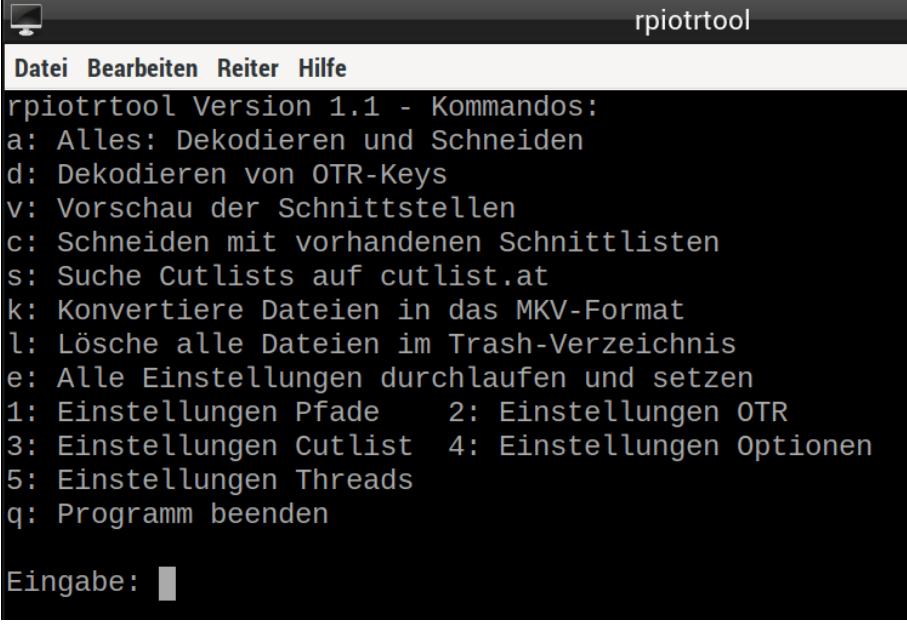


# rpiotrtool

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

## Bedienungsanleitung

Versionen 1.0 und 1.1



```
rpiotrtool
Datei Bearbeiten Reiter Hilfe
rpiotrtool Version 1.1 - Kommandos:
a: Alles: Dekodieren und Schneiden
d: Dekodieren von OTR-Keys
v: Vorschau der Schnittstellen
c: Schneiden mit vorhandenen Schnittlisten
s: Suche Cutlists auf cutlist.at
k: Konvertiere Dateien in das MKV-Format
l: Lösche alle Dateien im Trash-Verzeichnis
e: Alle Einstellungen durchlaufen und setzen
1: Einstellungen Pfade      2: Einstellungen OTR
3: Einstellungen Cutlist   4: Einstellungen Optionen
5: Einstellungen Threads
q: Programm beenden

Eingabe: █
```

# 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 Nutzung.....	11
1) Hardware- und Software-Voraussetzungen.....	11
2) Installation.....	11
3) Bedienungsanleitung.....	13
4) Arbeiten im interaktiven Modus.....	20
C) Manuell schneiden und Cutlists erstellen.....	21
1) Das Problem.....	21
2) Die Lösung.....	22
3) Navigieren im Video.....	24
4) Schnittpunkte setzen, löschen und speichern.....	25
5) Vorlauf und Nachlauf wegschneiden.....	27
6) Werbung und doppelte Szenen herausschneiden.....	28
7) Zeitverschobene Sendungen.....	30
8) Cutlists bearbeiten und hochladen.....	31
9) Der „otr“ Preset.....	31
10) Überblick über die neuen Tastatur-Kommandos.....	33
D) OTR-Workflow.....	35
1) Aufnahmen programmieren.....	35
2) Aufnahmen herunterladen, dekodieren und schneiden.....	37
E) Anhang.....	40
1) Standard-Tastaturkommandos in omxplayerGUI.....	40
Special omxplayerGUI Keyboard Controls.....	40
omxplayerGUI Keyboard Controls (Play Mode).....	41
2) Weitere Software des Autors.....	42
3) OTR-Nutzung.....	43

## **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 ist eine sinnvolle Verwendung von rpiotrtool auf dem Raspberry Pi 4B nicht möglich.

Deshalb wurde die Version 1.1 entwickelt. Sie unterscheidet 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.

Nachteile:

- 1) Die Schnittkontrolle mit VLC ist deutlich schwächer. So werden Abfang und Ende eines Videos „ungenauer“ angezeigt.
- 2) Das manuelle Schneiden ist ohne OSD (On Screen Display) deutlich schwieriger.

3) Das manuelle Schneiden des 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 kurzfristig nicht in Sicht.

Dieses Handbuch beschreibt sowohl die Version 1.0 für Stretch als auch die Version 1.1 für Buster. Die geringfügigen Unterschiede werden an den jeweiligen Stellen entsprechend gekennzeichnet.

## 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 Dienstansbieters [onlinetvrecorder.com](http://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](http://cutlist.at) oder von [onlinetvrecorder.com](http://onlinetvrecorder.com) herunterladen kann. Wenn man meine kweb suite installiert hat, kann man mit Hilfe einer speziellen 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](http://onlinetvrecorder.com) und die OTR-Community

[onlinetvrecorder.com](http://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](http://onlinetvrecorder.com) ist kostenpflichtig (außer in der Nacht), aber es existieren zahlreiche Mirror-Server, von denen man die Dateien kostenlos herunterladen kann.

Die 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 (80 - 99 ct + MWST im

Monat) kann man 120 Dateien pro Monat dekodieren (Ausdehnung auf weitere Dekodierungen ist möglich).

