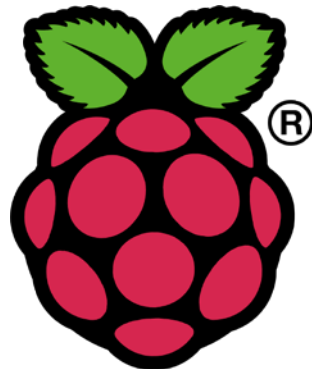


Projektbeschreibung Xmas TV

Beschreibung des Raspberry-Projekts „Christmas TV“ mit dem Raspberry Pi.



Inhaltsverzeichnis

1.	Raspberry Pi Projekt 2	1
2.	Idee.....	1
3.	Abstract (English)	2
4.	Fertige Lösung	2
5.	Hardware Aufbau.....	3
5.1	Frontpanel.....	5
5.2	Backpanel.....	5
6.	Pin Belegung Raspberry Pi.....	5
7.	Aufbau und Ansichten	6
8.	Software.....	8
8.1	Volumio Music-Player	8
8.2	Python mit GPIO	9
8.3	Xmas TV Programm.....	9
8.4	Station-ID	9
8.5	Soft-Shutdown	10
9.	Software Komponenten	10
9.1	Eintrag in /etc/rc.local	10
9.2	Musik auf USB-Stick.....	10
9.3	Code-Verfügbarkeit	11
10.	Links.....	11
11.	Web-Interface	11

1. Raspberry Pi Projekt 2

Der Raspberry Pi (genannt Pi) ist ein kreditkarten-kleiner und preisgünstiger (45 CHF) Linux Computer der sich seit seiner Einführung vor bald 4 Jahren grosser Beliebtheit erfreut und der bereits mehrere Millionen Mal verkauft wurde. Im Internet finden sich unzählige Projektbeschreibungen; alles Mögliche und Unmögliche wird mit diesem Micro-Computer gebaut. Der Initiator Eben Upton ist Engländer und in folgendem Video auf YouTube erklärt die Geschichte des Pi.

[Eben Upton on YouTube](#)

Der Pi hat 4 USB-Anschlüsse, einen Lan-Anschluss, ein HDMI-Anschluss für Monitor und einen Audio-Video-Ausgang. Ebenfalls auf dem Board ist ein Anschluss für die Pi-Camera - eine HD-fähige Kleinst-Camera für Fotos und Video. Was den Pi auszeichnet, ist die 40-polige Steckerleiste. Viele dieser Anschlüsse sind General Purpose Input/Output Pins, welche in Programmen angesteuert/gelesen werden können. Es gibt in der Zwischenzeit vielfältige Expansion Boards von unzähligen Anbietern. Der Pi konsumiert bloss 1 Watt Leistung: 5V und 200mA Strom.

Dieses Maschinchchen hat mich von Anfang an fasziniert, ich hatte mir in 2012 eines angeschafft. Nach der Realisierung des Switcher-Projektes im Sommer 2014 (siehe Kapitel Links) wurde ein weiteres Projekt in Angriff genommen: der Weihnachts-TV (Xmas TV).



Der Raspberry PI Model B+

2. Idee

Bei einem zufälligen Besuch in eine Geschäft in Zürich stiess ich auf den Christmas-TV, der mich sofort begeisterte - ich sah mein nächstes Raspi Projekt unmittelbar vor mir stehen. Der etwa 35 cm grosse Original-TV (eigentlich ein Spielzeug) spielt etwa 6 Weihnachtslieder ab einem kleinen Lautsprecher - der mit einem kleinen IC-Board verkabelt ist. Die Winter-Szene ist mit weissen Led's beleuchtet und der kleine Zug dreht seine Runden.

Eine perfekte Ausgangslage für einen Raspi Musikplayer.

3. Abstract (English)

I recently built what I call a Christmas TV based on a Raspberry Pi Linux Computer and Volumio-Software.

When I came across this lovely TV (see pic) in a shop in downtown Zurich, Switzerland (where I live) I had the idea to beef it up with a Pi based musicplayer. The device was originally able to play about 6 christmas songs through a one inch speaker that was hooked up to a small circuit board.

