

schlitzbäda

D O K U M E N T A T I O N

YAMuPlay

YAMuPlay – Yet Another MUsic Player für den Raspberry Pi



GNU General Public License v3
© 2016 - 2017 by schlitzbäda

Datum:
18.08.2017



Die Abbildung auf dem Display stammt aus dem Video zu folgendem Musikstück:

Riegler Hias feat. d' Hundskrippln *Gloana Bauer* bei 2:42

<https://www.hundskrippln.de/#filme>

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Tabellenverzeichnis	IV
1 Einführung	1
1.1 Kurzbeschreibung der Software	1
1.2 Rechtliche Hinweise zu Lizenz, Gewährleistung und Links	2
1.2.1 Software unter GPL v3	2
1.2.2 Dokumentation unter FDL v1.3	2
1.2.3 Externe Internet-Links	3
1.3 Konventionen dieser Dokumentation	3
1.4 Abkürzungsverzeichnis	4
1.5 Danksagung	4
2 Software YAMuPlay	5
2.1 Installation von YAMuPlay	5
2.2 Beschreibung und Bedienung von YAMuPlay	7
2.2.1 Bedienkonzept	7
2.2.2 Aufbau der GUI	7
2.2.3 Menü	9
2.2.4 Bedienung über die Tastatur	11
2.2.5 Aufruf im Terminalfenster und Kommandozeilenparameter	12
2.3 Erweiterungen und Verbesserungen der Software	13

Abbildungsverzeichnis

2.1	Erstmaliger Aufruf von YAMuPlay	6
2.2	Steuerelemente in der Benutzeroberfläche von YAMuPlay	8
2.3	Aboutbox von YAMuPlay	11
2.4	Video mit YAMuPlay in einem Fenster abspielen	12

Tabellenverzeichnis

1.1	Konventionen der Dokumentation	3
2.1	Zusätzliche Python-Module für den Betrieb von YAMuPlay	5
2.2	Steuerelemente in der Benutzeroberfläche von YAMuPlay	8
2.3	Funktionstastenbelegung in YAMuPlay	12
2.4	Kommandozeilenparameter von YAMuPlay	13

1 Einführung

Diese Dokumentation beschreibt die Installation und Anwendung von YAMuPlay V0.2 auf einem Raspberry Pi unter dem Betriebssystem *Raspbian Jessie with Desktop (PIXEL)* vom 05.07.2017, Imagedatei **2017-07-05-raspbian-jessie.img**. Um das SD-Image für eine SD-Karte mit nur 4GB Speicherkapazität anzupassen, wurde das Programmpaket *Mathematica/Wolfram* (ca. 683MB) gelöscht und das Dateisystem entsprechend verkleinert.

Die Software wurde auf einem Raspberry Pi 2B (Quad Core, 900MHz, 1GB RAM) erstellt und getestet. Auch die Inbetriebnahme auf einem Raspberry Pi Modell A+ (Single Core, 700MHz, 512MB RAM) verlief erfolgreich.

Nicht geprüft wurde hingegen die Lauffähigkeit unter der neuen Betriebssystemversion *Raspbian Stretch*, Imagedatei **2017-08-16-raspbian-stretch.img** oder neuer!

1.1 Kurzbeschreibung der Software

„YAMuPlay“ bzw. „Yet Another MUsic Player“ ist ein in **Python3** erstellter Wrapper für den Mediaplayer **omxplayer.bin**, welcher optimal auf die Hardware des Raspberry Pi zugeschnitten ist. Neben der Steuerung des nackten **omxplayer.bin** über Kommandozeilenparameter oder die Tastatur („hot keys“) ist auch eine Kommunikation mittels **D-Bus** möglich.

Die ursprüngliche Absicht war, eine Software zu erstellen, mit der eine einfache und intuitive Handhabung einer Musiksammlung möglich ist. Dabei sollte sich die Bedienung an einem klassischen CD-Spieler orientieren. Unter Linux und somit auch auf dem Raspberry Pi gibt es das hervorragende Softwarepaket **MPD** („Music Player Daemon“), das u. a. über ALSA alle auf dem Computer installierten Soundkarten unterstützt und zahlreiche Client-Anwendungen (MPC) zur Verfügung stellt. Sein einziger Nachteil liegt in der Verwaltung der Musikdateien über eine Datenbank und den damit verbundenen umständlichen und aufwändigen Aktualisierungsarbeiten bei neuen Musikdateien, die dem einfachen CD-Spieler-Prinzip entgegenstehen.

Gerne lasse ich mich hier vom Gegenteil überzeugen, falls es für den Raspberry Pi taugliche MPCs geben sollte, die obigen Anforderungen genügen 😊 (<mailto:schlizbaeda@gmx.de>).

Die Software **YAMuPlay** in Version V0.2 besteht aus einer grafisch einfach gehaltenen, mit dem Python-Modul **tkinter** erstellten Benutzeroberfläche (GUI) ähnlich einem Dateimanager (siehe Abbildung 2.1): Im linken Teil befindet sich ein sogenanntes Treeview-Steuerelement, in dem die hierarchische Ordnerstruktur von Laufwerken angezeigt wird, die unter **/media/pi** gemountet sind. Am Raspberry Pi neu angeschlossene USB-Laufwerke werden automatisch

erkannt und entsprechend in die Ordnerstruktur eingebunden. Mediendateien (nicht nur Musik, auch Videos) können durch einen Doppelklick in die Playlist eingefügt werden. Links oben ist ein Eingabefeld für die Titelsuche enthalten.

