

schlitzbäda

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

YAMuPlay

Mediaplayer YAMuPlay für den Raspberry Pi Modell B+



GNU General Public License v3

© 2016 schlitzbäda

Datum:

20.02.2016

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

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

<http://www.hundskripln.de/#filme>

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Tabellenverzeichnis	IV
1 Einführung	1
1.1 Kurzbeschreibung	1
1.1.1 Software	1
1.1.2 Hardware	2
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	3
1.2.3 Externe Internet-Links	3
1.3 Konventionen dieser Dokumentation	3
1.4 Abkürzungsverzeichnis	4
2 Hardware	5
2.1 Hinweise zum Nachbau der Hardware	5
2.2 Blockschaltbild	6
2.3 Stückliste	7
2.4 Ausblick und mögliche Verbesserungen	8
2.5 Bilder vom Aufbau	9
2.6 Gesamtaufbau der Musikanlage auf dem Faschingswagen	12
3 Software	17
3.1 Beschreibung und Bedienung von YAMuPlay	17
3.2 Bibliotheken und Module	20
3.3 Installation von YAMuPlay auf dem Raspberry Pi	20
3.3.1 Raspbian-Image auf eine 4GB SD-Karte aufspielen und updaten . . .	21
3.3.2 /boot/config.txt anpassen	22
3.3.3 YAMuPlay und alle notwendigen Python-Module installieren	23
3.3.4 Desktop des Raspberry Pi einrichten	24
3.3.5 Touchpanel-Tastatur <i>matchbox-keyboard</i> installieren	26
3.4 Erweiterungen und Verbesserungen der Software	27
3.4.1 Bedienung und grafische Oberfläche	27
3.4.2 Funktionalität von YAMuPlay	28

Abbildungsverzeichnis

2.1	Gefährliche elektrische Spannung	5
2.2	Blockschaltbild	6
2.3	Frontansicht	9
2.4	Innenansicht	10
2.5	Detailansicht Raspberry Pi und Ligawo 6518725 HDMI Extractor	10
2.6	Buchsen für Audio, USB und LAN	11
2.7	Schalter für Spannungsversorgung	11
2.8	Frequenzweiche	12
2.9	Blockschaltbild der Musikanlage	13
2.10	Bedieneinheit der Frequenzweiche	13
2.11	Gesamtanlage auf dem Faschingswagen	14
2.12	18. Gaudiwurm Tattenhausen 31.01.2016 Aufstellung	15
2.13	18. Gaudiwurm Tattenhausen 31.01.2016 Seitenansicht	15
2.14	Faschingszug Bad Aibling 07.02.2016	16
3.1	Hauptfenster von YAMuPlay mit geöffnetem Hauptmenü	18
3.2	Dateidialog von YAMuPlay zum Laden einer vorhandenen Playlist	19
3.3	About-Box von YAMuPlay	19
3.4	Ausgaben des Kommandos <code>tvservice</code>	22
3.5	Änderungen in der Datei <code>/boot/config.txt</code>	23
3.6	Neuer Eintrag im Startmenü des Raspberry Pi	25
3.7	Neues Icon auf dem Desktop des Raspberry Pi	25
3.8	Bildschirmschoner in Datei <code>/etc/lightdm/lightdm.conf</code> deaktivieren	26
3.9	Änderungen in der Datei <code>/home/pi/.config/lxpanel/LXDE-pi/panels/panel</code>	27

Tabellenverzeichnis

1.1	Konventionen der Dokumentation	3
2.1	Stückliste	7
3.1	In YAMuPlay eingebundene Bibliotheken und Module	20

1 Einführung

1.1 Kurzbeschreibung

1.1.1 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 von **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* 🍉 (<mailto:schlizbaeda@gmx.de>).

Die Software **YAMuPlay** in Version V0.1 besteht aus einer grafisch einfach gehaltenen, mit dem Python-Modul **tkinter** erstellten Benutzeroberfläche (GUI) ähnlich einem Dateimanager: Im linken Teil befindet sich ein sogenanntes Treeview-Steuerelement, in dem die hierarchische Ordnerstruktur von Laufwerken angezeigt wird, die unter **/media** gemountet sind. Am Raspberry Pi neu angeschlossene USB-Laufwerke werden automatisch erkannt und entsprechend in die Ordnerstruktur eingebunden. Musikdateien können durch einen Doppelklick in die Playlist eingetragen werden.

Der rechte Teil besteht aus der Playlist und Steuerelementen zum Entfernen und Verschieben der Musiktitel innerhalb der Playlist. Im unteren Teil befinden sich die Steuerelemente zum Abspielen der Musiktitel, wie man sie vom CD-Spieler kennt, außerdem ist noch ein Eingabefeld für die Titelsuche enthalten.

1.1.2 Hardware

Grundsätzlich ist für den Betrieb von **YAMuPlay** nur ein Raspberry Pi erforderlich, an den ein HDMI-Bildschirm mit Lautsprechern angeschlossen wird. Für die Bedienung benötigt man ferner Tastatur und Maus.

Der erste konkrete Anwendungsfall für **YAMuPlay** war jedoch ein Musikspieler auf dem Faschingswagen 2016 mit dem Thema „500 Jahre Reinheitsgebot“ von *Bauwong n.e.V.*, einer losen Gruppe von Freunden in Großkarolinenfeld. In der weisen Voraussicht, dass es auf dem Wagen während der Teilnahme an den Faschingsumzügen mitunter auch etwas grob zugehen könnte, beschloss ich, anstatt meines Laptops einen Aufbau mit robusteren und gleichzeitig kostengünstigeren Komponenten umzusetzen, die den mechanischen Anforderungen standhalten sollten:

Ein **SIMATIC Industrial Flat Panel** (19 Zoll) aus dem Hause Siemens – nicht wirklich billig, aber dank Vitamin B bin ich daran kostenlos herangekommen 🍋: Widerstandsfähig genug, um den Anforderungen im rauen Industrieinsatz zu genügen, sollte es auch den Einsatz auf dem Faschingswagen aushalten.

Auf dem rückseitigen Deckel des Panels wurden ein **Raspberry Pi Modell B+** und ein **Ligawo 6518725 HDMI Extractor** montiert. Letzterer ist notwendig, um aus dem HDMI-Anschluss ein analoges Audiosignal abzugreifen. Der analoge Audioausgang des Raspberry Pi ist nämlich echt besch...eiden 🍋. Das Ligawoteil ist mal Chinaware, die wirklich zu gebrauchen ist: Das erzeugte Audiosignal zeigt keinerlei Qualitätseinbußen. Selbst bei größerer Lautstärke an relativ guten Verstärkern und Lautsprechern, wo schlechte Tonqualität schnell auffällt, konnte ich als relativ anspruchsvoller HiFi-Fan keine Abstriche erkennen. 🍋

Das Ganze wurde in ein zugegebenermaßen rustikales Holzgehäuse eingebaut (siehe Bilder), Netzteile (24V und 5V) rein, 230V-, USB- und Ethernet-Buchsen sowie drei Schalter ins Holzgehäuse gebaut, das Zeug noch intern verkabelt und fertig.

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. Die Software darf frei kopiert und ohne Leistung eines Entgelts privat oder kommerziell verwendet werden. Bitte lesen Sie die GPL v3, die sich im Programmpaket in der Datei `bauwong.lic` befindet oder im Internet unter <https://www.gnu.org/licenses/gpl-3.0> abgerufen werden kann.

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 Links zu verschiedenen externen Seiten im Internet. Den Inhalt mache ich mir nicht zu eigen, da ich auf die Gestaltung dieser Seiten keinerlei Einfluss habe. Zum Zeitpunkt der Verlinkung enthielt keine Seite illegale Inhalte. Dies kann ich aber nur in größeren Abständen kontrollieren. Sollten Sie an einem Link etwas auszusetzen haben, so senden Sie bitte eine e-mail an <mailto:schlizbaeda@gmx.de>.

