

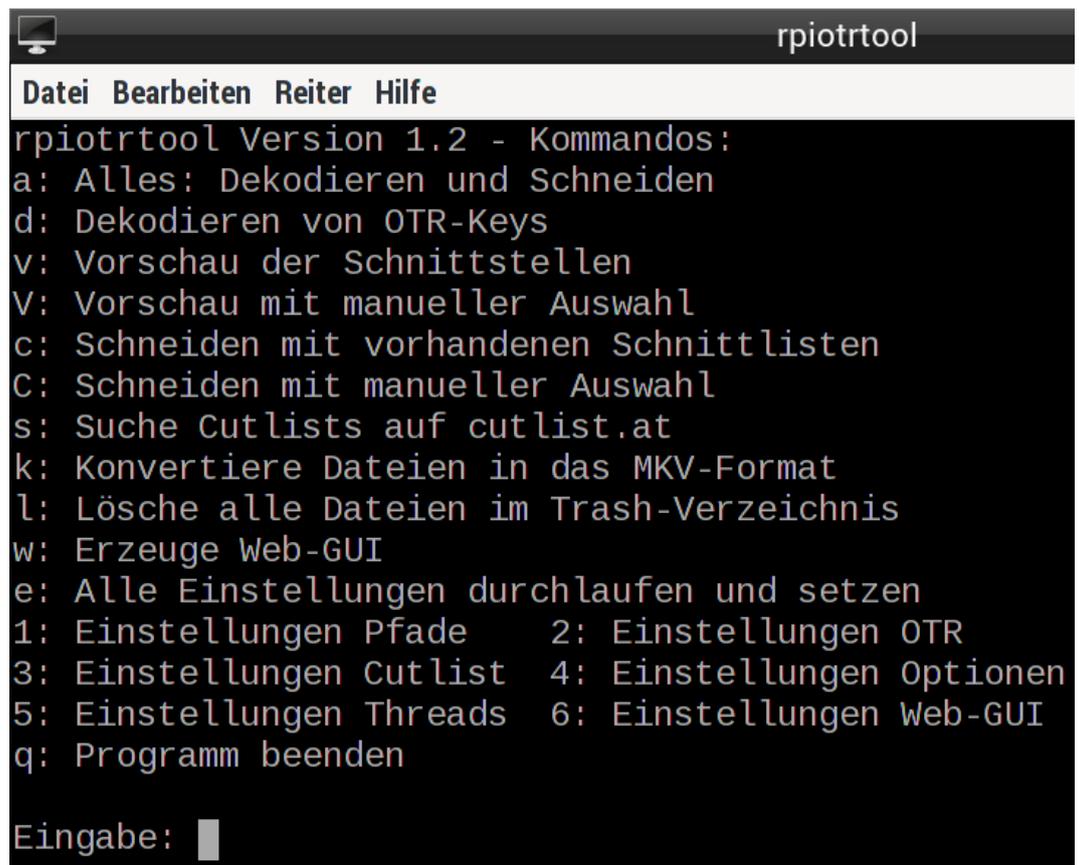
rpiotrtool

Werkzeug zum Dekodieren und Schneiden von TV-Aufzeichnungen
von onlinetvrecorder.com
mit dem Raspberry Pi

Bedienungsanleitung

3. erweiterte Auflage, Programmversion 1.2

Copyright 2020 by Günter Kreidl



Inhalt

Vorbemerkung: Versionsübersicht.....	3
A) Einleitung.....	5
1) Was kann man damit anfangen?.....	5
2) onlinetvrecorder.com und die OTR-Community.....	5
3) Warum Raspberry Pi?.....	7
4) Raspberry Pi und OTR.....	8
B) rpiotrtool: Installation und Einrichtung.....	11
1) Hardware- und Software-Voraussetzungen.....	11
2) Installation.....	11
C) Bedienungsanleitung: Text-Menü.....	13
1) Programm im Text-Menü-Modus starten.....	13
2) Alle Einstellungen.....	14
3) Hauptfunktionen des Programms.....	20
D) Bedienungsanleitung: Web-GUI.....	26
1) Was ist das und wie funktioniert es?.....	26
2) Die Funktionen im Einzelnen.....	28
Erste Gruppe: Aufnahmen.....	28
Zweite Gruppe: Schneiden.....	31
Dritte Gruppe: Verzeichnisse.....	34
Vierte Gruppe: Web-Video.....	36
Zusatzfunktionen.....	48
E) Bedienungsanleitung: Kommandozeilenprogramm.....	50
F) Eigene Cutlists erstellen.....	56
1) Das Problem.....	56
2) Die Lösung.....	58
3) Navigieren im Video.....	59
4) Schnittpunkte setzen, löschen und speichern.....	61
5) Vorlauf und Nachlauf wegschneiden.....	63
6) Werbung und doppelte Szenen herausschneiden.....	64
7) Zeitverschobene Sendungen.....	66
8) Cutlists bearbeiten und hochladen.....	66
9) Der „otr“ Preset.....	67
10) Überblick über die neuen Tastatur-Kommandos.....	69
E) Anhang.....	71
1) Standard-Tastaturkommandos in omxplayerGUI.....	71
Special omxplayerGUI Keyboard Controls.....	71
omxplayerGUI Keyboard Controls (Play Mode).....	72
2) Weitere Software des Autors.....	73
3) Lizenz und Copyright.....	74

Vorbemerkung: Versionsübersicht

rpiotrtool 1.0 wurde ursprünglich für das Betriebssystem Raspbian Stretch geschrieben und veröffentlicht. Für das neue Betriebssystem Raspbian Buster musste das Programm neu kompiliert und auch die Installationsmethode angepasst werden. So konnte z. B. der Decoder von OTR nur mit gewissen Tricks funktionfähig gemacht werden.

Die neue Version 1.0 für Raspbian Buster arbeitet ansonsten in identischer Weise wie die Stretch-Version. Allerdings gibt es ja nicht nur eine neue Betriebssystemversion, sondern auch einen nagelneuen Raspberry Pi 4B Computer, wahlweise mit 1, 2 oder 4 GB Arbeitsspeicher. Zu den neuen Features gehören unter anderem 2 USB3-Schnittstellen, die ein vielfach schnelleres Arbeiten mit USB-Festplatten oder SSDs ermöglichen, wovon z. B. gerade auch rpiotrtool stark profitiert. Die neue Hardware bringt aber auch eine Reihe von Schwierigkeiten und Problemen mit sich. So funktioniert z. B. der von rpiotrtool für verschiedene Funktionen genutzte Videoplayer omxplayer nur noch sehr eingeschränkt. Damit war eine sinnvolle Verwendung von rpiotrtool auf dem Raspberry Pi 4B nicht möglich.

Deshalb wurde die Version 1.1 entwickelt. Sie unterschied sich äußerlich und in den wesentlichen Funktionen nicht von der Version 1.0, umgeht aber einen Teil der Probleme, indem z. B. alternativ der Videoplayer VLC verwendet werden kann. Diese neue Version 1.1 kann nicht nur auf dem Raspberry Pi 4B, sondern auch auf allen älteren Raspberry Pi Modellen (2B, 3B, 3B+) mit dem Betriebssystem Raspbian Buster genutzt werden.

An dieser Stelle ist vielleicht noch ein Vergleich angebracht. Lohnt sich der Umstieg auf einen Raspberry Pi 4 für die Nutzung von rpiotrtool?

Vorteil: Deutlich höhere Geschwindigkeit, insbesondere beim Schneiden von Videos (bis zu sieben Mal schneller).

Nachteile:

1) Die Schnittkontrolle mit VLC ist deutlich schwächer. Zunächst einmal kann VLC die Original-AVI-Formate nur ruckelig anzeigen, was nur durch eine zusätzliche Konvertierung in das MKV-Format umgangen werden kann (Schnittvorschau). Bei der nachträglichen Schnittkontrolle wird der Anfang

eines Videos von VLC leicht verzögert angezeigt, was eine Kontrolle des ersten Schnitts erschwert.

2) Das manuelle Schneiden mit omxplayerGUI ist ohne OSD (On Screen Display) deutlich schwieriger.

3) Das manuelle Schneiden des alten AVI-Formates ist nicht mehr möglich.

Die Nachteile 1) und 2) könnten durch eine neue omxplayer-Version behoben werden, die OSD wieder ermöglicht. Eine solche Version ist aber derzeit nicht in Sicht.

Die neue Version 1.2 behebt einen Teil der Probleme mit dem Raspberry Pi 4B. So kann nun alternativ mplayer für die Schnittkontrolle verwendet werden und diese Lösung umgeht sämtliche Probleme mit VLC.

Darüber hinaus gibt es noch eine Reihe von weiteren Verbesserungen. Nun kann z. B. auch das MP4-Format von OTR geschnitten werden und auch das neue (derzeit noch experimentelle) HD-Format von OTR wird unterstützt. Bei der Schnitt-Vorschau und beim Schneiden gibt es nun zusätzlich die Möglichkeit der Einzelauswahl.

Die wichtigste Neuerung ist aber vielleicht das neue Web-GUI. Damit hat man nun alle OTR-Funktionen in einer Anwendung: Sendungen auswählen (integriertes Text-EPG) und die Aufnahme programmieren, Sendungen herunterladen, dekodieren, nach Cutlists suchen und schneiden (oder auch selber eine Cutlist erstellen). Auch das Hochladen eigener Cutlists und das Bewerten fremder Cutlists wird unterstützt.

Dazu gibt es noch eine Reihe von Zugaben: Suchen in den Mediatheken der öffentlich-rechtlichen Sender und Herunterladen von Sendungen. Der Downloader funktioniert aber außerdem auch für Youtube und hunderte andere Websites. Und zusätzlich kann man auf viele Live-TV-Streams unmittelbar zugreifen.

Das Web-Interface kann auch noch durch eigene Erweiterungen (Plugins) erweitert werden. Dazu sind keinerlei Programmierkenntnisse erforderlich.

A) Einleitung

1) Was kann man damit anfangen?

rpiotrtool ist ein Werkzeug für Raspberry Pi Computer zum Dekodieren und Schneiden von TV-Sendungen, die man mit Hilfe des Dienstbieters onlinetvrecorder.com aufgezeichnet hat. Man erhält dann abspielbare Videodateien, bei denen der Vorlauf und Nachlauf abgeschnitten sind und auch alle eventuell vorhandenen Werbeunterbrechungen entfernt wurden.

Das Schneiden geschieht mit Hilfe von Schnittlisten (Cutlists), die man von cutlist.at oder von onlinetvrecorder.com herunterladen kann. Wenn man meine kweb suite installiert hat, kann man mit Hilfe einer speziell konfigurierten Version von omxplayerGUI auch selbst Schnittlisten erstellen, die man dann auch weiterverteilen kann.

Zum framegenauen Schneiden wird die Methode "Smartmkvmerge" benutzt, die von den Programmierern von OTRVerwaltung++ entwickelt wurde. Der Programmcode für das Schneiden wurde zu großen Teilen aus OTRVerwaltung++ übernommen.

2) onlinetvrecorder.com und die OTR-Community

onlinetvrecorder.com (kurz OTR) ist, wie der Name schon sagt, ein Dienst für das Aufnehmen von TV-Sendungen von mehr als 100 TV-Sendern, also eine Art universeller Videorekorder, mit dem man praktisch beliebig viele Sendungen parallel aufnehmen kann.

Man muss in der Regel die Aufnahmen vorher auf der EPG-Seite von OTR programmieren. Einige Stunden später stehen sie dann zum Herunterladen zur Verfügung. Das Herunterladen von onlinetvrecorder.com ist kostenpflichtig (außer in der Nacht und außerdem haben Premium-User ein bestimmtes freies Downloadvolumen), aber es existieren zahlreiche Mirror-Server, von denen man die Dateien kostenlos herunterladen kann.

Die von den Mirror-Servern heruntergeladenen Dateien sind verschlüsselt, sogenannte "otrkeys", die zunächst mit einem von OTR zur Verfügung gestellten Dekodierprogramm entschlüsselt werden müssen. Dieses Programm nimmt über das Internet Kontakt zu OTR auf und dort wird überprüft, ob man registriert ist, die Aufnahme zuvor programmiert und die erlaubte Anzahl von Dekodierungen noch nicht überschritten hat. Das mag

kompliziert klingen, ist aber aus rechtlichen Gründen notwendig. Als Premium-User (1,99 Euro + MWST im Monat oder 20 Euro im Jahresabo) kann man 120 Dateien pro Monat bzw. 1440 Dateien pro Jahr dekodieren. Reicht das nicht aus, kann man das Abo vorzeitig verlängern.

Nach dem Dekodieren hat man bereits eine abspielfähige Filmdatei entweder im AVI- oder MP4-Format (Container). Die Videodateien haben in der Regel einen Vorlauf von 5 und einen Nachlauf von 8 Minuten. Damit wird sichergestellt, dass geringfügige Zeitverschiebungen oder Überziehungen von Sendezeiten ausgeglichen werden, also die komplette Sendung in der Datei enthalten ist. Man muss also jeweils vorspulen, um zum eigentlichen Sendungsanfang zu kommen, und hat am Ende einen weiteren überflüssigen Teil. Bei werbefinanzierten Sendern kann die Aufnahme auch noch größere Werbesequenzen enthalten, die man beim Abspielen natürlich durch entsprechendes Vorspulen überspringen kann.

Zum einmaligen (zeitversetzten) Anschauen mag das genügen. Will man das Video aber archivieren, empfiehlt es sich, die Aufnahme zu schneiden, also den Vor- und Nachlauf und die evtl. vorhandenen Werbesequenzen zu entfernen. Das erfordert spezielle Software und ist keine ganz triviale Aufgabe; insbesondere das passgenaue (framegenaue) Herausschneiden von Werbesequenzen incl. der oft vorhandenen doppelten Szenen kann eine ganz schön knifflige Angelegenheit sein.

Und hier kommt die OTR-Community ins Spiel. Wer mit einer geeigneten Software eine Schnittliste (Cutlist) erstellt hat, stellt diese über eine Austauschplattform (cutlist.at oder onlinetvrecorder.com) anderen Anwendern zur Verfügung. Damit wird das Schneiden ganz einfach. Die Schnittlisten enthalten eine Qualitätsangabe (0 - 5) und können von anderen Anwendern auch bewertet werden (nur auf cutlist.at).

OTR stellt die Aufnahmen in 2 - 4 verschiedenen Formaten zur Verfügung:

MP4: geringe Auflösung und Qualität und kleine Dateigröße, eigentlich nur zum Abspielen auf Smartphones geeignet.

HQ: H264 codierte SD Auflösung

Eine kleinere Anzahl von (Prime-Time) Aufnahmen steht auch in folgenden Formaten zur Verfügung:

HD: 720p50-Format, H264 kodiert.

AVI: Das alte Standardformat, H263 kodiert.

mpitool unterstützt das Schneiden von allen Formaten.

Auf dem Raspberry Pi 4B ist das manuelle Schneiden (Erstellen von Schnittlisten) von Dateien im H263-AVI-Format leider nicht möglich.

3) Warum Raspberry Pi?

Die Raspberry Pi Foundation ist eine gemeinnützige britische Stiftung, die sich zum Ziel gesetzt hat, den Computerunterricht an Schulen zu verbessern und Kinder und Jugendliche auch außerhalb der Schule zu ermutigen, sich mit der Programmierung von Computern zu beschäftigen und auch Hardware-Anwendungen zu entwickeln. Zu diesem Zweck wurden die extrem preiswerten (zwischen 5 und 55 \$ zuzügl. lokaler Mehrwertsteuer) Raspberry Pi Computerboards entwickelt.

Der Erfolg übertraf alle Erwartungen bei weitem und ging auch weit über den anvisierten Anwenderkreis hinaus. Mittlerweile wurden mehr als 30 Millionen Raspberry Pi Computer verkauft und rund um den Raspberry Pi hat sich eine riesige weltweite Community gebildet. Obwohl es inzwischen zahlreiche Nachahmer gibt, können sich diese bzgl. Support, Software-Unterstützung und der Größe der Community nicht annähernd mit dem Original vergleichen.

Alle Raspberry Pi Computer basieren auf einem Broadcom Videocore SOC (System on Chip), der nicht nur die vielfältigen Schnittstellen bereitstellt, sondern auch über eine leistungsfähige Multi-Media-Einheit verfügt (3D-Grafik, Video-Dekodierung und Encodierung und mehr). Die verschiedenen SOC's und Raspberry Pi Boards unterscheiden sich vor allem in den integrierten ARM-Prozessoren. Die älteren bzw. preiswertesten Modelle beinhalten einen ARMv6 Single-Core Prozessor mit einer Taktung von 700 – 1000 Mhz. Die neueren Modelle haben einen leistungsfähigen ARMv7/8 4-Kern-Prozessor mit Taktraten zwischen 900 Mhz und 1.5 Ghz. Des weiteren unterscheiden sich die verschiedenen Boards in der Anzahl der zusätzlich eingebauten Schnittstellen. Als preiswerter Massenspeicher dient eine microSD-Karte, aber man kann den Raspberry Pi auch mit einer externen USB-Festplatte betreiben.



Das aktuelle Spitzenmodell, der Raspberry Pi 4B, ist in Deutschland ab ca. 37 Euro erhältlich, abhängig von der Größe des Arbeitsspeichers. Er wird mit 1.5 Ghz getaktet und besitzt 2 oder 4 GB Arbeitsspeicher. An Schnittstellen stehen zwei HDMI-Anschlüsse für Monitor oder TV-Gerät, Ethernet und 4 USB-Ports (2 x USB3 und 2 x USB2) zur Verfügung, über die Tastatur, Maus, externe Festplatten oder andere Peripheriegeräte angeschlossen werden können. Zusätzlich sind WLAN und Bluetooth integriert. Über die Steckerleiste (identisch bei allen Raspberry Pis) und On-Board-Steckverbinder stehen zahlreiche weitere Schnittstellen zur Verfügung. Man benötigt noch ein Gehäuse, (USB-) Netzteil und eine microSD-Karte. Je nach Arbeitsspeicher, Gehäuseausführung und Speicherkartengröße erhält man für 60 – 90 Euro ein komplettes und erstaunlich leistungsfähiges Computersystem, das sich durch geringen Stromverbrauch (max. 15 Watt Leistungsaufnahme) auszeichnet.

Für den Raspberry Pi werden zahlreiche Betriebssysteme angeboten. Standard ist Raspbian, eine neu kompilierte Debian-Version, für die von der Raspberry Pi Foundation noch eine Menge von zusätzlicher Software bereitgestellt wird. Die aktuelle Version ist Raspbian Buster.

Mich hat von Anfang an die Idee fasziniert, meine zahlreichen Computeranwendungen (darunter auch einige Serveranwendungen, die rund um die Uhr erreichbar sein müssen) von stromfressenden Desktop-PCs auf Raspberry Pi Computer auszulagern. Inzwischen habe ich 4 Raspberry Pis in meinem Netzwerk laufen, die einen kompletten PC und einen Mediaplayer ersetzt haben. Mein einzig verbliebener Desktop PC wird nur noch wenige Stunden pro Woche gebraucht (für spezielle Programme oder wenn eine besonders hohe Leistung benötigt wird).

Zu diesem Zweck habe ich auch eine ganze Menge eigene Software entwickelt bzw. adaptiert und zum Teil auch veröffentlicht (siehe Anhang).

4) Raspberry Pi und OTR

Während meiner mehr als vierzehnjährigen Mitgliedschaft bei OTR habe ich eine riesige Menge von TV-Aufzeichnungen gesammelt (neben OTR nutze ich noch Mediathek-Downloads und mache eigene TV-Aufzeichnungen), die inzwischen komplett von meinen Raspberry Pis verwaltet werden. Für die Programmierung meiner OTR-Aufnahmen und das Herunterladen und

Dekodieren benutze ich auch nur noch einen Raspberry Pi. Zum Schneiden musste ich bislang die dekodierten Dateien übers Netzwerk auf meinen Ubuntu-Desktop-PC verschieben, dort mit OTR-Verwaltung++ schneiden und dann die geschnittenen Videodateien wiederum übers Netzwerk auf meine Raspberry Pis (die auch als NAS fungieren) verteilen. Das ist eine ziemlich umständliche Prozedur und war auch der Grund, warum ich rpiotrtool entwickelt habe. Mit den Versionen 1.0/1.1/1.2 von rpiotrtool kann man nun alle Arbeitsschritte direkt auf einem Raspberry Pi durchführen.

Ohne Anschluß einer externen USB-Festplatte (vorzugsweise 3.5 Zoll Festplatten mit eigener Stromversorgung) an den Raspberry Pi macht die Nutzung von OTR wenig Sinn. Der Speicherplatz von SD-Karten ist doch arg begrenzt und SD-Karten vertragen nur eine begrenzte Zahl von Schreibzugriffen. Nutzt man die SD-Karte zum Herunterladen, Dekodieren und Schneiden von OTR-Aufnahmen, wird die SD-Karte arg gestresst und wird sicher vorzeitig den Geist aufgeben. Und zum Archivieren der Dateien hat man da sicherlich nicht genug Speicherplatz. Bei dem neuen Raspberry Pi 4B ist zudem der Dateizugriff über eine USB3-Festplatte (oder SSD) um ein Vielfaches schneller als mit der SD-Karte.

Bei einigen Raspberry Pi Modellen kann man direkt von der Festplatte booten, benötigt also gar keine SD-Karte mehr. Bei der „klassischen“ Methode läßt man die (kleine) Boot-Partition auf der SD-Karte und verschiebt das Root-Dateisystem auf die Festplatte. Es ist aber auch möglich, das Betriebssystem komplett auf der SD-Karte zu belassen und die Festplatte als reinen Datenspeicher zu nutzen. In diesem Fall sollten die OTR-Verzeichnisse (siehe unten) und das Downloadverzeichnis auf die Festplatte ausgelagert werden.

Zum Abspielen der OTR-Aufzeichnungen ist der Raspberry Pi optimal geeignet und es können unterschiedliche Programme verwendet werden.

Der klassische Mediaplayer für den Raspberry Pi ist omxplayer, ein Kommandozeilenprogramm, das aber relativ umständlich zu bedienen ist. Deshalb habe ich dafür eine grafische Benutzeroberfläche entwickelt, omxplayerGUI (Teil meiner kweb suite, siehe Anhang), das nicht nur die Bedienung vereinfacht, sondern auch zahlreiche Funktionen hinzufügt. Zum

manuellen Schneiden und Erstellen von Cutlists dient nun eine erweiterte Version von omxplayerGUI.

Als Alternative steht das bekannte Multi-Media-Zentrum Kodi zur Verfügung, das direkt aus dem Raspbian Repository installiert werden kann.

Seit einiger Zeit enthält Raspbian einen neuen Standard-Desktop-Mediaplayer, eine spezielle Version von VLC mit Unterstützung der Hardware-Dekodierung durch die GPU. Leider kommt die aktuelle Version mit den ungeschnittenen AVI-Dateien von OTR nicht gut zurecht – sie werden nur ruckelhaft abgespielt. Ich habe deshalb in rpiotrtool eine Konvertierungsfunktion ins MKV-Format eingebaut. Das geht relativ fix, vor allem mit dem Raspberry Pi 4B. Bei den geschnittenen Dateien gibt es das Problem nicht, da sie sowieso schon im MKV-Format vorliegen.