Der rechte Teil der GUI besteht oben aus den Schaltflächen, die den klassischen Tasten eines CD-Spielers entsprechen: *Play/Pause*, *fast forward*, *rewind*, Titelsprung und *Stop*. Direkt darunter befindet sich eine in V0.2 noch nicht aktive Schiebeleiste, die später die aktuelle Position in der gerade laufenden Mediendatei anzeigen und auch ein schnelles Verschieben ermöglichen soll. Rechts unten ist die Playlist, in der alle zum Abspielen ausgewählten Mediendateien angezeigt werden sowie die Schaltflächen zum Entfernen und Verschieben der Mediendateien. Ein Doppelklick auf eine Mediendatei der Playlist beendet das laufende Stück und beginnt mit dem Abspielen des angeklickten Titels.

Das Menü der GUI ermöglicht unter **"Datei"** das Laden und Speichern von Playlists sowie das Beenden der Software. Im Dropdown-Menü **"Ansicht"** können diverse Einstellungen für die Wiedergabe von Videodateien eingestellt werden. Unter **"Hilfe"** ist derzeit nur die Anzeige einer Aboutbox möglich.

1.2 Rechtliche Hinweise zu Lizenz, Gewährleistung und Links

1.2.1 Software unter GPL v3

Die Software **YAMuPlay** wird unter der Lizenz GPL v3 der *Free Software Foundation* veröffentlicht. Sie darf frei kopiert und privat oder kommerziell gemäß der Lizenzvorgaben verwendet werden. Bitte lesen Sie die GPL v3, die sich im Programmpaket in der Datei COPYING befindet oder laden Sie den Originaltext aus dem Internet von <https://www.gnu.org/licenses/gpl-3.0> herunter.

schlitzbäda als Urheber und Copyrightinhaber stellt die Software „so wie sie ist“ ohne Garantie und ohne Zusicherung einer bestimmten Funktionalität zum Download bereit. Es steht Ihnen als Anwender frei, die Software zu benutzen oder es eben nicht zu tun. Entscheiden Sie sich für die Benutzung, so tun Sie dies auf eigenes Risiko und auf eigene Verantwortung. Der Autor übernimmt keine Garantie für die Software und deren Funktion und haftet auch nicht für Schäden, die aus der Installation und Verwendung der Software entstehen.

1.2.2 Dokumentation unter FDL v1.3

Diese Dokumentation wurde mit dem Textsatzsystem L^AT_EX erstellt.

Sie darf gemäß der Bestimmungen aus der Lizenz FDL v1.3 (oder nachfolgend) kopiert, verteilt und/oder geändert werden. Details siehe <http://www.gnu.org/licenses/fdl.html>

1.2.3 Externe Internet-Links

Einfach weil's sein muss:

In diesem Dokument befinden sich Hyperlinks zu verschiedenen externen Seiten anderer Anbieter im Internet, die außerhalb des Verantwortungsbereiches des Autors liegen. Durch die bloße Anbringung eines Links auf eine fremde Seite macht sich der Autor deren Inhalte nicht zu eigen, da er auf die Gestaltung dieser Seiten keinerlei Einfluss hat. Zum Zeitpunkt der Verlinkung waren auf den verlinkten Seiten keine illegalen Inhalte erkennbar. Sollten aktuelle oder künftige Inhalte jedoch rechtswidrig sein, so kann der Autor darüber per e-mail an <mailto:schlizbaeda@gmx.de> informiert werden. Es werden dann entsprechende Maßnahmen zur Beseitigung des/der betroffenen Links ergriffen.

1.3 Konventionen dieser Dokumentation

Folgende gestalterische Konventionen werden für die Dokumentation festgelegt:





 Hinweis  Text	Ein Hinweis enthält zusätzliche Information bzw. relevante Erläuterung zu einer bestimmten Funktionalität
 Achtung  Text	Ein Warnhinweis, dessen Nichtbeachtung zu Geräteschäden führen kann
[Schaltfläche]	Kennzeichnung von Schaltflächen der Software YAMuPlay
"Menüpunkt"	Kennzeichnung von Menüpunkten der Software YAMuPlay
Meldung	Kennzeichnung von Meldungen der Software YAMuPlay

Tabelle 1.1: *Konventionen der Dokumentation*

1.4 Abkürzungsverzeichnis

Folgende Abkürzungen werden in dieser Dokumentation verwendet:

ALSA	A dvanced L inux S ound A rchitecture
D-Bus	D esktop- B us
GUI	G raphical U ser I nterface
HDMI	H igh D efinition M ultimedia I nterface
MPC	M usic P layer C lient
MPD	M usic P layer D aemon
SoC	S ystem o n S ilicon: µP-Bausteine mit viel Peripheriemodulen wie z. B. der Broadcom 2835, 2836, 2837 auf dem Raspberry Pi

1.5 Danksagung

Mein Dank gilt insbesondere folgenden Personen

- ***Riegler Hias und d' Hundskrippln***

In dieser Dokumentation wird an verschiedenen Stellen auf das Musikstück *Gloana Bauer* von **Riegler Hias feat. d' Hundskrippln** verwiesen, weil ich dieses Lied einfach gut finde.

<https://www.hundskrippln.de> 93336 Steinsdorf

<http://www.riegler-hias.de> 93349 Hiendorf

Ich bedanke mich für die Erlaubnis, Screenshots aus dem Video zu diesem Musikstück verwenden zu dürfen.