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
<i>"Menüpunkt"</i>	Kennzeichnung von Menüpunkten der Software YAMuPlay
<i>Meldung</i>	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
DVI	D igital V isual I nterface
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
USV	U nterbrechnungsfreie S tromversorgung

2 Hardware

2.1 Hinweise zum Nachbau der Hardware

Dieses Kapitel der Dokumentation beschreibt die für meine Anwendung von YAMuPlay aufgebaute Hardware, ein Touchpanel mit einem Raspberry Pi Modell B+ und allen notwendigen Komponenten, die zur Verwendung als Audioquelle in der Musikanlage unseres Faschingswagens notwendig waren.



Abbildung 2.1: *Gefährliche elektrische Spannung*

Für den beschriebenen Aufbau wurde eine Spannungsversorgung über 12V-Bleiakkus verwendet (große LKW-Batterien). Über einen 12V/230V-Spannungswandler wird die Betriebsspannung für die im RPi-Touchpanel verbauten Netzteile erzeugt.



Vorsicht!

Ich appelliere an die Vernunft der Leserschaft, den Nachbau nur vorzunehmen, wenn man sich folgender Gefahren bewusst ist und das fachliche Wissen und handwerkliche Können besitzt, sie zu vermeiden:

- ausgehende Gefahr von einer 230V-Wechselspannung (gefährlicher Stromschlag)
- ausgehende Gefahr von hohen Strömen aus Autobatterien (Kurzschlussfall!)

Lizenz:

Der hier beschriebene elektrische Aufbau erreicht nicht die schöpferische Tiefe, um über eine Lizenz geschützt werden zu müssen. Vielmehr beruht er auf den allgemein bekannten Fakten für die Inbetriebnahme eines Raspberry Pi sowie auf den Bedienungsanleitungen der verwendeten Einzelkomponenten.

2.2 Blockschaltbild

Eine kurze funktionale Beschreibung des Aufbaus ist bereits in Kapitel 1.1.2 erfolgt. Aufgrund einer äußerst kurzfristigen Änderung der Rahmenbedingungen musste die Stromversorgung neu gestaltet werden. Daher wurde (gezwungenermaßen aus Zeit- und Materialmangel) auf eine saubere Netzabschaltung verzichtet. Die Spannungsversorgung der Einzelkomponenten (Siemens IFP, Ligawo 6518725 HDMI Extractor, Raspberry Pi) kann jedoch über einzelne Schalter getrennt zugeschaltet werden, um die Komponenten in einer definierten Reihenfolge in Betrieb zu nehmen. Siehe dazu auch Kapitel 2.4, Abschnitt „Power Sequencer“

Für einen vollständigen Überblick hier ein Blockschaltbild:

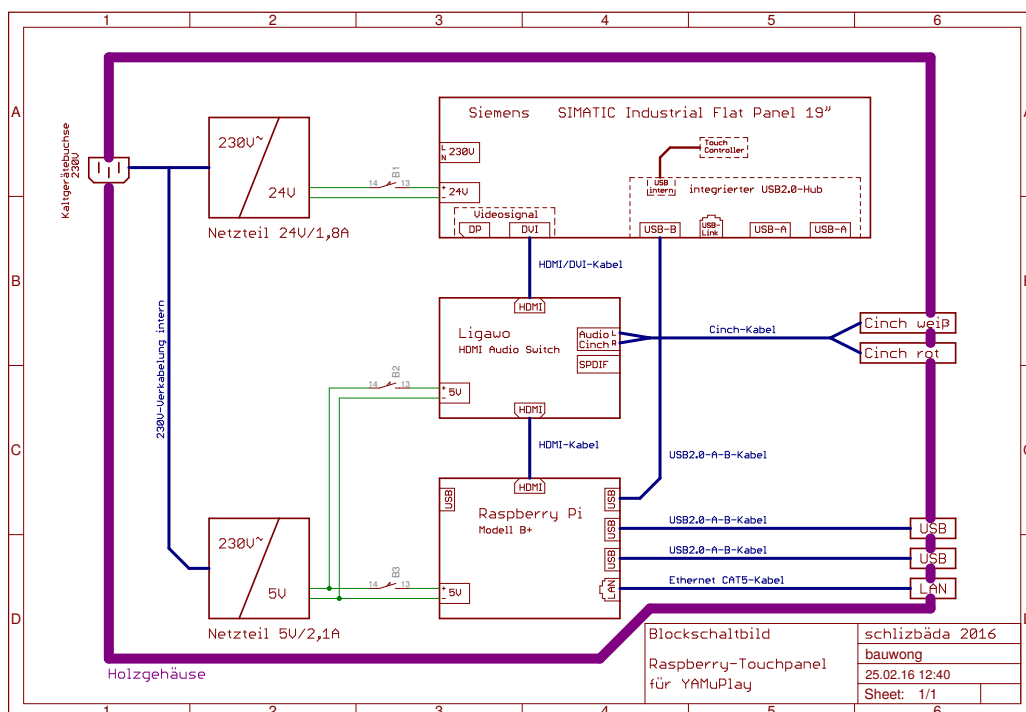


Abbildung 2.2: Blockschaltbild

2.3 Stückliste

Folgende Komponenten wurden für den elektrischen Aufbau verwendet. Das Material für den Holzrahmen (Holz, Schrauben, Montagewinkel) ist in der Stückliste nicht aufgeführt.

Anz.	Lieferant	Bestellnummer	Artikel
1	Vitamin B 🍌	—	Siemens IFP 19"MultiTouch
1	Vitamin B 🍌	—	Deutronic Einbaunetzteil 24V / 1.8A
1	amazon	Ligawo 6518725	Ligawo 6518725 HDMI Extractor
1	www.raspiprojekt.de	35001	Raspberry Pi Modell B+
1	reichelt	GOO 44009	5V-USB-Netzteil, 2A (USB-Ladeadapter)
1	reichelt	KES 1	Kaltgeräteeinbaustecker, waagrecht befestigt
3	reichelt	WIPPE 1801.1146	Marquardt Wippschalter 1-pol
1	reichelt	NEUTRIK NF2DB-2	Cincheinbaubuchse, Farbkennring rot
1	reichelt	NEUTRIK NF2DB-9	Cincheinbaubuchse, Farbkennring weiß
1	reichelt	NEUTRIK NE-8FDP	Flanschbuchse RJ45 auf RJ45
2	reichelt	NEUTRIK NAUSB-WB	USB-Einbaudurchgangsbuchse, innen USB-B, außen USB-A
1	reichelt	AK HDMI-DVI 1,0	HDMI-Stecker auf DVI-D Stecker 1,0m
1	reichelt	AK HDMI 0,50G ET	HDMI Kabel Stecker/Stecker 0,5m
1	reichelt	AVK 128	2x Cinchstecker auf 2x Cinchstecker 1,5m
3	reichelt	AK 672/HSF-0,5	USB2.0-Kabel, A-Stecker auf B-Stecker 0,5m
1	reichelt	PATCH-C7 05 MA	0,5m Cat.7 PiMF-Patchkabel, magenta, RJ45

Tabelle 2.1: *Stückliste*

2.4 Ausblick und mögliche Verbesserungen

Die hier beschriebene Umsetzung ist lediglich als Anregung zu verstehen: Es wurde hier ganz bewusst auf eine massive Bauweise gesetzt und hinsichtlich der mechanischen Beanspruchung für die geplante Anwendung (Faschingswagen!) auf robuste Komponenten wie ein industrielles Touchpanel Wert gelegt. Das Innenleben hinter dem Touchpanel im Holzrahmen konnte dann entsprechend filigraner umgesetzt werden.

Je nach Verwendung und Einsatzort eignet sich allerdings jedes beliebige andere Display mit HDMI-Anschluss mit einer Auflösung von mindestens 1366x768 Pixeln. Anstelle der Touchfunktionalität können auch eine ganz normale Tastatur und Maus benutzt werden.

