMedia*. 2019.
<https://sourceforge.net/projects/minidlna/>
besucht am 27. 12. 2019 (siehe Seite 282).
- [107] Github. *phue*. 2019.
<https://github.com/studioimaginaire/phue>
besucht am 27. 12. 2019 (siehe Seite 285).
- [108] LibreOffice. *Webseite*. 2019.
<http://de.libreoffice.org/>
besucht am 27. 12. 2019 (siehe Seite 292).
- [109] Wikipedia. *L^AT_EX*. 2019.
<http://de.wikipedia.org/wiki/LaTeX>
besucht am 27. 12. 2019 (siehe Seite 293).
- [110] Github. *Bewerbung*. 2019.
<https://github.com/Stefanqn/Bewerbung>
besucht am 27. 12. 2019 (siehe Seiten 296, 300).
- [111] Wikipedia. *DIN 5008*. 2019.
http://de.wikipedia.org/wiki/DIN_5008
besucht am 27. 12. 2019 (siehe Seite 296).
- [112] Latex Templates. *The best source of free quality LaTeX templates*. 2019.
<https://www.latextemplates.com>
besucht am 27. 12. 2019 (siehe Seiten 301, 303, 304).



2.1	Der ZOTAC CI660 ist ein luftgekühlter Intel-Bolide. Quelle: http://zotac.com [3].	1
2.2	Der Intel NUC7i7DNKE ist Hackintosh-kompatibel	3
2.3	Silentmaxx bietet schöne und lautlose PCs, die sich auf jeden Fall für den Einsatz im Wohnzimmer eignen. Quelle: http://silentmaxx.de [6]. . . .	4
2.4	Der Odroid-H2 ist preiswert, unterstützt aber kein MacOS. Quelle: https://www.hardkernel.com/shop/odroid-h2/ [7]	4
2.5	Der Lattepanada Alpha 864 bietet alles, was man für die Projekte in diesem Buch benötigt. Quelle: http://lattepanada.com [8].	5
3.1	Die Samsung MZ-V7E500BW 970 EVO ist eine beliebte NVMe-Festplatte. Quelle: https://www.samsung.com [14].	17
3.2	Win32diskimager schreibt unter Windows komfortabel ISO-Images auf den USB Stick.	18
3.3	gparted erlaubt es auf einfache Weise, Festplattenpartitionen anzulegen oder zu ändern.	19
3.4	Der <i>Clover Booth Disk Creator</i> erlaubt es, einen USB-Bootstick mit MacOS unter Windows zu erstellen.	21
3.5	So formatieren Sie den USB-Bootstick unter MacOS richtig.	24
3.6	Die Installation des Clover EFI Bootmanagers ist schnell erledigt.	25
3.7	Ändern Sie den Speicherort des Bootloaders, da er ansonsten auf der Festplatte landet.	26
3.8	Ändern Sie bitte die Optionen, die Sie für Ihr Board benötigen.	26
3.9	Das Festplatten-Dienstprogramm unter MacOS erlaubt das Partitionieren der Festplatte(n).	27
3.10	Das Festplatten-Dienstprogramm gibt nicht sofort alle Einstellungen preis.	28
3.11	Ein Klick auf <i>View</i> und <i>Show All Devices</i> gibt weitere Informationen aus.	28
3.12	Das Bootmenü des Clover Bootloaders lässt sich einfach konfigurieren.	29
3.13	Der Clover EFI Bootloader erlaubt es, grafisch auszuwählen, wohin der Bootloader installiert werden soll.	30