- **meigrafd**

Dieser Benutzer des größten deutschsprachigen Forums für den Raspberry Pi lud damals offenbar die Version V0.1 von YAMuPlay herunter und testete es anschließend ziemlich ausführlich durch. Er ließ mir äußerst konstruktive Kritik zukommen, an welchen Stellen mein ursprüngliches Python-Script noch zu verbessern wäre.

Insbesondere schlug er mir vor, meinen damaligen Code von klassischem Python mit unzähligen `global`-Deklarationen auf objektorientierte Programmierung umzustellen. Dazu hatte er *einfach so* mein komplettes Script angepasst und mir wieder zukommen lassen. Darauf aufbauend korrigierte und erweiterte ich das Script auf den jetzigen Versionsstand V0.2. <http://www.forum-raspberrypi.de/User-meigrafd>

2 Software YAMuPlay

2.1 Installation von YAMuPlay

YAMuPlay wird von schlizbäda unter GitHub unter <https://github.com/schlizbaeda/yamuplay> zum Download bereitgestellt. Das Softwarepaket besteht im Wesentlichen aus dem Python3-Script `yamuplay.py` und einigen Zusatzdateien sowie dem L^AT_EX-Quellcode dieser Dokumentation.

Damit das Python3-Script auf dem Raspberry Pi lauffähig wird, müssen weitere Python-Module installiert werden, die ebenfalls auf GitHub verfügbar sind oder als Paket in Raspbian enthalten sind:

Python-Modul	Lizenz	Quelle
python-omxplayer-wrapper	LGPL v3	https://github.com/willprice/python-omxplayer-wrapper.git
python3-dbus	MIT	<code>apt-get install python3-dbus</code>
pyudev v0.21.0	LGPL v2.1	https://github.com/pyudev/pyudev.git
python-magic	MIT	https://github.com/ahupp/python-magic.git

Tabelle 2.1: *Zusätzliche Python-Module für den Betrieb von YAMuPlay*

Installation

Auf einem Raspberry Pi, der mit dem Internet verbunden ist, kann die Installation in einem Terminalfenster durch Eingabe folgender Anweisungen durchgeführt werden:

Programmpaket **YAMuPlay**

```
cd /home/pi
git clone https://github.com/schlizbaeda/yamuplay.git
cd yamuplay
chmod 755 yamuplay.py
```

Programmpaket **python-omxplayer-wrapper**

```
cd /home/pi/yamuplay
git clone https://github.com/willprice/python-omxplayer-wrapper.git
cd python-omxplayer-wrapper
sudo python3 setup.py install
```

Programmpaket **python3-dbus**

```
cd /home/pi/yamuplay  
sudo apt-get install python3-dbus
```

Programmpaket **pyudev v0.21.0**

```
cd /home/pi/yamuplay  
git clone https://github.com/pyudev/pyudev.git  
cd pyudev  
sudo python3 setup.py install
```

Programmpaket **python-magic**

```
cd /home/pi/yamuplay  
git clone https://github.com/ahupp/python-magic.git  
cd python-magic  
sudo python3 setup.py install
```

Programmaufruf

Der Start von YAMuPlay erfolgt über das Kommando

```
/home/pi/yamuplay/yamuplay.py
```