Nachdem zunächst geplant war, eine 230V-Stromversorgung über ein am Bulldog angeschlossenes Zapfwellenaggregat herzustellen und mit großen Verstärkern zu arbeiten, musste ich dann feststellen, dass die Drehrichtung der Zapfwelle vorne genau falsch herum war. Da das Aggregat Drehstrom erzeugt, muss auch die Drehrichtung stimmen: Man denke an eine Drehstrom-Kreissäge, deren Sägeblatt sich falsch herum dreht... Saugefährlich! Daher schaltete bei unserem Aggregat die Sicherungseinrichtung ab. 🤨 Ein Anschluss des Aggregates an der Hinterseite des Bulldogs fiel aus, da ja dort der Faschingswagen hängt und das Aggregat der Deichsel im Wege ist. 🤨

Gott sei Dank haben wir das Ganze eine Woche vor dem ersten Einsatz geprüft und festgestellt, dass es nicht geht. Daher wurde ein 12V/230V-Sinus-Spannungswandler mit 1500W Dauerleistung bestellt, aber leider billigste Chinaware bei dem großen Online-Versand, der in Deutschland keine Steuern bezahlt. Ausgepackt, angeschlossen, geht nicht! 🤨 Das Glump daraufhin sofort wieder zurückgeschickt.

Also: Umbau am Tag vor dem ersten Faschingszug auf die althergebrachte Weise mit Autoendstufen und einem kleinen Spannungswandler mit (angeblich) 300 Watt. Deshalb auch die für manchen Leser wohl relativ chaotische und undurchdachte Spannungsversorgung im RPi-Touchpanel. Anschließend noch drei Endstufen mit Frequenzweiche in den Faschingswagen gepaxt, um halb sieben abends dann eingeschaltet, ausgepegelt und eine Halbe Bier aufgemacht. 🤨



Hinweis

Wenn die folgenden Punkte umgesetzt würden, könnte der ganze Aufbau sogar noch gut werden!

- **Saubere 230V-Versorgung mit Hauptschalter**
selbstredend!
- **Power Sequencer**

Derzeit tritt das Problem auf, dass *manchmal* nach dem Start des Raspberry Pi kein Ton ausgegeben wird, obwohl eigentlich alle Komponenten eingeschaltet sind. Ich vermute

die Ursache im Ligawo 6518725 HDMI Extractor. Im deutschsprachigen RaspberryPi-Forum schildert ein Anwender nämlich ein ähnliches Problem:

<http://www.forum-raspberrypi.de/Thread-problem-mit-hdmi-audio-extractor-ligawo?highlight=Ligawo+Audio>

Dies kann *meistens* umgangen werden, indem zuerst der HDMI-Bildschirm, dann der Ligawo 6518725 HDMI Extractor und zuletzt der Raspberry Pi und alles im richtigen Zeitabstand eingeschaltet wird. *So ganz sicher bin ich mir da allerdings nicht, da selbst ich schon drei aufeinanderfolgende fehlgeschlagene Boots erlebt habe.* 🤖

Möglicherweise könnte diese Einschaltreihenfolge mit einem Mikrocontroller automatisiert werden.

- **Sauberes Herunterfahren / USV**

Die obige Mikrocontrollerschaltung könnte so erweitert werden, dass beim Ausschalten die Stromversorgung zum Raspberry Pi erst dann getrennt wird, wenn dieser vollständig heruntergefahren ist. Alternativ wäre der Einbau einer USV für den Raspberry Pi noch eine Option, siehe <http://www.piusv.de>

- **Raspberry Pi Modell A oder Pi Zero verwenden** Wenn das verwendete Display einen eingebauten USB-Hub hat (so wie mein Siemens-Panel), kann auch ein Raspberry Pi-Modell mit nur einer USB-Buchse verwendet werden. Man stellt eine USB-Verbindung vom Raspberry Pi zum USB-Hub des Displays her und verbindet dessen Ausgangs-Ports mit den im Gehäuse verbauten USB-Buchsen. Da bringe ich dann endlich meinen Raspberry Pi Modell A+ sinnvoll unter. 🤖