Nach dem Dekodieren hat man bereits eine abspielfähige Filmdatei entweder im AVI- oder MP4-Format. 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 natürlich 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](http://cutlist.at) oder [onlinetvrecorder.com](http://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 (auf [cutlist.at](http://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.

rpiotrttool unterstützt nur das Schneiden in den Formaten HQ, HD und AVI.

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 27 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-IV 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 1, 2 oder 4 GB Arbeitsspeicher. An Schnittstellen stehen ein 2 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 zwölfjährigen 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 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

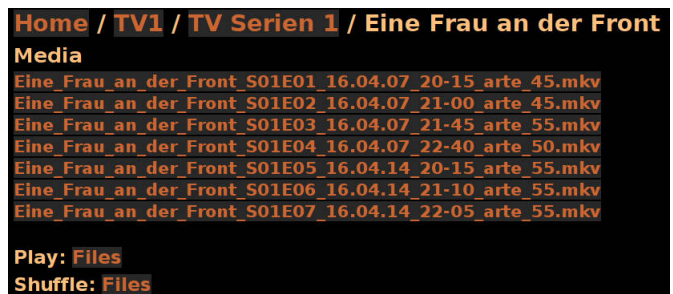
manuellen Schneiden und Erstellen von Cutlists dient nun eine erweiterte Version von omxplayerGUI.

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 ytdl-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 Nutzung**

### **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 Stretch (Version 1.0) oder Raspbian Buster (Version 1.1) 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, 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:

1) Raspbian Stretch, Version 1.0

```
wget http://steinerdatenbank.de/software/rpiotrtool10.tar.gz
tar -xzf rpiotrtool10.tar.gz
cd otr
./install
```

2) Raspbian Buster, Version 1.1

```
wget http://steinerdatenbank.de/software/rpiotrtool11.tar.gz
tar -xzf rpiotrtool11.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 nur zum manuellen Schneiden.

Im Home-Verzeichnis befindet sich nun ein neuer Ordner „otr“ mit einer Reihe von Unterordnern: „decoded“, „cut“, „converted“, „trash“, „tmp“, „upload“, „EPG“, „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 „Hör Zu“ ermöglichen als Hilfsmittel bei der Programmierung von TV-Aufnahmen. 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 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.

Wenn Sie sich während der Installation dafür entschieden haben, die kweb suite zu installieren (nötig zum manuellen Schneiden), sollten Sie nun noch folgende Einstellungen vornehmen:

Starten Sie kweb (Minimal Kiosk Browser) aus dem Startmenü. Auf der Startseite klicken Sie auf „Settings“. Im oberen Bereich dieser Seite sehen Sie eine Reihe von Buttons hinter dem Titel „Available Presets“. Einer dieser Buttons hat den Titel „otr“. Klicken Sie diesen Button an. Damit wird eine für das Schneiden von OTR-Dateien optimale Einstellung aktiviert.

Anschließend klicken sie oben auf „Configuration“. Es erscheint die Einstellseite für kweb. Setzen Sie ein Häkchen in den folgenden Feldern: „Enable Full Zoom“, „Enable JavaScript“, „Enable Cookies“, „Disable Private Browsing“. Klicken Sie anschließend unten auf den Button „Save Configuration“.

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

### 3) Bedienungsanleitung

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

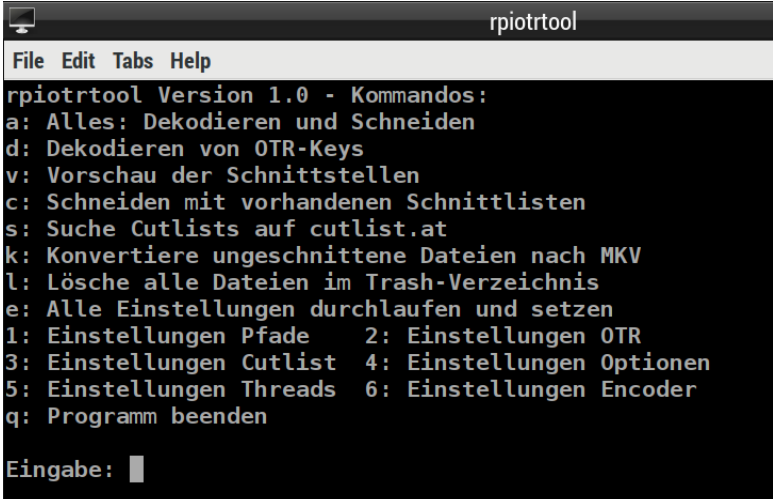
Installation wird eine

Desktop-Datei installiert und

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.



```
rpiotrtool
File Edit Tabs Help
rpiotrtool Version 1.0 - Kommandos:
a: Alles: Dekodieren und Schneiden
d: Dekodieren von OTR-Keys
v: Vorschau der Schnittstellen
c: Schneiden mit vorhandenen Schnittlisten
s: Suche Cutlists auf cutlist.at
k: Konvertiere ungeschnittene Dateien nach MKV
l: Lösche alle Dateien im Trash-Verzeichnis
e: Alle Einstellungen durchlaufen und setzen
1: Einstellungen Pfade      2: Einstellungen OTR
3: Einstellungen Cutlist   4: Einstellungen Optionen
5: Einstellungen Threads  6: Einstellungen Encoder
q: Programm beenden
Eingabe: █
```

Daneben kann rpiotrtool aber auch wie ein ganz gewöhnliches Kommandozeilenprogramm genutzt werden. Dabei gibt es eine Reihe von zusätzlichen Möglichkeiten. Ich werde zunächst diesen Modus näher beschreiben. Die Bedienung im interaktiven Modus ergibt sich daraus von selbst und wird im nächsten Abschnitt beschrieben.

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.