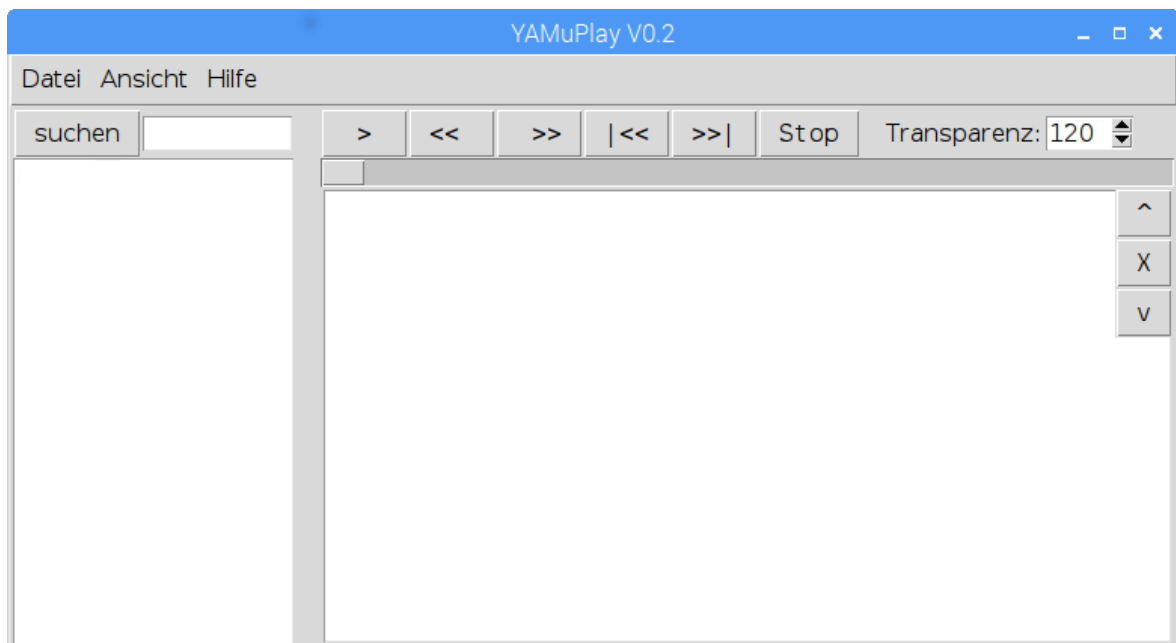


Abbildung 2.1: *Erstmaliger Aufruf von YAMuPlay*

2.2 Beschreibung und Bedienung von YAMuPlay

Die Software YAMuPlay V0.2 ist kein eigener Mediaplayer, sondern eine Bedieneroberfläche für den existierenden Kommandozeilen-Mediaplayer `omxplayer.bin`, der in den meisten(?) Betriebssystem-Distributionen für den Raspberry Pi standardmäßig enthalten ist. Diese Oberfläche ist quasi eine „Hülle“ – oder auf englisch – ein *Wrapper* für `omxplayer.bin`. Der Sinn für die Programmierung von YAMuPlay lag u. a. darin, eine Plattform unter **Python3** zu schaffen, mit der man relativ einfach Mediendateien (Musik und Videos) unter Zuhilfenahme von `omxplayer.bin` abspielen kann. Da `omxplayer.bin` als eigener Prozess gestartet wird und die Kommunikation mit YAMuPlay über **D-Bus** erfolgt, findet das Abspielen aus der Sicht des übergestülpten Python-Scripts im Hintergrund statt; in Python können währenddessen andere Aufgaben erledigt werden. Ein weiterer Vorteil liegt darin, dass `omxplayer.bin` hauptsächlich den GPU-Teil des SoC *Broadcom 283x* auf dem Raspberry Pi beansprucht und somit den CPU-Teil kaum belastet. Die CPU-Last beim Betrieb von YAMuPlay liegt auf einem Raspberry Pi 1 mit dem Broadcom-SoC 2835 bei ca. 25%-35%, es bleiben genügend CPU-Ressourcen frei.

Nachteilig ist jedoch, dass ALSA aufgrund der großen Hardwarenähe von `omxplayer.bin` nicht eingebunden ist und somit wirkungslos bleibt. Daher funktioniert weder der ALSA-Mixer von Raspbian, noch kann eine Soundkarte wie Hifiberry DAC+ eingesetzt werden. Die Audioausgabe kann nur über HDMI oder den Analoganschluss des Raspberry Pi erfolgen, eine Lautstärkeregelung muss am Audioverstärker vorgenommen werden!

2.2.1 Bedienkonzept

Es gibt bereits genügend kompliziert zu bedienende Mediaplayer vor allem hinsichtlich der Erstellung, Änderung und Verwaltung von Playlists. Bereits in Kapitel 1.1 wurde die Problematik des Music Player Daemons angedeutet, insbesondere wenn Änderungen in der Liste der abzuspielenden Stücke *schnell* vorgenommen werden sollen/müssen. Vielmehr ist in solchen Fällen eine intuitive Bedieneroberfläche erforderlich. YAMuPlay ist ein Versuch in diese Richtung, aber auch hier könnte noch viel getan werden, das will ich gar nicht abstreiten. Weitere geplante Verbesserungen dazu für künftige Programmversionen sind in Kapitel 2.3 aufgeführt.

2.2.2 Aufbau der GUI

Das Hauptfenster von YAMuPlay ist zweigeteilt, siehe Abbildung 2.2. Die wichtigsten Steuerelemente zur Bedienung von YAMuPlay befinden sich auf der grafischen Oberfläche und können direkt durch Anklicken bedient werden.