3.14	Das Launchpad zeigt unter MacOS alle installierten Programme.	31
3.15	Das Terminal-Programm finden Sie unter <i>Andere</i>	31
3.16	Im MacOS-Terminal können Sie Befehle eingeben.	32
3.17	Startbildschirm von Mojave mit der Information <i>Über diesen Mac</i>	34
3.18	Cyberlinks Blue-ray Advisor sagt Ihnen unter Windows, ob Ihr System HDCP unterstützt und wenn ja, in welcher Version.	36
3.19	iMovie ist in drei Bereiche aufgeteilt: Medien, Vorschau und Zeitleiste. . .	37
3.20	iMovie bietet eine Fülle an Übergängen, die dosiert eingesetzt werden sollten. .	38
3.21	iMovie stellt viele Audio-Schnipsel zum Einfügen zur Verfügung. Musik kann von iTunes importiert werden.	39
3.22	iMovie kann auf meinem Hackintosh Filme in UHD-Auflösung exportieren. .	40
3.23	Fortnite kann kostenlos von der Epic-Games-Seite heruntergeladen werden. .	41
3.24	Der Fortnite-Download nimmt viel Zeit in Anspruch.	42
3.25	Die empfohlenen Fortnite-Einstellungen für schwächere CPUs und GPUs. . .	42
4.1	Vor der Installation können Sie Ubuntu ausprobieren.	48
4.2	Ubuntu-Installation: Auswahl der Sprache.	49
4.3	Ubuntu-Installation: Auswahl der Tastatur.	49
4.4	Ubuntu-Installation: Auswahl des Netzwerkes.	50
4.5	Ubuntu-Installation: Updates und Software.	50
4.6	Ubuntu-Installation: Festlegen der Installationsart 1.	51
4.7	Ubuntu-Installation: Festlegen der Installationsart 2.	51
4.8	Ubuntu-Installation: Festlegen der Installationsart 3.	53
4.9	Ubuntu-Installation: Erstellen einer neuen Partition 1.	53
4.10	Ubuntu-Installation: Erstellen einer neuen Partition 2.	54
4.11	Ubuntu-Installation: Erstellen einer neuen Partition 3.	54
4.12	Ubuntu-Installation: Bearbeiten einer Partition.	54
4.13	Ubuntu-Installation: Speichern der Änderungen an der Partitionstabelle. .	56
4.14	Ubuntu-Installation: Auswahl der Zeitzone.	57
4.15	Ubuntu-Installation: Anlegen eines Benutzers.	58
4.16	Ubuntu-Installation: Während der Installation.	59
4.17	Ubuntu-Installation: Abschluss der Installation.	59
4.18	Der Ubuntu-Schreibtisch nach der Installation.	60
4.19	Die Programmvorschau unter Ubuntu.	61
4.20	Das Terminal-Fenster unter Ubuntu.	62
4.21	glxgears zeigt OpenGL-beschleunigte Zahnräder.	65
4.22	Aktivieren der Quelltext-Pakete in Ubuntu.	67
4.23	Grafische Installation von Ubuntu-Aktualisierungen.	69
4.24	Mit dem Programm <i>Ubuntu-Software</i> kann man Programme suchen, installieren und deinstallieren.	70
5.1	Die Netzwerkeinstellungen sind in der oberen Ubuntu-Leiste zu finden. .	73
5.2	Alle LAN-Einstellungen können nach einem Klick auf das Zahnrad vorgenommen werden.	74
5.3	So stellt man unter Ubuntu eine feste IP-Adresse ein.	74
5.4	Ubuntu stellt alle verfügbaren WLAN-Netze übersichtlich dar.	76
5.5	Für einen Zugangspunkt vergibt Ubuntu automatisch den Hotspot-Namen und den Zugangs-Code.	76
5.6	Die Hilfe zum Editor <i>joe</i> kann man im Editor selbst aufrufen.	78