Für den einfachen Zugriff auf meine Videosammlung nutze ich ein weiteres Tool aus meiner kweb suite (siehe Anhang), den Youtube-DL-Server. Ursprünglich wurde der programmiert, um die Nutzung von youtube-dl zum Abspielen von Webvideos zu beschleunigen, aber er kann auch zum einfachen Zugriff auf alle Mediadaten direkt aus dem Webbrowser verwendet werden. Er arbeitet sowohl mit omxplayerGUI als auch mit VLC. Ein simpler Klick auf eine angezeigte Mediendatei startet unmittelbar den gewünschten Player.



B) rpiotrtool: Installation und Einrichtung

1) Hardware- und Software-Voraussetzungen

Zur Nutzung von rpiotrtool benötigt man einen Raspberry Pi 2B, 3B, 3B+ oder 4B. Die Verwendung einer USB-Festplatte oder SSD wird dringend empfohlen.

Als Betriebssystem benötigt man Raspbian Buster in der Vollversion (Desktop-Version).

rpiotrtool benötigt eine Reihe von Hilfsprogrammen (ffmpeg, x264, mkvmerge usw.), die durch das Installationsskript automatisch installiert werden. Ebenso wird der otrpidecoder von onlinetvrecorder.com automatisch heruntergeladen und installiert.

Als Standard-Desktop-Mediaplayer ist auf Raspbian Stretch nun VLC installiert. Will man aber die ungeschnittenen Dateien ohne Konvertierung ins MKV-Format anschauen, sollte man zusätzlich Kodi oder omxplayerGUI (kweb suite, siehe Anhang) installieren.

Wenn man Videos manuell schneiden und eigene Cutlists erstellen will oder das Web-GUI benutzen will, muss man auf jeden Fall meine kweb suite installieren (wird bei der Installation abgefragt). Die mitgelieferte Spezialversion (mit Schnittfunktionen) von omxplayerGUI benötigt andere Bestandteile der kweb suite und wird deshalb nur installiert, wenn diese bereits installiert ist.

2) Installation

Zur Installation öffnet man ein Terminal und gibt nacheinander die folgenden Befehlszeilen ein:

```
wget http://steinerdatenbank.de/software/rpiotrtool12.tar.gz
tar -xzf rpiotrtool12.tar.gz
cd otr
./install
```

Während der Installation wird man gefragt, ob man die aktuelle Version der kweb suite installieren möchte (sofern sie nicht schon installiert ist). Benötigt wird sie zum manuellen Schneiden und für das Web-GUI. In diesem Fall

werden auch gleich optimale „Settings“ (kweb, omxplayerGUI) für die Nutzung mit rpiotrtool installiert.

Im Home-Verzeichnis befindet sich nun ein neuer Ordner „otr“ mit einer Reihe von Unterordnern: „decoded“, „cut“, „converted“, „trash“, „tmp“, „upload“, „EPG“, „web“, „sripts“, „src“, „installation“.

Die ersten 6 dieser Ordner werden von rpiotrtool zum Arbeiten benötigt. „EPG“ enthält eine Reihe von HTML-Dateien, die einen einfachen Zugriff auf das Text-EPG der TV-Zeitschrift „HÖRZU“ ermöglichen als Hilfsmittel bei der Programmierung von TV-Aufnahmen. Die Ordner „web“ und „scripts“ werden für das Web-GUI benötigt. Der Ordner „src“ enthält den Python-Quellcode von rpiotrtool, der mit nuitka kompiliert wurde. In „installation“ befinden sich Komponenten, die für die Installation benötigt werden.

Betreibt man den Raspberry Pi von Festplatte (zumindest das Root-Dateisystem), kann man alles so lassen und unmittelbar mit der Arbeit beginnen. Läuft das Betriebssystem auf der SD-Karte und verwendet man eine externe USB-Festplatte als reinen Datenspeicher, sollte man nun unbedingt den gesamten „otr“-Ordner auf die Festplatte verschieben. Ebenso sollte man auf der Festplatte ein Downloadverzeichnis anlegen, das als Standard-Download-Pfad für den benutzten Browser eingerichtet werden sollte (zum Herunterladen der OTR-Film-Dateien, otrkeys und cutlists). Anschließend muss man rpiotrtool mitteilen, wo sich die neuen Verzeichnisse befinden. Das könnte z. B. so aussehen: „/media/usbhd/otr“ und „/media/usbhd/Downloads“. (Den richtigen Pfadnamen finden Sie einfach im Dateimanager). Nun ruft man in einem Terminal das Programm folgendermaßen auf:

```
rpiotrtool init paths
```

Das Programm fragt dann zunächst das neue Downloadverzeichnis ab und anschließend den neuen OTR-Pfad.

Sie können diesen Schritt aber auch überspringen, denn er wird beim erstmaligen Starten von rpiotrtool bei Bedarf auch automatisch ausgeführt.

Das Programm ist nun einsatzbereit und alles Weitere erfahren Sie in der folgenden Bedienungsanleitung.

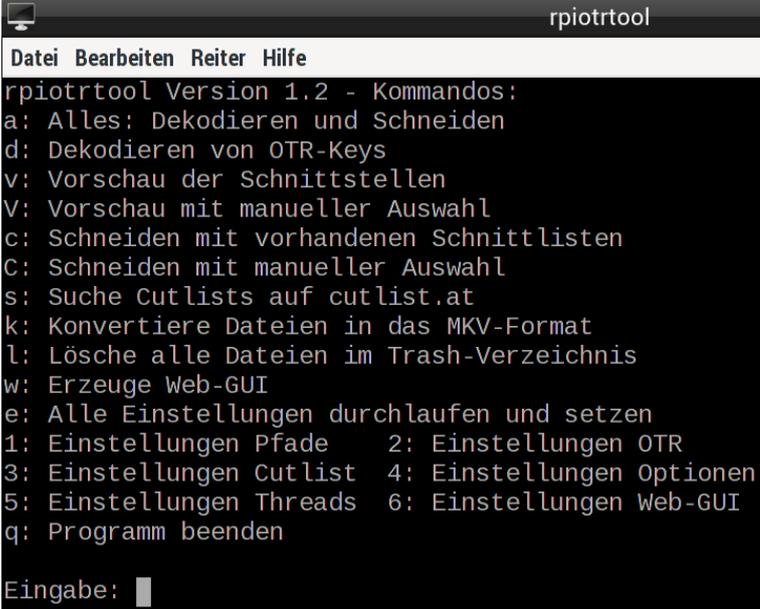
C) Bedienungsanleitung: Text-Menü

1) Programm im Text-Menü-Modus starten

rpiotrtool ist ein Kommandozeilenprogramm, besitzt also keine grafische Benutzeroberfläche (GUI). Trotzdem ist die Bedienung sehr einfach.

Kommandozeilenprogramme benötigen ein Terminal (Ixterminal) und man muss normalerweise den

Programmnamen eintippen, optional mit zusätzlichen Angaben wie Kommandos, Optionen usw. rpiotrtool bietet aber noch eine einfachere Variante an: einen interaktiven Modus mit einem einfachen Textmenü und Ein-Buchstaben-Kommandos. Bei der Installation wird eine Desktop-Datei installiert und



```
rpiotrtool
Datei Bearbeiten Reiter Hilfe
rpiotrtool Version 1.2 - Kommandos:
a: Alles: Dekodieren und Schneiden
d: Dekodieren von OTR-Keys
v: Vorschau der Schnittstellen
V: Vorschau mit manueller Auswahl
c: Schneiden mit vorhandenen Schnittlisten
C: Schneiden mit manueller Auswahl
s: Suche Cutlists auf cutlist.at
k: Konvertiere Dateien in das MKV-Format
l: Lösche alle Dateien im Trash-Verzeichnis
w: Erzeuge Web-GUI
e: Alle Einstellungen durchlaufen und setzen
1: Einstellungen Pfade      2: Einstellungen OTR
3: Einstellungen Cutlist   4: Einstellungen Optionen
5: Einstellungen Threads   6: Einstellungen Web-GUI
q: Programm beenden

Eingabe: █
```

man sollte nun im Programm-Menü unter Unterhaltungsmedien den Eintrag „RPI-OTR-Tool“ finden. Wählt man den an, öffnet sich Ixterminal mit rpiotrtool im interaktiven Modus.

Alternativ dazu kann man zunächst ein Terminal öffnen und „rpiotrtool menu“ oder noch kürzer „rpiotrtool -i“ eintippen. Auch dann wird rpiotrtool im interaktiven Modus gestartet.

Bei jedem Programmstart wird überprüft, ob alle benötigten Verzeichnisse vorhanden sind. Ist das nicht der Fall, gelangt man unmittelbar zur Pfadeinstellung (wird weiter unten beschrieben).

Das Programm wartet nun auf Eingaben. Diese bestehen alle aus einem einzelnen Buchstaben gefolgt von einem Zeilenvorschub (Return-Taste). Dann wird das entsprechende Kommando ausgeführt und am Ende landet man wieder im Menü. Einfacher geht es kaum.

2) Alle Einstellungen

Bevor man mit rpiotrtool sinnvoll (oder überhaupt) arbeiten kann, müssen eine Reihe von individuellen Einstellungen vorgenommen werden. Man kann das auf zweierlei Weise tun:

Mit dem Kommando „e“ werden allen Einstellungen nacheinander durchlaufen. Mit den Ziffern „1“ bis „6“ kann man die Einstellungen für bestimmte Bereiche einzeln vornehmen. Wir wollen nun im folgenden sämtliche Einstellungen genauer beschreiben.

1: Einstellungen Pfade:

Das Programm benötigt zwei Pfade.

Den Pfad zum Verzeichnis „otr“, das bei der Programminstallation angelegt (und dann eventuell verschoben) wurde.

Den Pfad zum Downloadverzeichnis, das von Ihrem Browser (und evtl. anderen Programmen) standardmäßig zum Herunterladen von Dateien genutzt wird. Dort sammelt rpiotrtool die heruntergeladenen OTR-Filmdateien, otrkeys und Schnittlisten ein.

Es muss jeweils der komplette Pfad beginnend mit dem Wurzelverzeichnis eingegeben werden, also z. B. „/home/pi/otr“. Das Programm zeigt (wie bei allen Einstellungen) die gerade gültigen Werte an. Mit einer leeren Eingabe kann man die Pfade unverändert lassen.

2: Einstellungen OTR

Für den Betrieb des Dekoders benötigt das Programm Ihre bei OTR hinterlegte Email-Adresse und Ihr OTR-Passwort. Beide können hier eingegeben werden.

Vor allem das Speichern des Passwortes stellt ein Sicherheitskiko dar. Sie können deshalb einen oder beide Werte leer lassen. Dann müssen Sie vor jedem Dekodieren den jeweiligen Wert manuell eingeben.

3: Einstellungen Cutlist

Wenn Sie bei cutlist.at nicht registriert sind, können Sie hier mit den voreingestellten Grundeinstellungen arbeiten. Allerdings sind Sie dann auf das Suchen und Herunterladen von Cutlists eingeschränkt. Sie können diese

Cutlists nicht bewerten und auch keine Cutlists hochladen und damit anderen Nutzern zur Verfügung stellen. Auch beim Web-GUI stehen dann die entsprechenden Funktionen nicht zur Verfügung.

Ich möchte deshalb jedem Nutzer von rpiotrtool dringend raten, sich bei cutlist.at zu registrieren. Dabei werden keinerlei persönliche Daten gespeichert! Sie erhalten stattdessen einen persönlichen Link, auch FRED genannt, mit dem Ihnen dann alle Funktionen von cutlist.at zur Verfügung stehen. Diesen FRED müssen Sie dann sorgfältig aufheben, denn selbst bei cutlist.at können Sie ihn nicht mehr nachträglich restaurieren (perfekter Datenschutz!).

Gehen wir nun die Einstellungen im einzelnen durch:

Cutlist-Server: Hier geben Sie Ihren FRED ein, sobald sie registriert sind. Andernfalls steht hier <http://cutlist.at>.

Autoren-Name für eigene Cutlists: Geben Sie hier den Namen ein, der bei Ihren cutlists als Autor erscheinen sollte.

Cutlist-Bewertung: Geben Sie hier die voreingestellte Qualitätsstufe für Ihre eigenen Cutlists ein (1 .. 5). Wenn Sie hier „0“ eingeben (Voreinstellung) wird die Qualität bei jeder Cutlist-Erstellung abgefragt.

Cutlist-Kommentar: Geben Sie hier den voreingestellten Kommentar zu Ihren eigenen Cutlists ein. Standardeinstellung ist „Geschnitten mit rpiotrtool“.

4: Einstellungen Optionen

Hier kann man eine Reihe von unterschiedlichen Programmoptionen einstellen, die wir hier nun im einzelnen vorstellen wollen:

OTR-Keys nach dem Dekodieren ohne Rückfrage löschen? Die Antwort kann „ja“ (j) oder „nein“ (n) lauten. Es werden nur otrkeys gelöscht, wenn die Dekodierung erfolgreich war. Deshalb kann man hier gefahrlos mit „ja“ antworten.

Dateinamen der geschnittenen Datei (hinter Laufzeit) kürzen? rpiotrtool enthält keine Optionen zum automatischen oder manuellen Umbenennen der geschnittenen Datei. Es kann aber der Dateinamen ab einer bestimmten Stelle gekürzt werden, ab dem „_TVOON“ bis zum Dateiformatkürzel „mkv“.

Schnittvorschau vor jedem Schneiden? Soll vor jedem Schneidebefehl automatisch eine Schnittvorschau abgespielt werden? Wenn dabei die Cutlist negativ bewertet wird, unterbleibt das Schneiden. Man kann die Schnittvorschau alternativ auch manuell gesteuert mit einem eigenen Befehl aufrufen und auf das automatische Abspielen verzichten. Das ist eine Frage der persönlichen Vorliebe oder Arbeitsmethode.

Geschnittene Videos nach dem Schneiden anzeigen? Dies betrifft die Schnittkontrolle nach dem Schneiden. Das ist nur unmittelbar nach dem Schneiden möglich, also nicht mehr im Nachhinein. Und dies ist die einzige Methode, die Qualität der Schnitte wirklich zu beurteilen. Deshalb wird empfohlen, diese Frage mit „ja“ (j) zu beantworten, falls Ihnen Qualität wichtig ist.

Zeitfenster für Vorschau und Anzeige. Mögliche Werte sind hier: 8, 10, 12, 15, 20, 25 oder 30 Sekunden. Da die Keyframes bei OTR-Aufnahmen bis zu 10 Sekunden auseinanderliegen können, sollte man den Wert nicht geringer als 12 wählen. Größere Werte machen Sinn, wenn man genau auf doppelte Szenen achten will, aber dann dauert es eben auch länger. Dieser Wert wird sowohl für die Schnitt-Vorschau als auch für die Schnittkontrolle nach dem Schneiden verwendet.

Player für die Schnittvorschau und Schnittkontrolle auswählen. Folgende 5 Einstellungen sind möglich: „omxplayer“, „vlc“, „mplayer“, „vlc, omxplayer wenn möglich“, „mplayer, omxplayer wenn möglich“. Auf Raspberry Pis 2B, 3B, 3B+ sollte man hier grundsätzlich „omxplayer“ auswählen. Auf einem Raspberry Pi 4B muss man eine der anderen Möglichkeiten wählen, vorzugsweise „mplayer“ oder „mplayer, omxplayer wenn möglich“.

Nach dem Dekodieren automatisch nach MKV konvertieren? Beantwortet man diese Frage mit „ja“ (j), werden alle neu dekodierten Dateien automatisch nach MKV konvertiert, außer MP4-Dateien. Dies ist nötig, wenn man die Dateien unmittelbar mit VLC anschauen möchte oder VLC für die Schnittkontrolle ausgewählt hat. Das Konvertieren kann aber jederzeit auch manuell gestartet werden.

5: Einstellungen Threads

Unter diesem Punkt gibt es nur eine einzige Einstellung. Soll das Schneiden single-threaded (nur ein Programm zur gleichen Zeit) oder multi-threaded

(mehrere Programme zur gleichen Zeit) durchgeführt werden. Multi-threaded geht deutlich schneller, erfordert aber mehr Arbeitsspeicher :deshalb ist die Antwort modell-abhängig: Nur wenn der Raspberry Pi mindestens 2 GigaByte Arbeitsspeicher hat, sollte man multi-threaded wählen.

6: Einstellungen Web-GUI

Diese Gruppe von Einstellungen bezieht sich ausschließlich auf das Web-GUI (siehe nächstes Hauptkapitel). Das kann man nur nutzen, wenn meine kweb suite installiert ist.

Welcher Browser soll für OTR und die Mediatheksuche verwendet werden?

Ausgewählt werden können „kweb“, „chromium-browser“, „firefox“ und „vivaldi“. Auf allen Modellen mit weniger als 2 GigaByte Arbeitsspeicher rate ich dringend dazu, kweb zu benutzen. „firefox“ oder „vivaldi“ darf man natürlich nur auswählen, wenn man den entsprechenden Browser zuvor installiert hat.

Wählen Sie Layout-Format und -Größe: Grundsätzlich gibt es zwei Layout-Formate: „normal“ und „kompakt“. Das Format „normal“ eignet sich für Bildschirmgrößen mit 1920x1080 Pixeln. Bei kleineren Bildschirmgrößen sollte man „kompakt“ wählen, um unnötiges Scrolling zu vermeiden. Zusätzlich gibt es diese Formate noch mit verschiedenen Zoom-Größen. Für Menschen mit Sehbehinderung empfiehlt sich die Einstellung „kompakt150“ auf einem 1920x1080-Bildschirm. Die beiden restlichen Einstellungen sind nur sinnvoll, wenn Sie Ihren Raspberry Pi 4B an einen 4K Fernseher angeschlossen haben: „normal200“ und „kompakt300“.

Wählen Sie die Hintergrundfarbe des Web-GUI. Hier stehen 3 Layoutvarianten zur Auswahl: weißer, dunkelgrauer oder schwarzer Hintergrund. Dies betrifft die Hintergrundfarbe des Hauptmenüs und etlicher eingebauter Funktionen. Das EPG wird unabhängig von dieser Einstellung immer mit schwarzem Hintergrund dargestellt, da es von der „HÖRZU“ entsprechend geliefert wird. Ebenso erscheinen die Einblendungen von cutlist.at immer mit weißem Hintergrund, weil diese Webseiten eben so konfiguriert sind.

Was soll im Web-GUI rechts beim Programmstart angezeigt werden? Das Web-GUI besteht aus zwei Bereichen: Einer Liste von Programmfunktionen auf der linken Seite und einem weitaus größeren Bereich rechts, in dem

entsprechende (interne oder externe) Inhalte angezeigt werden. Dieser rechte Bereich kann beim Programmstart entweder leer sein, das EPG anzeigen oder die Plugin-Seite, die vom Benutzer selbst gestaltet werden kann.

Soll nach dem Ausführen eines rpiotrtool-Befehls immer auf eine Tastatureingabe gewartet werden (oder nur beim Dekodieren)? Diese Einstellung bedarf einer zusätzlichen Erläuterung. Wenn man die eigentlichen rpiotrtool-Funktionen (z. B. „Dekodieren“ oder „Schneiden“) aus dem Web-GUI aufruft, wird ein Terminal gestartet, in dem der gewählte Prozeß abläuft. Das ist auch unbedingt notwendig, da die jeweilige Programmfunktion ja unter Umständen Benutzereingaben verlangt. Normalerweise würde das Terminal nach Beendigung der Programmfunktion automatisch geschlossen. Beim Dekodieren könnte das aber fatale Auswirkungen haben, weil Ihnen unter Umständen Fehlermeldungen des Dekoders entgehen. Deshalb wartet das Programm nach Beendigung noch auf eine Tastatureingabe (Zeilenvorschubtaste), bevor das Terminal geschlossen wird. Bei den anderen Funktionen ist das nicht so kritisch. Deshalb können Sie mit dieser Option festlegen, ob auch bei allen anderen Funktionen auf eine Tastatureingabe des Benutzers gewartet werden soll oder nicht.

Wie sollen die OTR-Verzeichnisse angezeigt werden? „Im Browser“ oder „Im Datei-Manager“? Wenn Sie die Verzeichnisse im Browser anzeigen lassen, können Sie Videos zwar direkt starten (z. B. zum manuellen Schneiden) oder sich auch den Inhalt von Cutlists im Browser anschauen, aber weitere Funktionen (z. B. Dateien umbenennen oder löschen) können Sie nur im Datei-Manager nutzen. Es ist eine Frage der persönlichen Arbeitsweise, welche Option Ihnen besser gefällt. Allerdings funktioniert die Verzeichnisanzeige im Browser aus technischen Gründen nur bei weißem Hintergrund. Deshalb wird bei anderen Layoutfarben automatisch auf die Anzeige im Datei-Manager umgeschaltet.

Optionsvorgabe im YTDL-Downloader: Legt die Standardvorgabe für das Optionsfeld im Youtube-DL Web Video Downloader fest. Läßt man das Feld leer, versucht Youtube-DL das Format mit der besten Videoauflösung und der besten Audioqualität herunterzuladen. Das führt oft dazu, dass zwei separate Dateien heruntergeladen werden, die dann noch zusammengemischt werden müssen (geschieht automatisch). Mit der Option „-f best“ wird das beste

Format heruntergeladen, bei dem Video und Audio in einer einzelnen Datei geliefert werden. Bei Youtube ist das z. B. Die 720p-Auflösung. Mit der Option „-f mp4“ wird die bestmögliche Videoqualität in einer einzelnen Datei im MP4-Format heruntergeladen. Die Option „mp4/best“ kombiniert diese Möglichkeiten noch und liefert entweder MP4 oder, wenn das Format nicht existiert, die beste Qualität in einem anderen zur Verfügung gestellten Format.

Standard-Player für den YTDL-Server: „omxplayerGUI“ oder „VLC“. Sie können den YTDL-Server (Teil der kweb suite) aus dem Web-GUI starten. Mit dieser Option können Sie entscheiden, welcher Player dabei zur Wiedergabe voreingestellt werden soll. Näheres dazu erfahren Sie im folgenden Kapitel.

Preset-Wahl für den Youtube-DL-Server. „Presets“ sind komplette Einstellungen für die kweb suite. Sie enthalten vielfältige Einstellungen für kweb, omxplayerGUI und den Youtube-DL-Server. Presets können auf der Settings-Seite von kweb gewählt, konfiguriert und erzeugt werden. Der mit rpiotrtool installierte Preset „otr“ mag nicht unbedingt ideal für den Betrieb des YTDL-Servers sein. Deshalb kann hier optional ein anderer Preset dafür gewählt werden. Man muss den Namen des Presets so eintragen, wie er auf der Settings-Seite von kweb angezeigt wird.

Formatvorgabe für den Youtube-DL-Server. Der Youtube-DL-Server dient ja vor allem dazu, Videos aus dem Internet unmittelbar mit omxplayerGUI oder VLC abspielen zu können. Das funktioniert nur bei Videoformaten, bei denen Audio und Video in einem gemeinsamen Stream übertragen werden. Dazu benötigt der Server bestimmte Formatvorgaben. Läßt man diese Option leer, wird die Formateinstellung aus den aktuellen Settings oder dem gewählten Preset (s. o.) übernommen. Man kann aber auch beim Starten des Servers ein Format einstellen. Welches das ist, bestimmt diese Option. Mit der Option „-f best“ wird das beste Format abgespielt, bei dem Video und Audio in einer einzelnen Datei geliefert werden. Bei Youtube ist das z. B. die 720p-Auflösung. Mit der Option „-f mp4“ wird die bestmögliche Videoqualität in einer einzelnen Datei im MP4-Format abgespielt. Die Option „mp4/best“ kombiniert diese Möglichkeiten noch und liefert entweder MP4 oder, wenn das Format nicht existiert, die beste Qualität in einem anderen zur Verfügung gestellten Format.