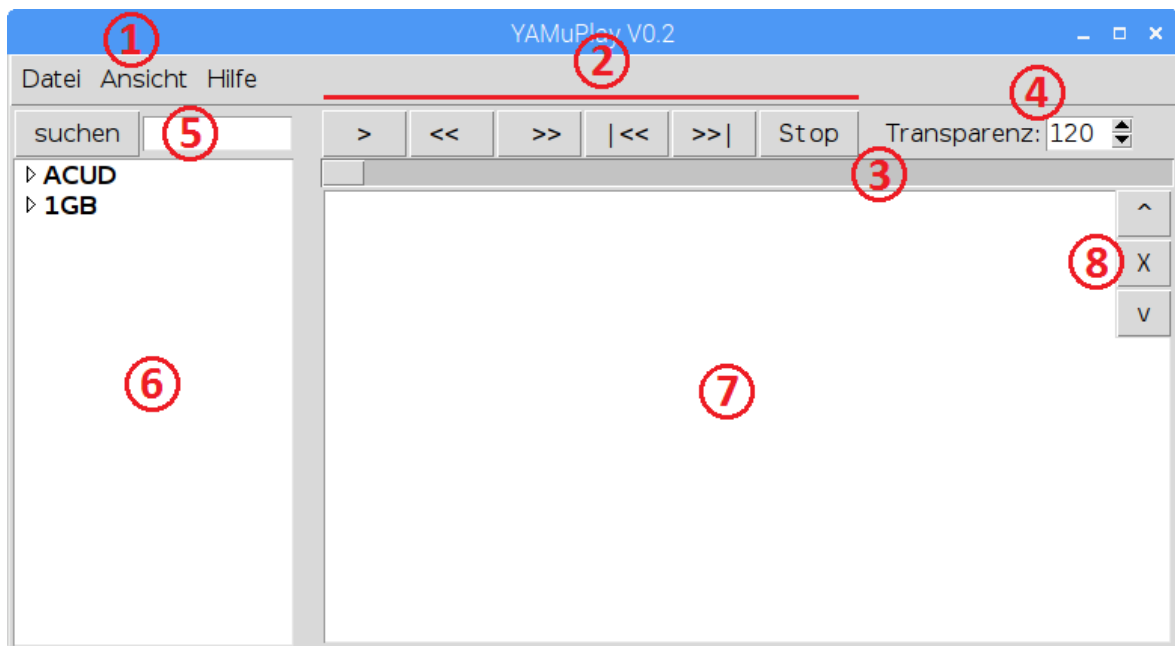


Abbildung 2.2: Steuerelemente in der Benutzeroberfläche von YAMuPlay

- (1) Hauptmenü
- (2) an einen CD-Spieler angelehnte Schaltflächen
- (3) Fortschrittsbalken für die laufende Mediendatei – **derzeit noch inaktiv**
- (4) Einstellung für den Alphawert (Transparenz) von angezeigten Videos
- (5) Titelsuche
- (6) Anzeige der Dateien auf den angesteckten USB-Laufwerken
- (7) Playlist
- (8) Steuerelemente zum Löschen oder Verschieben von Elementen der Playlist

Tabelle 2.2: Steuerelemente in der Benutzeroberfläche von YAMuPlay

In der linken Hälfte werden alle auf den angeschlossenen USB-Laufwerken enthaltenen Dateien und Verzeichnisse in einer hierarchischen Baumstruktur angezeigt, gewöhnliche Dateien in Normalschrift und Verzeichnisse in fetter Schrift. Ein Doppelklick auf ein Verzeichnis öffnet oder schließt es, eine gewöhnliche Datei wird der Playlist hinzugefügt. Mehrmaliges Hinzufügen der gleichen Datei zur Playlist ist natürlich möglich. YAMuPlay kann derzeit jedoch noch nicht unterscheiden, ob es sich bei der gewählten Datei um eine abspielbare Mediendatei oder um einen anderen Dateityp (z. B. eine Textdatei) handelt.

Im oberen Bereich befinden sich Steuerelemente zur Dateisuche. Die Suche berücksichtigt derzeit nur die **Dateinamen**, nicht die Metadaten (z. B. ID3-Tags) der Mediendateien. Dies würde ja eine Datenbank erfordern, auf die aus Performancegründen bewusst verzichtet wurde.

Wird ein USB-Laufwerk entfernt oder ein weiteres angeschlossen, so wird die Baumstruktur entsprechend aktualisiert. Dateieinträge in der Playlist bleiben davon unangetastet!

Der rechte Teil enthält oben die Schaltflächen *Play/Pause*, *seek*, *prev*, *next* und *Stop*, die den Tasten eines CD-Spielers nachempfunden sind und (hoffentlich) keiner Erklärung bedürfen 🟡 Direkt darunter ist ein derzeit noch inaktiver Fortschrittsbalken, der in folgenden Programmversionen die aktuelle Position des laufenden Stücks anzeigen und auch ein Verschieben ermöglichen soll.

Neben den CD-Schaltflächen befindet sich ein Eingabefeld für die Transparenz (den sogenannten alpha-Wert) von Videos, die zwischen 0 (vollständig transparent, d. h. unsichtbar) und 255 (vollständig deckend) liegen kann. Grundsätzlich reagiert der Desktop auch bei deckender Anzeige von Videos auf Maus- bzw. Touchereignisse ganz normal, die Eingabe muss allerdings „blind“ erfolgen. Der Standardwert von 120 für die Transparenz lässt den Desktop des Raspberry Pi und YAMuPlay noch leicht durchscheinen, so dass eine optische Bedienung möglich ist. Der untere Teil wird vollständig von der Playlist ausgefüllt, in der die aktivierten Mediendateien angezeigt werden. Am rechten Rand befinden sich Schaltflächen, um einzelne Dateien in der Liste zu verschieben oder wieder ganz aus der Liste zu entfernen. Ein Doppelklick auf einen Titel in der Playlist springt sofort dorthin und spielt diesen Titel ab.

2.2.3 Menü

DropDown-Menü **Datei**

- **Datei→Playlist öffnen**

Laden einer gespeicherten Playlist:

Es wird der Standarddialog des Betriebssystems zur Dateiauswahl angezeigt, mit dem der Dateiname ausgewählt werden kann. Das Dateiformat ist `m3u`, eine Textdatei, in der jede Zeile eine Mediendatei enthält.

💡 **Hinweis**

Die bestehende Playlist in YAMuPlay wird dabei nicht gelöscht, sondern die in der Playlist-Datei enthaltenen Mediendateien werden hinten angehängt.

- **Datei→Playlist speichern**

Speichern der aktuellen Playlist:

Es wird der Standarddialog des Betriebssystems zur Dateiauswahl angezeigt, mit dem der Dateiname der `m3u`-Datei ausgewählt werden kann. Es werden alle Elemente der aktuellen Playlist von YAMuPlay abgespeichert.

- **Datei→Beenden**

YAMuPlay verlassen

DropDown-Menü **Ansicht**



Hinweis

Dieses Menü beinhaltet Punkte zur Steuerung der Videodarstellung. Für reine Musikdateien wird es nicht benötigt.