5.7	<i>xemacs</i> ist ein Editor, der von der grafischen Oberfläche Gebrauch machen kann.	79
5.8	Mit <i>putty</i> kann man sich von einem Windows-System mit dem Wohnzimmer-PC erbinden.	80
5.9	<i>MobaXterm</i> bringt eine praktische X11-Oberfläche mit.	81
5.10	Die grafische Oberfläche von <i>MobaXterm</i> unterstützt sogar OpenGL-beschleunigte Ausgaben, hier am Beispiel von <i>glmark2</i>	81
5.11	Der <i>vncviewer</i> speichert die einmal verbundenen Rechner.	82
5.12	Der <i>vncviewer</i> macht den Ubuntu-Desktop unter anderen Betriebssystemen sichtbar.	83
5.13	Die Installation von <i>dropbox</i> lädt weitere Pakete aus dem Internet nach. .	84
5.14	Die Eingabe der <i>dropbox</i> -Benutzerdaten erfolgt per Web-Interface.	84
5.15	Nach der Installation erscheint das <i>dropbox</i> -Logo in der oberen Ubuntu-Leiste.	85
5.16	Der LINUX-Dateimanager <i>nautlius</i> zeigt alle <i>dropbox</i> -Dateien an.	85
5.17	<i>OpenVPN Connect</i> unter Windows zeigt neben dem Verbindungsstatus auch die übertragene Datenmenge grafisch an.	94
5.18	Mit <i>Winscp</i> können Dateien von Ubuntu nach Windows kopiert werden (Quelle: winscp.net [35]).	95
5.19	Mit Hilfe der <i>gnome-tweaks</i> kann das Aussehen der Ubuntu-Oberfläche komplett geändert werden.	107
5.20	Mit <i>McMojave</i> sieht der Ubuntu-Desktop aus wie <i>MacOS Mojave</i>	107
5.21	Wählen Sie im Erscheinungsbild das soeben heruntergeladene MacOS-Thema aus.	108
6.1	Quelle: supermariorun.com [42]	111
6.2	Ein PS4-Controller eignet sich hervorragend zum Spielen mit <i>cemu</i> . Sogar das Touchpad funktioniert einwandfrei. Quelle: saturn.de	112
6.3	Verbindung eines Bluetooth-Controllers unter Windows.	113
6.4	<i>DS4Windows</i> sorgt dafür, dass ein PS4-Controller unter Windows mit <i>cemu</i> verwendet werden kann.	113
6.5	<i>cemu</i> macht Super Mario 3D World unter Windows spielbar.	114
6.6	<i>cemu</i> -Einstellungen für einen PS4-Controller, der mit <i>DS4Windows</i> eingebunden ist.	114
6.7	Der Start-Bildschirm von Super Mario 3D World in <i>cemu</i>	115
6.8	Viel Spaß macht Super Mario 3D World erst im Vollbild-Modus.	116
6.9	Quelle: Wikipedia.	116
6.10	Konfiguration der Windows-Version für <i>wine</i>	117
6.11	Konfiguration der Windows-Version für <i>wine</i> - Fortsetzung.	118
6.12	Diese Warnung müssen Sie mit <i>Abbrechen</i> quittieren, bevor <i>cemu</i> startet. .	120
6.13	Unter Ubuntu kann man einen PS4-Controller ohne Probleme per Bluetooth verbinden.	121
6.14	PS4-Controller-Einstellungen für <i>cemu</i> unter <i>wine</i>	121
6.15	Super Mario 3D World für <i>cemu</i> unter <i>wine</i>	122
6.16	Super Mario Odyssey unter <i>yuzu</i> . Quelle: https://yuzu-emu.org/ [48]. .	123
6.17	Auch für den Switch-Emulator <i>yuzu</i> können Bluetooth-Controller einfach konfiguriert werden.	124
6.18	Super Mario Odessey mit <i>yuzu</i> auf dem Ubuntu-Desktop.	125
6.19	Super Mario Galaxy im <i>dolphin</i> -Emulator. Quelle: http://de.dolphin-emu.org [49].	126