3) Hauptfunktionen des Programms

Nun kommen wir zu den eigentlichen Programmfunktionen. Sie werden jeweils durch den im Menü angezeigten Buchstaben (+ Returntaste) gestartet. Während eine Funktion abläuft, wird der Ablauf durch entsprechende Textausgaben im Terminal angezeigt. Bei einigen Funktionen muss der Benutzer eingreifen und entsprechende Fragen durch Eingaben beantworten. Ich beschreibe zunächst die Einzelfunktionen und zum Schluß das kombinierte Kommando „a: Alles“.

d: Dekodieren von OTR-Keys

Je nach Einstellung werden zuvor Emailadresse und Passwort abgefragt (wenn man sie nicht gespeichert hat). Das Programm sucht dann im Downloadverzeichnis nach neuen Otrkeys und ruft zur Dekodierung den von OTR zur Verfügung gestellten otrpidecoder auf. Die Ausgaben dieses Programms werden im Terminal angezeigt. Man kann daran den Fortschritt des Programmes erkennen und ebenso, ob Fehler bei der Dekodierung auftreten. Nach dem erfolgreichen Dekodieren werden die Otrkeys automatisch oder nach Abfrage im Downloadverzeichnis gelöscht. Das ist insofern wichtig, als man sonst beim nächsten Dekodier-Befehl evtl. die gleichen Otrkeys erneut dekodieren würde. Ich empfehle deshalb, das automatische Löschen in den Optionen einzustellen.

Bei der Dekodierung können Fehler auftreten. Deshalb ist es wichtig, dass man sich das Ergebnis immer anschaut. Kommt es zu Netzwerkfehlern (der Dekoder kann keine Verbindung zum OTR-Server herstellen) sollte man es einfach zu einem späteren Zeitpunkt erneut versuchen.

Andere Fehler sind dramatischer. Sie bedeuten in der Regel, dass der Otrkey korrupt ist. Man sollte ihn im Downloadverzeichnis explizit löschen und die Datei von einem anderen Mirror-Server neu herunterladen.

So ganz nebenbei werden bei diesem Befehl auch Cutlists und OTR-Aufnahmen eingesammelt, die (von OTR) bereits dekodiert heruntergeladen wurden (AVI- oder MP4-Dateien).

v: Vorschau der Schnittstellen

Das Programm zeigt nacheinander eine Vorschau aller ungeschnittenen Videos an, für die eine Cutlist existiert. Die Anzeige erfolgt - je nach

Voreinstellung - mit omxplayer in einem Overlay in der oberen linken Ecke des Bildschirms, mit VLC im Vollbildmodus oder in einem mplayer-Fenster. Die Wiedergabe beginnt einige Sekunden vor der ersten Schnittstelle. Der Schnitt selbst wird durch einen Untertitel „Start Cut0“ angezeigt (für zwei Sekunden). Einige Sekunden später springt die Wiedergabe zu einem Zeitpunkt einige Sekunden vor dem Ende des ersten Schnittbereichs, der mit dem Untertitel „Ende Cut0“ angezeigt wird. Enthält die Cutlist weitere Schnittbereiche, wird das so fortgeführt, wobei die Schnittnummer hochgezählt wird („Start Cut1“, „Ende Cut1“ usw.). Einige Sekunden nach der letzten Schnittstelle wird die Wiedergabe beendet. Dann erscheint im Terminalfenster folgende Abfrage:

```
Sind die Schnitte OK (j,n)?  
Soll die Vorschau erneut angezeigt werden (w)?  
Vorschau wiederholen mit neuem Zeitfenster (8,10,12,15,20,25,30)?
```

Eingabe:

Antwortet man mit „n“, wird die Cutlist ins Trash-Verzeichnis verschoben und steht somit zum Schneiden nicht mehr zur Verfügung. Mit „j“ sollte man antworten, wenn man die Cutlist akzeptiert. Man kann die Vorschau auch wiederholen („w“), bei Bedarf auch mit einem größeren oder kleineren Zeitfenster (siehe Ziffern oben). Alle Eingaben müssen mit „Return“ abgeschlossen werden.

V: Vorschau mit manueller Auswahl

Manchmal ist es praktischer, sich nur die Schnittvorschau einer bestimmten Aufnahme anzuschauen. Bei dieser Funktion wird zunächst eine Liste der Videos angezeigt, für die eine Cutlist existiert.

Wählen Sie eine Datei für die Schnittvorschau aus:

- (1) `Das_letzte_Problem_20.01.31_20-15_arte_90_TVOON_DE.mpg.HD.avi`
- (2) `Doubt_Glaubensfrage_20.02.03_20-15_arte_100_TVOON_DE.mpg.HD.avi`
- (3) `Geisterfjord_20.02.01_20-15_tele5_130_TVOON_DE.mpg.HQ.avi`
- (4) `Mitternachtsspitzen_20.02.01_21-45_wdr_60_TVOON_DE.mpg.HQ.avi`
- (5) `Silent_Witness_20.02.03_21-00_ukbbc_60_TVOON_DE.mpg.HQ.avi`

Eingabe (Ziffer):

Man wählt nun mit der entsprechenden Ziffer die Datei aus, deren Schnittvorschau man anzeigen möchte. Dann erscheint die Schnittvorschau mit anschließender Abfrage exakt wie oben beschrieben.

Mit jeder anderen Eingabe (einfach Return-Taste drücken) kehrt man zum Hauptmenü zurück.

c: Schneiden mit vorhandenen Schnittlisten

Je nach Einstellung führt das Programm verschiedene Schritte aus. Ist die Vorschau aktiviert, erfolgt zunächst eine Schnittvorschau aller Videos, für die eine Cutlist existiert, exakt wie oben beschrieben.

Anschließend werden alle entsprechenden Dateien geschnitten. Das kann je nach Dateigröße und Anzahl der Schnitte zwischen 2 und 5 Minuten pro Datei dauern; auf dem Raspberry Pi 4B geht es bis zu 7 mal schneller. Nach dem Schneiden werden die ungeschnittene Datei und die Cutlist in das Trash-Verzeichnis verschoben. Das geschnittene Video befindet sich im Verzeichnis „cut“.

Nach dem Schneiden (aller Dateien) können die Schnitte noch begutachtet werden, wenn man die Einstellung „Schnittanzeige“ aktiviert hat. Die Videos werden ähnlich wie bei der Schnittvorschau - je nach Einstellung – mit omxplayer in der oberen linken Ecke des Bildschirms angezeigt, mit VLC im Vollbildmodus oder in einem mplayer-Fenster. Wurden nur Vor- und Nachlauf entfernt, werden die ersten paar Sekunden des geschnittenen Videos angezeigt und anschließend die letzten paar Sekunden, so dass man Anfang und Ende überprüfen kann. Enthält das Video weitere Schnittpunkte (Werbung entfernt), werden diese ebenfalls angezeigt. Die Wiedergabe springt dann vor bis zu einem Zeitpunkt einige Sekunden vor dem ersten Schnitt. Der Schnittpunkt wird dann durch einen Untertitel „Interner Schnitt 1“ angezeigt. Das wird für alle Schnittpunkte wiederholt, wobei die Schnittnummer hochgezählt wird. Damit kann man die Qualität der Schnitte überprüfen. Zuletzt werden dann die letzten paar Sekunden des Videos angezeigt. Nach jeder Videoanzeige erscheint eine Abfrage im Terminal:

Sind die Schnitte OK (j,n)?

Sollen die Schnitte erneut angezeigt werden (w)?

Anzeige wiederholen mit neuem Zeitfenster (8,10,12,15,20,25,30)?

Eingabe:

Antwortet man mit „n“, wird die geschnittene Datei gelöscht und die ungeschnittene wieder aus dem Trash-Verzeichnis restauriert (zum erneuten Schneiden). Mit „j“ sollte man antworten, wenn man den Schnitt akzeptiert.

Man kann die Schnittanzeige auch wiederholen („w“), bei Bedarf auch mit einem größeren oder kleineren Zeitfenster (siehe Ziffern oben). Alle Eingaben müssen mit „Return“ abgeschlossen werden.

Nach dem erfolgreichen Schneiden werden evtl. vorhandene Kopien im MKV-Format (im Ordner „converted“) automatisch gelöscht.

Falls OTR eine zusätzliche separate AC3-Tonspur anbietet (nur für HD-Aufnahmen) und man den entsprechenden otrkey heruntergeladen und dekodiert hat, wird diese Datei ebenfalls geschnitten und in die geschnittene Datei als zweite Tonspur einkopiert. Dies funktioniert auch mit HQ-Videodateien.

C: Schneiden mit manueller Auswahl

Manchmal ist es wünschenswert, nur eine bestimmte Datei zu schneiden anstatt alle möglichen auf einmal. Wenn Sie dieses Kommando eingeben erscheint zunächst eine Auswahl:

Wählen Sie eine Datei zum Schneiden aus:

- (1) `Das_letzte_Problem_20.01.31_20-15_arte_90_TV00N_DE.mpg.HD.avi`
- (2) `Doubt_Glaubensfrage_20.02.03_20-15_arte_100_TV00N_DE.mpg.HD.avi`
- (3) `Geisterfjord_20.02.01_20-15_tele5_130_TV00N_DE.mpg.HQ.avi`
- (4) `Mitternachtsspitzen_20.02.01_21-45_wdr_60_TV00N_DE.mpg.HQ.avi`
- (5) `Silent_Witness_20.02.03_21-00_ukbbc_60_TV00N_DE.mpg.HQ.avi`

Eingabe (Ziffer):

Man wählt nun mit der entsprechenden Ziffer die Datei aus, die man schneiden möchte. Das läuft genau so ab wie oben beschrieben, also je nach Options-Einstellung auch mit Schnittvorschau und nachträglicher Schnittkontrolle.

Mit jeder anderen Eingabe kehrt man zum Hauptmenü zurück.

s: Suche Cutlists auf cutlist.at

Das Programm erstellt eine Suchliste für alle ungeschnittenen Dateien als HTML-Seite und öffnet diese in einem Browser. Ist die kweb suite installiert, wird die Seite in kweb geöffnet, ansonsten wird ein passender Browser gesucht.

Suche Cutlists mit deinem FRED

[Das_letzte_Problem_20.01.31_20-15_arte_90](#)
[Doubt_Glaubensfrage_20.02.03_20-15_arte_100](#)
[Geisterfjord_20.02.01_20-15_tele5_130](#)
[Mitternachtsspitzen_20.02.01_21-45_wdr_60](#)
[Nomis_Die_Nacht_des_Jaegers_20.02.03_22-15_zdf_90](#)
[Silent_Witness_20.02.03_21-00_ukbbc_60](#)

Der Zusatz „mit deinem Fred“ in der Überschrift erscheint nur, wenn man bei cutlist.at registriert ist und den Registrierungslink (FRED) in rpiotrtool eingegeben hat. Dann kann man die heruntergeladenen Cutlists auf cutlist.at später bewerten.

Klickt man auf einen der Links, wird cutlist.at mit der entsprechenden Suchanfrage in einem neuen Fenster oder Tab geöffnet. Es wird nach dem verkürzten Dateinamen gesucht. Man findet also Cutlists für alle Formate. rpiotrtool kann auch cutlists verarbeiten, die für ein anderes Format erstellt worden sind. Die cutlists sollten im Standard-Downloadordner abgelegt werden (den man in rpiotrtool eingestellt hat) und werden dann von rpiotrtool automatisch eingesammelt. Man sollte aber immer nur eine Cutlist herunterladen und die dann zunächst ausprobieren (Vorschau oder Schneiden.) Erst wenn man diese verworfen hat sollte man die nächste ausprobieren.

k: Konvertiere Dateien in das MKV-Format

Es werden alle ungeschnittenen Dateien (für die das noch nicht geschehen ist) in das MKV-Format konvertiert. Es handelt sich dabei um Kopien, die im Verzeichnis „converted“ abgelegt werden. MP4-Dateien werden nicht konvertiert. Es handelt sich dabei nur um eine Umwandlung in ein anderes Containerformat; es wird also nichts neu enkodiert. Deshalb geht das Ganze recht schnell.

Diese Funktion wurde eingebaut, weil VLC die Original AVI-Dateien von OTR nur ruckelig abspielen kann. Wenn man für die Vorschau omxplayer oder mplayer (auf dem Rpi 4) verwendet, ist das Konvertieren nicht nötig.

Nach einem erfolgreichen Schneiden werden die MKV-Dateien automatisch gelöscht.

Wenn man die entsprechende Option eingestellt hat, kann das Konvertieren auch automatisch nach dem Dekodieren durchgeführt werden.

l: Lösche alle Dateien im Trash-Verzeichnis

Der Mülleimer (ungeschnittene Dateien und Cutlists) wird geleert. Es erscheint zudem eine Abfrage, ob man auch alle nach MKV konvertierten Dateien löschen will.

w: Erzeuge Web-GUI

Mit diesem Befehl wird eine neues Web-GUI erzeugt. Das funktioniert nur, wenn man die kweb suite installiert hat.

Man muss diese Funktion jedesmal aufrufen, wenn man die Einstellungen der Pfade, für die Cutlists oder für das Web-GUI geändert hat.

q: Programm beenden

Beendet das Programm. Das Terminal wird automatisch geschlossen, wenn man rpiotrtool aus dem Programm-Menü gestartet hat.

Man kann stattdessen auch einfach das Terminal schließen.

a: Alles: Dekodieren und Schneiden

Mit diesem Kommando werden eine Reihe von Einzelprozessen nacheinander ausgeführt:

- 1) Dekodieren neu gefundener Otrkeys und Einsammeln von heruntergeladenen Cutlists und unkodierter Dateien (Direkt-Download von OTR).
- 2) Optional (falls eingestellt) Konvertieren nach MKV.
- 3) Optional (falls eingestellt) Schnitt-Vorschau aller Aufnahmen, für die eine Cutlist vorhanden ist.
- 4) Schneiden aller Dateien, für die eine Cutlist vorhanden ist.
- 5) Optional (falls eingestellt) Schnittkontrolle aller neu geschnittenen Dateien.

D) Bedienungsanleitung: Web-GUI

1) Was ist das und wie funktioniert es?

Ein GUI (Graphical User Interface = Graphische Benutzer-Schnittstelle) ist eine Programmoberfläche die in einem Fenster läuft und (weitestgehend) mit der Maus bedient werden kann.

Ein Web-GUI ist eine solche Programmoberfläche, die in einem Web-Browser läuft. Dazu ist in der Regel ein Server-Backend nötig, entweder als separate Anwendung oder in ein Programm integriert. Bei meinem „Minimal Kiosk Browser“ (kweb) ist das allerdings anders. Er kann aus lokalen HTML-

The screenshot displays the RPi-OTR-Tool Web-GUI (kweb) interface. The browser address bar shows the file path: `file:///home/pi/otr/web/index.html`. The interface is divided into a left sidebar with navigation menus and a main content area. The sidebar includes sections like 'Aufnahmen', 'Schneiden', 'Verzeichnisse', and 'Web-Video'. The main content area shows a grid of TV channels and a 'Senderauswahl' section with dropdown menus for day, sender, and time, and a list of program listings for ZDFneo on Sunday, 16.02.2020.

Dateien externe Programme ansteuern und eignet sich deshalb sehr gut zur Erstellung einfacher Web-GUIs. Eine solche kweb-Anwendung ist also auch das Web-GUI von rpiotrtool. Das setzt natürlich voraus, dass meine kweb

suite installiert ist, aber die braucht man ja auch zum Erstellen eigener Cutlists mit omxplayerGUI (manuelles Schneiden).

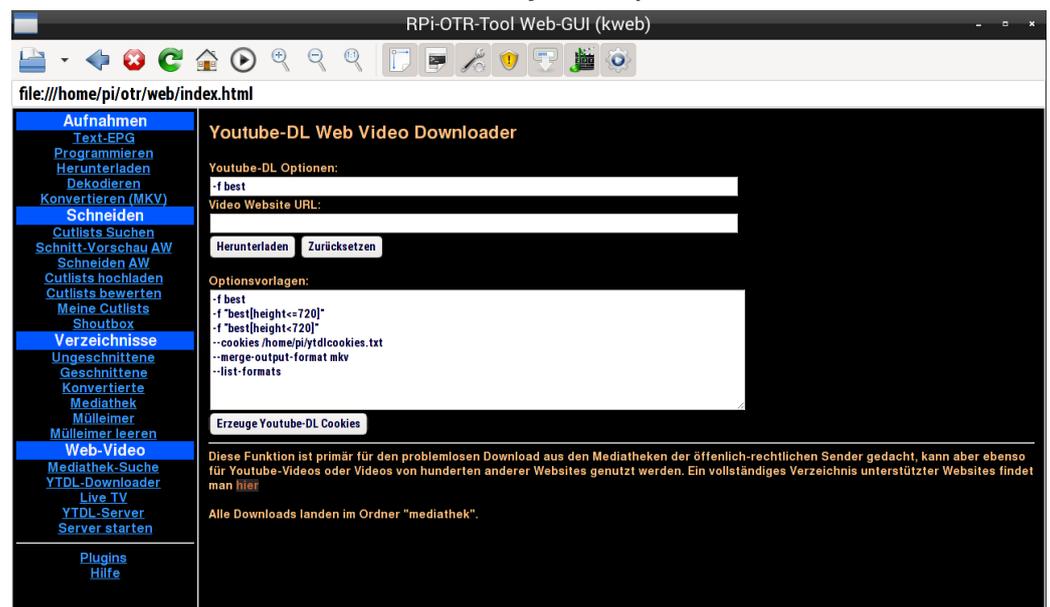
Bei der Arbeit mit OTR benötigen wir ja einerseits Internetfunktionen (z. B. die Webseiten von OTR zum Programmieren und Herunterladen von Aufnahmen und cutlist.at zum Suchen, Herunterladen, Hochladen und Bewerten von Cutlists), zum Anderen die eigentlichen Programmfunktionen von rpiotrtool (Dekodieren, Schneiden usw.). Beides wird nun mit dem Web-GUI in einer einzigen Anwendung zusammengefasst. Darüber hinaus bietet es noch eine Reihe von zusätzlichen Funktionen: So kann ein einfaches Text-EPG der Zeitschrift „HÖRZU“ eingeblendet werden, passend für die Benutzung mit OTR konfiguriert. Diese EPG verzichtet auf allen grafischen Schnickschnack und ermöglicht so das schnelle Auffinden von Sendungen, die man zur Aufnahme programmieren möchte.

Weitere Funktionen dienen dem Zugriff auf die Mediatheken der öffentlich-rechtlichen TV-Sender (und auch auf andere Videoquellen aus dem Internet) und auf Live-TV-Streams. Eine Plugin-Funktion ermöglicht dem Anwender den Einbau eigener Erweiterungen.

Bevor man das Web-GUI nutzen kann, muss man es zunächst erzeugen. Dazu dient eine entsprechende Funktion in rpiotrtool (w). Verändert man Einstellungen in den Gruppen : „1: Pfade“, „3: Cutlists“ oder „6: Web-GUI“, muss das Web-GUI anschließend immer neu generiert werden. Das geht aber blitzschnell.

Für das Web-GUI stehen zwei unterschiedliche Layouts („normal“ und

„kompakt“) zur Verfügung. Das Kompaktformat benutzt Links statt Buttons (spart eine Menge Platz). Außerdem stehen noch unterschiedliche Zoomstufen zur Verfügung, was



die Anpassung an unterschiedliche Bildschirme erleichtert. Des Weiteren können drei unterschiedliche Farbgestaltungen (Hintergrundfarbe weiß, (dunkel-) grau oder schwarz) gewählt werden.

Grundsätzlich kann man die Funktionen des Web-GUI in vier unterschiedliche Gruppen einteilen:

a) Externe Webseiten in separater Browser-Instanz. Dies sind die Webseiten von OTR und die Mediatheksuche (mediathekviewweb.de). Sie werden in einem neuen Fenster geöffnet, wozu ein separater Browser gestartet wird. Standardmäßig ist das kweb; optional können aber auch andere Browser verwendet werden, wenn man ausreichend Arbeitsspeicher zur Verfügung hat (empfohlen ab 2 GB).

b) Web-Inhalte die rechts im Fenster eingeblendet werden. Das sind z. B. das Text-EPG und alle Funktionen von cutlist.at.

c) Lokale Inhalte, die selbst wieder eigene kweb-Anwendungen sind wie der Youtube-DL Web Video Downloader oder die Live-TV-Seite. Auch diese werden im rechten Teil des Web-GUI eingeblendet.

d) Die eigentlichen rpiotrtool Funktionen (Dekodieren, Schneiden usw.). Diese werden in einem Terminal gestartet (das automatisch geöffnet wird). Dies dient einerseits der Kontrolle (z. B. ob alle Dekodierungen erfolgreich waren) und zum Anderen verlangen ja einige Funktionen Benutzereingaben wie etwa die Schnittkontrolle.

2) Die Funktionen im Einzelnen

Das Funktionsmenü in der linken Fensterspalte ist in vier Gruppen unterteilt.

Erste Gruppe: Aufnahmen Aufnahmen

Diese Gruppe enthält die eigentlichen OTR-Funktionen: Aufnahmen programmieren, herunterladen und dekodieren (sofern es sich um verschlüsselte Otrkeys handelt). Das Text-EPG stellt eine nützliche Hilfe beim Programmieren zur Verfügung. Und die Konvertierfunktion packt die OTR-AVI Dateien in einen MKV-Container, da sie sonst mit VLC nicht sauber abgespielt werden können.

Text-EPG

Klickt man diesen Button (Link) an, erscheint im rechten Teil des Fensters

das Text-EPG der Fernsehzeitschrift „HÖRZU“. Genau genommen handelt es sich zunächst einmal um lokale Webseiten, in die das Text-EPG (aus dem Internet) in unterschiedlichen Arten eingeblendet wird. Gestartet wird im Menü-Modus, bei dem links eine Auswahl von 50 TV-Sendern erscheint. Es handelt sich dabei um die Sender (mit der Ausnahme von RTL), die von OTR aufgenommen werden und für die zugleich ein Text-EPG zur Verfügung steht. Klickt man einen Sender an, erscheint das heutige Programm dieses Senders im rechten Rahmen.

Es gibt jedoch noch weitere Optionen, die man in der Kopfzeile anwählen kann: Klickt man auf „Heute“, „Morgen“, „Übermorgen“ usw. erscheint das Programm des jeweiligen Tages für alle 50 Sender untereinander im rechten Kasten.

Von Interesse mag auch noch die Suchfunktion sein, mit der man im Text-EPG nach bestimmten Sendungen suchen kann.

Programmieren

Klickt man diesen Button an, öffnet sich die Aufnahme-Seite von OTR (EPG) in einer neuen Browser-Instanz (in einem neuen Fenster). Standardmäßig wird dabei kweb verwendet, es wird aber nicht nur ein neues Fenster geöffnet, sondern ein neues Browserprogramm gestartet. Wenn die OTR-Website (die ja bekanntermaßen recht langsam lädt) dann ein Weilchen zum Laden braucht, kann man zwischenzeitlich zum Web-GUI-Fenster zurückschalten und schon mal das Text-EPG durchforsten.

Es ist auch möglich, statt kweb einen anderen „großen“ Browser wie Chromium, Firefox oder Vivaldi für diese Funktion zu nutzen, aber die brauchen dermaßen viel Speicherplatz, dass ich davon abrate, wenn man nicht wenigstens 2 GB Arbeitsspeicher zur Verfügung hat.

Wenn man diese Funktion (oder irgendeine andere OTR-Funktion) zum ersten Mal nutzt, muss man auf die Hauptseite von OTR wechseln und sich anmelden. Das wird dann in einem Cookie abgelegt, so dass man später die Aufnahmeseite immer direkt erreicht.