- **Ansicht→Vollbild**

Wechsel zwischen der Anzeige eines Videos als Vollbild oder in einem Fenster

- **Ansicht→AspectMode**

Einstellung der Darstellungsanpassung von Videos mit den drei Modes *letterbox*, *fill* und *stretch*:

letterbox:

Vollständige Skalierung in das Videofenster ohne Verzerrung. Es entstehen Ränder an der zu großen Seite.

fill:

Skalierung in das Videofenster auf die kleinere Kante. Zu große Bereiche werden abgeschnitten und sind unsichtbar.

stretch:

Anpassung an die Fenstergröße mit Verzerrung

- **Ansicht→Hintergrundfarbe**

Über den Standarddialog „Farbauswahl“ des Betriebssystems kann eine Hintergrundfarbe für das Videofenster gewählt werden. Bei alpha-Werten kleiner als 255 schimmert die Hintergrundfarbe anteilig durch das Video. Damit können nette(?) Effekte erzeugt werden 🍷

- **Ansicht→Videogröße automatisch anpassen**

Normalerweise wird die Fenstergröße bei Neustart einer Videodatei auf die Originalgröße des Videos angepasst und mittig dargestellt. Da dies aber auch unerwünscht sein kann, werden alle Videos an die vom Anwender eingestellte Größe unter Berücksichtigung des „Aspect Mode“ angepasst.

DropDown-Menü **Hilfe**

- **Hilfe→Info**

Anzeige einer sogenannten „Aboutbox“, in der Informationen über die Software und die Lizenz von YAMuPlay angezeigt werden (siehe Abbildung 2.3)

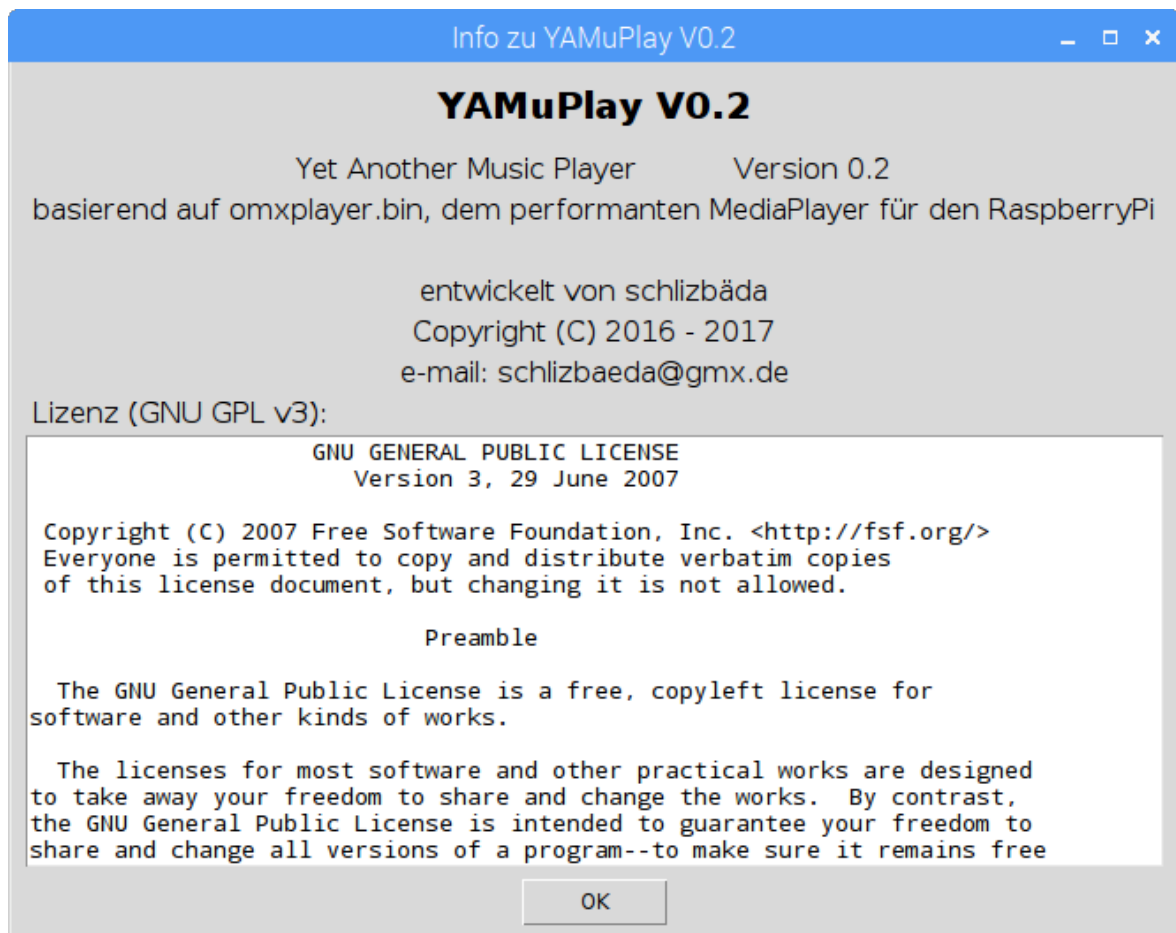


Abbildung 2.3: Aboutbox von YAMuPlay

2.2.4 Bedienung über die Tastatur

Die Bedienung von YAMuPlay ist weitestgehend auch über die Tastatur möglich. Die Taste [TAB] ermöglicht (wie bei den meisten Programmen mit einer GUI) das Wechseln des aktiven Steuerelementes. In der Playlist und im Treeview-Steuerelement und kann mit den Cursortasten eine Datei gewählt werden. Ein Directory kann dabei mit [Cursor links] und [Cursor rechts] auf- oder zugeklappt werden.