6.20	Der <i>dolphin</i> -Emulator auf dem Ubuntu-Desktop.	127
6.21	Super Mario 64 ist ein Klassiker aus dem Jahre 1996. Damals servierte Nintendo noch im 4:3-Format.	128
6.22	Der Nintendo 64-Emulator <i>mupen64plus</i> auf dem Ubuntu-Desktop. . . .	129
6.23	Zelda in der Oberfläche von M64Py. Quelle: http://m64py.sourceforge.net/ [51].	130
6.24	Einstellung des Controllers in der Oberfläche von <i>M64Py</i>	131
6.25	Supertuxkart erinnert ein wenig an Mario Kart.	132
6.26	<i>lutris</i> ist eine Open-Source-Spieleplattform für LINUX. Quelle: https://lutris.net/ [52].	133
7.1	Das VDR-Portal ist die erste Anlaufadresse für Fragen rund um den VDR (Quelle: Jan Grell).	137
7.2	Der Sundtek-Stick bringt seine eigenen Treiber mit (Quelle: Sundtek). . . .	138
7.3	Die DVBSky S960 V2-Box benötigt eine Firmware-Datei (Quelle: amazon.de).142	
7.4	Philips Universalfernbedienung (Quelle: Philips).	145
7.5	Der <i>Flirc</i> USB-Empfänger setzt Fernbedienungs-Befehle in Tastatur-Kommandos um. (Quelle: https://flirc.tv/ [61]).	145
7.6	Die grafische <i>Flirc</i> -Oberfläche erlaubt ein einfaches Anlernen der Fernbedienung.	146
7.7	Mit dieser Einstellung versteht Windows UTC.	163
7.8	<i>gnome-session-properties</i> zeigt alle Programme an, die automatisch gestartet werden.	165
7.9	Eine per USB eingebundene 2,5 Zoll-Festplatte erweitert den Speicherplatz des Wohnzimmer-PCs. Aber Vorsicht: Für die externe Festplatte kann eine externe Spannungsversorgung erforderlich sein.	166
7.10	<i>VDRAdmin-am</i> erlaubt das Programmieren von Aufzeichnungen vom Internet aus (Quelle: vdr-wiki.de).	168
7.11	Metrix HD ist ein wunderschönes Skin für den VDR. (Quelle: vdr-portal.de [56]).	169
7.12	<i>tvscraper</i> zeigt in einer schönen Darstellung Filmplakate und Inhalte an. Quelle: vdrportal.de [56]	173
7.13	<i>osdteletext</i> zeigt den Teletext im OSD des VDR. Quelle: vdrportal.de . . .	175
7.14	<i>femon</i> zeigt Signaldaten eines Senders an, hier am Beispiel eines DVB-T-Senders. Quelle: vdr-wiki.de [55].	176
7.15	<i>tvguideng</i> stellt das TV-Programm übersichtlich dar, hier zusammen mit dem Shady-Skin. Quelle: http://anthra.de/shady-skin [72].	176
7.16	3,5 Zoll HDMI-Bildschirme sind bereits für 25 € zu haben (Quelle: https://www.amazon.de [73]).	180
7.17	<i>scraper2vdr</i> stellt Bilder und weitere Informationen zur laufenden Sendung zur Verfügung (Quelle: https://wiki.easy-vdr.de [74]).	184
7.18	Beim Aufruf von <i>osd2web</i> ohne Parameter können verschiedene Farbthemen ausgewählt werden.	186
7.19	<i>osd2web</i> macht das OSD des VDR im Webinterface sichtbar.	186
7.20	<i>osd2web</i> -Darstellung einer Aufnahme auf einem angeschlossenen LCD. .	187
7.21	Das Eisgold VFD leuchtet sehr hell und kann auch aus größerer Entfernung gut abgelsen werden [77].	193
7.22	Das Logo des VLC Media-Players (Quelle: https://www.vlc.de/ [78]). . . .	195
7.23	Die Firma Digital Devices bietet ein CI an, das mit den hauseigenen Karten DVB kompatibel ist. Quelle: https://digitaldevices.de [79].	196