Herunterladen

Klickt man diesen Button an, öffnet sich in einer neuen Browser-Instanz die Seite „Meine Aufnahmen“ von OTR. Hier gilt das gleiche, was bereits oben zur Programmier-Funktion gesagt wurde.

Verwendet man kweb für diese Funktion, muss man einige Besonderheiten beachten. kweb verwendet für alle Downloads ein externes Programm, den Downloadmanager uget. Es hängt von der Konstruktion der Website ab, wie ein solcher Download gestartet werden kann:

Beim Direktdownload von OTR kann man die Downloadfunktion wie gewohnt mit der linken Maustaste aufrufen.

Bei den meisten Mirrorservern muss man aber mit der rechten Maustaste auf den Downloadlink klicken und dann „Verweisziel herunterladen“ anwählen. Bei manchen Mirrorservern startet der Download auch automatisch, aber nicht bei allen.

Dekodieren

Klickt man diesen Button (Link) an, öffnet sich ein Terminal, in dem rpiotrtool läuft und alle neu gefundenen OTR-Keys dekodiert. Dabei werden auch gleich neue Cutlists aus dem Downloadordner eingesammelt und ebenso von OTR direkt heruntergeladene (nicht verschlüsselte) OTR-Aufnahmen.

Nachdem das Dekodieren fertig ist, wartet das Programm auf eine Benutzereingabe (Return-Taste). Man sollte immer erst kontrollieren, ob alle Dekodierungen erfolgreich waren, bevor man das Programm beendet (Return-Taste oder einfach Fenster schließen).



```
rpiotrtoolcmd.sh
Datei Bearbeiten Reiter Hilfe
Im_Visier_des_Killers_20.02.17_22-15_zdf_100_TV00N_DE.mpg.HQ.avi.otrkey
OTRpiDecoder, v1.2
Verbinde zum Server... OK
Dekodieren...
[=====] 100%
Fertig
Navy_CIS_S17E07_20.02.19_01-20_sat1_45_TV00N_DE.mpg.HQ.avi.otrkey
OTRpiDecoder, v1.2
Verbinde zum Server... OK
Dekodieren...
[=====] 100%
Fertig
Navy_CIS_L_A_S11E07_20.02.19_02-05_sat1_45_TV00N_DE.mpg.HQ.avi.otrkey
OTRpiDecoder, v1.2
Verbinde zum Server... OK
Dekodieren...
[=====] 100%
Fertig
Mit Return-Taste beenden
```

Konvertieren (MKV)

Startet den Konvertiervorgang nach MKV. Auch hier wird wieder ein Terminal geöffnet, in dem die entsprechende rpiotrtool-Funktion läuft. Ob nach der Beendigung auf eine Benutzereingabe gewartet wird, hängt von Web-GUI-Einstellungen ab, die man vorgenommen hat.

Hinweise:

Je nach Einstellungen unter „Optionen“ kann das Konvertieren auch automatisch nach dem Dekodieren durchgeführt werden.

Das Konvertieren nach MKV ist nur dann nötig (oder empfehlenswert), wenn man VLC zum Anschauen der ungeschnittenen Videos oder zur Schnittvorschau verwenden möchte.

Zweite Gruppe: **Schneiden**

Diese Gruppe enthält alle wichtigen Funktionen zum Schneiden von OTR-Aufnahmen: Cutlists suchen, Schnittvorschau und das eigentliche Schneiden mit optionaler nachträglicher Schnittkontrolle. Hat man sich bei cutlist.at registriert und seinen „FRED“ (Registrierungslink) eingetragen, kann man auch eigene Cutlists hochladen, heruntergeladene Cutlists bewerten und weitere cutlist.at-Funktionen nutzen.

Cutlists Suchen

Ruft man diese Funktion auf, erscheint im rechten Teil des Fensters kurzfristig eine Meldung: **Erstelle Suchliste ...**

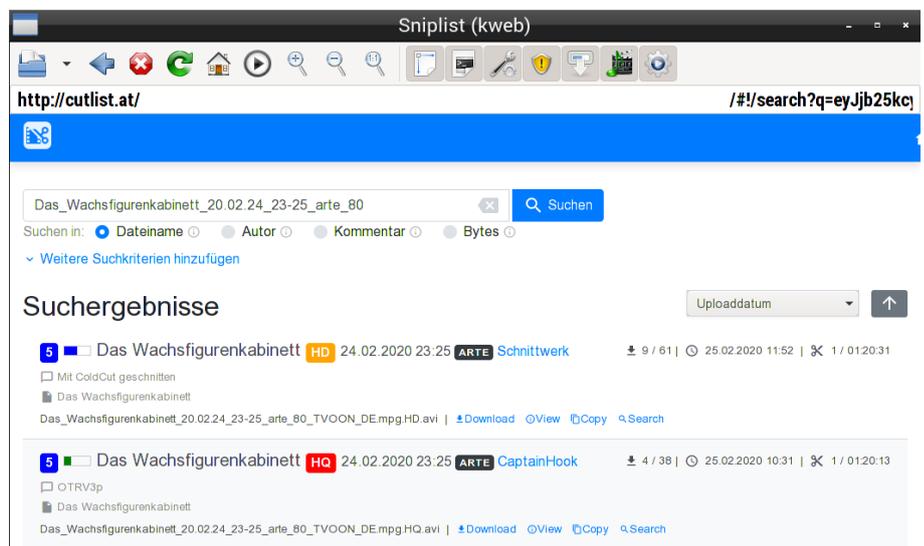
Im Hintergrund wird von rpiotrtool eine neue Suchliste erstellt, falls ungeschnittene Aufzeichnungen ohne passende Cutlists existieren.

Diese Suchliste wird dann angezeigt und zwar als Links zum



Aufrufen von cutlist.at mit oder ohne Registrierungslink (FRED). Klickt man

einen dieser Links an, wird in einem neuen Browserfenster die entsprechende Suchseite von cutlist.at geöffnet. Da immer nach dem gekürzten Dateinamen gesucht wird, werden Suchergebnisse aller Dateiformate angezeigt.



Optimal ist natürlich eine Cutlist für das gleiche Format (HQ, HD usw.), aber in der Regel kann man problemlos auch Cutlists von anderen Formaten verwenden. Klickt man bei einer Cutlist auf „Download“, öffnet sich ein weiteres kweb-Fenster und dann auch gleich uget, der Downloadmanager,

der standardmäßig von kweb verwendet wird. Nach dem erfolgreichen Download kann man beide kweb-Fenster wieder schließen.

Man kann dies nun für alle Einträge in der Suchliste wiederholen oder gleich zu den nächsten Schritten (Schnittvorschau und Schneiden) übergehen.

Schnitt-Vorschau **AW**

Die Schnittvorschau dient der Qualitätskontrolle der heruntergeladenen Cutlists. Sind Anfang und Ende der Aufnahme perfekt geschnitten? Wurden Werbeblöcke und doppelte Szenen entfernt? Dabei hat man zwei Möglichkeiten:

Klickt man auf „Schnittvorschau“, erhält man nacheinander eine Vorschau aller Aufnahmen, für die eine Cutlist existiert (heruntergeladen oder selbst erstellt). Man kann dann für jeden Einzelfall entscheiden, ob man die Schnittliste akzeptiert oder verwirft (und evtl. eine andere ausprobiert).

Klickt man stattdessen auf „AW“ (steht für „Auswahl“), kann man explizit eine einzelne Aufnahme für die Schnittvorschau auswählen.

In beiden Fällen handelt es sich um rpiotrtool-Funktionen, die in einem Terminal ablaufen. Die visuelle Schnittkontrolle kann je nach den Optionseinstellungen entweder mit omxplayer, VLC oder mplayer durchgeführt werden. Auf dem Raspberry Pi 4B kann omxplayer nicht verwendet werden; bei allen älteren Modellen wird dies aber explizit empfohlen.



```
pi@raspberrypi3: ~/rpiotrtool
Datei Bearbeiten Reiter Hilfe
Schnittvorschau: Navy_CIS__L_A_S11E07_20.02.18_21-15_sat1_60_TV00N_DE.mpg.HQ.avi
Sind die Schnitte OK (j,n)?
Soll die Vorschau erneut angezeigt werden (w)?
Vorschau wiederholen mit neuem Zeitfenster (8,10,12,15,20,25,30)?
Eingabe: w
Sind die Schnitte OK (j,n)?
Soll die Vorschau erneut angezeigt werden (w)?
Vorschau wiederholen mit neuem Zeitfenster (8,10,12,15,20,25,30)?
Eingabe: j
```



Schneiden **AW**

Klickt man „Schneiden“ an, werden alle Dateien, für die eine Cutlist vorliegt,



```
rpiotrtool
Datei Bearbeiten Reiter Hilfe
Schneide: Navy_CIS__L_A_S11E07_20.02.18_21-15_sat1_60_TV00N_DE.mpg.HQ.avi
Erfolgreich geschnittene Datei: Navy_CIS__L_A_S11E07_20.02.18_21-15_sat1_60.mkv
Schnitte anzeigen: /home/pi/otr/cut/Navy_CIS__L_A_S11E07_20.02.18_21-15_sat1_60.mkv
Sind die Schnitte OK (j,n)?
Sollen die Schnitte erneut angezeigt werden (w)?
Anzeige wiederholen mit neuem Zeitfenster (8,10,12,15,20,25,30)?
Eingabe: █
```

nacheinander geschnitten. Dies ist eine rpiotrtool-Funktion, die in einem Terminal abläuft. Hat man die Schnittkontrolle nach dem Schneiden aktiviert, werden anschließend alle Schnitte der jeweiligen Aufnahme angezeigt: Anfang und Ende des Videos und auch interne Schnitte, wenn z. B. Werbesequenzen entfernt wurden. Bei jedem Video kann dann entschieden werden, ob das Schnitteergebnis akzeptiert wird oder nicht.

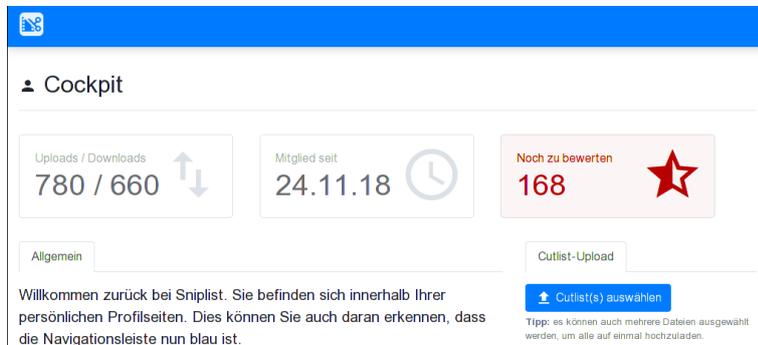


Will man nur eine einzelne Datei schneiden, klickt man auf „AW“ (für Auswahl). Dann kann man (im Terminal) die Datei auswählen, die geschnitten werden soll.

Wichtiger Hinweis: Die folgenden vier Funktionen stehen nur zur Verfügung, wenn man sich bei cutlist.at registriert hat und seinen Registrierungslink (FRED) in den cutlist-Einstellungen von rpiotrtool eingegeben hat.

Cutlists hochladen

Klickt man diese Funktion an, erscheint im rechten Teil des Fensters die (persönliche) Cockpit-Seite von cutlist.at. Klickt man dort auf „Cutlist(s) auswählen“, öffnet sich ein Dateiauswahlfenster und man kann eine oder mehrere Cutlists zum



Hochladen markieren. Man muss dazu in das Verzeichnis „upload“ im otr-Verzeichnis navigieren. Es ist deshalb praktisch, den Ordner „otr“ im normalen Dateimanager als Favoriten einzutragen.

Cutlists bewerten

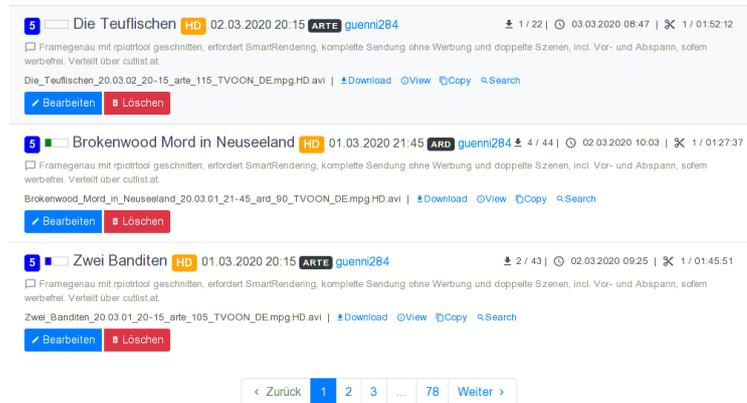
cutlist.at bietet die Möglichkeit, Cutlists, die man zuvor (als registrierter Benutzer) heruntergeladen hat, nachträglich zu bewerten. Die



meisten Cutlists enthalten ja eine Qualitätsangabe des Autors (von 0 bis 5) und man kann hier seine eigene Bewertung eintragen. Das sollte man natürlich nur machen, wenn man die Cutlist zuvor geprüft hat, am besten mit der Schnittkontrolle nach dem Schneiden.

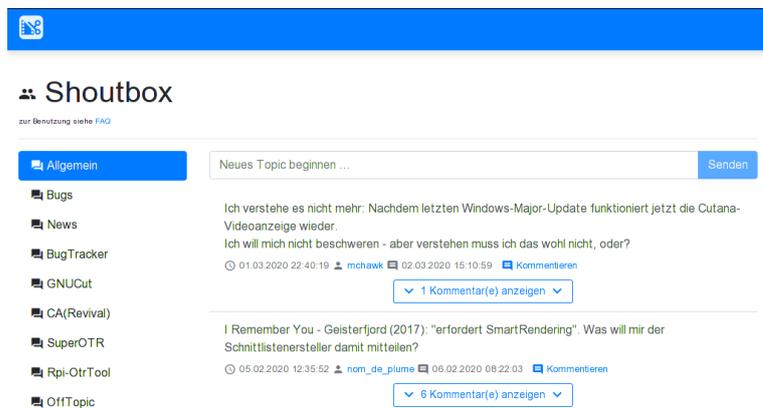
Meine Cutlists

Will man z. B. sehen, wie oft die eigenen Cutlists heruntergeladen wurden (nur von cutlist.at, sie werden aber oft auch von OTR selbst übernommen und angeboten) oder wie sie bewertet wurden, kann man diese Funktion benutzen. Es öffnet sich dann im rechten Teile des Fensters die „Uploads“-Seite von cutlist.at.



Shoutbox

Wenn man diese Funktion aufruft, erscheint die Shoutbox von cutlist.at im rechten Teil des Fensters. Die Shoutbox ist die interne Kommunikationsplattform der bei cutlist.at registrierten Benutzer. Dort kann man eigene Mitteilungen veröffentlichen oder die von anderen Benutzern kommentieren. Hier findet man auch oft nützliche Informationen zu den verschiedenen Programmen, die man zum Bearbeiten der OTR-Aufnahmen braucht, darunter auch rpiotrtool.



Dritte Gruppe: Verzeichnisse Verzeichnisse

Die Funktionen in dieser Gruppe bieten den Zugriff auf die wichtigsten OTR-Verzeichnisse. Dabei gibt es zwei Varianten, ja nach Optionseinstellung:

Name	Size	Date Modified
The_Rookie_20.03.06_22-00_zdfneo_40_TVOON_DE.mpg.HD.avi	648,7 MB	09:23:55 07.03.2020
The_Rookie_20.03.06_22-40_zdfneo_40_TVOON_DE.mpg.HD.avi	621,9 MB	09:24:22 07.03.2020
Paulette_Die_etwas_andere_Oma_20.03.06_21-00_one_80_TVOON_DE.mpg.HO.avi	760,2 MB	09:23:26 07.03.2020
Der_Garten_des_Boesen_20.03.06_22-50_bay3_95_TVOON_DE.mpg.HO.avi.cutpoints	30 Bytes	09:28:30 07.03.2020
Der_Garten_des_Boesen_20.03.06_22-50_bay3_95_TVOON_DE.mpg.HO.avi	790,7 MB	09:22:28 07.03.2020
Paulette_Die_etwas_andere_Oma_20.03.06_21-00_one_80_TVOON_DE.mpg.HO.avi.cutpoints	30 Bytes	09:33:08 07.03.2020
The_Rookie_20.03.06_22-00_zdfneo_40_TVOON_DE.mpg.HD.avi.cutpoints	30 Bytes	09:35:08 07.03.2020
Der_Garten_des_Boesen_20.03.06_22-50_bay3_95_TVOON_DE.mpg.HO.avi.cutlist	893 Bytes	09:37:53 07.03.2020
The_Rookie_20.03.06_22-40_zdfneo_40_TVOON_DE.mpg.HO.avi.cutpoints	29 Bytes	09:37:33 07.03.2020
Paulette_Die_etwas_andere_Oma_20.03.06_21-00_one_80_TVOON_DE.mpg.HO.avi.cutlist	900 Bytes	09:37:56 07.03.2020
The_Rookie_20.03.06_22-00_zdfneo_40_TVOON_DE.mpg.HD.avi.cutlist	885 Bytes	09:37:58 07.03.2020
The_Rookie_20.03.06_22-40_zdfneo_40_TVOON_DE.mpg.HD.avi.cutlist	885 Bytes	09:38:00 07.03.2020

1) Die Verzeichnisse werden im rechten Teil des Browserfensters angezeigt. Aus technischen Gründen funktioniert das nur, wenn man als Hintergrundfarbe „weiß“ gewählt hat. Videos in den Verzeichnissen kann man direkt durch Anklicken mit omxplayerGUI öffnen (z. B. zum manuellen Schneiden). Außerdem können sich dort ja auch Cutlists oder Cutpoint-Dateien befinden. Dabei handelt es sich um Textdateien, die unmittelbar im Browserfenster angezeigt werden können.

2) Die entsprechenden Verzeichnisse werden im Dateimanager angezeigt. Das hat den Vorteil, dass man Dateien auch gleich löschen, umbenennen oder verschieben kann. Wählt man in den Web-Optionen eine andere Hintergrundfarbe als „weiß“, wird dieser Modus automatisch eingeschaltet.

Des Weiteren findet man in dieser Gruppe noch eine Funktion zum Löschen des Mülleimers.

Ungeschnittene

Zeigt den Inhalt des Ordners „otr/decoded“ an, in dem sich die ungeschnittenen OTR-Aufnahmen befinden.

Geschnittene

Zeigt den Inhalt des Ordners „otr/cut“ an, in dem sich die geschnittenen OTR-Aufnahmen befinden.

Konvertierte

Zeigt den Inhalt des Ordners „otr/converted“ an, in dem sich in das MKV-Format konvertierte (ungeschnittene) OTR-Aufnahmen befinden (sofern man diese Funktion nutzt).

Mediathek

Zeigt den Inhalt des Ordners „otr/mediathek“ an, in dem sich alle Videodateien befinden, die man mit dem Youtube-DL-Downloader heruntergeladen hat.

Mülleimer

Zeigt den Inhalt des Ordners „otr/trash“ an, in den nach dem Schneiden die ungeschnittenen OTR-Aufnahmen sowie die Cutlists und Cutpoint-Dateien verschoben werden.

Mülleimer leeren

Dies ist eine rpiotrtool-Funktion die in einem Terminal-Fenster ausgeführt wird. Es werden alle Dateien im „otr/trash“-Verzeichnis unwiederbringlich gelöscht. Außerdem wird abgefragt, ob Dateien im Verzeichnis „otr/converted“ ebenfalls gelöscht werden sollen (falls dieser Ordner nicht leer ist).

Vierte Gruppe: Web-Video Web-Video

Diese Gruppe enthält keine spezifischen OTR-Funktionen. Aber wer OTR nutzt, wird wohl häufig auch die Mediatheken der öffentlich-rechtlichen Sender nutzen. Deshalb wurde hier eine Mediathek-Suche eingebaut und eine kweb-Anwendung zum Herunterladen von Web-Videos (nicht nur aus den Mediatheken) mit Hilfe von Youtube-DL. Außerdem kann man auf viele TV-Live-Streams im Internet zugreifen und sie mit omxplayerGUI oder VLC abspielen. Ebenso wird der Youtube-DL-Server unterstützt (Teil meiner kweb suite), der es ermöglicht, Web-Videos von hunderten von Websites unmittelbar mit omxplayerGUI oder VLC anzuschauen und der auch für den Zugriff auf die lokale Mediensammlung eingerichtet werden kann.

Mediathek-Suche

Ruft man diese Funktion auf, öffnet sich die Mediathek-Suchmaschine mediathekviewweb.de in einer neuen Browser-Instanz. Je nach Einstellung ist das entweder kweb oder ein anderer gewählter Browser.

Sender	Thema	Titel	Datum	Zeit	Dauer	Video
ZDF	Unterleuten	Unterleuten - Das zerrissene Dorf (3) - Nach dem Roman "Unterleuten" von Juli Zeh	12.03.2020	20:15	93:55	
ZDF	Unterleuten	Unterleuten - Das zerrissene Dorf (3) - Nach dem Roman "Unterleuten" von Juli Zeh (Audiodeskription)	12.03.2020	20:15	93:55	
ZDF	Unterleuten	Unterleuten - Das zerrissene Dorf (2) - Nach dem Roman "Unterleuten" von Juli Zeh	11.03.2020	20:15	94:40	
ZDF	Unterleuten	Unterleuten - Das zerrissene Dorf (2) - Nach dem Roman "Unterleuten" von Juli Zeh (Audiodeskription)	11.03.2020	20:15	94:40	
ZDF	Unterleuten	Wer ist wer?	09.03.2020	20:15	01:40	
ZDF	Unterleuten	Unterleuten - Das zerrissene Dorf (1) - Nach dem Roman "Unterleuten" von Juli Zeh (Audiodeskription)	09.03.2020	20:15	93:51	
ZDF	Unterleuten	Unterleuten - Das zerrissene Dorf (1) - Nach dem Roman "Unterleuten" von Juli Zeh	09.03.2020	20:15	93:51	
ZDF	Unterleuten	Darum geht's	19.02.2020	13:30	00:30	
ARD	Nordmagazin	"Unterleuten" von Juli Zeh in Neustrelitz	17.05.2019	19:30	03:00	
NDR	Nordmagazin	"Unterleuten" von Juli Zeh in Neustrelitz	17.05.2019	19:30	03:01	

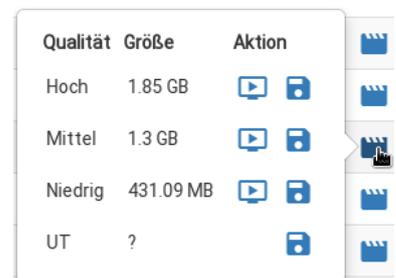
Die Suchmaschine brauchte 4,20 ms. Zeige Treffer 1 bis 10 von insgesamt 10 Treffern.
Filmliste zuletzt um 14:16 Uhr aktualisiert

Man kann mit dieser Suchmaschine in allen Mediatheken der öffentlich-rechtlichen Sender nach verfügbaren Videos suchen, auch nach Beiträgen, die noch gar nicht gesendet worden sind („Zukünftige“ aktivieren).

Insbesondere ZDF, ZDF Neo und Arte stellen oft komplette Serien vor der TV-Ausstrahlung zur Verfügung.