Ferner sind einige Funktionstasten wie folgt belegt:

💡 Hinweis

Ein hoher alpha-Wert (wenig Transparenz) kann vor allem bei Vollbildanzeige oder großen Videofenstern die Bedienung einschränken. Durch Drücken von [F9] wird der alpha-Wert auf einen einstellbaren Standardwert (normalerweise 120) gesetzt. Damit ist eine Bedienung der Oberfläche wieder einigermaßen möglich, da die transparente Videodarstellung den Desktop durchscheinen lässt.

- [F1] Anzeige der Aboutbox (Menüpunkt Hilfe→Info)
- [F2] Debugausgabe im Konsolenfenster: `def omxplayerDebugPrint(self):`
- [F9] Transparenz auf Defaultwert setzen (Kommandozeilenparameter `-alpha`)
- [F10] Öffnen des Menüs (offenbar ein internes TKinter-Feature)
- [F11] Wechsel zwischen Videoanzeige im Fenster und Vollbild
- [F12] Wechsel der `aspect modes`: `letterbox`, `fill`, `stretch`

Tabelle 2.3: Funktionstastenbelegung in YAMuPlay

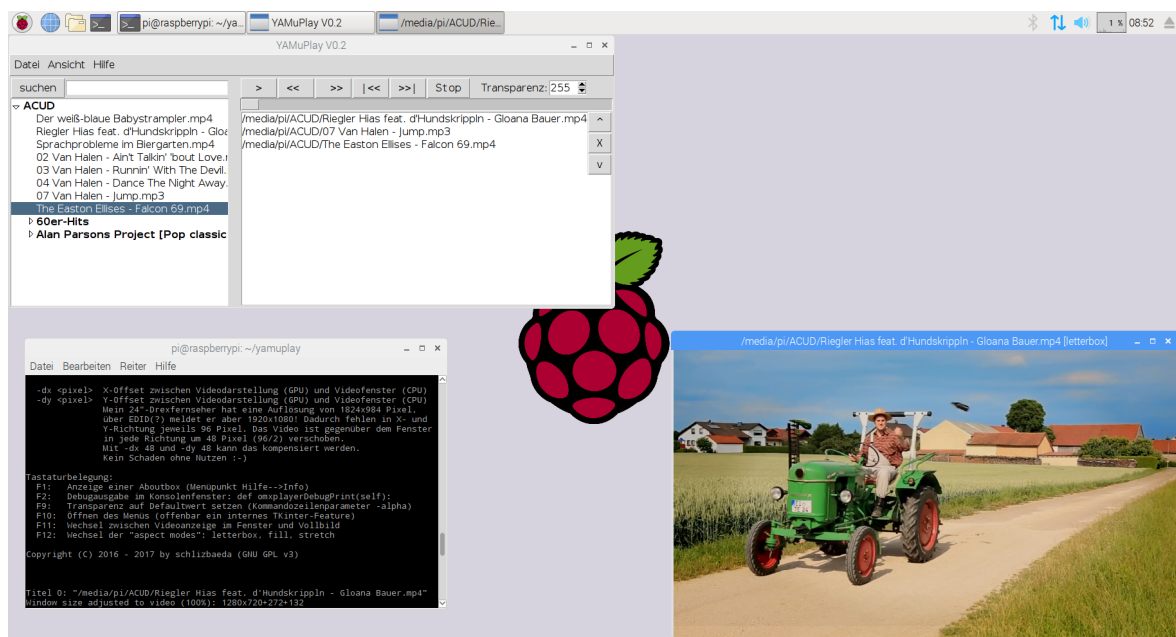


Abbildung 2.4: Video mit YAMuPlay in einem Fenster abspielen

2.2.5 Aufruf im Terminalfenster und Kommandozeilenparameter

Der Start von YAMuPlay erfolgt im Terminal über die Kommandos

```
cd /home/pi/yamuplay
./yamuplay.py [Parameter] [Mediadatei(en)]
```

Nach dem Programmstart wird im Terminalfenster eine kurze Hilfe ausgegeben, in der u. a. die Aufrufparameter beschrieben werden.

<code>-f <bool></code>	0=Videoanzeige im Fenster, 1=Videoanzeige als Vollbild
<code>-a <mode></code>	<i>aspect mode</i> mit den Optionen <i>letterbox</i> , <i>fill</i> und <i>stretch</i>
<code>-k <bool></code>	0=Videofenster an die Größe des Videos anpassen, 1=Größe des Videofensters nicht anpassen
<code>-c <backcol></code>	Hintergrundfarbe des Videofensters festlegen
<code>-alpha <int></code>	alpha-Standardwert für Druck von [F9] festlegen
<code>-dx <pixel></code>	X-Versatz zwischen GPU und CPU
<code>-dy <pixel></code>	Y-Versatz zwischen GPU und CPU

Tabelle 2.4: Kommandozeilenparameter von YAMuPlay

**Hinweis**

Bei Betrieb des Raspberry Pi an einem FullHD-Fernsehgerät mit einer nominalen Auflösung von 1920x1080 anstelle eines Computermonitors ist ein Versatz von `-dx 48` und `-dy 48` zu verwenden, um die Videoausgabe durch die GPU mit dem von der CPU angezeigten Videofenster in Deckung zu bringen. Die *echte* Auflösung eines FullHD-Fernsehers beträgt offenbar nur 1824x984 Pixel.

Im weiteren Betrieb von YAMuPlay dient das Terminalfenster auch zur Ausgabe von Debuginformationen. Mit der Taste [F2] erfolgt die in der Routine `def omxplayerDebugPrint(self):` programmierte Debugausgabe im Terminal. Außerdem machen viele Routinen Debugausgaben der von ihnen ermittelten Werte.