7.24	Die Firma Digital TBS Technology bietet USB-Sticks mit integriertem CI. Quelle: https://www.tbs-technology.de [80].	197
7.25	USB-CI-Module sind heute kaum noch zu finden. Quelle: https://www.amazon.de [82]	198
8.1	Das KODI Mediacenter spielt Filme, zeigt Bilder und vieles mehr.	209
8.2	LibreELEC – Just enough OS for KODI (Quelle: https://www.kodinerds.net/ [89]).	214
8.3	Estuary ist der Standard-Skin für KODI (Quelle: kodi.wiki).	214
8.4	<i>tvheadend</i> wird komfortabel mit Hilfe eines Webinterfaces konfiguriert.	223
8.5	Mit Hilfe von <i>samba</i> können die Logo-Dateien schnell an ihren Bestim- mungsort kopiert werden.	225
8.6	Die Verzeichnisse für die Senderlogos können im <i>tvheadend</i> Web-Frontend eingestellt werden.	226
8.7	Mit Senderlogos sieht <i>tvheadend</i> gleich viel schöner aus.	227
8.8	Amazon VOD.	228
8.9	Als Wiedergabemethode im <i>Amazon VOD</i> Add-on muss <i>Input Stream</i> gewählt werden.	228
8.10	Die Verzeichnisse für die Senderlogos können im <i>tvheadend</i> Web-Frontend eingestellt werden.	229
8.11	Die <i>Kore</i> App unter Android.	230
8.12	Auf einem Apple-Gerät taucht <i>shairport-sync</i> unter meinem Rechnernamen auf.	232
8.13	Delock 61961 ist eine preiswerte USB-Soundkarte (Quelle: Tragant).	232
8.14	Der Delock 62988 ist ein USB-C auf HDMI-Adapter mit PD-Funktion (Quelle: Reichelt).	233
9.1	Ambilight verlängert das Fernsehbild über die Ränder des TVs hinaus (Quelle: Stephan Legachev, wikipedia.de [92]).	237
9.2	Der APA102(C) LED-Streifen eignet sich hervorragend, um ein Ambilight nachzubauen (Quelle: https://amazon.de [93]).	238
9.3	Einen Arduino Nano-Nachbau gibt es günstig im Dreierpack. Der Elegoo Nano V3 mit CH340 USB-Chip wird von <i>LibreELEC</i> ohne Probleme erkannt (Quelle: https://www.amazon.de [94]).	239
9.4	Die Pins des Arduino Nano (Quelle: Arduino Forum, user pighixxx [95]).	240
9.5	So verkabeln Sie Ihr Ambilight richtig	241
9.6	So schneiden Sie den LED-Streifen richtig.	241
9.7	Der LED-Streifen kann mit Hilfe von Klebehaken und Kabelbindern hinter dem Fernseher befestigt werden.	242
9.8	<i>LibreELEC</i> stellt das <i>Hyperion</i> -Add-on unter <i>Dienste</i> bereit.	247
9.9	Mit dem Java-Tool <i>Hypercon</i> erstellt man schnell und komfortabel eine Konfigurationsdatei <i>hyperion.config.json</i> für <i>Hyperion</i> (Quelle: wiki.libreelec.tv [100]).	251
9.10	Ambilight sieht selbst hinter einem curved-Fernseher gut aus.	251
9.11	<i>Hyperion remote</i>	252
9.12	Die Struktur eines KODI-Add-ons.	253
9.13	Das eigene <i>Hyperion</i> -Add-on.	255
9.14	433 MHz Sender (rechts) und Empfänger (links) (Quelle: https://www.amazon.de) [102].	256

9.15	Der 433 MHz-Sender kann mit Jumperkabeln schnell mit dem Arduino Nano verbunden werden.	256
10.1	Der <i>CUPS</i> -Startbildschirm im Webbrowser.	264
10.2	Die <i>CUPS</i> -Verwaltung im Webbrowser.	265
10.3	So sieht der minimalistische CherryPy-Server auf Firefox aus.	268
10.4	Die <i>Apache</i> Standard-Seite sagt „Hallo!“.	269
10.5	Die <i>PHP</i> -Infoseite listet alle Standard-Einstellungen auf.	270
10.6	Quelle: mysql.de	271
10.7	Quelle: phpmyadmin.net	271
10.8	Die <i>phpMyAdmin</i> -Oberfläche verwaltet die <i>MySQL</i> -Datenbank.	273
10.9	Die weitere Konfiguration von <i>WordPress</i> erfolgt aus dem Browser heraus.	277
10.10	Nach erfolgreicher Installation teilt <i>WordPress</i> das mit.	278
10.11	Das Dashboard von <i>WordPress</i> erlaubt weitere Einstellungen.	278
10.12	Unter der Rubrik <i>Appearance</i> können verschiedene Themen für die eigene Webseite gewählt werden.	279
10.13	Die Standard Themen von <i>WordPress</i> hinterlassen einen guten Eindruck.	280
10.14	Es gibt unzählige Themen für <i>WordPress</i> , die im Internet kostenlos zur Verfügung gestellt werden. Das Thema <i>Alizee</i> ist eines davon.	281
10.15	Der Status von MiniDLNA kann in einem Browser über den Port 8200 abgefragt werden.	283
10.16	Unter VLC sind die Einstellungen zum Abspielen von DLNA-Inhalt unter der Rubrik <i>Ansicht</i> versteckt.	284
10.17	Auch die iPhone-App JuP&P erkennt meinen DNLA-Server sofort.	285
11.1	Das kostenlose Office-Paket <i>LibreOffice</i>	289
11.2	Das Startmenü von <i>LibreOffice</i> erlaubt die einfache Auswahl der zu startenden Büro-Applikation.	290
11.3	<i>LibreWriter</i> erlaubt das Erstellen von Texten ähnlich wie <i>WORD</i>	291
11.4	<i>LibreWriter</i> kann <i>WORD</i> -Dateien lesen und speichern.	292
11.5	\LaTeX kann sich sehen lassen: Das einfache Wikipedia-Dokument macht einen guten Eindruck (Quelle: http://de.wikipedia.org/wiki/LaTeX).	295
11.6	Mit <i>aspell</i> gibt es die Möglichkeit, die Rechtschreibung einer Datei zu überprüfen.	301
11.7	Ein einseitiger Lebenslauf mit \LaTeX	302
11.8	Das <i>The Legrand Orange Book-Template</i> ist ein professionelles \LaTeX -Template zum Schreiben von Büchern. Quelle: www.latextemplates.com [112].	303
11.9	Beim ersten Start sieht <i>PlayOnLinux</i> noch sehr leer aus.	304
11.10	Unter <i>PlayOnLinux</i> sind viele Programme zur Installation bereits vorkonfiguriert.	305
11.11	Bei der Installation von Windows-Programmen gibt der <i>PlayOnLinux</i> -Assistent einige Warnungen aus.	306
11.12	Vor der Installation gibt der <i>PlayOnLinux</i> -Installer einige letzte Informationen aus.	306
11.13	<i>PlayOnLinux</i> erlaubt die Installation von DVD oder aus einem Verzeichnis heraus.	307
11.14	Nach der Installation zeigt <i>PlayOnLinux</i> alle installierten Windows-Programme an.	308
11.15	Microsoft <i>WORD</i> fügt sich nahtlos in den LINUX-Desktop ein.	308