2.5 Bilder vom Aufbau



Abbildung 2.3: Frontansicht

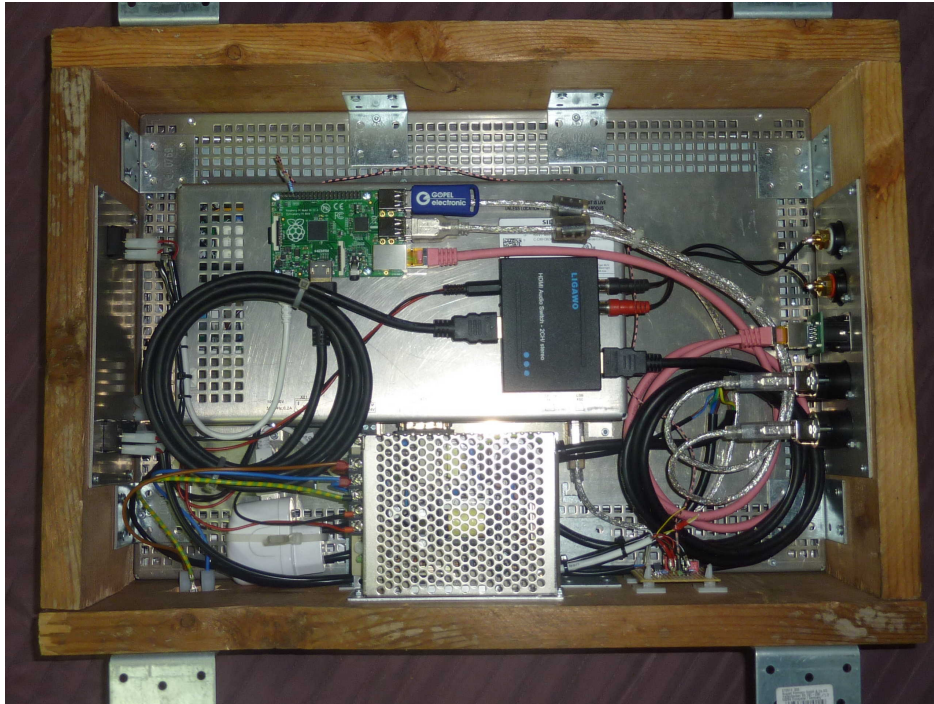


Abbildung 2.4: Innenansicht

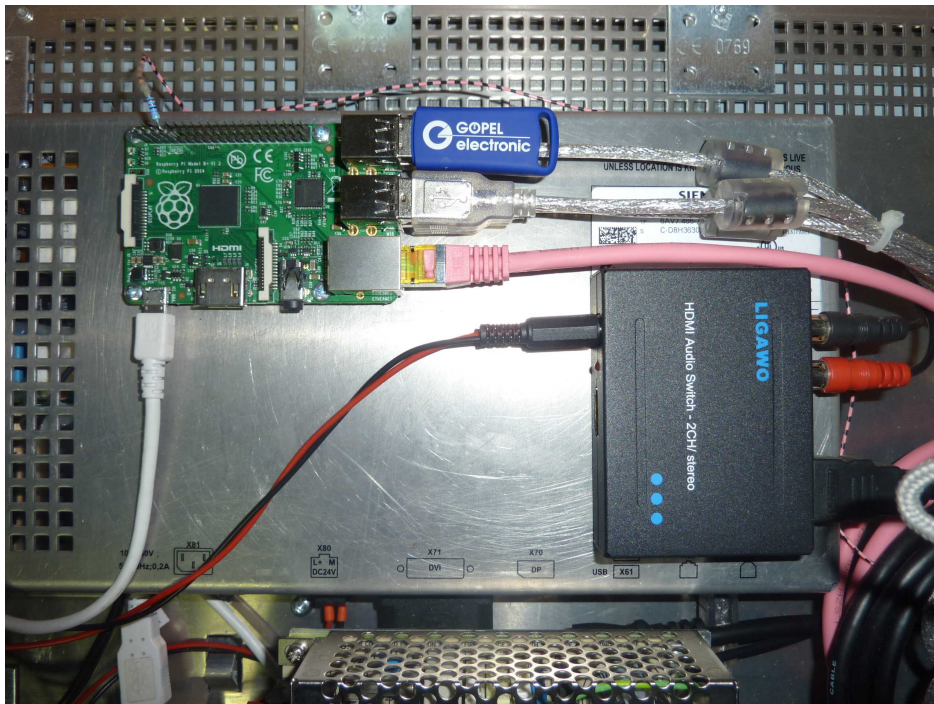


Abbildung 2.5: Detailansicht Raspberry Pi und Ligawo 6518725 HDMI Extractor



Abbildung 2.6: Buchsen für Audio, USB und LAN

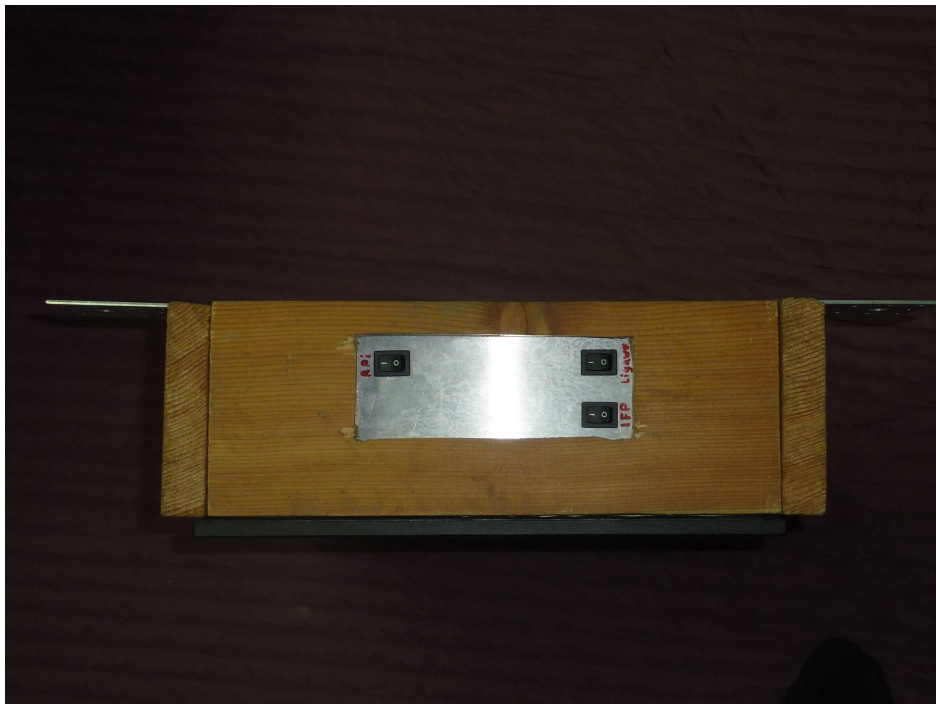


Abbildung 2.7: Schalter für Spannungsversorgung

2.6 Gesamtaufbau der Musikanlage auf dem Faschingswagen

Dieser Abschnitt ist etwas „off topic“, denn er behandelt keinerlei Themen bezüglich des Raspberry Pi. Vielmehr wird hier kurz der Gesamtaufbau der Musikanlage bestehend aus dem vorgestellten RPi-Touchpanel und klassischen Car-Hifi-Komponenten beschrieben. Wer sich dafür nicht interessiert, kann diesen Teil bedenkenlos überspringen...

Auf dem Faschingswagen wurde im Prinzip eine Car-Hifi-Anlage aufgebaut, deren Lautstärke und Klang jedoch nicht von einem normalen Autoradio (head unit), sondern mit Hilfe der Frequenzweiche Sony XEC-505 gesteuert wurde. Aufgrund der Verwendung der Software `omxplayer.bin` auf dem Raspberry Pi greift dort der ALSA-Mixer nicht. Eine Lautstärkeregelung war deshalb softwaretechnisch nicht so leicht zu lösen. *Auch hier lasse ich mich gerne eines Besseren belehren!* 🍀 (<mailto:schlizbaeda@gmx.de>) Vorteilhaft an dieser Lösung war jedenfalls, dass alle Lautsprecherpaare einzeln geregelt und ausgepegelt werden konnten. Die Funktion der Sony-Frequenzweiche ist auf deren Gehäuse als Blockschaltbild dargestellt und zwar dermaßen gut, dass dieses Bild mehr sagt als 1000 Worte, siehe Abbildung 2.8.

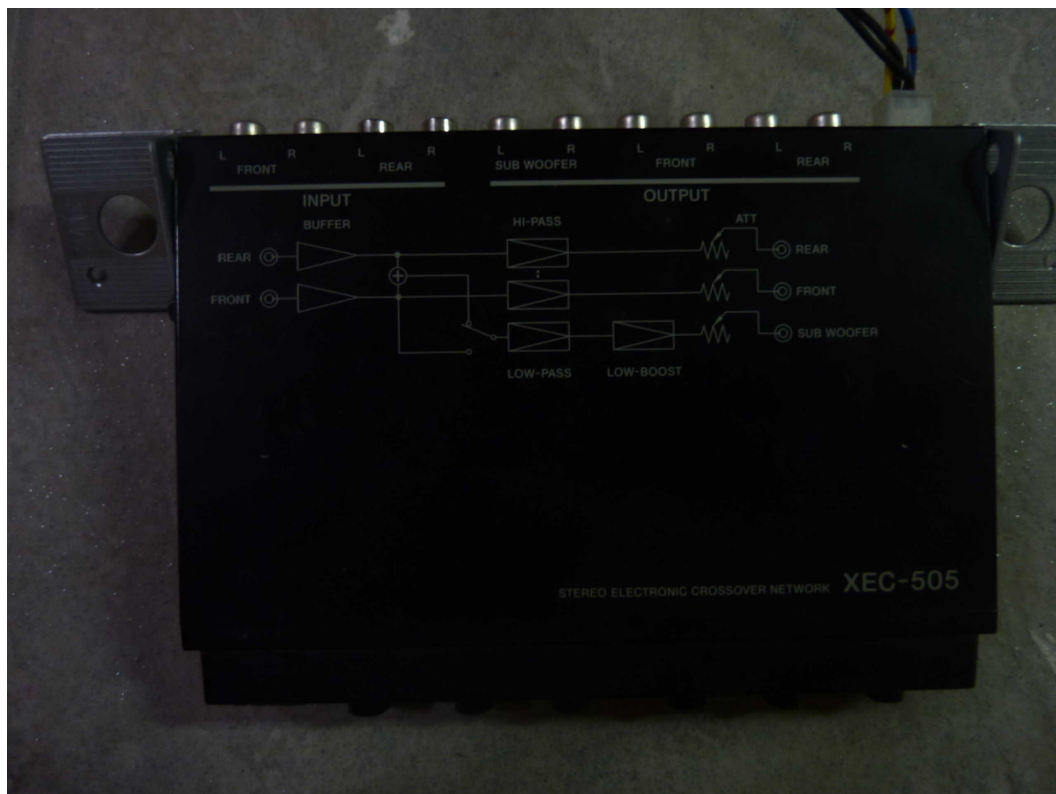
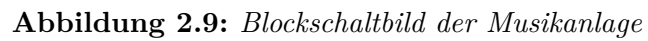


Abbildung 2.8: Frequenzweiche

Aus dieser Skizze ergibt sich die Verkabelung der gesamten Musikanlage auf Abbildung 2.9. Es ist dann – Achtung: Unwort! – *nur* noch Fleißarbeit, das Ganze auf dem Faschingswagen aufzubauen.



Zuletzt noch einige Bilder vom Faschingswagen:



Abbildung 2.11: Gesamtanlage auf dem Faschingswagen



Abbildung 2.12: 18. Gaudiwurm Tattenhausen 31.01.2016 Aufstellung



Abbildung 2.13: 18. Gaudiwurm Tattenhausen 31.01.2016 Seitenansicht



Abbildung 2.14: *Faschingszug Bad Aibling 07.02.2016*

3 Software

3.1 Beschreibung und Bedienung von YAMuPlay

Die Software YAMuPlay V0.1 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-Programms im Hintergrund statt; in Python können währenddessen andere Aufgaben erledigt werden. Ein weiterer Vorteil liegt darin, dass `omxplayer.bin` hauptsächlich den GPU-Teil des Broadcom 2835 auf dem Raspberry Pi beansprucht und somit den CPU-Teil kaum belastet. Die CPU-Last beim Betrieb von YAMuPlay liegt bei 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!