Wer will, kann sich die Sendungen natürlich im Browser anschauen, aber da ist (je nach Browser) gerade auf dem Raspberry Pi die Qualität oft nicht überragend (manche sind auch gar nicht nutzbar). Viel besser ist es, die Sendungen in einem hardwarebeschleunigten Videoplayer (omxplayerGUI, VLC, Kodi) anzuschauen. Wie das geht, wird weiter unten näher beschrieben.

Man kann die Streams aber auch herunterladen, archivieren und später offline anschauen. In der Regel bietet mediathekviewweb.de die Sendungen auch gleich in drei verschiedenen Formaten (Qualitätsstufen) zum Herunterladen an. Dieses Verfahren hat folgende Vorteile:



Qualität	Größe	Aktion
Hoch	1.85 GB	 
Mittel	1.3 GB	 
Niedrig	431.09 MB	 
UT	?	

1) Man kann auf einfache Weise die Qualitätsstufe wählen.

2) Es funktioniert mit allen Sendern (außer Ländereinschränkungen)

Es hat aber auch Nachteile:

1) Die Dateien erhalten sehr kryptische Dateinamen, man muss sie also direkt nach dem Download umbenennen, sonst weiß man später gar nicht mehr was man da heruntergeladen hat.

2) Die Dateien landen (in der Regel) im Standard-Download-Verzeichnis und nicht in unserem Mediathek-Ordner.

Als alternatives Verfahren bietet sich der Download mit dem YTDL-Downloader an, der im nächsten Abschnitt näher beschrieben wird.

Zuvor aber noch ein paar Hinweise zur Nutzung von mediathekviewweb.de mit kweb. Bedingt durch die veraltete Webkit-Engine zeigt kweb das Download-Menü nicht ganz richtig an (es fehlen die Icons). Man kann es trotzdem nutzen, muss aber etwas „blind“ navigieren. Dafür hat man zwei Optionen zur Verfügung. Klickt man mit der rechten Maustaste auf das (unsichtbare) Downloadsymbol, erscheint



ein Pop-Up-Menü. Wählt man darin „Verweis in neuem Reiter öffnen“, öffnet sich tatsächlich ein neues (leeres) Browserfenster, aber gleich darauf omxplayerGUI und man kann den Videostream live ansehen. Wählt man „Verweisziel herunterladen“, öffnet sich der Downloadmanager uget und lädt das Video herunter.

Nicht in allen Fällen führt die Suche mit mediathekviewweb.de zum Erfolg. Es kann sich dann lohnen, die betreffenden Senderseiten direkt zu öffnen und dort zu suchen. Findet man dort die gewünschte Sendung, kann man ebenfalls den YTDL-Downloader zum Herunterladen nutzen.

YTDL-Downloader

Diese Funktion zeigt den Youtube-DL-Downloader im rechten Teil des Fensters an. Man kann diese Webseite stattdessen auch in einem neuen Fenster öffnen, wenn man mit der rechten Maustaste auf den Button (Link) klickt und „Verweis in neuem Reiter öffnen“ auswählt, oder auch gleich in einer neuen Browser-Instanz, wenn man beim Draufklicken (mit der linken Maustaste) die Shift-Taste gedrückt hält (das funktioniert im Übrigen bei allen internen Seiten so).

Der Youtube-DL-Downloader ist wiederum eine typische kweb-Anwendung: Ein Web-Frontend (Web-GUI) für das



Kommandozeilenprogramm youtube-dl. Dieses wird bei der Installation der kweb suite automatisch installiert.

youtube-dl wurde (wie der Name schon sagt) ursprünglich entwickelt, um das Herunterladen von Youtube-Videos zu ermöglichen. Mittlerweile kann die in Python geschriebene Anwendung aber viel mehr: Sie enthält hunderte von Modulen für die unterschiedlichsten Websites (darunter auch die meisten öffentlich-rechtlichen deutschen Mediatheken). Entsprechend kompliziert kann auch manchmal die Benutzung sein.

kweb nutzt youtube-dl bereits seit der allerersten Version (1.0 von 2013), aber nicht um Videos herunterzuladen, sondern um die Video-URL aus einer

Video-Webseite zu extrahieren und diese an omxplayer (später omxplayerGUI oder auch VLC) weiterzuleiten. Damit konnte man schon auf den ersten und einfachsten Raspberry Pi Computern (z. B. Rpi B oder Rpi Zero mit nur 512 MB Arbeitsspeicher und einfachem 1-Kern-Prozessor) Youtube-Videos oder auch die Sendungen der Mediatheken in perfekter Qualität abspielen.

Hier geht es aber darum, youtube-dl in seiner eigentlichen Funktion als Web-Video-Downloader zu nutzen und die Bedienung so einfach wie möglich zu



gestalten. Das Web-GUI besteht im wesentlichen aus einem einfachen Formular mit zwei Textfeldern und zwei Buttons.

In das obere Feld kann man eine oder mehrere youtube-dl Optionen einfügen. Dies sind einfach die gleichen Optionen, die man auch bei einer Nutzung als Kommandozeilenprogramm verwenden würde. Das wichtigste dabei sind die Qualitätsvorgaben. Man kann das Feld auch komplett leer lassen. Dann versucht youtube-dl das Format mit der besten Video- und der besten Audioqualität zu finden und herunterzuladen. Unter Umständen werden Video und Audio in getrennten Streams geliefert und auch als zwei getrennte Dateien heruntergeladen, die dann noch zusammengemischt werden müssen. Das kann youtube-dl alles automatisch (benötigt aber unter Umständen bestimmte Hilfsprogramme).

Für das Optionsfeld gibt es eine Reihe von Format-Vorgaben, die mit den Web-Gui-Optionen voreingestellt werden können. Über die Vielzahl der weiteren Optionen kann man sich im beigefügten youtube-dl-Handbuch (unter „Hilfe“) näher informieren.

In das zweite Textfeld „Video Website URL:“ fügt man die URL der Webseite ein, auf der das Video (im Browser) angezeigt wird. In der Suchmaschine hat man z. B. links vor jedem Suchergebnis einen Link zur Senderseite des Videos. Den kann man z. B. mit der rechten Maustaste anklicken und „Verweisziel kopieren“ anwählen. Dann klickt man in das Textfeld mit der

rechten Maustaste und wählt „Einfügen“. Befindet man sich im Browser selbst auf einer Video-Webseite, kann man die URL aus der Adresszeile des Browsers herauskopieren und hier einfügen.

Nun kann man

„Herunterladen“

anklicken, um den

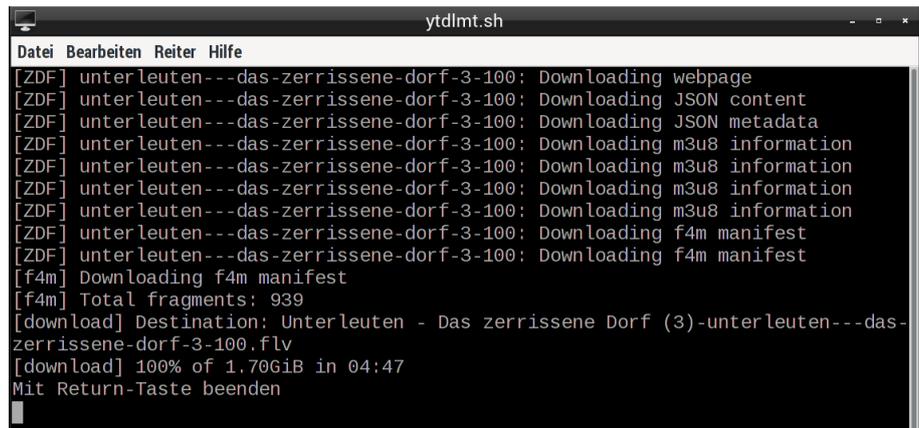
Download zu starten.

Wie bei den rpiotrtool-

Funktionen wird dabei

wieder ein Terminal

geöffnet, in dem man



```
ytdlmt.sh
Datei Bearbeiten Reiter Hilfe
[ZDF] unterleuten--das-zerrissene-dorf-3-100: Downloading webpage
[ZDF] unterleuten--das-zerrissene-dorf-3-100: Downloading JSON content
[ZDF] unterleuten--das-zerrissene-dorf-3-100: Downloading JSON metadata
[ZDF] unterleuten--das-zerrissene-dorf-3-100: Downloading m3u8 information
[ZDF] unterleuten--das-zerrissene-dorf-3-100: Downloading m3u8 information
[ZDF] unterleuten--das-zerrissene-dorf-3-100: Downloading m3u8 information
[ZDF] unterleuten--das-zerrissene-dorf-3-100: Downloading f4m manifest
[ZDF] unterleuten--das-zerrissene-dorf-3-100: Downloading f4m manifest
[f4m] Downloading f4m manifest
[f4m] Total fragments: 939
[download] Destination: Unterleuten - Das zerrissene Dorf (3)-unterleuten---das-zerrissene-dorf-3-100.flv
[download] 100% of 1,70GiB in 04:47
Mit Return-Taste beenden
```

den eigentlichen Download-Vorgang verfolgen kann. Falls etwas schiefgeht, erhält man die entsprechenden Fehlermeldungen.

Der Button „Zurücksetzen“ leert das URL-Feld im Formular und füllt das Optionsfeld mit dem Vorgabewert.

Das Textfeld „Optionsvorlagen“ enthält eine Reihe von praktischen Optionen, die man mit Copy und Paste bei Bedarf in das Optionsfeld übernehmen kann. Den Inhalt kann man selbst erweitern. Im „web“-Verzeichnis befindet sich eine Datei mit dem Namen „ytdloptionlist.txt“. Die kann man mit einem beliebigen Texteditor öffnen und weitere Optionen hinzufügen. Nutzt man z. B. eine Videowebseite, bei der man sich mit Benutzernamen und Passwort einloggen muss, könnte man hinzufügen „-u Benutzername -p Passwort“. Dann hat man diese Option jederzeit zur Verfügung, wenn man von der entsprechenden Website etwas herunterladen will. Nach jeder Änderung dieser Datei muss man in rpiotrtool das WebGUI neu generieren.

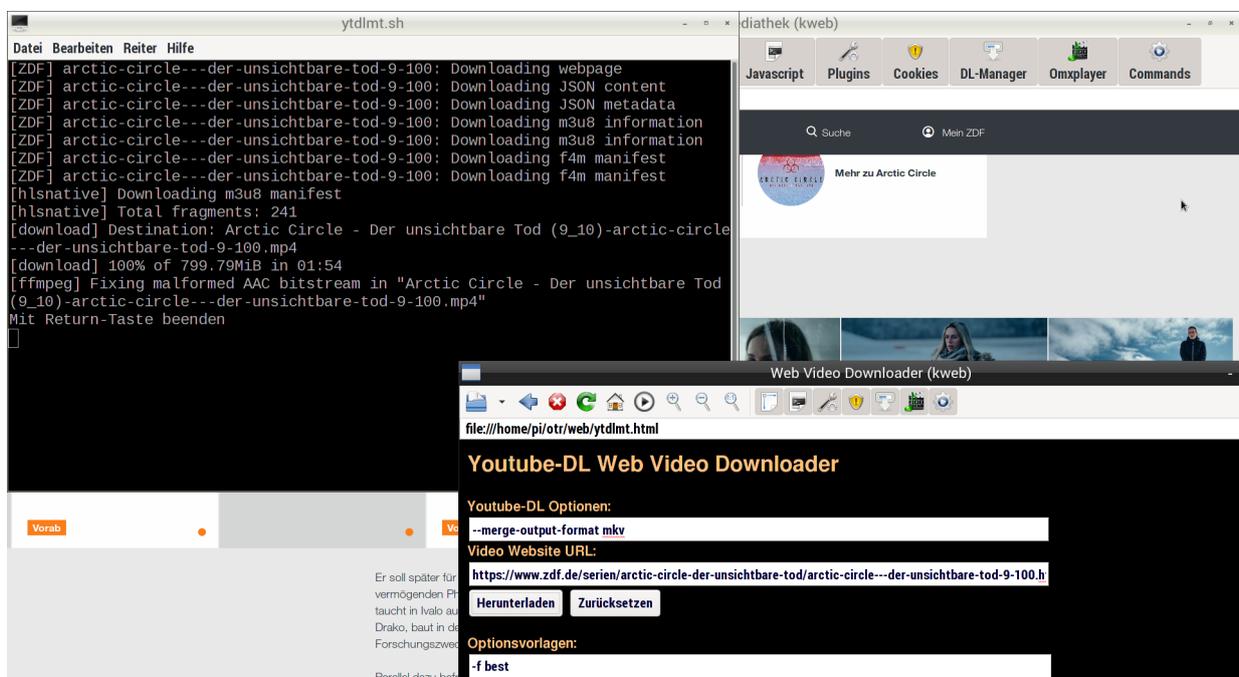
Bei manchen Websites wird nach dem Einloggen ein Cookie generiert, das beim nächsten Besuch dafür sorgt, dass man sich nicht neu einloggen muss. Oder man muss einen Anti-Robot-Test über sich ergehen lassen, dessen Ergebnis dann ebenfalls in einem Cookie abgelegt wird. Macht man das in kweb, kann man anschließend durch Anklicken von „Erzeuge Youtube-DL Cookies“ die Cookie-Liste von kweb in ein für youtube-dl verständliches Format konvertieren. Dann kann man die Option „--cookies /home/pi/ytdlcookies.txt“ in der Optionszeile hinzufügen und youtube-dl kann das passende Cookie dann beim Runterladen verwenden.

Es gibt noch eine besondere Option, mit der man sich vor dem Herunterladen alle verfügbaren Formate und Qualitäten anzeigen lassen kann: „--list-formats“ . Gibt man nur diese Option und eine URL ein, erhält man oft eine sehr lange Liste mit verfügbaren Formaten. Eine Zeile darin könnte z. B. so aussehen:

```
hds-1798                                flv                                unknown                                1798k
```

Das Downloadformat wäre in diesem Falle „flv“ und die Bitrate 1798 KBit. Ganz links steht immer die spezifische Formatvorgabe. Entscheidet man sich für dieses Format, gibt man im Optionsfeld ein „-f hds-1798“ und kann dann genau dieses Format herunterladen.

Das folgende Bild zeigt noch einmal den YTDL-Downloader (als separate Anwendung, im Hintergrund eine ZDF-Seite) in Aktion:



Live TV

Dies ist eine weitere kweb-Anwendung, die das Abspielen von einer Reihe von TV-Live-Streams mit omxplayerGUI oder VLC ermöglicht. Dabei wird auf playlists von <https://github.com/jnk22/kodinerds-iptv> zurückgegriffen.

Playlisten mit Live TV und Radio Streams

Sendergruppe	Abspielen mit:	
Alle TV-Sender	omxplayergui	VLC
TV-Hauptsender	omxplayergui	VLC
Regionale TV-Sender	omxplayergui	VLC
Lokale TV-Sender	omxplayergui	VLC
Extra TV-Sender	omxplayergui	VLC
Shopping TV-Sender	omxplayergui	VLC
AT/CH TV-Sender	omxplayergui	VLC
US/UK TV-Sender	omxplayergui	VLC
Internationale TV-Sender	omxplayergui	VLC
Radiosender	omxplayergui	VLC

Nicht alle Sender können jederzeit und von jedem Ort abgespielt werden.
Die Playlisten stammen von <https://github.com/jnk22/kodinerds-iptv>

„Alle TV-Sender“ enthält die komplette Liste. Darunter befinden sich die einzelnen Teillisten. „Radiosender“ enthält alle Radiosender und wird nicht weiter in Gruppen unterteilt.

VLC eignet sich eigentlich besser zum Abspielen der Streams, weil es schneller umschaltet und auch bei der Bedienung flüssiger reagiert als omxplayerGUI. Allerdings kann derzeit (während ich das Handbuch schreibe) VLC die ZDF-Streams nicht richtig abspielen (nur Ton, kein Bild), wohl aber omxplayerGUI.

YTDL-Server

und

Server starten

Der Youtube-DL-Server ist ein Bestandteil der kweb suite. Er ist ein spezieller kleiner Webserver mit einer Fülle von Mediafunktionen. Klickt man auf „YTDL-Server“, wird seine Hauptseite im rechten Teil des Fensters angezeigt. Dazu muss der Server aber erst einmal gestartet werden. Das kann man durch Anklicken von „Server starten“. Der Server läuft dann als Hintergrundprozess. Er bleibt aktiv, auch wenn man das rpiotrtool Web-GUI schließt, kann aber auf seiner Hauptseite selbst jederzeit wieder abgeschaltet (geschlossen) werden (Button „Stop Server“).

Was hat aber nun der Youtube-DL-Server mit rpiotrtool zu tun? Das ist nicht ganz so einfach zu erklären, vor allem weil sich dieses kleine Tool inzwischen zu einer „eierlegenden Wollmilchsau“ entwickelt hat, das eine Fülle von Multimediafunktionen bietet. Eine ausführliche Beschreibung findet man im omxplayerGUI Handbuch. Ich will mich hier auf die wichtigsten Funktionen beschränken, die für den normalen Anwender relevant sind:

- 1) Beschleunigt das Extrahieren von Web-Videos zum Abspielen mit omxplayerGUI oder VLC (anstatt im Browser).
- 2) Die Hauptseite des Webservers kann in jedem Browser-Programm dazu genutzt werden, Web-Video in omxplayerGUI oder VLC abzuspielen.
- 3) Die in kweb integrierte „Play“-Funktion (= spiele das Video aus dieser Webseite in omxplayerGUI oder VLC ab) kann auf einfache Weise in jedes Browser-Programm integriert werden.
- 4) Zugriff auf die komplette lokale Mediensammlung aus jedem beliebigen Browser-Programm.

Wer sich für diese Funktionen näher interessiert, erfährt in den folgenden Abschnitten mehr darüber. Wenn das nicht der Fall ist, kann man sie einfach überspringen.

1) Beschleunigtes Extrahieren von Web-Videos

Bereits die erste Version 1.0 von kweb (im Jahr 2013) enthielt die Möglichkeit, Web-Videos mit dem hardwarebeschleunigten Videoplayer omxplayer abzuspielen. Das war damals überhaupt die einzige Möglichkeit dazu. Man navigiert im Browser auf die Webseite mit dem Video (z. B. auf Youtube) und klickt auf den „Play“ Button in der Toolbar. Dann wird mit Hilfe von youtube-dl die URL des Videos aus der Webseite extrahiert und an den Player weitergeleitet. So funktioniert es auch heute noch, allerdings mit erheblichen Verbesserungen. In Version 1.5 kam omxplayerGUI hinzu, das diese Funktion ebenfalls in seinem Frontend enthält.

Mit der Zeit wurde youtube-dl aber immer langsamer, weil es immer mehr Websites unterstützte. Dazu mussten beim Programmstart immer mehr Module geladen werden (mittlerweile sind das etliche hundert) und das machte sich auf einem kleinen Rechner wie dem Raspberry Pi (vor allem bei den älteren Modellen) doch arg bemerkbar. So habe ich nach immer besseren Möglichkeiten gesucht, diese Funktion zu beschleunigen. Die letzte Ausbaustufe davon war der Youtube-DL-Server. Die Idee dahinter ist ganz einfach: Statt hunderte von Modulen bei jedem Programmaufruf von youtube-dl zu laden, werden sie nun nur einmal geladen, nämlich beim Start des Servers (und z. T. bei der ersten Nutzung des Servers).

In kweb funktioniert die „Play“-Funktion auch heute noch ohne den Youtube-DL-Server, aber wenn er läuft, wird er benutzt. Wenn man das omxplayerGUI-Frontend startet, wird der Server automatisch im Hintergrund gestartet (und bei Beendigung des Programms auch wieder gestoppt).

2) Youtube-DL-Server im Browser

Da es sich beim Youtube-DL-Server um einen kleinen speziellen Webserver handelt, kann man seine Startseite in jedem Browser aufrufen (also nicht nur in kweb). Man gibt dazu folgende URL ein: <http://localhost:9192/>. Die Abbildung zeigt ihn so, wie er bei mir konfiguriert ist (näheres dazu weiter unten).

Die obere Zeile (Local Media) wollen wir zunächst vernachlässigen (sie wird im vierten Abschnitt beschrieben). Um Web-Videos abzuspielen, kopiert man die URL der Webseite, die das Video enthält in das obere Eingabefeld und klickt auf „Extract & Play“. Dann wird der Videoplayer (omxplayerGUI oder VLC) gestartet und das Video abgespielt.

Im Browserfenster erscheint dann eine Anzeige wie in der nebenstehenden Abbildung, die auch ein omxplayerGUI-Fenster in niedriger Auflösung (512 Bildzeilen) anzeigt. Um im Browser auf die Startseite zurückzukehren, klickt man auf den „Go Back“ Button.

Man kann auch einen direkten Link auf ein Video oder den Dateipfad eines lokalen Videos in das zweite Eingabefeld („Direct Video“) eingeben und dieses durch Klicken auf den „Play“ Button abspielen.

Mit „Switch Player“ kann man zwischen omxplayerGUI und VLC zum Abspielen der Videos umschalten. Insbesondere auf dem Raspberry Pi 4B kann die Verwendung von VLC interessant sein, weil dort etliche der Hardware-Codecs fehlen, auf die omxplayer(GUI) angewiesen ist und zusätzlich auch mit H265 kodierte Videos mit Hardwarebeschleunigung in VLC abgespielt werden können. Der Server kann auch so gestartet werden, dass VLC als Standardplayer beim Programmstart eingestellt ist. Diese und

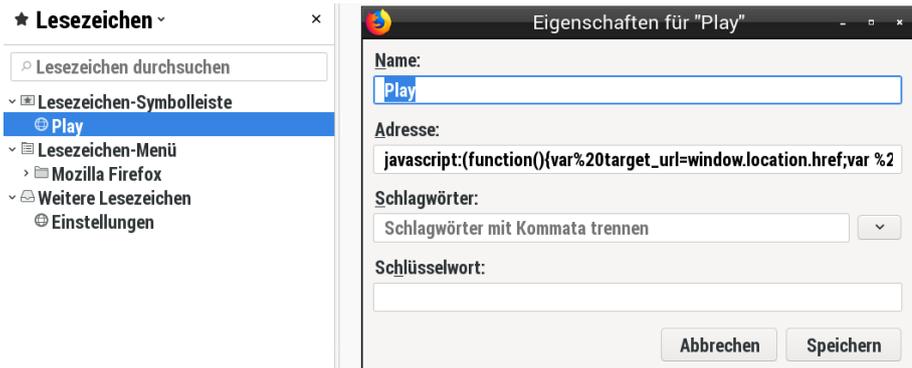


weitere Optionen (z. B. ein bestimmter „Skin“ = CSS-Datei) können bei den Web-Gui-Einstellungen in rpiotrtool ausgewählt werden.

3) „Play“-Funktion in jeden Browser integrieren

Die oben beschriebene „Play“-Funktion von kweb (Extrahiere Web Video und spiele es in VLC oder omxplayerGUI ab) kann man nun ganz leicht in jeden beliebigen Browser einbauen, also etwa in den Chromium-Browser, Firefox oder Vivaldi. Zur Nutzung dieser Funktion muss der Youtube-DL-Server laufen. Ich möchte das im Folgenden am Beispiel von Firefox illustrieren.