```
pi@raspberrypi4:~ $ rpiotrtool help
rpiotrtool Vers. 1.0 - Copyright 2019 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).*

**Nur in der Version 1.1:** 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.

#### **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 mit omxplayer in einem Overlay in der oberen linken Ecke des Bildschirms oder optional (nur Version 1.1) mit VLC im Vollbildmodus. 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.

**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. 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 in der oberen linken Ecke des Bildschirms angezeigt oder optional (Version 1.1) mit VLC im Vollbildmodus. Wurden nur Vor- und Nachlauf entfernt, werden die ersten paar Sekunden des geschnittenen Videos angezeigt und anschließend die letzten paar Sekunden, sodaß 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 die Cutlist 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“ angeschlossen werden.

**Nur Version 1.1:** 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.

### **search**

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

**Diese Seite wird dann in einem Web-Browser geöffnet.**

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.

Welcher Browser geöffnet wird, hängt davon ab, ob man zusätzliche Browser installiert hat. Es wird nach folgenden Browsern gesucht: kweb, Vivaldi, Firefox, Chromium. Der zuerst gefundene wird verwendet.

### **Version 1.0:**

#### **convert [Datei]**

**Konvertiert eine oder alle ungeschnittenen Dateien ins MKV-Format. Es werden nur Dateien konvertiert, für die keine Cutlist vorliegt.**

Dies ist eine Hilfsfunktion, um das Abspielen von ungeschnittenen Dateien mit VLC zu ermöglichen, das mit dem Original-AVI-Format nicht klarkommt. Die konvertierten Dateien befinden sich anschließend im Verzeichnis „converted“. Nach der Konvertierung wird abgefragt, ob die ungeschnittenen Original-Dateien in den „trash“-Ordner verschoben werden oder ob sie im „decoded“-Ordner verbleiben sollen (um sie z. B. später noch zu schneiden). Es findet keine neue Kodierung statt, sondern nur eine Konvertierung in ein anderes Containerformat (MKV). Deshalb geht das recht schnell.

### **Version 1.1:**

#### **convert [Datei]**

**Konvertiert eine oder alle Dateien ins MKV-Format.**



*Hier werden alle 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. Eine Abfrage zum Löschen der Original-AVI-Dateien gibt es nicht mehr. Nach dem Schneiden einer AVI-Datei, wird die MKV-Kopie automatisch gelöscht,*

*In dieser Version kann die Konvertierung 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 fünf (Version 1.0) bzw. sieben (Version 1.1) 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 „15“ (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.

#### **Nur in der Version 1.1:**

6) Soll VLC für den Preview und die Schnittkontrolle nach dem Schneiden verwendet werden (anstelle von omxplayer)?

Anmerkung: Auf dem Raspberry Pi 4B muß VLC ausgewählt werden. Mit den älteren Modellen ist beides möglich. omxplayer kann 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, weil VLC die Schnittvorschau 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! Die Kombination von ffmpeg als Encoder und Multithreading ist besonders kritisch. 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 beschleunigt werden.

#### **Nur in Version 1.0:**

#### **'encoder': Wähle Encoder für das Smart Rendering (x264 oder ffmpeg)**

Wenn eine Videodatei an beliebigen Stellen (nicht nur an keyframes) geschnitten wird, müssen kleinere Sequenzen an den Schnittstellen neu kodiert werden, um einen nahtlosen Übergang zu ermöglichen. Dieses Verfahren nennt man „Smart Rendering“. Für diese Neukodierung können entweder x264 oder ffmpeg verwendet werden und genau diese Auswahl kann man mit „init encoder“ treffen. ffmpeg arbeitet etwas schneller, benötigt aber mehr Arbeitsspeicher als x264 (default), das auch besser getestet ist.

In der Version 1.1 wurde die Encoder-Auswahl entfernt, weil ffmpeg unter Buster bei HQ-Dateien korrumpierte Bildformate liefert.

## **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 Kommand „clean“ komplett löschen kann (ähnlich dem „Papierkorb leeren“ im Dateimanager).*

**Version 1.1:** 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 otrlkeys 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.*

**Version 1.1:** Bei entsprechend aktivierter Option werden nach dem Dekodieren alle Dateien auch gleich in das MKV-Format konvertiert.

## **Optionen:**

**'-l': Lösche 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**

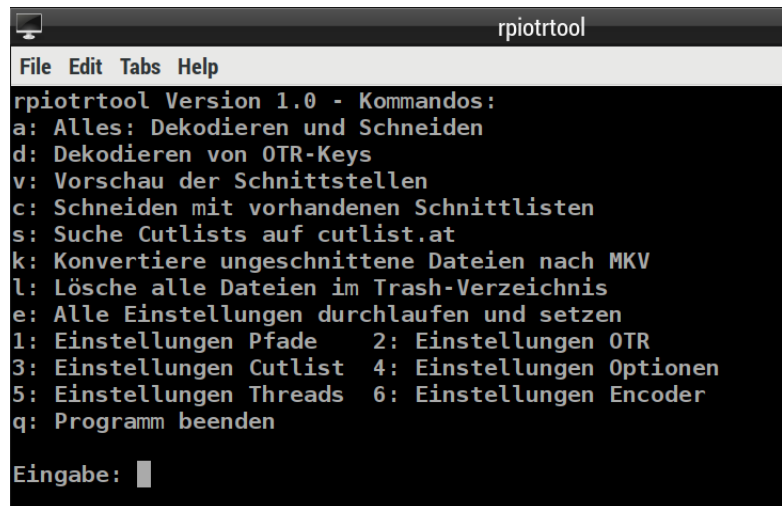
**Im interaktiven (menu) Modus werden die Optionen nicht ausgewertet.**

**Sie können aber mit 'Einstellungen Optionen' bzw. ' Einstellungen Threads' eingestellt 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.*