Es gibt bereits genügend kompliziert zu bedienende Mediaplayer vor allem hinsichtlich der Erstellung, Änderung und Verwaltung von Playlists. Bereits in Kapitel 1.1.1 in der Einführung habe ich die Problematik des Music Player Daemons angedeutet. So etwas ist nicht zu gebrauchen, wenn Änderungen *schnell* vorgenommen werden sollen/müssen. Vielmehr ist eine intuitive Bedieneroberfläche erforderlich. YAMuPlay ist ein Versuch in diese Richtung, aber auch hier kann diesbezüglich noch viel getan werden, das will ich gar nicht abstreiten. Mehrere Punkte dazu sind in Kapitel 3.4 aufgeführt.

Das Hauptfenster von YAMuPlay ist zweigeteilt, siehe Abbildung 3.1. 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,

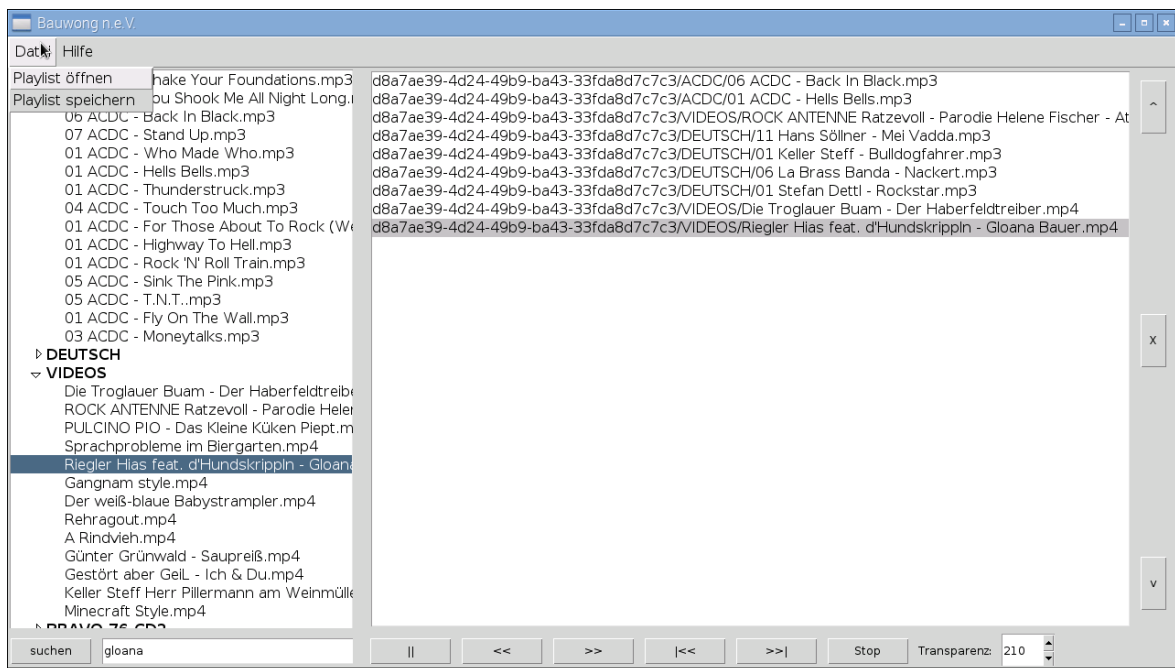


Abbildung 3.1: Hauptfenster von YAMuPlay mit geöffnetem Hauptmenü

ob es sich bei der gewählten Datei um eine abspielbare Mediendatei oder um einen anderen Dateityp (z. B. eine Textdatei) handelt.

Im unteren 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 die Playlist mit den aktivierten Mediendateien. Am rechten Rand befinden sich Schaltflächen, um einzelne Dateien in der Liste zu verschieben oder wieder ganz aus der Liste zu entfernen. Unterhalb der Playlist sind die Schaltflächen *Play/Pause*, *seek*, *prev*, *next* und *Stop*, die den Tasten eines CD-Spielers nachempfunden sind. Ein Doppelklick auf einen Titel in der Playlist springt sofort dorthin und spielt diesen Titel ab.

Zusätzlich ist unten 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 210 für die Transparenz lässt den Desktop des Raspberry Pi und YAMuPlay noch leicht durchscheinen, so dass eine Bedienung möglich ist.

Über das Menü von YAMuPlay können in der Leiste **"Datei"** Playlists geladen und gespeichert werden. Es wird der Standarddialog des Betriebssystems zur Dateiauswahl angezeigt, mit dem der Dateiname festgelegt werden kann. Das Dateiformat ist **m3u**, eine Textdatei, in

der jede Zeile eine Mediendatei enthält. Beim Abspeichern einer Playlist werden immer **alle** Elemente der Playlist in die Datei geschrieben, beim Laden werden alle Einträge der Datei an die aktuell bestehende Playlist **hinten angehängt**.

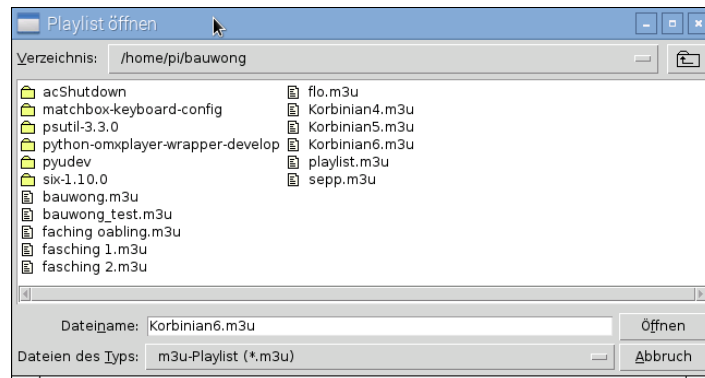


Abbildung 3.2: Dateidialog von YAMuPlay zum Laden einer vorhandenen Playlist

Der Menüpunkt **"HilfeInfo"** zeigt eine sogenannte „About-Box“ an, in der Informationen über die Software und die Lizenz angezeigt werden, siehe 3.3.

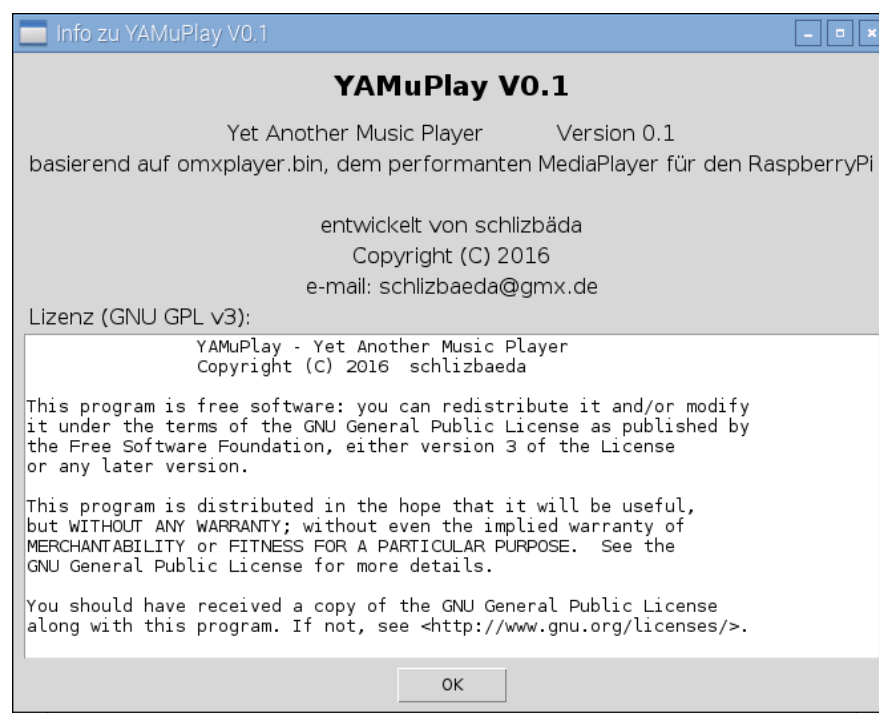


Abbildung 3.3: About-Box von YAMuPlay

💡 Hinweis

Da am Ende die Zeit davonlief (ich weiß, eine schlechte Ausrede 🤪), wurde die gesamte grafische Oberfläche von YAMuPlay für eine Displaygröße von 1366x768 Pixeln „optimiert“, d.h. es wird von einer Fensterhöhe von ca. 720 Pixeln ausgegangen. Die Breite ist unkritischer.