Man muss zunächst in Firefox die Lesezeichen-Sidebar öffnen. Man klickt dann dort auf „Lesezeichen-Symbolleiste“ (muss

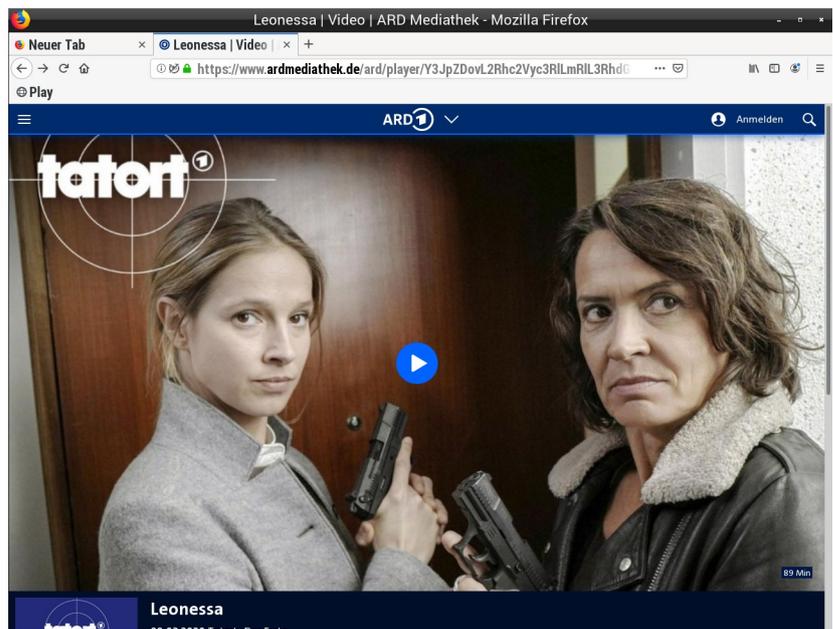


blau unterlegt sein). Mit einem Rechtsklick kann man dann „Neues Lesezeichen ...“ auswählen. Es öffnet sich dann das Formular „Neues Lesezeichen“. Als Name gibt man z. B. „Play“ ein und im Feld „Adresse“ die folgende Zeile:

```
javascript:(function(){var%20target_url=window.location.href;var %20ytsvr='http://localhost:9192/play?url=';var %20final=ytsvr.concat(encodeURIComponent(target_url));var %20myWindow=window.open(final, '_top')}}());
```

Anschließend klickt man dann auf „Speichern“. Die „Play“-Funktion wird nun oben im Firefox angezeigt.

Man kann nun nun auf eine Videowebseite gehen, z. B. in die ARD -Mediathek. Dort klickt man nun auf den „Play“-Link oben links im Browser und das Video wird nun im gewählten Video-Player angezeigt.



Im Browser wird nun „Playing:“ gefolgt vom Titel des Videos angezeigt. Durch Anklicken von „Go Back“ kehrt man auf die Video-Webseite zurück.



Die nebenstehende Abbildung zeigt dies am Beispiel von omxplayerGUI, wobei

ich der Darstellung wegen eine kleine Bildgröße (576 Bildzeilen) eingestellt habe. Standardmäßig wird omxplayerGUI mit maximiertem Videofenster angezeigt. Das ist aber einstellbar. Man kann z. B. auch im Fullscreen-Modus starten.

Es gibt noch eine weitere Möglichkeit, die sogenannte „**Video-Casting**“ ermöglicht. Ich kann eine solche Link-Adresse nämlich auch in einem Browser in jedem anderen Computer im lokalen Netzwerk (incl. Handies oder Tablets) eingeben, wenn man in dieser Adresse „localhost“ durch die IP oder den Hostnamen des Raspberry Pi Computers im lokalen Netzwerk ersetzt. Dann kann man von diesem Computer aus das Abspielen des Videos auf dem Raspberry Pi (der z. B. an den Fernseher im Wohnzimmer angeschlossen ist) starten.

4) Web-Interface für lokale Medien-Sammlung

Es gibt sicherlich viele OTR-Mitglieder, die ihre heruntergeladenen TV-Aufzeichnungen archivieren. Im Laufe der Jahre kann daraus eine sehr große Video-Sammlung entstehen, die vielleicht sogar mehrere Festplatten benötigt. Diese kann man z. B. an den Raspberry Pi direkt anschließen (evtl. über einen USB-Hub), sie können sich aber auch auf anderen Computern im lokalen Netzwerk oder in einem NAS-Server befinden, die man dann als lokale Verzeichnisse auf seinem Raspberry Pi „mounten“ kann. Und wenn man halbwegs ordentlich ist, hat man seine Sammlung auch vernünftig in Haupt- und Unterverzeichnisse gegliedert.

Den Youtube-DL-Server kann man nun so konfigurieren, dass man darüber auf seine komplette Mediensammlung (incl. Audiodateien) auf einfachste Weise zugreifen kann. Die genaue Beschreibung würde den Rahmen dieses Handbuches sprengen. Man findet alles nötige dazu in Kapitel 7 des omxplayerGUI-Handbuchs. Hier soll nur die Funktionsweise behandelt werden.

Hat man den Server entsprechend konfiguriert, sieht man in der obersten Zeile der Startseite den Eintrag „Local Media:“, gefolgt von einer Reihe von Links. Diese führen in die Hauptverzeichnisse der lokalen Mediensammlung.

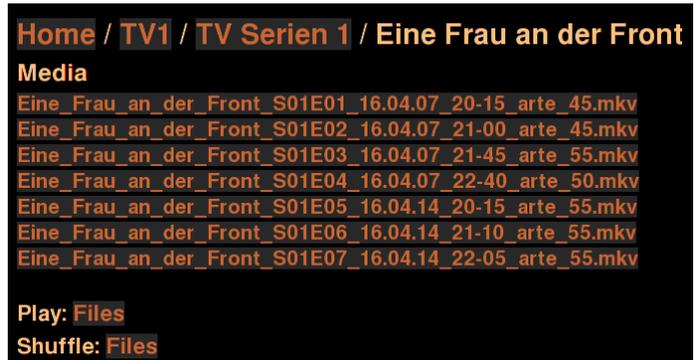
Die Abbildung weiter oben zeigt dies am Beispiel meiner eigenen Mediensammlung. Von dort aus

kann man dann in einfachster

Weise weiter navigieren. Ein

Beispiel sagt oft mehr als tausend

Worte und ist in der Abbildung



angezeigt. Wir befinden uns hier im Hauptverzeichnis „TV1“, Unterverzeichnis

„TV Serien 1“, Unterverzeichnis „Eine Frau an der Front“. Der Inhalt dieses

Unterverzeichnisses wird angezeigt (unter dem Titel „Media“). Eventuell

vorhandene weitere Unterverzeichnisse werden als „Directories“ angezeigt

(das ist hier aber nicht der Fall) und können durch Anklicken geöffnet werden.

Nun kann ich jede angezeigte Mediendatei durch einfaches Anklicken

abspielen, je nach Einstellung mit VLC oder omxplayerGUI.

Klicke ich auf „Play:“, „Files“, wird eine Playlist generiert und an den

Medienplayer geschickt. Dann werden alle Medien in der angezeigten

Reihenfolge nacheinander abgespielt. Klicke ich stattdessen auf „Shuffle:“,

„Files“ werden alle Medien in zufälliger Reihenfolge abgespielt. Bei einer TV-

Serie macht das wenig Sinn, aber z. B. bei einer großen Audiosammlung.

Befinden sich im aktuellen Verzeichnis nur Unterverzeichnisse, wird

stattdessen „Play: Recursive“ bzw. „Shuffle: Recursive“ angezeigt. Klickt man

die entsprechenden Links an, werden alle Medien aus den jeweiligen

Unterverzeichnissen (und auch deren Unterverzeichnissen usw.) abgespielt.

Befinden sich im aktuellen Verzeichnis sowohl Medien als auch

Unterverzeichnisse, erscheinen neben „Play:“ und „Shuffle“ jeweils zwei

Links: „Files“ und „Recursive“. Klickt man den ersten an, werden alle Medien im aktuellen Verzeichnis abgespielt. Klickt man den zweiten an, werden sowohl alle Medien im aktuellen Verzeichnis als auch die in allen Unterverzeichnissen (und deren Unterverzeichnissen etc.) abgespielt.

In der Kopfzeile erscheinen die einzelnen Pfadelemente als einzelne Links, was ein sehr einfaches Navigieren ermöglicht. Wenn man auf „Home“ klickt, kehrt man auf die Startseite zurück.

Das Ganze ist sehr schnell und intuitiv, erfordert keine Datenbank und somit auch keine Datenbank-Updates. Alle Inhalte werden „on the fly“ generiert und sind immer aktuell.

Es sollte noch erwähnt werden, dass omxplayerGUI ja auch einen speziellen kleinen Audioplayer enthält. Aber natürlich ist auch VLC als Audioplayer geeignet.

Zusatzfunktionen

Unterhalb des Trennstriches findet man noch zwei weitere Funktionen.

Plugins

Klickt man darauf, öffnet sich im rechten Teil des Fensters eine nahezu leere Seite. Im Prinzip ist das eine kweb-Anwendung ohne Inhalt. Diese kann man selbst bearbeiten und so dem Web-GUI eigene Funktionen hinzufügen. Dazu stellt die kweb suite einen speziellen Editor zur Verfügung.

Wir navigieren in einem Terminal in den Ordner `.../otr/web` und geben folgendes ein:

```
kweb_edit.py plugins.txt
```

Dann öffnet sich der „Kweb Editor for User Pages“. Was man damit tun kann und wie man eigene kweb-Anwendungen erzeugt, würde den Inhalt dieses Handbuchs sprengen. Man findet alles nähere dazu in den Kapiteln 12 – 14 des kweb-Handbuchs.

Man kann auch rein interessehalber mal folgendes Kommando im Terminal eingeben:

```
kweb_edit.py index.txt
```

Dann öffnet sich der Editor mit der Textdatei, aus der das Web-GUI selbst generiert wird. Man wird daran sehen, dass das Ganze gar nicht so kompliziert ist.

Hilfe

Klickt man auf den Hilfe-Button, öffnet sich im rechten Teil des Fensters die Hilfe-Seite. Sie enthält im Wesentlichen Links zu den Handbüchern. Neben meinen eigenen Handbüchern (rpiotrtool, kweb,

Handbücher zu den Programmen

[RPI-OTR-Tool](#)

[OmxplayerGUI](#)

[Minimal Kiosk Browser \(kweb\)](#)

[Youtube-DL](#)

Support

Wer Hilfe zu rpiotrtool benötigt, Fehler melden will oder Verbesserungen vorschlagen möchte, kann den Autor in zwei Foren erreichen. Dort findet man auch immer aktuelle News und die neueste Version.

[im OTR-Forum](#)

[im Raspberry Pi Forum](#)

omxplayerGUI) habe ich auch das Handbuch zu Youtube-DL beigefügt.

Klickt man einen der Links an, öffnet sich das jeweilige Handbuch in einem geeigneten PDF-Reader.

Weiter unten findet man noch zwei Supportlinks zu den entsprechenden Forenbeiträgen im Raspberry-Pi-Forum und im OTR-Forum.

E) Bedienungsanleitung: Kommandozeilenprogramm

rpiotrtool kann auch wie ein ganz gewöhnliches Kommandozeilenprogramm benutzt werden. In dieser Art wird es z. B. auch vom Web-GUI genutzt. Dabei gibt es eine Reihe von zusätzlichen Möglichkeiten.

rpiotrtool enthält eine eingebaute Anleitung, die mit „rpiotrtool help“ (oder „-h“ bzw. --help) aufgerufen werden kann. Ich gebe die Ausgabe hier wieder und füge (in kursiver Schrift) weitere Erläuterungen hinzu.

\$ rpiotrtool help

rpiotrtool Vers. 1.2 - Copyright 2020 by Günter Kreidl

Aufruf: rpiotrtool [Optionen] [Kommando] [Wert1] [Wert2]

Verfügbare Kommandos:

menu (oder -i)

Startet rpiotrtool im interaktiven Modus (einfaches Textmenü).

Der interaktive Modus ermöglicht den Aufruf der verschiedenen Programmbefehle mit einfachen Tastaturkommandos (ein Buchstabe bzw. eine Ziffer + Return). Befehle und zugehörige Buchstaben werden in einem einfachen Textmenü angezeigt. Einige wenige erweiterte Befehle sind im interaktiven Modus nicht möglich.

decode

Dekodiert alle neuen OTR-Keys

Das Programm durchsucht das Downloadverzeichnis nach neu heruntergeladenen otrkeys und dekodiert sie. Erfolgreich dekodierte Dateien werden entweder nach Abfrage oder automatisch gelöscht. Die dekodierten Videodateien befinden sich anschließend im Verzeichnis „decoded“. Auch neu heruntergeladene Cutlists werden gesucht und in das gleiche Verzeichnis kopiert (zum automatischen Schneiden). Und ebenso werden alle AVI-Dateien, die man direkt von OTR heruntergeladen hat, aus dem Downloadverzeichnis in das Verzeichnis „decoded“ verschoben, wo sie dann zur weiteren Bearbeitung zur Verfügung stehen.

Abhängig von einer Einstellungsoption können nun die dekodierten Dateien auch automatisch gleich in das MKV-Format konvertiert werden (Kopie im Ordner „converted“). Dies ist insbesondere auf dem Raspberry Pi 4B sinnvoll, da diese Kopien von der Schnittvorschau mit VLC verwendet werden können anstelle des AVI-Formates, das von VLC nur leicht ruckelig abgespielt wird. Verwendet man dort aber mplayer für die Vorschau, ist das nicht notwendig.

preview [Datei] [Cutlist]

Vorschau der Schnittstellen aller Dateien, für die eine Cutlist existiert.

Man kann auch eine bestimmte Datei und evtl. Cutlist für die Vorschau angeben.

Das Programm zeigt nacheinander eine Vorschau aller ungeschnittenen Videos an, für die eine Cutlist existiert. Die Anzeige erfolgt je nach Einstellung mit omxplayer in einem Overlay in der oberen linken Ecke des Bildschirms, mit VLC im Vollbildmodus oder in einem mplayer-Fenster. Die Wiedergabe beginnt einige Sekunden vor der ersten

Schnittstelle. Der Schnitt selbst wird durch einen Untertitel „Start Cut0“ angezeigt (für zwei Sekunden). Einige Sekunden später springt die Wiedergabe zu einem Zeitpunkt einige Sekunden vor dem Ende des ersten Schnittbereichs, der mit dem Untertitel „Ende Cut0“ angezeigt wird. Enthält die Cutlist weitere Schnittbereiche, wird das so fortgeführt, wobei die Schnittnummer hochgezählt wird („Start Cut1“, „Ende Cut1“ usw.). Einige Sekunden nach der letzten Schnittstelle wird die Wiedergabe beendet. Dann erscheint im Terminalfenster folgende Abfrage:

**Sind die Schnitte OK (j,n)?
Soll die Vorschau erneut angezeigt werden (w)?
Vorschau wiederholen mit neuem Zeitfenster
(8,10,12,15,20,25,30)?**

Eingabe:

Antwortet man mit „n“, wird die Cutlist ins Trash-Verzeichnis verschoben und steht somit zum Schneiden nicht mehr zur Verfügung. Mit „j“ sollte man antworten, wenn man die Cutlist akzeptiert. Man kann die Vorschau auch wiederholen („w“), bei Bedarf auch mit einem größeren oder kleineren Zeitfenster (siehe Ziffern oben). Alle Eingaben müssen mit „Return“ abgeschlossen werden.

previewse1

Eine Datei zur Schnittvorschau auswählen.

Es wird zunächst eine Liste aller Videos angezeigt, für die eine Cutlist existiert. Daraus kann man dann eine auswählen, deren Schnittvorschau man sich ansehen möchte. Das geschieht genau so, wie oben beschrieben.

cut [Datei] [Cutlist]

Schneidet alle Dateien für die eine Cutlist vorhanden ist.

Man kann auch eine bestimmte Datei und evtl. Cutlist zum Schneiden angeben.

Je nach Einstellung führt das Programm verschiedene Schritte aus. Ist die Vorschau aktiviert, erfolgt zunächst eine Schnittvorschau aller Videos, für die eine Cutlist existiert, exakt wie oben unter „preview“ beschrieben. Anschließend werden alle entsprechenden Dateien geschnitten. Das kann je nach Dateigröße und Anzahl der Schnitte zwischen 2 und 5 Minuten pro Datei dauern; auf dem Raspberry Pi 4B geht es bis zu 7 mal schneller. Nach dem Schneiden werden die ungeschnittene Datei und die Cutlist in das Trash-Verzeichnis verschoben. Die geschnittenen Videos befinden sich im Verzeichnis „cut“.

Nach dem Schneiden (aller Dateien) können die Schnitte noch begutachtet werden, wenn man die Einstellung „Schnittanzeige“ aktiviert hat. Die Videos werden ähnlich wie bei der Schnittvorschau je nach Einstellung mit omxplayer in der oberen linken Ecke des Bildschirms angezeigt, mit VLC im Vollbildmodus oder in einem mplayer-Fenster (empfohlen für Raspberry Pi 4B Modelle). Wurden nur Vor- und Nachlauf entfernt, werden die ersten paar Sekunden des geschnittenen Videos angezeigt und anschließend die letzten paar Sekunden, so dass man Anfang und Ende überprüfen kann. Enthält das Video weitere Schnittpunkte (Werbung entfernt), werden diese ebenfalls angezeigt. Die Wiedergabe springt dann vor bis zu einem Zeitpunkt einige Sekunden vor dem ersten Schnitt. Der Schnittpunkt wird dann durch einen Untertitel „Interner Schnitt 1“ angezeigt. Das wird für alle Schnittpunkte wiederholt, wobei die Schnittnummer hochgezählt wird. Damit kann man die Qualität der Schnitte überprüfen. Zuletzt werden dann die letzten paar Sekunden des Videos angezeigt. Nach jeder Videoanzeige erscheint eine Abfrage im Terminal:

Sind die Schnitte OK (j,n)?

**Sollen die Schnitte erneut angezeigt werden (w)?
Anzeige wiederholen mit neuem Zeitfenster
(8,10,12,15,20,25,30)?**

Eingabe:

Antwortet man mit „n“, wird die geschnittene Datei gelöscht und die ungeschnittene wieder aus dem Trash-Verzeichnis restauriert (zum erneuten Schneiden). Mit „j“ sollte man antworten, wenn man den Schnitt akzeptiert. Man kann die Schnitthanzeige auch wiederholen („w“), bei Bedarf auch mit einem größeren oder kleineren Zeitfenster (siehe Ziffern oben). Alle Eingaben müssen mit „Return“ abgeschlossen werden. Nach dem erfolgreichen Schneiden werden evtl. vorhandene Kopien im MKV-Format (im Ordner „converted“) automatisch gelöscht.

Man kann auch den Dateinamen (nur den Namen, ohne Pfad!) einer bestimmten Datei zum Schneiden eingeben und optional ebenso den Namen einer Cutlist-Datei. Falls OTR eine zusätzliche separate AC3-Tonspur anbietet (nur für HD-Aufnahmen) und man den entsprechenden otrkey heruntergeladen hat, wird diese Datei ebenfalls geschnitten und in die geschnittene Datei als zweite Tonspur einkopiert. Dies funktioniert auch mit HQ-Videodateien.

cutsel

Es wird zunächst eine Liste aller Videos angezeigt, für die eine Cutlist existiert. Daraus kann man dann eine zum Schneiden auswählen. Das geschieht genau so, wie oben beschrieben.

search

Generiert eine HTML-Seite mit Cutlist-Suchlinks für alle Dateien ohne Cutlist.

Diese Seite wird gespeichert als ...otr/web/search.html.

Die integrierte Cutlist-Suche bietet zwei Vorteile. Man kann zu jedem beliebigen Zeitpunkt nach Cutlists für alle noch nicht geschnittenen

Suche Cutlists mit deinem FRED

**Das_Kindermaedchen_19.01.03_21-45_zdfneo_90
Frau_im_Mond_19.01.14_00-00_arte_165**

Dateien suchen, ohne mühsam etwas in das Suchfeld von cutlist.at eingeben zu müssen. Und außerdem kann man seinen registrierten cutlist.at-Link (FRED genannt) für die Suche verwenden, wenn man ihn zuvor bei den Einstellungen (cutlist) eingetragen hat. Damit werden dann registrierte Downloads generiert, die man später auf cutlist.at bewerten kann. Bei der Suche werden verkürzte Dateinamen verwendet. Es werden dann cutlists für alle verschiedenen Formate angezeigt. Prinzipiell kann rpiotrtool jede Cutlist verwenden. Die eventuell falschen Frameangaben werden automatisch korrigiert.

Die Datei search.html kann mit jedem beliebigen Browserprogramm geöffnet werden.

convert [Datei]

Konvertiert eine oder alle Dateien ins MKV-Format.

Hier werden alle ungeschnittenen Dateien in das MKV-Format konvertiert (sofern noch nicht geschehen). Die konvertierten Dateien können problemlos mit VLC abgespielt werden und werden dann ebenfalls für die Cutlist-Vorschau mit VLC verwendet, das die

Original-AVIs nur leicht ruckelig abspielt. Nach dem Schneiden einer AVI-Datei wird die MKV-Kopie automatisch gelöscht,

Die Konvertierung kann auch automatisch nach dem Dekodieren erfolgen, wenn man die entsprechende Einstellungsoption aktiviert hat,

init [Sektion]

Ohne weiteres Argument werden alle möglichen Einstellungen durchlaufen.

Verfügbare Sektionen:

'paths': Verzeichniseinstellungen

Mit „init paths“ werden die vom Programm benötigten Verzeichnispfade gesetzt. Dies sind das Wurzelverzeichnis für alle vom Programm benötigten Unterverzeichnisse (default: /home/pi/otr) und das Downloadverzeichnis, das vom verwendeten Webbrowser zum Herunterladen von otrkeys und cutlists verwendet wird (default: /home/pi/Downloads). Die Ordner müssen existieren. Falls die vom Programm benötigten Unterordner nicht vorhanden sind, werden sie neu angelegt.

'otr': OTR-Einstellungen für Dekoder

Zum Dekodieren benötigt das Programm die bei OTR hinterlegte Emailadresse und das OTR-Passwort des Anwenders. Werden sie nicht eingegeben oder gelöscht (leere Eingabe) werden sie vor dem Dekodieren interaktiv abgefragt. Das Speichern des Passwortes stellt ein Sicherheitsrisiko dar!

'cutlist': Einstellungen zum Suchen und Erzeugen von Cutlists

Mit „init cutlist“ werden Einstellungen vorgenommen, die für die Cutlist-Suche und zum Erstellen eigener Cutlists benötigt werden.

Zunächst einmal benötigt das Programm die Internetadresse des Cutlist-Servers. Hier sind nur zwei Einstellungen möglich: <http://cutlist.at> (Voreinstellung) oder der von cutlist.at vergebene registrierte Link (auch FRED genannt). Man benötigt diesen FRED, wenn man Cutlists bewerten oder eigene Cutlists hochladen möchte.

Für eigene Cutlists wird ein Autorennamen benötigt. Man sollte zuvor auf cutlist.at sicherstellen, dass er nicht von einem anderen verwendet wird (Autorensuche).

Dritter Einstellungspunkt in diesem Bereich ist die Qualitätseinstellung für eigene Cutlists. Voreinstellung ist „0“. Dann wird bei jeder Cutlisterstellung die Qualitätsstufe abgefragt. Wenn man alle seine Cutlists mit einer bestimmten Qualitätsstufe generiert (1 -5), kann man diesen Wert hier fest eingeben. Dieser Wert wird dann automatisch übernommen.

'options': diverse Programm-Einstellungen

Mit „init options“ können sieben weitere Einstellungen vorgenommen werden:

1) Sollen die dekodierten otrkeys automatisch oder erst nach Abfrage gelöscht werden?

2) Sollen die Dateinamen der geschnittenen Dateien gekürzt werden (hinter der Laufzeitdauer)?