5.1 Die wichtigsten *vi*-Befehle lernt man schnell. 77

7.1 Die Tabelle zeigt alle Einstellmöglichkeiten der Datei *lirc hardware.conf*.
Quelle: <https://wiki.ubuntuusers.de/Lirc/> [62]. 149

11.1 Die wichtigsten LINUX-Befehle. 314



Index

Symbole

Übersetzungsfehler	157
L ^A T _E X	292
Aufrufen	294
Bewerbungen	296
Buchvorlage	303
Installieren	293
Rechtschreibung	300
Zusätzliche Pakete	293
2 Screens auf einer Grafikkarte	191
433 MHz-Sender	257

A

AirPlay	230
AirPrint	261
Installieren	261, 266
Ambilight	237
APA102(C)	238
App	252
Funksteckdose	256
Klebehaken	238
KODI Add-on	252
KODI App	252
LibreELEC	237
Netzteil	238
Pegelwandler	241
Sketch für Arduino	246
Software	242
AMD	11, 63
Treiber installieren	63

Apache	268
Apache2 installieren	268
aptitude	
Installieren	71
Programm suchen	71
Arduino Nano	238
at	
Installieren	199
Starten	199
Audio-Device auflisten	160
Autostart von Programmen	164

B

Benutzernamen herausfinden	263
Bewerbungen schreiben	296
BIOS	22
Bootloader installieren	31
Bootstick	20, 23
Buchaufbau	XV
Buchinhalt	XIII

C

C-Beispielprogramm	96
C-Programm übersetzen	96
cemu	112
cherrypy installieren	267
Clover	20
Boot-Eintrag	218
Boot-Partition	218

Default festlegen 33
 config.plist 34
 CUPS
 Starten 262
 Webinterface starten 264

D

Datei suchen 311
 Dateiinhalte anschauen 150
 Dateisystem erstellen 166
 Datenbank 181, 261
 Datensicherung 17
 ddci2-Plugin 196
 DLNA
 Installieren 282
 Starten 284
 dolphin 127
 dos2unix
 Aufrufen 275
 Installieren 275
 Dropbox installieren 83
 DS4 113
 DVB-Sticks 138

E

EDID auslesen 201
 Editor
 joe 77
 vi 77
 EFI 25
 Partition einbinden 33
 Partition kopieren 32, 33
 Einfuhrumsatzsteuer 6
 Einträge in der Datei *remote.conf* 161
 Embedded-Boards 4
 epg2vdr
 Einbinden 183
 Installieren 183
 epgd
 Automatisch starten 183
 Datenbank erstellen 182
 Manuell starten 182
 übersetzen 182
 epgsearch
 Patch 179
 Übersetzen 180

ethtool 87, 88
 ethtool WOL setzen 88
 externalplayer
 Aktivierung des Plugins 174
 Installation des Plugins 174
 Konfiguration des Plugins 174
 Konfigurieren des Plugins 174
 Externe Festplatte 166