## 4) Arbeiten im interaktiven Modus

Die bequemste Art, rpiotrtool zu nutzen, ist sicherlich der interaktive Modus. Man kann das Programm aus dem Startmenü aufrufen (Rpi-OTR-Tool in Unterhaltungsmedien), muß sich keine Befehle, Parameter oder Optionen merken und kann



```
rpiotrtool
File Edit Tabs Help
rpiotrtool Version 1.0 - Kommandos:
a: Alles: Dekodieren und Schneiden
d: Dekodieren von OTR-Keys
v: Vorschau der Schnittstellen
c: Schneiden mit vorhandenen Schnittlisten
s: Suche Cutlists auf cutlist.at
k: Konvertiere ungeschnittene Dateien nach MKV
l: Lösche alle Dateien im Trash-Verzeichnis
e: Alle Einstellungen durchlaufen und setzen
1: Einstellungen Pfade      2: Einstellungen OTR
3: Einstellungen Cutlist   4: Einstellungen Optionen
5: Einstellungen Threads   6: Einstellungen Encoder
q: Programm beenden

Eingabe: █
```

nacheinander verschiedene Programmfunktionen nutzen, ohne das Programm zu verlassen. Zum Aufruf einer Programmfunktion muß man nur einen einzigen Buchstaben oder eine einzige Ziffer eintippen (gefolgt von einem Return) und man weiß auch immer, was man eintippen muß. Die Nachteile sind gering: Man kann z. B. nicht eine einzelne Datei zum Schneiden aussuchen und die Kommandozeilen-Optionen sind nicht verwendbar. Die Funktionen selbst sind ansonsten vollkommen identisch mit denen, die im vorherigen Teil für die Kommandozeilen-Nutzung beschrieben worden sind.

## C) Manuell schneiden und 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. 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. rpiotrttool nutzt diese Möglichkeit z. B. für die Schnittstellen-Vorschau und die Schnittkontrolle in geschittenen 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 Handbuches.

### ***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 nicht mehr 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 kann einige Zeit dauern, bis das jemand machen kann.

## **2) Die Lösung**

Ich habe nun die aktuellste Version von omxplayerGUI genommen und um eine Reihe von Zusatzfunktion 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.

Ich will nicht verhehlen, daß dieses Verfahren 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 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. Spring 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 ,“.

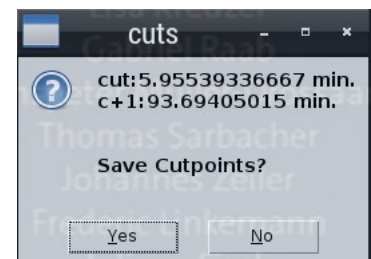
Es gibt noch zwei weitere Tastatur-Kommandos, mit denen man Schnittstellen markieren kann. Diese decken Sonderfälle ab, die man nur in Ausnahmefällen braucht. Mit „Alt ;“ wird der Anfang (der erste Frame) des Videos als Schnittstelle markiert. Dies muß das erste Schnitt-Kommando sein. Mit „Alt :“ wird 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 starten omxplayerGUI entweder mit einem Doppelklick auf die Videodatei, die wir schneiden möchten, oder mit Rechtsklick und wählen dann omxplayerGUI aus dem Menü aus. Es öffnet sich das Videofenster und die Filmdatei wird abgespielt. Wenn man das otr-Preset aktiviert hat, öffnet sich das Videofenster in einer Auflösung von 512 Bildzeilen (im Video). Wir können das Fenster nun mit „Alt m“ maximieren.

Nun spulen wir durch wiederholtes Drücken der rechten Pfeiltaste 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. 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.

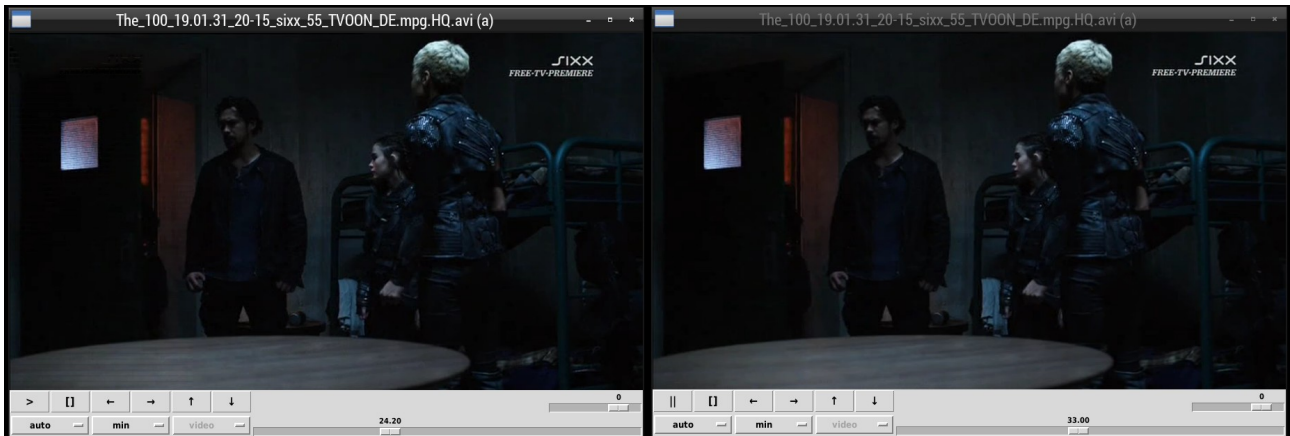
Nun haben wir Anfang und Ende der Sendung markiert und können die „Cutpoints“ abspeichern. Wir drücken „Alt x“ und dann den „Yes“-Button. Wenn wir dann anschließend in rpiotrtool den „cut“ Befehl eingeben, erscheint zunächst (für jede Cutpoints-Datei) eine Meldung: „Erzeuge Cutlist für “ gefolgt vom 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 dem 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 beschrieben 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 mit dem gleichen Video! Verwendet man das „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 eigenen Kommentar einfügen statt des Standard-Kommentars („Mit rpiotrtool geschnitten“)

Cutlist-Dateien sind einfache Textdateien, die man mit Leafpad 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. Die von rpiotrtool erzeugten Cutlists können mit allen anderen Schnittprogrammen verwendet werden.