3.2 Bibliotheken und Module

Zur Installation von YAMuPlay benötigt man folgende Bibliotheken und Module:

Modul	Version	Lizenz	Quelle
python3-pip	aktuell(?)	MIT-Lizenz	apt-get install python3-pip
python-omxplayer-wrapper	0.0.2	LGPL v3	https://github.com/willprice/python-omxplayer-wrapper
python3-dbus	1.2.2-1	MIT-Lizenz	apt-get install python3-dbus
pyudev	0.18	LGPL v2.1	https://github.com/pyudev
six	1.10.0	MIT-Lizenz	https://pypi.python.org/pypi/six
matchbox-keyboard	aktuell(?)	LGPL v...	https://github.com/mwilliams03/matchbox-keyboard.git

Tabelle 3.1: In YAMuPlay eingebundene Bibliotheken und Module

YAMuPlay selbst benötigt für seinen Betrieb nur die Module `python-omxplayer-wrapper`, `python3-dbus`, `pyudev` und `six`. Das Modul `python3-pip` muss auf dem Raspberry Pi installiert werden, damit die vier genannten Python-Module wiederum richtig installiert werden können.

Die Software `matchbox-keyboard` ist nur dann notwendig, wenn keine reale Tastatur verwendet werden soll, so wie auf dem Faschingswagen.

3.3 Installation von YAMuPlay auf dem Raspberry Pi

Dieser Abschnitt beschreibt die **vollständige** Installation von YAMuPlay auf einem jungfräulichen Raspbian-System mit dem Ausgabestand *wheezy* vom 05.05.2015, Imagedatei `2015-05-05-raspbian-wheezy.img`.

Ich habe die Beschreibung möglichst detailliert gehalten, damit alle Leser die Sache durch- und nachvollziehen können, egal auf welchem Erfahrungsstand sie sich befinden. An den beschriebenen Punkten hatte ich anfangs als grimmiger Linux- und Python-Noob selbst meine Probleme!

Grundsätzlich kann diese Installationsanleitung in fünf Bereiche aufgliedert werden:

- Raspbian-Image auf eine SD-Karte (4GB oder größer) aufspielen und updaten
- `/boot/config.txt` anpassen

- YAMuPlay und alle notwendigen Python-Module installieren
- Desktop des Raspberry Pi einrichten
- Touchpanel-Tastatur *matchbox-keyboard* installieren



Hinweis

Für die Installation muss der Raspberry Pi zwingend einen Internetzugang erhalten. Zudem sollten zumindest am Anfang Tastatur und Maus angeschlossen sein. Die Interaktion mit dem Raspberry Pi kann im späteren Verlauf der Inbetriebnahme wahlweise auch über `ssh` erfolgen.

3.3.1 Raspbian-Image auf eine 4GB SD-Karte aufspielen und updaten

Die gesamte Installation wurde auf *raspbian wheezy*, Ausgabestand 05.05.2015 durchgeführt. Mittlerweile ist die Nachfolgeversion *jessie* veröffentlicht, mit der die Installation ebenso funktionieren sollte.

In dieser Anleitung wird davon ausgegangen, dass ein aktuelles Raspbian-Image auf die SD-Karte aufgespielt wurde und das allererste, automatisch startende `raspi-config` bereits abgeschlossen wurde. Ferner ist vom Raspberry Pi aus eine Internetverbindung verfügbar. **Wenn alle diese Bedingungen erfüllt sind, kann ab jetzt auch mit `ssh` gearbeitet werden.**

- **raspi-config ausführen**

```
sudo raspi-config
```

Menüpunkt 1 **Expand Filesystem** ausführen, um Platz für die Backups während der System-Updates zu schaffen

Spracheinstellungen im Menüpunkt 4 **Internationalisation Options** durchführen

- **Raspbian-Update ohne Kernel und Broadcom-Firmware durchführen**

```
sudo apt-get update
```

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

- **Kernel- und Broadcom-Firmware-Update durchführen**

`uname -a` # liefert die aktuelle Kernel-Version, z. B. :

```
Linux raspberrypi 3.18.11+ 781 PREEMPT Tue Apr 21 18:02:18 BST 2015 armv6l
GNU/Linux
```

`vcgencmd version` # liefert die aktuelle Firmware-Version, z. B. :

```
Apr 21 2015 14:42:19
```

```
Copyright (c) 2012 Broadcom
```

```
version 2d5ad04b63af4233440c3f7c8587108223201102 (clean) (release)
```

Update durchführen:

```
sudo apt-get install rpi-update
```

```
sudo rpi-update
```

```
sudo reboot
```

Update kontrollieren:

```
uname -a
```

```
Linux raspberrypi 4.1.18+ 845 Thu Feb 18 19:37:13 GMT 2016 armv6l GNU/Linux
```

```
vcgencmd version
```

```
Feb 25 2016 18:51:26
```

```
Copyright (c) 2012 Broadcom
```

```
version dea971b793dd6cf89133ede5a8362eb77e4f4ade (clean) (release)
```

3.3.2 /boot/config.txt anpassen

Bei Verwendung der Hardwarekomponente **Ligawo 6518725 HDMI Extractor** wird empfohlen, die HDMI-Einstellungen in der Datei `/boot/config.txt` anzupassen.

- **Unterstützte Auflösungen des angeschlossenen Bildschirms ermitteln**

`tvservice --modes=CEA` # ermittelt die unterstützten Modi für die HDMI-Gruppe CEA (Consumer Electronics Association)

`tvservice --modes=DMT` # ermittelt die unterstützten Modi für die HDMI-Gruppe DMT (Display Monitor Timing)

```

pi@raspberrypi: ~
Datei Bearbeiten Reiter Hilfe
pi@raspberrypi ~ $ tvservice --modes=CEA
Group CEA has 6 modes:
  (native) mode 1: 640x480 @ 60Hz 4:3, clock:25MHz progressive
            mode 2: 720x480 @ 60Hz 4:3, clock:27MHz progressive
            mode 3: 720x480 @ 60Hz 16:9, clock:27MHz progressive
            mode 4: 1280x720 @ 60Hz 16:9, clock:74MHz progressive
            mode 17: 720x576 @ 50Hz 4:3, clock:27MHz progressive
            mode 18: 720x576 @ 50Hz 16:9, clock:27MHz progressive
pi@raspberrypi ~ $ tvservice --modes=DMT
Group DMT has 7 modes:
            mode 4: 640x480 @ 60Hz 4:3, clock:25MHz progressive
            mode 6: 640x480 @ 75Hz 4:3, clock:31MHz progressive
            mode 9: 800x600 @ 60Hz 4:3, clock:40MHz progressive
            mode 16: 1024x768 @ 60Hz 4:3, clock:65MHz progressive
            mode 36: 1280x1024 @ 75Hz 5:4, clock:135MHz progressive
  (prefer) mode 81: 1366x768 @ 60Hz 16:9, clock:85MHz progressive
            mode 85: 1280x720 @ 60Hz 16:9, clock:74MHz progressive
pi@raspberrypi ~ $

```

Abbildung 3.4: Ausgaben des Kommandos `tvservice`

Den beiden Ausgaben kann man die optimale Auflösung entnehmen: Im Beispiel aus Abbildung 3.4 handelt es sich um den Modus 81 aus der HDMI-Gruppe DMT

- **Editieren der Datei `/boot/config.txt`**

Eintrag `hdmi_group=2` # CEA = Gruppe 1 / DMT = Gruppe 2

Eintrag `hdmi_mode=81` # 1366x768 Pixel

Eintrag `hdmi_drive=2` # Audio-Signal über HDMI erzwingen

In Abbildung 3.5 sind die Anpassungen für das verwendete 19-Zoll Siemens Industrial Flat Panel gezeigt.