F

Fehlersuche 311
 femon
 Aktivierung des Plugins 175
 Installation des Plugins 175
 Fernbedienung anlernen 150
 ffmpeg übersetzen 156
 Firmware DVB 144
 flatpak 133
 Flirc
 Abhängigkeiten 146
 GUI 146
 Start 146
 Fortnite 40
 Funksteckdose 257

G

gcc installieren 95
 git installieren 89
 glxgears 65
 gnome-tweaks installieren 106
 gparted 18
 Grafikanalysen auflisten 200
 grep 312
 grub
 Einstellungen 66
 Einstellungen editieren 66
 Änderungen übernehmen 66
 gst-play 65
 gstreamer-Tools 64

H

Hackintosh 2, 15
 HDCP 35
 Hinweise XV

Hypercon starten 250
 Hyperion
 Ein- und ausschalten 254
 Konfigurationsdatei 250

Scancodes 222
 Senderlogos 225
 USB-Stick 215
 Übersetzen 213

I

iMovie 36
 Infrarot-Befehle anschauen 151
 Intel 10, 62
 IP-Adressen anzeigen 75

J

Java installieren 250
 joe
 Dokument bearbeiten 78
 Installieren 77
 Mit aptitude installieren 78

K

Kernel
 Gepackte Pakete 101
 Installieren 101
 Konfiguration ändern 101
 Konfigurieren 100
 Quelltexte laden 100
 Version abfragen 100
 Kernel-Option auflisten 200
 KODI 209
 Abhängigkeiten 212
 ALSA Audio Passthrough 234
 Aufrufen 213
 Blu-ray-Datenbank 227
 Build-Verzeichnis 213
 Eigenes Add-on 252
 Fernbedienung 219, 220
 Fernbedienungsdatei 221
 Image entpacken 215
 Installation 210
 Installieren 213
 Konfigurationsdatei 222
 lirc-Datei 220, 221
 lirc-KEYs 222
 Protokoll Fernbedienung 221
 Quelltexte 213

L

LattePanda Alpha 864(s) 13
 Lebenslauf 301
 libdvdcss 212
 libdvdcss Quelltexte 212
 LibreELEC 214
 Baudrate setzen 257
 Daten kopieren 217
 Festplatte 216
 Starten 216
 Systempartition 216
 USB, Festplatte 216
 LibreOffice 289
 LibreOffice installieren 289
 LibreWriter 291
 libwebsockets installieren 185
 LINUX 313
 LINUX-Befehle 313
 Lirc

 Auslösen von Tastatur-Events ... 151
 Dienst starten 151
 Dienst stoppen 150
 Hardwarekonfiguration 148
 Input-Event 149
 Installation 147, 148
 Installation, Fehler 147
 Installation, Kopieren Installationsda-
 tei 147
 lircd.conf aufzeichnen 150
 lircd.conf Schlüssel 150
 Treiber 150
 Lirc Konfigurations-Datei kopieren .. 151
 lm-sensors installieren 171
 Lutris 133

M

M64Py 130
 MAC-Adresse ausgeben 89
 MacOS 2, 11
 Makefile

Aufrufen	97
Erstellen	97
Makefile editieren	158
mariadb	
apparmor deaktivieren	181
Benutzer hinzufügen	272
Installieren	181, 271
Terminal	272
Micro-Controller	238
Mojave Installation	28
MP3	233
mpv	
Beispiel	64
Installieren	64
Optionen	64
mupen64plus	128
MySQL	271

N

N64	127
net-tools installieren	75
Netzwerk-Manager neu starten	76
Netzwerke	73
NTFS	167
NTFS-Partitionen einbinden	167
NVIDIA	11, 63
Manuelle Installation	63
Treiber installieren	63

O

Office-Paket	289
openbox installieren	192
OpenGL	7
openvpn	
Installations-Skript klonen	90
Interface anzeigen	92
Server-Konfigurationsdatei	92
osd2web	
Einbinden in VDR-Startdatei	185
Installieren	185
osdteletext	
Aktivieren des Plugins	175
Installation des Plugins	174