3) Soll die Vorschau der Schnittstellen vor dem Schneiden aktiviert sein (Preview beim Schneiden)?

4) Soll die Anzeige der Schnittstellen nach dem Schneiden aktiviert werden?

5) Einstellung des Zeitfensters für die Schnittvorschau und die nachträgliche Anzeige. Voreinstellung ist „12“ (Sekunden). Man kann einen kleineren Wert wählen, aber da omxplayer immer nur zu keyframes springen kann und die in den OTR-Dateien teilweise recht weit auseinanderliegen können, wird dann evtl. ein Schnitt nicht korrekt angezeigt. Ein größerer Wert ist ebenfalls möglich, wenn man z. B. sehr genau auf doppelte Szenen achten will.

6) Welcher Media-Player soll für den Preview und die Schnittkontrolle nach dem Schneiden verwendet werden: omxplayer, VLC oder mplayer?

Anmerkung: Auf dem Raspberry Pi 4B muss VLC oder (bevorzugt) mplayer ausgewählt werden. Bei den älteren Modellen sollte man grundsätzlich omxplayer verwenden. omxplayer kann dort aber nur mit dem Legacy Grafiktreiber verwendet werden.

7) Sollen nach dem Dekodieren alle Videodateien automatisch in das MKV-Format konvertiert werden?

Dies ist insbesondere sinnvoll auf dem Raspberry Pi 4B, wenn man VLC für die Schnittvorschau verwendet, das die AVIs nur ruckelig anzeigt. Es werden Kopien im Ordner „converted“ erzeugt, die nach dem Schneiden automatisch gelöscht werden.

'threads': Single- oder Multithreading beim Schneiden

Mit „init threads“ stellt man ein, ob alle für den Schnitt benötigten Prozesse nacheinander (singlethreaded) oder teils parallel (multithreaded) ausgeführt werden. Multithreaded ist natürlich schneller, benötigt aber viel mehr Arbeitsspeicher und kann dazu führen, dass das System Arbeitsspeicher auslagern muss (Swappen). Dabei geht nicht nur Arbeitsgeschwindigkeit verloren, sondern das System kann für eine Weile regelrecht einfrieren. In diesem Fall sollte man geduldig warten und keinesfalls den Stecker ziehen! Singlethreaded ist die empfohlene Einstellung für alle Modelle mit 1 GB Arbeitsspeicher. Auf einem Raspberry Pi 4B mit 2 oder 4 GB kann Multithreading gewählt und damit das Schneiden noch einmal deutlich beschleunigt werden.

clean Löscht alle Dateien im Trash-Verzeichnis

Nach dem Schneiden werden die ungeschnittenen Dateien und die Cutlists nicht gelöscht, sondern in den Ordner „trash“ verschoben. Damit ist sichergestellt, dass man das Schneiden bei Bedarf wiederholen kann, etwa mit einer anderen Cutlist. Dazu muss die entsprechende Videodatei aus dem „Trash“ Verzeichnis wieder in den Ordner „decoded“ verschoben werden.

Auf die Dauer sammeln sich so im Ordner „trash“ natürlich viele Dateien an, die man mit dem Kommando „clean“ komplett löschen kann (ähnlich dem „Papierkorb leeren“ im Dateimanager).

Sollten sich noch MKV-Dateien im Ordner „converted“ befinden, folgt noch eine Abfrage, ob auch diese Dateien gelöscht werden sollen.

get
Holt alle neuen Cutlists aus dem Download-Verzeichnis
Dies geschieht auch automatisch bei jedem 'cut' oder 'decode' Kommando

help
Zeigt diese Hilfe an. Funktioniert auch mit '-h' und '--help'

**Wird das Programm ohne jedes Argument gestartet, werden folgende Kommandos nacheinander aufgerufen: get, decode, (convert), (preview), cut, (Anzeigen).
In Klammern bedeutet: Falls aktiviert (in Optionen)**

Der einfache Aufruf von „rpiotrtool“ ohne jeden Parameter ist der normale Arbeitsmodus. Es werden alle neuen Cutlists eingesammelt, alle neuen otrkeys dekodiert, und dann alle Dateien geschnitten, für die eine Cutlist existiert. Wenn die entsprechenden Optionen aktiviert sind, werden dabei vor dem Schneiden zunächst die Schnittstellen angezeigt (preview) und ebenso nach dem Schneiden die Schnittpunkte nochmals zur Überprüfung.

Bei entsprechend aktivierter Option werden nach dem Dekodieren alle Dateien auch gleich in das MKV-Format konvertiert.

Optionen:

- '-l': Lösche erfolgreich dekodierte OTR-Keys ohne Rückfrage
- '-v': Vorschau der Schnittpunkte anzeigen
- '-a': Schnittpositionen in geschnittenen Videos anzeigen
- '-r': Dateinamen der geschnittenen Datei kürzen
- '-s': Single-Threaded schneiden
- '-m': Multi-Threaded schneiden

Im interaktiven (menu) Modus werden die Optionen nicht ausgewertet.

Sie können aber unter 'Einstellungen' (Optionen, Threads) festgelegt werden.

Mit diesen Kommandozeilenoptionen kann man bestimmte feste Einstellungen (z. B. mit „init options“ gesetzt) überstimmen, z. B. die Schnittvorschau aktivieren, wenn sie nicht fest eingestellt wurde.

F) Eigene Cutlists erstellen

1) Das Problem

Um es gleich vorweg zu sagen: Ein komfortables Schnittprogramm bzw. ein Schnitt-Editor (wie etwa VirtualDub oder der in OTR-Verwaltung++ integrierte Schnitteditor) ist derzeit auf dem Raspberry Pi nicht (oder nur sehr schwer) zu realisieren, zumindest auf älteren Modellen (vor dem Raspberry Pi 4B). Für eine flüssige Videowiedergabe ist die Unterstützung durch die GPU unerlässlich. Es gibt aber nur wenige Programme bzw. Bibliotheken, die die GPU nutzen können und keines dieser Werkzeuge verfügt über einen Einzelbild-Modus oder kann einzelne Frames exakt anspringen. An hardwarebeschleunigten Abspielprogrammen stehen nur das Medienzentrum Kodi, VLC und omxplayer zur Verfügung. Die Verwendung von VLC oder Kodi kommt für unseren Zweck nicht in Frage. Es bleibt also nur omxplayer.

omxplayer ist ein Kommandozeilenprogramm, das vom Benutzer durch Tastatureingaben kontrolliert werden kann. Es ist immer noch der effektivste Videoplayer, weil das von der GPU dekodierte Video unmittelbar von der GPU in einem Bildschirmoverlay angezeigt wird. Das ist so effektiv, dass man sogar auf den ältesten oder billigsten Raspberry Pi Computern (B, B+, Zero, ZeroW) Videos in HD-Qualität ohne irgendwelche Einschränkungen abspielen kann.

Programmierer können zudem omxplayer über DBUS-Kommandos fernsteuern. rpiotrtool nutzt diese Möglichkeit z. B. für die Schnittstellen-Vorschau und die Schnittkontrolle in geschichteten Videos.

Für den normalen Anwender ist aber der Aufruf und die Nutzung von omxplayer wenig komfortabel. Deshalb habe ich schon vor Jahren ein GUI dafür geschrieben, das die Bedienung mit der Maus erlaubt und die Fähigkeiten des Players um etliche Funktionen erweitert (Playlist, Wiedergabe von Webvideos u. v. m.). In gewissem Sinne ist dieses Programm aber ein „Fake“. Die Videoanzeige erfolgt immer noch in einem Overlay, was man leicht daran erkennt, dass das Video „durchscheint“, wenn man ein anderes Fenster darüber schiebt. Aber omxplayer kann so exakt kontrolliert werden, dass das Video immer innerhalb des GUI-Fensters von omxplayerGUI erscheint, auch wenn man das Fenster verschiebt, vergrößert

oder verkleinert. Wenn man das Fenster verbirgt, läuft auch das Video unsichtbar im Hintergrund weiter (man hört noch den Ton).

Auch wenn man omxplayerGUI komplett mit der Maus bedienen kann, unterstützt es den gesamten Satz an Tastatur-Kommandos von omxplayer. Das ist auch unerlässlich, wenn man z. B. auf Vollbilddarstellung umschaltet. Darüber hinaus verfügt omxplayerGUI über einen zweiten Satz an Tastaturkommandos in Kombination mit der (linken) Alt-Taste. Eine komplette Liste beider Sätze von Tastaturkommandos findet man im Anhang des omxplayerGUI-Handbuchs und auch im Anhang dieses Handbuchs.

Anmerkungen zu omxplayer auf dem Raspberry Pi 4B:

omxplayer kann ausschließlich Videos abspielen, deren Decodierung von der GPU unterstützt wird. Auf allen vorherigen Modellen standen die folgenden Codecs zur Verfügung: H264 H263 WVC1 MPG4 MPG2 VP8 VP6 VORB THRA MJPG FLAC PCM (wobei man die Lizenzen für MPG2 und WVC1 separat käuflich erwerben konnte). Auf dem Raspberry Pi 4 werden etliche dieser Codecs nicht mehr unterstützt. Es verbleiben nur noch: H264 VP8 VP6 VORB THRA MJPG FLAC PCM. Die Raspberry Pi Foundation argumentiert damit, dass die fehlenden Formate nun auch per Software dekodiert werden können, weil der Raspberry Pi 4 dafür schnell genug ist. Das ist zwar richtig, aber dann leider nicht mehr mit omxplayer(GUI). Das betrifft auch eines der OTR-Formate, nämlich das alte AVI-Format, weil hier die H263-Kodierung verwendet wird. Deshalb ist manuelles Schneiden diese Formates auf dem Raspberry Pi 4 derzeit nicht möglich.

Es gibt noch ein weiteres Problem: Die OSD-Funktion (On-Screen-Display) von omxplayer musste auf dem Raspberry Pi 4B komplett abgeschaltet werden, weil sie nicht mit den neuen Grafiktreibern kompatibel ist. Das betrifft nicht nur die Untertitel (die wir ja für die Schnittkontrolle brauchen), sondern auch solche nützlichen Dinge wie Positions- und Geschwindigkeitsanzeige, die beim manuellen Schneiden extrem hilfreich sind. Auf dem Raspberry Pi 4B ist das ein wenig wie das Fischen im Trüben. Allerdings ist dieses Problem lösbar, wenn die OSD-Funktionen umprogrammiert werden, aber es sieht derzeit nicht danach aus.

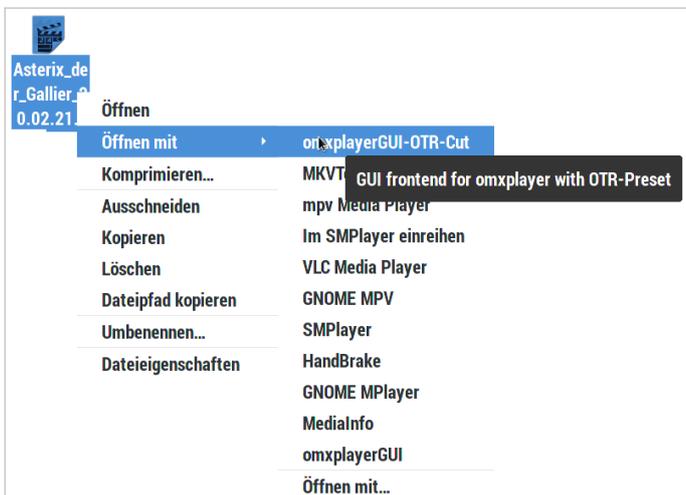
Da inzwischen die von OTR gelieferten Videoformate auch mit mplayer (ohne Hardware-Unterstützung) abgespielt werden können, plane ich für die

nächste Version einen separaten, auf mplayer basierenden Schnittditor. Das ist aber noch Zukunftsmusik.

2) Die Lösung

Ich habe nun die aktuellste Version von omxplayerGUI genommen und um eine Reihe von Zusatzfunktionen erweitert. Bedient werden diese neuen Funktionen alle über zusätzliche Tastaturkommandos des zweiten Befehlssatzes (mit der linken Alt-Taste). Mithilfe dieser neuen Funktionen (und einigen Standard-Tastaturkommandos von omxplayer) ist es möglich, Schnittstellen exakt anzufahren und Schnittpunkte zu setzen. Die gesetzten Schnittpunkte können in einer „cutpoints“-Datei gespeichert werden. rpiotrtool erkennt diese Cutpoints-Dateien automatisch und wandelt sie in reguläre Cutlists um.

Damit das Erstellen von Cutlists mit omxplayerGUI optimal funktioniert, muss man es mit einer bestimmten Konfiguration (Preset) starten. Um dies zu erleichtern wird bei der Installation eine separate Desktop-Datei angelegt, so dass praktisch ein neues Programm im Menü erscheint: omxplayerGUI-OTR-Cut.



Um eine Cutlist zu erzeugen, geht man in das Verzeichnis `.../otr/decoded`, macht einen Rechtsklick auf die gewünschte OTR-Datei (`.avi` oder `.mp4`), sucht unter „Öffnen mit“ nach „omxplayerGUI-OTR-Cut“ und öffnet damit das Video. Erscheint das noch nicht in der Auswahl (das kann beim ersten Mal passieren), sucht man es unter „Öffnen mit ...“ in der Kategorie „Unterhaltungsmedien“. Hat man Kopien im MKV-Format erzeugt, kann man auch diese zum Erzeugen der Cutlist verwenden. Man geht dazu in das Verzeichnis `„converted“` und öffnet das Video mit „omxplayerGUI-OTR-Cut“

Ich will nicht verhehlen, dass das Erzeugen von Cutlists mit omxplayerGUI ein wenig Übung erfordert. Aber ich verwende es nun selbst seit einiger Zeit und kann damit praktisch genau so schnell Schnittpunkte setzen und abspeichern wie zuvor mit VirtualDub (kombiniert mit otrverwaltung++).

Wie soll das nun funktionieren, wenn omxplayer über keinen Einzelschritt-Modus verfügt und man einzelne Frames auch nicht gezielt ansteuern kann? Die wichtigste Funktion von omxplayer, die wir hier nutzen, ist die Reduzierung der Wiedergabegeschwindigkeit. Mit der Taste „1“ kann man die Geschwindigkeit schrittweise reduzieren bis zu einem Sechzehntel der Originalgeschwindigkeit (angezeigt als 0.062, aber eigentlich 0.0625). Bei einer Bildfrequenz von 25 Hz wird jedes Einzelbild dann für 0,64 Sekunden angezeigt, bei 50 Hz (HD-Dateien) sind es 0.32 Sekunden. Drückt man dann nochmals die Taste „1“, bleibt das Bild stehen. Mit der Taste „2“ kann man die Geschwindigkeit in den gleichen Schritten wieder erhöhen bis zur normalen Abspielgeschwindigkeit (und seltsamerweise auch bis zu 1.25, obwohl das gar keine sichtbare Wirkung hat). Die Tasten „1“ und „2“ sind also so etwas wie Brems- und Gaspedale.

Man navigiert also möglichst nahe an den gewünschten Schnittpunkt (siehe nächsten Abschnitt), reduziert die Geschwindigkeit auf 0.062 (Bildschirmanzeige) und hält dann an der Schnittstelle an. Das erfordert aber eine ziemliche Reaktionsgeschwindigkeit, vor allem bei HD-Dateien. Deshalb gibt es noch eine weitere Funktion, die die Arbeit wesentlich erleichtert. Man hält die Wiedergabe kurz vor der Schnittstelle komplett an (Geschwindigkeit 0). Mit dem Tastatur-Befehl „Alt n“ springt die Wiedergabe um genau einen Frame vorwärts. omxplayer kann dies eigentlich gar nicht. Es wird dadurch realisiert, dass vom Programm die Geschwindigkeit kurzfristig wieder auf 0.062 erhöht wird und dann wieder auf 0 gesetzt wird, sobald das nächste Bild erreicht ist. Man sieht das auch an der Bildschirmanzeige (OSD, leider aber nicht auf dem Raspberry Pi 4).

Leider funktioniert das nur in einer Richtung. Das Rückwärtsspringen in den vorherigen Frame ist aus technischen Gründen leider nicht möglich (oder würde extrem lange dauern).

3) Navigieren im Video

Nachdem wir nun wissen, wie man das Video gezielt an einer bestimmten Stellen anhalten kann, um dort Schnittpunkte zu setzen (siehe folgender Abschnitt), kommt nun alles darauf an, diese möglichen Schnittpunkte (Anfang und Ende der eigentlich aufgezeichneten Sendung,

Werbesequenzen usw.) möglichst schnell und effektiv zu finden, d. h. innerhalb des Videos zeitlich zu navigieren.

Es wurde bereits erwähnt, dass omxplayer nur zu Keyframes springen kann. Es kommt aber noch eine Besonderheit hinzu, die man verstehen muss. Springt man zeitlich nach vorne, sucht omxplayer den nächsten Keyframe hinter (zeitlich nach) der angepeilten Sprungstelle. Springt man rückwärts, sucht omxplayer den nächsten Keyframe, der zeitlich vor dem Sprungziel liegt. Springt man also die gleiche Stelle einmal „von vorne“ und einmal „von hinten“ an, gelangt man zu unterschiedlichen Keyframes.

Die wichtigsten Navigationswerkzeuge sind die linke und rechte Pfeiltaste bzw. die entsprechenden Buttons im omxplayerGUI-Interface. Omxplayer springt dann um ca. 30 Sekunden nach vorne oder nach hinten. Hält man eine der beiden Pfeiltasten dauerhaft gedrückt, wird sehr schnell vor- oder zurückgespult, erkenntlich an der Zeitanzeige im OSD (On Screen Display). Drückt man z. B. die Taste „→“ rhythmisch (etwa ein mal pro Sekunde), bis jeweils das neue Bild erscheint, kann man sehr schnell den Programmanfang (grob) oder eine Werbesequenz finden.

Zum genauen Navigieren sind 30 Sekunden aber zu groß. Es gibt deshalb vier weitere Tastaturkommados zum feineren Navigieren:

„Alt →“: Springe um eine Sekunde vorwärts.

„Alt ←“: Springe um eine Sekunde rückwärts.

„Alt ↑“: Springe um einen Frame vorwärts.

„Alt ↓“: Springe um einen Frame rückwärts.

Wegen der bereits angesprochenen Besonderheit von omxplayer, nur Keyframes anspringen zu können, spult man aber nicht wirklich um eine Sekunde oder einen Frame vor oder zurück. Effektiv springt man mit diesen Kommandos zum nächsten oder vorherigen Keyframe und das ist die kleinste Sprungweite, die überhaupt möglich ist.

Es gibt noch eine weitere Navigationsmöglichkeit. Mit dem unteren Schieber im omxplayerGUI-Interface kann man gezielt eine bestimmte Position anfahren. Er hat eine Auflösung von 3 Sekunden. Die Zeit wird in Minuten mit Nachkommastellen (.05 = 3 Sekunden) angezeigt und das passt gut zu den Zeitangaben von OTR-Dateien. Klickt man rechts oder links neben den Schieber, springt er (und das Video) um 3 Sekunden vor oder zurück.

Die Position des Schiebers wird während des Abspielens nicht automatisch aktualisiert (und das ist Absicht!). Man kann aber mit dem Tastaturkommando „Alt g“ den Schieber an die aktuelle Position holen. Und mit „Alt p“ positioniert man das Video an der aktuellen Schieberposition. Man kann damit den Schieber auch als „Sprunganker“ (festes Sprungziel) verwenden.

Das alles liest sich komplizierter als es ist; mit ein wenig Übung lernt man damit sehr effizient zu arbeiten.

4) Schnittpunkte setzen, löschen und speichern

Für jeden Schnitt benötigt man zwei Schnittpunkte, die den Anfang und das Ende markieren. Die so markierten Bereiche werden dann in das geschnittene Video übernommen. Die Schnittpunkte markieren den ersten und den letzten Frame des Bereiches.

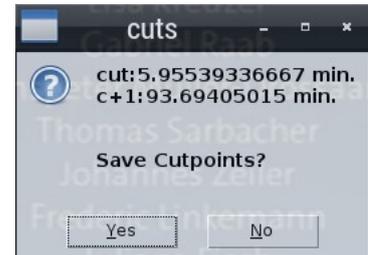
Hat man eine gewünschte Schnittstelle angefahren und das Bild angehalten, kann man mit „Alt .“ den Schnittpunkt setzen. Es gibt aber noch eine zweite Variante mit „Alt ,“. Dann wird nicht der aktuelle Frame zum Schneiden benutzt, sondern der vorangehende Frame. Das ist besonders am Ende eines Schnittbereiches praktisch. Man klickt sich solange vorwärts, bis das erste Bild erscheint, das nicht mehr in den Schnittbereich gehört, und markiert den Schnitt dann mit „Alt ,“.

Es gibt noch zwei weitere Tastatur-Kommandos, mit denen man Schnittstellen markieren kann. Diese decken Sonderfälle ab, die man nur in Ausnahmefällen braucht. Mit „Alt ;“ wird der Anfang (der erste Frame) des Videos als Schnittstelle markiert. Dies muss das erste Schnitt-Kommando sein. Mit „Alt :“ wird der letzte Frame eines Videos als Schnittstelle markiert. Dies muss das letzte Schnittkommando sein. Beide Stellen sind nur schwer oder gar nicht manuell anzufahren und deshalb wurden diese Kommandos eingebaut.

Generell ist festzuhalten, dass Schnittstellen nur nacheinander in der Zeit gesetzt werden können. Es ist also nicht möglich, Schnittpunkte vor bereits vorhandenen Schnittstellen zu setzen. Man arbeitet sich also immer vorwärts durch das Video.

Mit „Alt d“ kann der letzte Schnittpunkt gelöscht werden. Man kann also nur „rückwärts“ löschen, so wie man Schnittpunkte auch immer nur „vorwärts“ setzen kann.

Mit „Alt x“ kann man sich die gesetzten Schnittpunkte anzeigen lassen. Es erscheint ein kleines Dialogfenster. Normalerweise würde dieses Fenster vom Video (Overlay!) überdeckt. Deshalb wird das Video für die Dauer der Anzeige auf „hochgradig transparent“ gesetzt. Die Abbildung zeigt ein Beispiel für eine Datei, bei der Vorlauf und Nachlauf weggeschnitten wurden. Die erste Textzeile („cut: ..“) zeigt einen Schnittpunkt, der mit „Alt .“ gesetzt wurde. Die zweite Zeile („c+1: ..“) zeigt einen Schnittpunkt, der mit „Alt ,“ gesetzt wurde (und beim Erstellen der Cutlist durch den vorangehenden Frame ersetzt wird). Die Zeitangabe erfolgt in Minuten mit Nachkommastellen. Das passt gut zum Anzeigeformat des Positionsschiebers. Intern werden diese Werte als Mikrosekunden gespeichert.



Die Option „Save Cutpoints?“ mit den Buttons „Yes“ und „No“ erscheint nur, wenn eine gerade Anzahl von Schnittpunkten gesetzt wurde. Klickt man auf „Yes“, wird eine Cutpoints-Datei gespeichert, die dann später von rpiotrtool zum Erzeugen der eigentlichen Cutlist-Datei verwendet wird.

Es gibt noch eine Reihe von zusätzlichen Tastatur-Kommandos, die aber weniger wichtig sind. Mit diesen Kommandos kann man zu den bereits gesetzten Schnittpunkten springen, aber immer in den Grenzen, die von omxplayer gesetzt sind (und im vorherigen Kapitel beschrieben wurden):

„Alt e“: Springe zum ersten Schnittpunkt.

„Alt l“: Springe zum letzten Schnittpunkt.

„Alt o“: Springe zum nächsten Schnittpunkt.