## **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. Ich möchte diese (von der Default-Einstellung abweichenden) Einstellungen hier im Einzelnen aufführen 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. Bisher 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. Man kann auch damit OTR-Dateien schneiden, wenn man folgendes beachtet:

Der FPS-Wert ist standardmäßig auf 25 gesetzt. Damit kann man OTR-HQ-Dateien und Standard-Avis 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 (erkenntlich am „(a)“ im Titelbalken). Das kann man auch generell machen, wenn man Videos schneiden will ohne den OTR-Preset 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.

<b>Befehl</b>	<b>Funktion</b>	<b>Modus, Bedingung</b>
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

## D) OTR-Workflow

Die Nutzung von OTR mag für den Anfänger kompliziert aussehen. Deshalb möchte ich zum Schluß noch zeigen, wie das bei mir abläuft (neudeutsch „Workflow“ genannt).

Zur Bedienung benötigt man natürlich einen Webbrowser. Der Standardbrowser auf Raspbian Stretch ist Chromium. Man kann aber auch Firefox installieren oder Vivaldi. Wer sich dazu entschlossen hat, meine kweb suite zu installieren (zum manuellen Schneiden erforderlich), der kann auch meinen kleinen „Minimal Kiosk Browser“ (kweb) verwenden. Er ist kein Ersatz für einen vollwertigen modernen Browser, benötigt aber wenig Ressourcen und ist für die Arbeit mit OTR gut geeignet. Von besonderem Interesse kann sein, daß er Downloads nicht selbst durchführt, sondern an den Download-Manager uGet weiterleitet. Ich werde im folgenden auf einige Besonderheiten bei kweb eingehen.

### 1) Aufnahmen programmieren

Zum Programmieren der Aufnahmen muß man natürlich die EPG-Seite von [onlinetvrecorder.com](http://onlinetvrecorder.com) öffnen (Menü: TV-Programm). Diese Seite ist aber unübersichtlich und langsam und nicht besonders gut geeignet, um die Sendungen herauszufinden, die mich interessieren könnten. Deshalb öffne ich eine weitere Seite mit einem einfachen Text-EPG der Zeitschrift „Hör Zu“. Zu diesem Zweck habe ich mir eine eigene HTML-Seite konfiguriert, die alle deutschsprachigen Senderprogramme, die von OTR und dem Text-EPG unterstützt werden, untereinander anzeigt. Bei mir kommt diese Seite von meinem lokalen TV-Server, aber ich habe passende statische Webseiten im Ordner EPG beigefügt. Um einen Überblick über die heutigen Programme der meisten von OTR unterstützten Sender zu bekommen, öffnen Sie einfach mal die Datei

#### Sender Das Erste für Mittwoch, den 12.12.2018 .

- \* 05:00 Uhr . FAKT . Information .
- \* 05:30 Uhr . ARD-Morgenmagazin . Information .
- \* 09:00 Uhr . Tagesschau . Information .
- \* 09:05 Uhr . Live nach Neun . Information .
- \* 09:55 Uhr . Sturm der Liebe . Serie .
- \* 10:44 Uhr . Tagesschau . Information .
- \* 10:45 Uhr . Meister des Alltags . Show .
- \* 11:15 Uhr . Wer weiß denn sowas? . Show .
- \* 12:00 Uhr . Tagesschau . Information .
- \* 12:15 Uhr . ARD-Buffer . Show .
- \* 13:00 Uhr . ZDF-Mittagsmagazin . Information .
- \* 14:00 Uhr . Tagesschau . Information .
- \* 14:10 Uhr . Rote Rosen . Serie .
- \* 15:00 Uhr . Tagesschau . Information .
- \* 15:10 Uhr . Sturm der Liebe . Serie .
- \* 16:00 Uhr . Tagesschau . Information .
- \* 16:10 Uhr . Verrückt nach Meer . Serie .
- \* 17:00 Uhr . Tagesschau . Information .
- \* 17:15 Uhr . Brisant . Information .
- \* 18:00 Uhr . Wer weiß denn sowas? . Show .
- \* 18:50 Uhr . Rentnercops . Serie .
- \* 19:45 Uhr . Wissen vor acht - Werkstatt . Information .
- \* 19:50 Uhr . Wetter vor acht . Information .
- \* 19:55 Uhr . Börse vor acht . Information .
- \* 20:00 Uhr . Tagesschau . Information .
- \* 20:15 Uhr . Anne Burda - Die Wirtschaftswunderfrau . Spielfilm .
- \* 21:45 Uhr . Anne Burda - Die Königin der Kleider . Information .
- \* 22:15 Uhr . Tagesthemen . Information .
- \* 22:45 Uhr . Maischberger . Show .
- \* 00:00 Uhr . Tagesschau . Information .
- \* 00:10 Uhr . Anne Burda - Die Wirtschaftswunderfrau . Spielfilm .
- \* 01:40 Uhr . Anne Burda - Die Königin der Kleider . Information .
- \* 02:13 Uhr . Tagesschau . Information .
- \* 02:15 Uhr . Das Glück am anderen Ende der Welt . Spielfilm .
- \* 03:45 Uhr . Maischberger . Show .