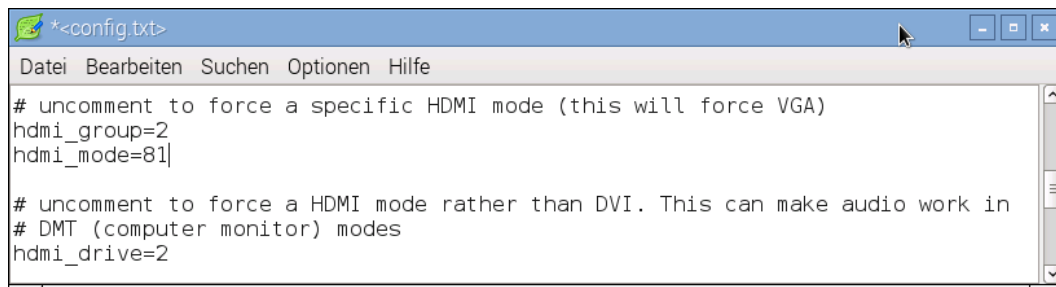


Abbildung 3.5: Änderungen in der Datei `/boot/config.txt`

- **raspi-config ausführen**

```
sudo raspi-config
```

Menüpunkt 8 **Advanced Options** ausführen

Es öffnet sich ein Untermenü:

Menüpunkt A9 **Audio** öffnen und Option 2 **Force HDMI** wählen

3.3.3 YAMuPlay und alle notwendigen Python-Module installieren

- **Package Installer für Python3 installieren**

```
sudo apt-get install python3-pip
```

Es wird das pip-Installationsprogramm für Python3-Module installiert.

- **GitHub-Download des Python-Moduls `python-omxplayer-wrapper`**

```
cd /home/pi
```

Die Original-Software wurde für Python2.x erstellt (<https://github.com/willprice/python-omxplayer-wrapper>). Um es für Python3 anzupassen habe ich auf GitHub einen sogenannten *Fork* angelegt, eine Projektabspaltung. Download und Entpacken in das Unterverzeichnis `python-omxplayer-wrapper` mit

```
git clone https://github.com/schlizbaeda/python-omxplayer-wrapper.git
cd python-omxplayer-wrapper
```

Das folgende Kommando für den *Package Installer für Python3* übernimmt die aktuelle Version des Moduls in die Python3-Bibliothek. Dieser Aufruf ist immer notwendig, wenn Änderungen an diesem Modul gemacht wurden!

```
sudo python3 setup.py install
sudo apt-get install python3-dbus # notwendig für die Dbus-Kommunikation des
Wrappers mit omxplayer.bin
```

- **GitHub-Download des Python-Moduls pyudev**

Dieses Modul wird für die *Plug-and-Play*-Erkennung am USB-Anschluss benötigt. Verwendung von Version 0.19 oder höher:

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

- **GitHub-Download von YAMuPlay**

YAMuPlay ist derzeit auf GitHub noch unter dem Projektnamen (Repository) *bauwong* abgelegt!

```
cd /home/pi
git clone https://github.com/schlizbaeda/bauwong.git
cd bauwong
chmod a+x bauwong.py # Die Python-Programmdatei ausführbar machen
```

Es ist so weit!

Jetzt ist der Zeitpunkt gekommen, die neu installierte Software **YAMuPlay** auszuprobieren 🍷. Geben Sie im Terminalfenster bitte folgende Kommandos ein:

```
cd /home/pi/bauwong # Da sollten Sie sich bereits befinden!
./bauwong.py
```

Es sollte die grafische Oberfläche von YAMuPlay aus Abbildung 3.1 gestartet werden. Nun muss nur noch ein USB-Memorystick mit Musikdateien angesteckt werden und es kann losgehen!

3.3.4 Desktop des Raspberry Pi einrichten

Nachdem die Software nun läuft, kann der Desktop des Raspberry Pi etwas benutzerfreundlicher eingerichtet werden 🍷. Im GitHub-Repository befinden sich im Unterverzeichnis **desktop** dafür vorbereitete Dateien.

- **YAMuPlay ins Startmenü aufnehmen**

```
sudo cp bauwong.desktop /usr/share/raspi-ui-overrides/applications
```

Wenn die Datei *bauwong.desktop* nicht verändert wird, erscheint im Startmenü unter *Zubehör* der Eintrag *Bauwong n.e.V.*, siehe Abbildung 3.6.

- **Auf dem Desktop ein Icon für YAMuPlay anlegen**

Durch Erstellung eines symbolischen Links auf die Datei [bauwong.desktop wird ein Desktop-Icon angelegt. Dieses Icon wird sofort sichtbar, siehe Abbildung 3.7.

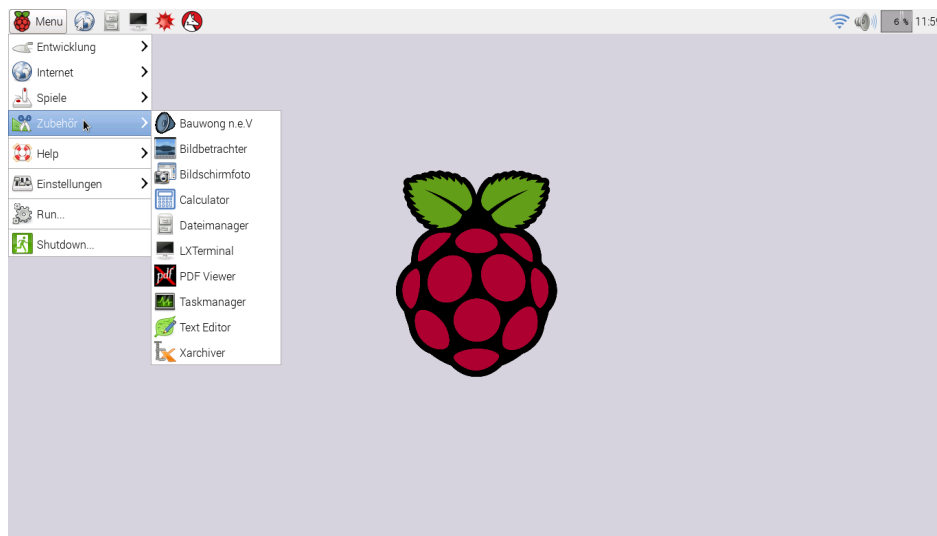


Abbildung 3.6: Neuer Eintrag im Startmenü des Raspberry Pi

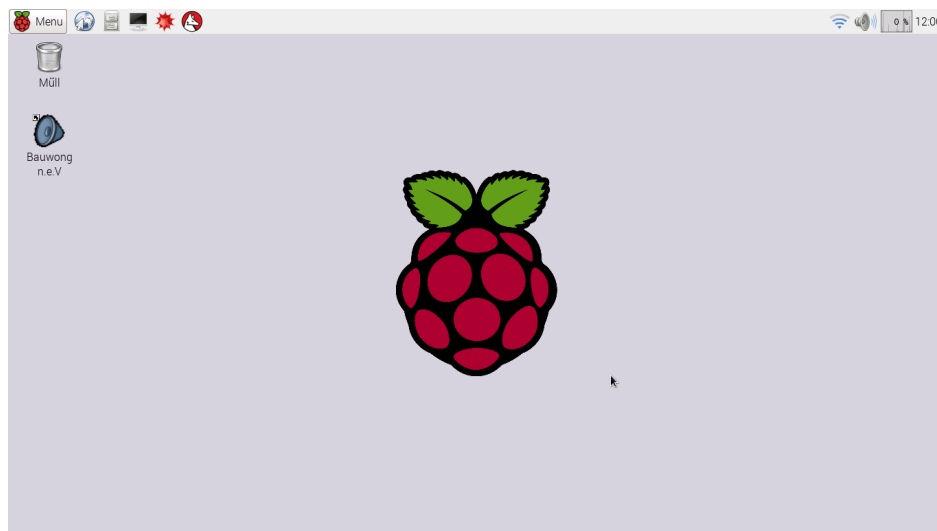


Abbildung 3.7: Neues Icon auf dem Desktop des Raspberry Pi

```
cd /home/pi/Desktop
ln -s /usr/share/raspi-ui-overrides/applications/bauwong.desktop
```