„Alt b“: Springe zum vorherigen Schnittpunkt.

Die beiden letzten Kommandos arbeiten „Im Kreis“. Hat man den ersten Schnittpunkt angesprungen, geht es ja nicht weiter zurück. Dann springt man mit „Alt b“ zum letzten Schnittpunkt. Und mit „Alt o“ springt man wieder zum ersten Schnittpunkt, wenn man zuvor den letzten erreicht hatte. Wegen der Ungenauigkeit von omxplayer beim Springen zu definierten Positionen und

dem unterschiedlichen Verhalten je nach Sprungrichtung (siehe letztes Kapitel) sind diese Funktionen nur von begrenztem Wert.

Wir kennen nun die wichtigsten Werkzeuge zum Setzen von Schnittpunkten. In den nächsten beiden Kapiteln werde ich zeigen, wie man sie in der Praxis einsetzt.

5) Vorlauf und Nachlauf wegschneiden

Die ungeschnittenen Dateien befinden sich im Verzeichnis „decoded“ (im otr-Verzeichnis). Wir navigieren mit dem Dateimanager dorthin und öffnen die Videodatei, die wir schneiden möchten, mit omxplayerGUI-OTR-Cut. Es öffnet sich das Videofenster in einer Auflösung von 512 Bildzeilen (im Video) und die Filmdatei wird abgespielt. Wir können das Fenster nun mit „Alt m“ maximieren.

Nun spulen wir durch wiederholtes Drücken der rechten Pfeiltaste oder mit dem Schieber zum Beginn der eigentlichen Sendung vor. Sind wir schon in der Sendung gelandet, springen wir mit der linken Pfeiltaste wieder zurück und benutzen die Tastatur-Kommandos „Alt → „ und „Alt ← „, um zum letzten Keyframe vor dem eigentlichen Sendungsbeginn zu gelangen. Es macht Sinn, sich den Übergang zum Sendungsbeginn einmal in Echtzeit anzuschauen, weil wir dann besser abschätzen können, ab welcher Stelle wir die Abspielgeschwindigkeit reduzieren müssen. Wir springen dann mit „Alt ← „ wieder zurück und ab der passenden Stelle reduzieren wir die Geschwindigkeit dann schrittweise durch wiederholtes Drücken der Taste „1“. Kurz vor der Schnittstelle reduzieren wir die Geschwindigkeit auf 0. Nun können wir im Einzelschrittmodus arbeiten, also mit der Tastenkombination „Alt n“. Dabei müssen wir immer abwarten, bis die Geschwindigkeitsanzeige wieder „0“ anzeigt (auf dem Raspberry Pi 4 leider „blind“). Haben wir den ersten Frame der Sendung erreicht, drücken wir „Alt .“ um die Schnittstelle zu speichern. Wir können dann noch auf „Alt g“ klicken, um den Videoschieber an die aktuelle Position zu holen. Das ist nicht unbedingt notwendig, hilft aber beim Finden des Sendungsendes.

Nun benutzen wir den Positionsschieber um ungefähr ans Ende der eigentlichen Sendung zu gelangen. Wir kennen ja die Startposition und die Laufzeit können wir am Dateinamen erkennen, der oben im Fensterbalken angezeigt wird. Nun benutzen wir wieder die Pfeiltasten, auch in Kombination

mit der Alt-Taste, um möglichst nahe an das Sendungsende zu kommen und benutzen dann wieder die Geschwindigkeitsreduktion und den Einzelschrittmodus, um zu der gewünschten Schnittstelle zu gelangen. Dabei haben wir aber zwei unterschiedliche Optionen: Wir können zu dem letzten Frame navigieren, der noch im Video bleiben soll, und den wieder mit „Alt.“ abspeichern. Oder wir navigieren zum ersten Frame, der nicht mehr im Video enthalten sein soll, und speichern die Schnittstelle mit „Alt,“. Das zweite Verfahren ist oft einfacher.

Nun haben wir Anfang und Ende der Sendung markiert und können die „Cutpoints“ abspeichern. Wir drücken „Alt x“ und dann den „Yes“-Button. Wenn wir dann anschließend in rpiotrtool den „cut“ Befehl eingeben, erscheint zunächst (für jede Cutpoints-Datei) eine Meldung: „Erzeuge Cutlist für“ gefolgt vom Dateinamen. Hat man die Qualitätsstufe nicht fest eingestellt, erfolgt auch noch eine Abfrage der Schnittqualität (0-5).

Hat man die Schnittvorschau aktiviert, werden dann die Schnittstellen angezeigt. Beantwortet man die Abfrage mit „j“, wird die Datei geschnitten. Hat man die (nachträgliche) Anzeige der Schnitte aktiviert, werden nach dem Schneiden Anfang und Ende der Datei zur Kontrolle angezeigt.

6) Werbung und doppelte Szenen herausschneiden

Im Prinzip könnte man mit der im vorherigen Kapitel beschriebenen Methode auch Werbesequenzen herausschneiden, wenn nur die vermaledeiten doppelten Szenen nicht wären. Man kann sich da leider nie sicher sein. Für das exakte Schneiden doppelter Szenen ist das oben beschriebene Verfahren leider nicht geeignet. Ohne exaktes Hin- und Herspringen und Vergleichen zwischen den Schnittpunkten geht es leider nicht und das ist eben mit omxplayer nicht möglich.

Aber wo ein Wille ist, da ist auch ein Weg, und die Lösung erweist sich sogar als sehr komfortabel. Wir starten einfach zwei Instanzen von omxplayerGUI-OTR-Cut mit dem gleichen Video! Mit dem „otr“-Preset (s. u.) wird der Player in einer Fenstergröße gestartet, die es auf einem normalen HD-Bildschirm leicht ermöglicht, die beiden Videofenster schön nebeneinander zu positionieren. Wir verwenden das linke Videofenster zum Schneiden und das rechte zum Vergleichen.



Zu Beginn setzen wir das rechte Videofenster auf „Pausiert“ (GUI-Symbol oder Leertaste) und setzen im linken Fenster zunächst den Schnittpunkt am Sendungsbeginn, wie im letzten Kapitel beschrieben. Dann spulen wir vor (rechte Pfeiltaste) bis zur ersten Werbeeinblendung und pausieren die Wiedergabe. Es ist praktisch, sich mit „Alt g“ den Positionsschieber an die aktuelle Stelle zu holen, damit wir im rechten Fenster gleich an die passende Position vorspulen können. Wir wechseln nun zum rechten Fenster, spulen an die gleiche Position vor und von da aus zu den ersten Szenen nach der Werbesequenz. Nun können wir durch Vergleichen der letzten Szenen vor der Werbeeinblendung (im linken Fenster) und den ersten Szenen nach der Werbeeinblendung (im rechten Fenster) überprüfen, ob doppelte Szenen vorhanden sind. Ist das nicht der Fall, kann man im linken Fenster schneiden, wie im letzten Kapitel beschrieben. Aber wir gehen jetzt davon aus, dass es doppelte Szenen gibt. Dann gehen wir wie folgt vor:

Im linken Fenster frieren wir das Bild an einer geeigneten Schnittstelle ein (optimal ist immer ein Szenenwechsel). Dann wechseln wir zum rechten Fenster und frieren das Video beim exakt gleichen Bild nach der Werbesequenz ein. Das geht relativ leicht, weil wir ja das Vergleichsbild im linken Fenster haben.

Wir wechseln wieder zum linken Fenster und setzen den Schnittpunkt mit „Alt ,“! Das ist extrem wichtig, um später Störgeräusche in der Tonspur zu vermeiden. Anschließend navigieren wir zum entsprechenden Bild nach der Werbung und frieren dort die Wiedergabe ein. Mit Hilfe des Vergleichsbildes im rechten Fenster geht das relativ einfach. Dann setzen wir den Schnittpunkt mit „Alt .“.

Das wiederholen wir nun für alle Werbeeinblendungen und zuletzt markieren wir das Ende der Sendung wie im letzten Kapitel beschrieben. Dann können

wir uns die Schnittstellen mit „Alt x“ anschauen und speichern. Wie gut wir unsere Schnittpunkte gesetzt haben, können wir später beim Schneiden mit der Vorschau und vor allem der „Nachschau“ sehr gut beurteilen.

7) Zeitverschobene Sendungen

Gelegentlich kommt es vor, dass eine Aufnahme nicht komplett in der OTR-Datei enthalten ist, wenn z. B. ein Sender kurzfristig das Programm geändert hat. Als Premium-User kann man dann die nachfolgende Sendung herunterladen, die den Rest des gewünschten Programmes enthält.

Das oben beschriebene Zwei-Fenster-Prinzip kann auch dazu verwendet werden, beide Dateien so zu schneiden, dass man sie hinterher nahtlos aneinanderfügen kann. Wir öffnen die Originalsendung und gleichzeitig die Folgesendung (linkes Fenster Original, rechtes Folgesendung). Wir gehen hier der Einfachheit halber davon aus, dass es sich um eine Sendung ohne Werbeeinblendungen handelt.

Im linken Fenster setzen wir den Schnittpunkt für den Sendungsbeginn. Dann suchen wir uns kurz vor dem Ende einen geeigneten Schnittpunkt aus. Den gleichen Frame suchen wir auch im rechten Fenster (Folgesendung). Im linken Fenster setzen wir den Schnittpunkt mit „Alt ,“ und speichern die Schnittliste ab. Im rechten Fenster setzen wir den Schnittpunkt mit „Alt .“. Dann suchen wir das Sendungsende, setzen dort den zweiten Schnittpunkt und speichern das Ergebnis mit „Alt x“.

Nach dem Schneiden haben wir zwei Dateien, die nahtlos zusammengefügt werden können. Ich verwende dafür MKVToolNixGUI. Man zieht die erste Datei auf das MKVToolNixGUI-Fenster, anschließend dann die zweite Datei. Dabei erscheint eine Abfrage, wie damit verfahren werden soll. Man wählt „Anhängen“ (Append). Dann startet man das „Muxen“. Nach ein oder zwei Minuten hat man eine neue Datei, die beide Teile enthält.

8) Cutlists bearbeiten und hochladen

Wer manuell schneidet und damit Cutlists erstellt, sollte diese auch anderen Anwendern zur Verfügung stellen. So funktioniert die Community.

Von allen selbst erstellten Cutlists werden Kopien im Upload-Verzeichnis erzeugt. Sie sind so, wie sie erzeugt werden, bereits zum Weiterverteilen geeignet. Evtl. möchte man sie aber noch bearbeiten, z. B. einen besonderen

Kommentar einfügen statt des Standard-Kommentars (den man in rpiotrtool einstellen kann.)

Cutlist-Dateien sind einfache Textdateien, die man mit Leafpad (oder jedem anderen reinen Texteditor) öffnen und modifizieren kann. Dabei darf aber die Struktur nicht zerstört werden. Nur Einträge in der Sektion [Info] dürfen geändert werden. Man kann z. B. den Kommentar ändern oder einen Namensvorschlag für die geschnittene Datei machen. rpiotrtool wertet dies zwar nicht aus, aber andere Programme tun dies. Näheres zum Cutlist-Format findet man hier: <http://cutlist.at/faq/cutlist.html>.

Dann kann man die cutlists mit einem Browser seiner Wahl auf cutlist.at hochladen. Dazu benötigt man einen registrierten Link (FRED). Man geht dann auf die Cockpit-Seite und startet den Upload (im Web-GUI direkt anwählbar). Die von rpiotrtool erzeugten Cutlists können mit allen anderen Schnittprogrammen verwendet werden.

Im Web-GUI sind alle dazu nötigen Funktionen bereits eingebaut, auch für das Bewerten von Cutlists, die man zuvor heruntergeladen hat.

9) Der „otr“ Preset

omxplayerGUI teilt mit anderen Programmen der kweb suite ein gemeinsames System von Einstellungen. Sie können in kweb (Minimal Kiosk Browser) auf einer Webseite bearbeitet werden. Standardmäßig wird kweb mit seiner Menüseite gestartet. Hat man eine andere Startseite eingestellt, kann man die Menüseite aus dem „Open“-Menü aufrufen (oder mit „:m“). Dort klickt man auf „Settings“, um auf die Einstellseite zu kommen.

Die Einstellungen sind ziemlich komplex und werden in beiden Handbüchern (kweb, omxplayerGUI) ausführlich beschrieben. Um die Arbeit zu vereinfachen, gibt es sogenannte Presets, die man mit einem Mausklick aufrufen kann. Man kann auch jederzeit die aktuelle Einstellung als Preset mit einem eigenen Namen speichern. Bei der Installation von rpiotrtool wird ein neuer Preset „otr“ hinzugefügt (wenn zuvor kweb installiert wurde). Dieser enthält Einstellungen, die optimal für das Schneiden von OTR-Videos geeignet sind (wird automatisch verwendet mit omxplayerGUI-OTR-Cut). Ich möchte diese (von der Default-Einstellung abweichenden) Einstellungen hier im Einzelnen auführen und erläutern.

Gewöhnlich wird ein omxplayerGUI-Videofenster geschlossen, wenn das Abspielen eines Videos (oder einer Playlist von Videos und/oder Audiodateien) beendet ist. Springt man beim Schneiden versehentlich zum Ende eines Videos, wären dann alle bereits gesetzten Schnittpunkte verloren und man müsste von vorne beginnen. Deshalb wurde „autofinish“ auf „False“ gesetzt. Dann bleibt das Videofenster geöffnet (Playlist-Modus) und man kann das Video erneut starten (mit einem Doppelklick auf den Namen). Alle Schnittpunkte sind dann noch vorhanden.

Standardmäßig wird omxplayerGUI mit maximierter Fenstergröße gestartet. Alternativen dazu sind „minimale Fenstergröße“ und „Full Screen“. Der OTR-Preset nutzt die „min“-Einstellung (screenmode = min). Gleichzeitig wurde die minimale Fenstergröße auf ein Videoformat mit 512 Bildzeilen gesetzt (videoheight = 512). Das ist eine optimale Größe für den Zwei-Fenster-Modus beim Schneiden von Videos mit doppelten Szenen.

Die letzte Abweichung von der Grundeinstellung ist die wichtigste und bedarf einer Erklärung. OmxplayerGUI benötigt einige Informationen über die Video-Datei (oder den Video-Stream), die es abspielen soll (Seitenverhältnis, Dauer usw.). Normalerweise holt es sich diese Informationen von omxplayer mittels Dbus-Abfragen. Bis diese Informationen vorliegen, sind das Videofenster und einige Einstellungselemente „eingefroren“ (erkenntlich am (f) im Fensterbalken). Es gibt noch eine zweite Möglichkeit, bei der ein externes Programm zur Analyse des Videos verwendet wird. Bislang war das eine zweite omxplayer-Instanz in einem bestimmten Modus. Das hat in der Vergangenheit gelegentlich zu Problemen geführt, weil sich die omxplayer-Instanzen irgendwie gegenseitig blockiert haben. Deshalb wurde diese Methode nur als „Notlösung“ eingesetzt. In der aktuellen Fassung wird stattdessen ffprobe zur Videoanalyse benutzt und damit treten keine Probleme mehr auf. Der OTR-Preset nutzt nun diese alternative Methode (get_DAR=True, durch ein (a) im Fensterbalken erkenntlich), weil nur damit der FPS-Wert (Frames per Second) ermittelt werden kann und dieser Wert für einige der neuen Funktionen benötigt wird. Ein kleiner Nachteil dieses Verfahrens ist, dass das Video einem Moment länger braucht, um zu starten.

Wer omxplayerGUI vorwiegend als Medienplayer einsetzen möchte, der ist mit dem Default-Preset besser bedient. Deshalb wurde ja ein neuer Programmstarter „omxplayerGUI-OTR-Cut“ installiert, der omxplayerGUI mit

dem otr-Preset startet. Man kann auch mit dem „normalen“ omxplayerGUI OTR-Dateien schneiden, wenn man folgendes beachtet:

Der FPS-Wert ist standardmäßig auf 25 gesetzt. Damit kann man OTR-HQ- und MP4-Dateien problemlos schneiden. Für HD-Dateien kann man den benötigten FPS-Wert (50) auch per Tastatur-Befehl „Alt z“ einstellen. Mit „Alt y“ kann er wieder auf 25 Hz. gesetzt werden. Will man Aufzeichnungen von US-Sendern schneiden, die eine andere Bildwiederholrate haben, hilft nur das Kommando „Alt r“. Damit werden die Videoinformationen neu eingelesen und gleichzeitig die Methode gewechselt (erkennlich am „(a)“ im Titelbalken). Das kann man auch generell machen, wenn man Videos schneiden will ohne den OTR-Preset bzw. omxplayerGUI-OTR-Cut zu nutzen, denn damit bekommt man immer die aktuelle Bildwiederholrate (FPS).

10) Überblick über die neuen Tastatur-Kommandos

Die Standard-Tastaturkommandos von omxplayerGUI findet man im Anhang des omxplayerGUI-Handbuchs und auch hier im Anhang. Zur Realisation der Schneidefunktionen wurden nun in dieser Spezialversion eine Reihe von zusätzlichen Tastaturkommandos eingebaut. Diese wurden oben im Textzusammenhang bereits erwähnt, aber sie sollen auf der nächsten Seite der besseren Übersicht wegen nochmals zusammenhängend dargestellt werden. Die neuen Kommandos lassen sich in zwei Gruppen einteilen: Erweiterte Navigationskommandos und Befehle zum Bearbeiten von Schnittpunkten (Setzen, Löschen, Anzeigen und Speichern).

Zuvor noch einige Erläuterungen. Einige der Funktionen stehen nur unter einer bestimmten Bedingung zur Verfügung: Die Playlist darf nur ein Element enthalten und es muss sich dabei um eine Datei handeln (kein Webstream). Das wird im folgenden mit „cut_enabled“ bezeichnet. Eine weitere Einschränkung ist der „Modus“: Player (Abspielfenster mit Video), Playlist oder beide (keine Einschränkung).

Bei allen Navigationsbefehlen ist zu beachten, dass omxplayer immer nur zu Keyframes springen kann. Ausnahme ist der Befehl „Alt n“, der aber nur im „eingefrorenen“ Modus (Geschwindigkeit 0) zuverlässig funktioniert.

Befehl	Funktion	Modus, Bedingung
Alt →	Springe um eine Sekunde vorwärts	Player
Alt ←	Springe um eine Sekunde rückwärts	Player
Alt ↑	Springe um einen Frame vorwärts	Player
Alt ↓	Springe um einen Frame rückwärts	Player
Alt g	Hole Schieber an aktuelle Position	Player
Alt p	Springe zur Schieberposition	Player
Alt n	Springe zum nächsten Bild (exakt)	Player, cut_enabled
Alt e	Springe zum ersten Schnittpunkt	Player, cut_enabled
Alt l	Springe zum letzten Schnittpunkt	Player, cut_enabled
Alt o	Springe zum nächsten Schnittpunkt	Player, cut_enabled
Alt b	Springe zum vorherigen Schnittpunkt	Player, cut_enabled
Alt .	Setze Schnittpunkt im aktuellen Frame	Player, cut_enabled
Alt ,	Setze Schnittpunkt im vorherigen Frame	Player, cut_enabled
Alt ;	Setze Schnittpunkt im ersten Frame	Player, cut_enabled
Alt :	Setze Schnittpunkt im letzten Frame	Player, cut_enabled
Alt d	Lösche letzten Schnittpunkt	Player, cut_enabled
Alt x	Schnittpunkte anzeigen und speichern	Beide, cut_enabled
Alt y	Setze Bildwiederholrate (FPS) auf 25 Hz	Beide, cut_enabled
Alt z	Setze Bildwiederholrate (FPS) auf 50 Hz	Beide, cut_enabled

E) Anhang

1) Standard-Tastaturkommandos in omxplayerGUI

Die folgenden Tabellen stammen aus dem englischen Originalhandbuch zu omxplayerGUI.

Special omxplayerGUI Keyboard Controls

Key	Mode	Action
ALT c, q	any	quit player
ALT k	playing	kill omxplayer instance, when blocking
ALT m	any	toggle window between max and min
ALT f	any	toggle window between full and max
ALT 0	any	window min size
ALT 1...9	any	other window sizes
ALT s	playlist	save playlist
ALT u	playlist	toggle between simple (f) and extended mode
ALT h	any	hide / show controls
ALT a	any	set "Mode:" (aspect ratio) to "auto"
ALT r	any	Toggle get_DAR and reload video information if video is running.
ALT KP+, PgUp	playlist	layer number + 1
ALT KP-, PgDown	playlist	layer number - 1
ALT +	any	next Mode: (aspect ratio)
ALT -	any	previous Mode: (aspect ratio)

omxplayerGUI Keyboard Controls (Play Mode)

omxplayerGUI uses almost the same keyboards commands as omxplayer

omxplayer GUI	Action
1	decrease speed
2	increase speed
<	rewind
>	fast forward
z	show info
j	previous audio stream
k	next audio stream
i	previous chapter
o	next chapter
n	previous subtitle stream
m	next subtitle stream
s	toggle subtitles
d	decrease subtitle delay (- 250 ms)
f	increase subtitle delay (+ 250 ms)
q	exit omxplayer
p, space	pause/resume
-	decrease volume
+, =	increase volume
left arrow	seek -30 seconds
right arrow	seek +30 seconds
down arrow	seek -600 seconds
up arrow	seek +600 seconds
	previous playlist item
	next playlist item

2) Weitere Software des Autors

Um mit dem Raspberry Pi all das machen zu können, was ich möchte, habe ich auch eine ganze Menge eigene Software geschrieben und einiges davon auch veröffentlicht.

Die „kweb suite“ enthält einen kleinen auf Webkit aufbauenden Webbrowser (Minimal Kiosk Browser = kweb) mit einer Vielzahl von Spezialfunktionen, eine grafische Bedieneroberfläche für den Command Line Videoplayer omxplayer (omxplayerGUI) und weitere Tools. Näheres dazu erfährt man hier: <https://www.raspberrypi.org/forums/viewtopic.php?t=40860>

Zu beiden Hauptprogrammen gibt es jeweils ein ausführliches Handbuch, das man auch separat herunterladen kann, wenn man sich erst mal informieren möchte: http://steinerdatenbank.de/software/kweb_manual.pdf und http://steinerdatenbank.de/software/omxplayerGUI_manual.pdf.

Mein „rtranscode V. 4.0 TV package“ enthält einen Echtzeit-Transcoder für DVB TV-Streams. Damit kann man z. B. lokales Digital-TV (DVB-T/S/C) über das Internet anschauen. Näheres dazu findet man hier:

<https://www.raspberrypi.org/forums/viewtopic.php?t=123876>

Auch hierzu gehört ein ausführliches Handbuch, das ebenfalls online verfügbar ist: http://steinerdatenbank.de/software/rtranscode4_manual.pdf

3) Lizenz und Copyright

Die Nutzung von rpiotrtool ist kostenlos. Es wird als Open Source Projekt unter der GNU General Public License Version 3 veröffentlicht. Der Quellcode der mit Nuitka compilierten Version ist beigefügt.

Das Handbuch kann ebenfalls frei verteilt werden, unterliegt aber dem Urheberrecht und darf weder modifiziert noch kommerziell verwertet werden.

Licence:

rpiotrtool - decode and cut OTR-recordings on Raspberry Pi

Copyright 2020 by Günter Kreidl

Version 1.2

Some parts of the code are based on otrverwaltung++ by Markus Liebl.

Copyright (C) 2013 Markus Liebl

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License version 3, as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranties of MERCHANTABILITY, SATISFACTORY QUALITY, or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details: <http://www.gnu.org/licenses>