2.3 Erweiterungen und Verbesserungen der Software

Zuletzt noch eine Liste von Punkten, um die das Python3-Script `yamuplay.py` ergänzt werden könnte. Hier handelt es sich um ein *Brainstorming*. Die Reihenfolge soll keine Gewichtung darstellen!

- **FEHLER:**

Das Schließen des Videofensters mit dem roten „X“ führt zu einer Script-Fehlermeldung!

Ein Umschalten von Vollbild in Fensteransicht ist nicht mehr möglich, da das Fenster geschlossen wurde.

→ nicht schließen, sondern minimieren?

- **Tastaturbedienung erweitern**

Durch Drücken von [ENTER] oder [Space] soll der aktuelle Listeneintrag (Treeview, Playlist) aktiviert werden. Identisch zu Doppelklick

- **Scrolling durch Wischgesten wie an einem Smartphone**
horizontal und vertikal: `Treeview.xview` bzw. `Treeview.yview`
- **Scrollbar für Treeview und Playlist**
horizontal und vertikal, da die Schrift der Listeneinträge wegen der vorgesehenen Touchbedienung relativ groß ist.
- **Auf die Gesamtdauer eines Stückes skaliertes „Fortschrittsbalken“**
einfache Verschiebemöglichkeit mit Zeitanzeige wie bei den meisten Mediaplayern. Das Steuerelement existiert bereits, hat aber noch keine Funktion!
- **Anzeige von Titeldaten und aktueller Laufzeit**
wie bei den meisten klassischen CD-Spielern
- **Drag + Drop von Mediendateien aus der Baumstruktur in die Playlist**
Damit hätte man die Möglichkeit, neue Titel irgendwo in der Mitte der bestehenden Playlist einzufügen. Momentan werden alle neuen Titel hinten angehängt.
- **Lautstärke über `omxplayer.bin` einstellen**
WICHTIG:
ALSA funktioniert wegen der Hardwarenähe von `omxplayer.bin` nicht!
- **`omxplayer`-eigenes Fading beseitigen (falls möglich)**
Der `omxplayer.bin` macht ein kurzes Fading (< 1 Sekunde) beim Start einer neuen Musikdatei. Dies ist manchmal wirklich störend!
Komischerweise nicht bei Videodateien
- **PLAYLIST**
 - * bereinigen um nicht mehr vorhandene oder ungültige Dateien
 - * Dateien über Menü auch aus Orten ungleich `/media/pi/...` laden
 - * bei zweitem Aufruf von YAMuPlay die Dateien aus den Kommandozeilenparametern an die Playlist anhängen.
 - * Bei angegebenen Kommandozeilenparametern sofort nach dem Programmstart mit dem Abspielen beginnen
 - * BILDER als Diashow anzeigen (Zeit über Menü/cmdlin einstellbar)
- **Schriftgröße anpassen auf unterschiedliche Displaygrößen**
schwierig:
Schriftgröße vor allem fürs Treeview-Steuerelement parametrierbar machen
- **Erkennung anderer USB-Gerätetypen (Smartphones)**
Derzeit wird nur der USB-Gerätetyp „Mass Storage Device“ unterstützt. Viele neuere Smartphones stellen ihre Daten mitunter nur noch über MTP (Media Transfer Protocol), eine Weiterentwicklung von PTP (Picture Transfer Protocol) zur Verfügung.

- **Dateitypen berücksichtigen**

Derzeit werden alle vorhandenen Dateien angezeigt. Über die Dateieindung oder eine „magische Dateinummer“ am Dateianfang nur die Mediendateien auflisten.

- * Klären, welche Dateitypen von `omxplayer.bin` überhaupt unterstützt werden.

- * Playlists (*.mpu), Bilddateien, Textdateien, und pdf berücksichtigen?

→ Python-Modul `python-magic` wurde für die als Kommandozeilenparameter angegebenen Dateien bereits implementiert. Bei Auswahl in Treeview ebenfalls berücksichtigen...

- **Einbinden der vorhandenen Dateien in die Baumansicht**

Derzeit wird ein USB-Laufwerk, nachdem es erkannt wurde, immer **komplett** (rekursiv) eingelesen! Dies kann bei großen Laufwerken mit vielen Einzeldateien mitunter recht lange dauern! Besser wäre es, nur das gerade geöffnete Verzeichnis **flach** und nicht rekursiv einzulesen. Über dieses Vorgehen werden die Dateien stückweise registriert und der Vorgang dauert nie arg lange.

Zu berücksichtigen ist das aber bei der Dateisuche, da zum Suchzeitpunkt nicht zwingend schon alle Unterverzeichnisse komplett eingelesen wurden!

→ Einlesevorgang bei riesigen USB-Speichern als Hintergrundthread?

- **„Skins“** (richtiges Wort?)

Neben dem GUI-Modus für Tastatur-/Mausbedienung (bzw. Touchdisplay) auch Modus für:

- * Bedienung durch Kinder: grafische Aufbereitung (Metadaten, Zusatzgrafikdateien)

- * Bedienung im Auto:

Bedienung über Taster an GPIOs

KEINE VIDEOS! (bzw. alpha-Wert=0)

Hierbei handelt es sich um die noch nicht abgearbeiteten Punkte aus meiner Schmierzettelsammlung. Diese Liste erhebt aber keinen Anspruch auf Vollständigkeit. 🤪

schlizbaeda