- **Autostart von YAMuPlay einrichten**

```
cd /home/pi/.config
mkdir autostart # falls dieses Verzeichnis noch nicht existiert
cd autostart
ln -s /usr/share/raspi-ui-overrides/applications/bauwong.desktop
```

- **Bildschirmschoner des Raspberry Pi ausschalten**

In der Datei /etc/lightdm/lightdm.conf muss folgende Änderung vorgenommen werden:

```
sudo nano /etc/lightdm/lightdm.conf
```

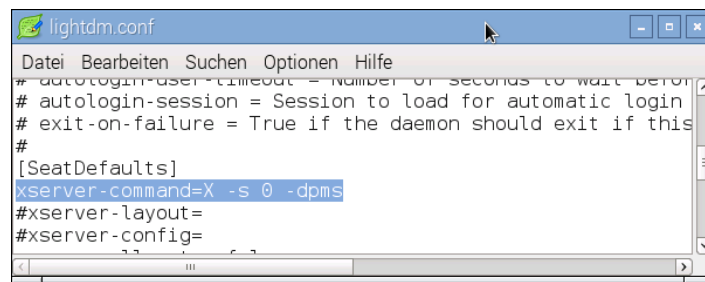


Abbildung 3.8: Bildschirmschoner in Datei `/etc/lightdm/lightdm.conf` deaktivieren

3.3.5 Touchpanel-Tastatur *matchbox-keyboard* installieren

In diesem Abschnitt wird die Installation der virtuellen Touchpanel-Tastatur *matchbox-keyboard* beschrieben. Das Vorgehen wurde weitestgehend von <http://ozzmaker.com/virtual-keyboard-for-the-raspberry-pi> übernommen.

- **Compilierung von *matchbox-keyboard* vorbereiten**

Softwarepakete für die Compilierung installieren:

```
sudo apt-get install libfakekey-dev libpng-dev libxft-dev autoconf libtool
-y
```

- ***matchbox-keyboard* compilieren**

```
git clone https://github.com/mwilliams03/matchbox-keyboard.git
cd matchbox-keyboard
./autogen.sh
make
sudo make install
```

- **shared libraries *erst nach* der Compilierung installieren!**

```
sudo apt-get install libmatchbox1 -y
```

- **Icon Toggle Matchbox Keyboard in der Schnellstartleiste erstellen**

```
cd /home/pi/bauwong/keyboard
```

Eintrag im Startmenü unter *Zubehör*:

```
sudo cp matchbox-keyboard.sh /usr/local/bin
sudo chmod a+x /usr/local/bin/matchbox-keyboard.sh
sudo cp matchbox-keyboard.desktop /usr/local/share/applications/inputmethods
Shellscript toggle-matchbox-keyboard.sh hinzufügen:
sudo cp toggle-matchbox-keyboard.sh /usr/bin
sudo chmod a+x /usr/bin/toggle-matchbox-keyboard.sh
sudo cp toggle-matchbox-keyboard.desktop /usr/share/applications
Eintrag des Shellscripts toggle-matchbox-keyboard.sh in der Schnellstartleiste:
cp panel /home/pi/.config/lxpanel/LXDE-pi/panels
```

Die kopierte Datei *panel* entspricht dem Original mit folgender Ergänzung: Im Ab-

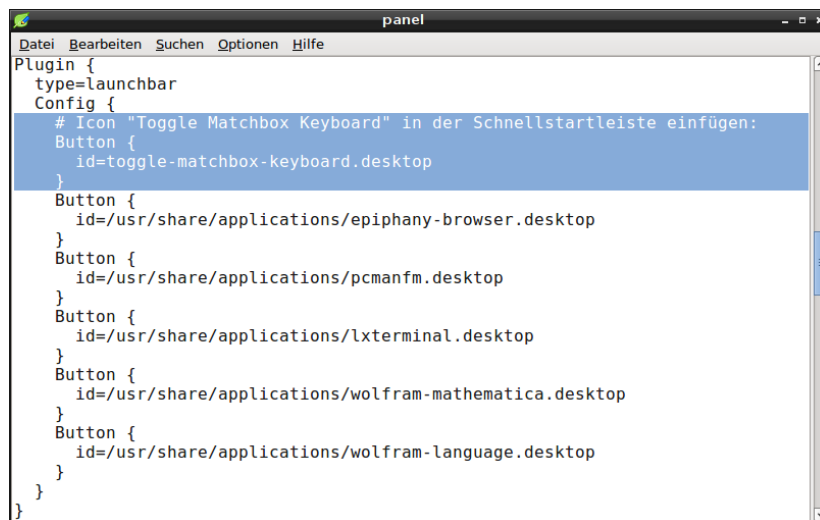


Abbildung 3.9: Änderungen in der Datei `/home/pi/.config/lxpanel/LXDE-pi/panels/panel`

schnitt `Plugin{ Config{ type=launchbar } }` müssen die markierten Zeilen aus Abbildung 3.9 ergänzt werden.

- **Tastaturlayout**

Ein passendes Tastaturlayout befindet sich in der Datei

`/home/pi/bauwong/keyboard/keyboard-bauwong.xml`. Dieses Layout kann natürlich noch angepasst werden. Nach der Änderung muss es nach `/usr/local/share/matchbox-keyboard` kopiert werden:

```
sudo cp keyboard-bauwong.xml /usr/local/share/matchbox-keyboard
```

3.4 Erweiterungen und Verbesserungen der Software

Zuletzt noch eine Liste von Punkten, um die die Software YAMuPlay ergänzt werden könnte. Hier handelt es sich um ein *Brainstorming*. Die Reihenfolge soll keine Gewichtung darstellen!

3.4.1 Bedienung und grafische Oberfläche

- **Scrolling durch Wischgesten wie an einem Smartphone**
horizontal und vertikal: `Treeview.xview` bzw. `Treeview.yview`
- **Anzeige von Titlenummer und aktueller Laufzeit**
wie bei den meisten klassischen CD-Spielern
- **Auf die Gesamtdauer eines Stückes skaliertes „Fortschrittsbalken“**
einfache Verschiebemöglichkeit mit Zeitanzeige wie bei den meisten Mediaplayern

- **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.
- **evtl „Cursortasten“ für die Baumansicht und <ENTER> zum Aktivieren**
Ist dieses Thema mit der virtuellen Tastatur erschlagen?

3.4.2 Funktionalität von YAMuPlay

- **Anpassen auf unterschiedliche Displaygrößen**
Derzeit ist eine Displaygröße von mindestens 1366x768 Pixeln notwendig, da die Fensterhöhe im Code derzeit fest eingestellt ist. Die Fensterbreite ist unkritischer.
* Fenstergröße flexibler programmieren!
* Schriftgröße vor allem fürs Treeview-Steuerelement parametrierbar machen
* Alles in einer config-Datei ablegen
- **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.
- **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!
- **Dateitypen berücksichtigen**
Derzeit werden alle vorhandenen Dateien angezeigt. Über die Dateierweiterung 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?
- **omxplayer.bin**
Der omxplayer macht ein kurzes Fading (< 1 Sekunde) beim Start einer neuen Musikdatei. Dies ist manchmal wirklich störend!

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