„heute.html“ im Browser Ihrer Wahl. Wer nur wenige Sender durchsuchen möchte, kann stattdessen auch die Datei „menu.html“ verwenden.

**Wichtiger Hinweis:** Chromium, Vivaldi und vermutlich auch Firefox weigern sich neuerdings, die iFrames mit den Senderinhalten darzustellen, weil sie angeblich nicht sicher sind. Um das zu umgehen, öffnet man zunächst die Datei „menu.html“. Dann klickt man mit der rechten Maustaste auf irgendeinen Sendernamen in der linken Spalte und wählt „Öffne in neuem Tab“. Dort kann man dann erzwingen, daß die Seite angezeigt wird. Die Browser merken sich die Einstellungen und zukünftig sollte man dann alle EPG-Seiten problemlos öffnen können. Kweb kennt dieses Problem nicht und öffnet die Seiten anstandslos.

Hat man ein Programm gefunden, das man aufzeichnen möchte, wechselt man zur OTR-Seite (in einem weiteren Tab oder Fenster) und programmiert die Aufnahme.

Ich habe aber stets noch eine weitere Website geöffnet: fernsehserien.de. Da kann ich dann schnell überprüfen, ob es sich bei einer angebotenen Serienfolge um eine neue Folge oder um eine der unzähligen Wiederholungen handelt. Leider funktioniert fernsehserien.de neuerdings nicht mehr ohne Javascript und mit JS ist es elend langsam geworden.

Falls man sich entscheidet, kweb dafür zu nutzen, kann man sich auch eine Keyboard-Shortcut basteln, der gleich alle benötigten Webseiten auf einmal öffnet. Man startet kweb (Minimal Kiosk Browser), klickt auf „Editor“ und scrollt nach unten zum „Keyboard Editor“. Man wählt eine Nummer (z. B. „1“) aus dem Popup-Menü und trägt folgendes in die ersten 3 Felder ein (den ersten Pfad muß man evtl. anpassen):

file:///home/pi/otr/EPG/heute.html

))http://www.onlinetvrecorder.com/v2/?go=schedule&dbc=my

))http://www.fernsehserien.de/

Dann klickt man auf „Create KBD Command“. Nun kann man mit ALT+1 jederzeit die 3 Seiten auf einmal öffnen. „fernsehserien.de“ ist natürlich optional und kann auch weggelassen oder durch eine andere passende Website ersetzt werden. Das „))“ vor den zusätzlichen URLs bewirkt übrigens, daß diese beiden Webseiten in separaten Browser-Instanzen geladen

werden, die auf unterschiedlichen Prozessorkernen laufen. Das beschleunigt die Darstellung.

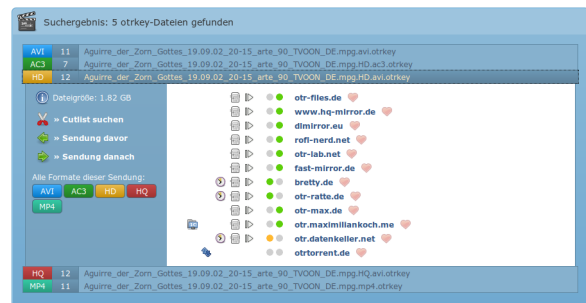
## 2) Aufnahmen herunterladen, dekodieren und schneiden

Wenn man es nicht gerade besonders eilig hat, ist es sinnvoll, einige Tage zu warten, bevor man die Aufnahmen herunterlädt. Dann ist die Chance relativ groß, daß man gleich passende Cutlists findet. Auf der OTR-Website wählt man „Menü: Meine Aufnahmen“ und sieht eine Liste seiner Aufnahmen. Klickt man dann eine Aufnahme an, kann es z. B. so aussehen wie in dieser Abbildung.



Klickt man oben rechts auf „Mirror“, erscheint im rechten Bereich eine Liste der Formate. Klickt man eines davon, erhält man eine Liste der Mirror-Server zum kostenlosen Download. Ich bevorzuge aber eine andere Methode.

Klickt man auf den Link „Auf otrkeyfinder.com anzeigen“, öffnet sich eine weitere Website, otrkeyfinder.com, und man wählt dort das gewünschte Format (hier HD). Man findet eine mehr oder weniger große Anzahl von Mirror-Servern, die die gewünschte Datei (den verschlüsselten otrkey) zum Download anbieten. Links sieht man aber auch einen Link „Cutlist suchen“. Klickt man den an, gelangt man zur Suchseite von cutlist.at, wo evtl. vorhandene Cutlists zum Download angeboten werden. Klickt man dort auf „View“, erhält man detaillierte Informationen zu der jeweiligen Cutlist. Man kann so parallel zu seinen Aufnahmen (otrkeys) auch gleich die passende Cutlist herunterladen.

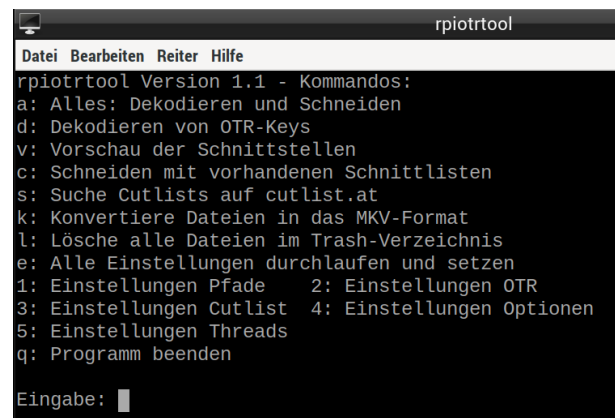


Dieses Verfahren hat aber auch einen großen Nachteil: Man kann später die heruntergeladenen Cutlists auf cutlist.at nicht bewerten. Dazu benötigt man „registrierte“ Downloads. Und selbst erstellte Cutlists hochladen kann man ebenfalls nur als registrierter Anwender. cutlist.at verwendet ein vollkommen anonymes Registrierungsverfahren, bei dem keine persönlichen Daten

gespeichert werden. Man erhält einen personalisierten Link (auch FRED) genannt), den man in seinem Browser (am besten als Link in den Bookmarks gespeichert) zum Aufruf von cutlist.at (sniplist) verwendet. Diesen FRED kann man auch in rpiotrtool eingeben und für die integrierte Cutlist-Suche verwenden. Dann werden die Downloads registriert und können später auf cutlist.at (mit FRED aufrufen) bewertet werden. Das ist ein wichtiger Beitrag zur Funktion der Community.