P

Philips Hue	286
PHP	270
Installieren	270
Skript	270
php.ini	274
phpMyAdmin installieren	271
PlayOnLinux installieren	304
Programm ausführen	96
Programme de-installieren	96
PS4	112
Python	
Installieren	98
Programm aufrufen	98
Webserver	267

Q

Quelltexte	XV
------------------	----

R

Rechner herunterfahren	89
Rechnername abfragen	75
Rechtschreibprüfung	300

S

Samba	
Installieren	85
Kennwort erstellen	86
Konfigurieren	86
Korrekte Syntax	87
Neu starten	87
scp (secure copy)	94
scraper2vdr	
Einbinden	184
Installieren	184
Senderlogos herunterladen	172
Server	261
shairport-sync	231
Abhängigkeiten	231
Auschecken	231
Konfigurieren	231
Starten	231, 232
Übersetzen	231

skindesigner		Programm de-installieren	70
VDR-Start-Skript	170	Software-Quellen	68
Abhängigkeiten	170	Suchen von Programmen	70
Auschecken	169	Update	99
CPU-Temperatur auslesen	171	Version upgraden	69
VDR-Auslastung auslesen	172	Ubuntu-Sourcen freigeben	211
Übersetzen	170	UEFI	19
Sound Devices	232	USB-Festplatte dauerhaft einbinden .	167
Source-Code-Patch	157	USB-Festplatte einbinden	167
Spiele	111, 131	USB-Port Initialisierung	257
ssh-Server installieren	74	USB-Soundstick	232
streamdev		USB-Stick	21
Aufbau der URL	195	Daten synchronisieren	215
In Start-Skript eintragen	194	Einschieben	215
Konfigurieren	194	Unmounten	215
Verbindungen zulassen	195	UUID Festplatte	217
Übersetzen	194		
Super Mario World	115		
svdrpsend installieren	165		
Switch	123		
Symbolische Links	152		
syslinux.cfg Änderungen	217		
Systemausgaben überwachen	182		

T

Temperatursensoren auslesen	171
Terminal-Verbindung aufbauen	75
Terminalausgabe umleiten	312
texlive	293
tvguideng	
Aktivierung des Plugins	176
Herunterladen des Plugins	177
Patch 1	177
Patch 2	178
Patches anwenden und übersetzen ...	179
tvscraper	
Abhängigkeiten installieren	173
Einbinden des Plugins	173
Installieren	173

U

Ubuntu	47
Aktualisieren	68, 212
Grafiktreiber	61
Installation	48

V

Vaainfo	9
vaapidevice Patches	157
VDR	
<i>commands.conf</i>	155
<i>diseqc.conf</i>	155
<i>remote.conf</i>	155
Abhängigkeiten	153
Ausgabe-Plugin	159
Ausgabe-Plugin Abhängigkeiten	158
Ausgabe-Plugin übersetzen	156, 158
Autostart-Skript	164
Einfaches Start-Skript	160
Fernsteuern	195
Installation	159
Installation aus den Quellen	152
Kanal-Liste	154
Kanalliste	154
Komplettes Start-Skript	162
Komplettes Stopp-Skript	163
Konfigurationsdateien kopieren	153
Mit Lirc-Unterstützung	160
NVidia-Abhängigkeiten	158
Plugin weatherforecast installieren ..	172
Plugins	169
Start-Skript ausführbar machen	160
Starten	159, 160
Video-Verzeichnis erstellen	153
Video-Verzeichnis löschen	166
Übersetzen	153

VDRAdmin-am	
Abhängigkeiten installieren	168
Auschecken	168
Browser-Aufruf	169
Installation	168
Konfiguration	169
Starten	169
veracrypt	
Abhängigkeiten installieren	101
Container aushängen	106
Container mounten	104
Neuer Container	102
Quelltexte entpacken	102
Übersetzen	102
Verzeichnis wechseln	63
vi installieren	77
VNC-Server aktivieren	82
VPN installieren	76

W

wakeonlan installieren	89
weatherforecast-Plugin aktivieren . . .	172
Wii U	111
Win32diskimager	18
winepak	134
Wohnzimmer-PC	
Auswahl	1
Eigenbau	3
WOL Beispiel	89
WordPress	
Apache2	275
Installieren	274
Konfiguration	275

X

XBMC Start-Skript	165
xeamcs installieren	78

Z

Zielgruppe	XIV
ZIP-Datei auspacken	255