*Hinweis: Verwendet man kweb zum Herunterladen der Otrkeys (und auch der Cutlists), werden alle Downloads an den Download-Manager uGet weitergeleitet. In der Regel funktionieren „automatische“ Downloads in kweb nicht. Man muß den Download explizit mit einem Rechtsklick auf den Download-Link und Auswahl im PopUp-Menü starten. Eine Ausnahme bilden Cutlist-Downloads, die man mit der linken Maustaste anklicken muß.*

Hat man alle seine Aufnahmen eingesammelt, ruft man rpiotrtool auf. Ich gehe hier mal davon aus, daß der Normalanwender den interaktiven Modus bevorzugen wird (kann aus dem Programm-Menü gestartet werden). Hat man schon Cutlists mit heruntergeladen, kann man dann „a“ (Alles) anwählen und es wird dekodiert und geschnitten.



```
rpiotrtool
Datei Bearbeiten Reiter Hilfe
rpiotrtool Version 1.1 - Kommandos:
a: Alles: Dekodieren und Schneiden
d: Dekodieren von OTR-Keys
v: Vorschau der Schnittstellen
c: Schneiden mit vorhandenen Schnittlisten
s: Suche Cutlists auf cutlist.at
k: Konvertiere Dateien in das MKV-Format
l: Lösche alle Dateien im Trash-Verzeichnis
e: Alle Einstellungen durchlaufen und setzen
1: Einstellungen Pfade      2: Einstellungen OTR
3: Einstellungen Cutlist   4: Einstellungen Optionen
5: Einstellungen Threads
q: Programm beenden
Eingabe: █
```

Da ich selbst die meisten Aufnahmen archiviere und auf gute Schnittqualität Wert lege, gehe ich selbst anders vor. Ich lade zunächst nur die Otrkeys herunter. In rpiotrtool lasse ich dann zunächst nur dekodieren. Anschließend starte ich die Suchfunktion, suche nach passenden Cutlists und lade sie herunter. Das Programm erstellt eine Suchseite für alle dekodierten Dateien, für die noch keine Cutlist heruntergeladen wurde.

Anschließend starte ich das Schneiden. Ich habe die Vorschau fest eingeschaltet, das heißt, ich sehe mir wirklich erst die Schnitte an. Zum Teil entscheide ich dann schon bei der Vorschau, daß mir eine Cutlist nicht gefällt. Anschließend werden alle (positiv bewerteten) Dateien geschnitten. Auch die nachträgliche Kontrolle der Schnittpunkte habe ich aktiviert. Gefallen mir dann Start oder Ende oder auch interne Schnittpunkte nicht, lehne ich das

Ergebnis ab. Dann kann ich später eventuell noch eine andere Cutlist suchen und ausprobieren.

Die geschnittenen Dateien im Ordner „cut“ verschiebe ich dann anschließend in mein Archiv. Ich besuche auch [cutlist.at](http://cutlist.at) (mit meinem FRED) und bewerte die Schnittlisten, die ich heruntergeladen habe.

Es werden in der Regel einige ungeschnittene Dateien übrigbleiben. Man kann dann nach ein paar Tagen mit der Suchfunktion nochmal überprüfen, ob es inzwischen eine Cutlist gibt, oder sich auch gleich dazu entscheiden, selbst manuell zu schneiden und die dabei erzeugte Cutlist hochzuladen und anderen Anwendern zur Verfügung zu stellen. Je mehr Leute das machen, desto besser funktioniert die Community.

Mit der Spezialversion von omxplayerGUI werden die Dateien nicht geschnitten, sondern es werden nur Schnittpunkte markiert und abgespeichert. Man ruft dann anschließend in rpiotrtool wieder die Cut-Funktion auf, die die eigentlichen Cutlists erstellt und dann die Dateien schneidet. Auch hier ist die anschließende Schnittkontrolle wichtig, um die wirkliche Qualität der Schnitte und damit auch der Cutlist zu überprüfen.

## 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

<b>omxplayer GUI</b>		<b>Action</b>
1	1	decrease speed
2	2	increase speed
<	<	rewind
>	>	fast forward
z	z	show info
j	j	previous audio stream
k	k	next audio stream
i	i	previous chapter
o	o	next chapter
n	n	previous subtitle stream
m	m	next subtitle stream
s	s	toggle subtitles
d	d	decrease subtitle delay (- 250 ms)
f	f	increase subtitle delay (+ 250 ms)
q	q, Esc	exit omxplayer
p, space	p, space, Return, Enter	pause/resume
-	-, KP-	decrease volume
+, =	+, KP+	increase volume
left arrow	left arrow	seek -30 seconds
right arrow	right arrow	seek +30 seconds
down arrow	PgDown, ", "	seek -600 seconds
up arrow	PgUp, .	seek +600 seconds
	up arrow	previous playlist item
	down arrow	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](http://steinerdatenbank.de/software/kweb_manual.pdf) und [http://steinerdatenbank.de/software/omxplayerGUI\\_manual.pdf](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](http://steinerdatenbank.de/software/rtranscode4_manual.pdf)

### 3) OTR-Nutzung

Der folgende Text stammt von onlinetvrecorder.com. Ich möchte betonen, daß ich weder mit der Website noch der dahinter stehenden Firma in irgendeiner Weise verbunden bin. Ich bin nur ein einfacher langjähriger Benutzer dieses Dienstes und Mitglied der OTR-Community.

#### 1. Was ist Onlinetvrecorder.com (OTR)?

Kurz gesagt: Ein Videorecorder, der seit 2005 rund um die Uhr ALLE Sendungen ALLER Sender aufzeichnet. Während Sie dies lesen werden über 100 TV-Sender aus Deutschland, USA, UK und anderen Ländern aufgenommen. Sie können diese Aufnahmen später gratis herunterladen, auf DVD brennen oder sofort unterwegs im Handy oder im Büro ansehen (streamen). Es gibt auch Apps für iOS und Android.

#### 2. Wie kann man OTR kostenlos benutzen?

MP4-Streams (Handy-Format) sind immer kostenlos (sofort ansehbar). Zwischen 0 und 8 Uhr sind auch Downloads auf OTR gratis (HappyHour). Wenn Sie über Mirror-Server (das sind externe Server) laden oder bei uns per Torrent, sind alle Downloads rund um die Uhr gratis. Ansonsten kosten Downloads und Streams bei OTR 12 Cent pro Gigabyte (1.024 MB). User mit dem "Beginner"-Status können 10 Aufnahmen pro Monat gratis sehen (die ersten 5 sind gratis, die folgenden 5 kosten 5 Cent). User mit dem "Premium"-Status können 120 Aufnahmen pro Monat ansehen, dieser Status kostet 99 Cent/Monat, bzw. 80 Cent/Monat bei jährlicher Zahlung. Durch Klicken von speziellen Werbe-Bannern (gekennzeichnet) oder dem Werben anderer User z.B. via Facebook-Postings, können Sie den Premium-Status auch gratis erhalten oder ihr Konto gratis auffüllen.

#### 3. Was ist ein Mirror?

Mirror sind externe Webseiten, die OTRKEY-Dateien kostenlos zum Download anbieten.

#### 4. Was ist eine OTRKEY-Datei?

Das ist eine verschlüsselte Video-Datei Ihrer Aufnahme, die Mirror-Server anbieten.

Der Vorteil bei Mirror Servern ist, dass der Download rund um die Uhr kostenlos ist.

Mit unserer kostenlosen Dekoder-Software ([hier](#)) wird aus der OTRKEY-Datei innerhalb von 2 Minuten ein abspielbares Video erzeugt.

Von unseren Servern können Sie auch direkt abspielbare Videos runterladen, die Datei muss dann nicht extra mit dem Dekoder dekodiert werden.

Allerdings ist hier der Download nur zwischen 0 und 8 Uhr kostenlos, ansonsten kostet es 12 Cent pro 1 GB.

#### 5. Wozu brauche ich eine Dekoder-Software?

Um eine OTRKEY-Datei in eine abspielbare Video-Datei umzuwandeln.

Die Verschlüsselung ist aus urheberrechtlichen Gründen nötig.

Das kommt darauf an, wo und wie Sie diese Sendung ansehen wollen.

Empfehlungen:

-Auf dem Handy: MP4

-Auf dem PC und normalen Fernsehern: AVI/DivX  
-Auf HDTV-Fernsehern: HQ  
User mit Status Beginner können nur MP4 herunterladen.

## **7. Wie geht das mit den Bannern?**

Einige Banner, die auf OTR angezeigt werden, können Ihren Kontostand erhöhen, wenn Sie sie anklicken. Da diese Banner von anderen OTR-Mitgliedern geschaltet werden, variiert die Anzahl anklickbarer Banner je nach Tageszeit.

## **8. Was ist eine Wishlist?**

Eine Wishliste ist ein wiederkehrender Auftragsauftrag, z.B. für Ihre Lieblingsserien. Sie müssen so nicht jede Sendung einzeln aufnehmen, sondern es werden auch zukünftige Sendungen automatisch aufgezeichnet. Für diese Komfort-Funktion benötigen Sie den Premium-Status.

Als Beginner können Sie nur auf Sendungen zugreifen, die Sie vorher einzeln zur Aufnahme angeklickt haben.

## **9. Wie werde ich Premium?**

Wenn Sie mindestens Cent auf Ihrem Konto haben, können Sie den Premium-Status auf dieser Seite starten: [hier](#). Diese 99 Cent bekommen Sie am einfachsten und schnellsten durch eine direkte Einzahlung oder kostenlos durch Bannerklicks.

## **10. Wie kann man OTR am komfortabelsten benutzen?**

99 Cent Monatsgebühr für den Premium-Status einzahlen und z.B. 1 Euro für den Download von 8 Gigabyte. Das reicht je nach Format für 16 Spielfilme.

Als Premium-User sind dann sämtliche Sendungen sämtlicher Sender ab Beginn der Mitgliedschaft downloadbar.

Premium-User müssen nicht mehr jede Sendung vorher programmieren und sich vorher überlegen, was wohl interessant sein könnte. Sie haben Zugriff auf das gesamte TV-Angebot der Vergangenheit. Sie müssen also zukünftig Ihre Freizeitaktivitäten nicht mehr wegen guter Spielfilme unterbrechen. Sie können ALLES später ansehen und sogar werbefrei!

Premium-User können auch hochauflösende TV-Formate ansehen.

Zudem haben Sie Zugriff auf Sender aus den USA, UK und anderen Ländern.