The lovely 3D winter scenery ist illuminated with a number of led's. A small train runs around in a circle with a tunnel. Just a tad kitschy but lovely. I call it Christmas TV.

So here is what I ended up with: I added a frontpanel with a number of pushbuttons and a backpanel with switches. The Pi is stacked with a HiFiBerry DAC+ and a Mini Jambox provides the sound output. It plays all my >200 xmas songs from Frank Sinatra, Elvis, Nat King Cole and many more (and plenty of Grateful Dead titles too). The mp3 songs are stored on a small usb stick.

The interface board carries all the resistors, the ADC chip für the volume and three connectors to the switches and led's.

There are two python scripts (that could be combined into just one): one running several threads with callbacks for the pushbuttons. The other script initiates a shutdown if the shutdown pushbutton on the back panel is pressed. Before shutdown a 3 sec message is played: ‚thank you for listening, good bye‘.

The main scripts randomly plays a 6 sec. station identification: ‚all day, all night, Xmas TV, stay tuned for more‘ that I recorded on my Mac. This and the whole thing is just for fun

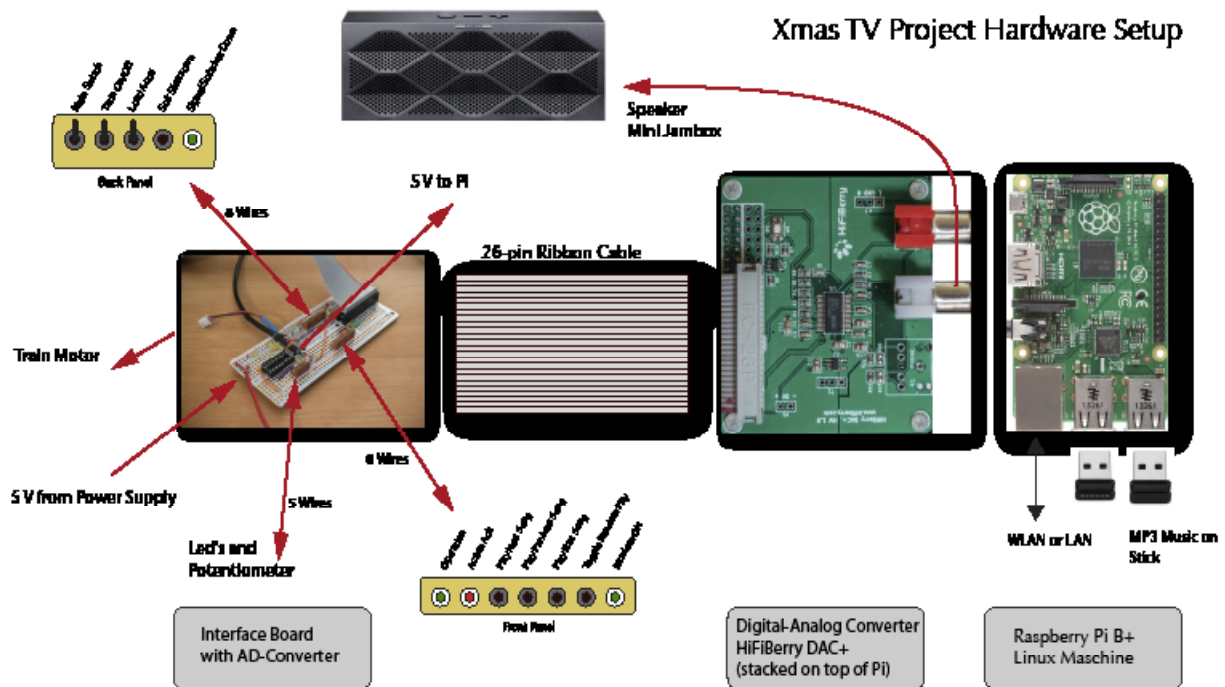
4. Fertige Lösung

So präsentiert sich der Xmas TV



Xmas TV Front View

5. Hardware Aufbau



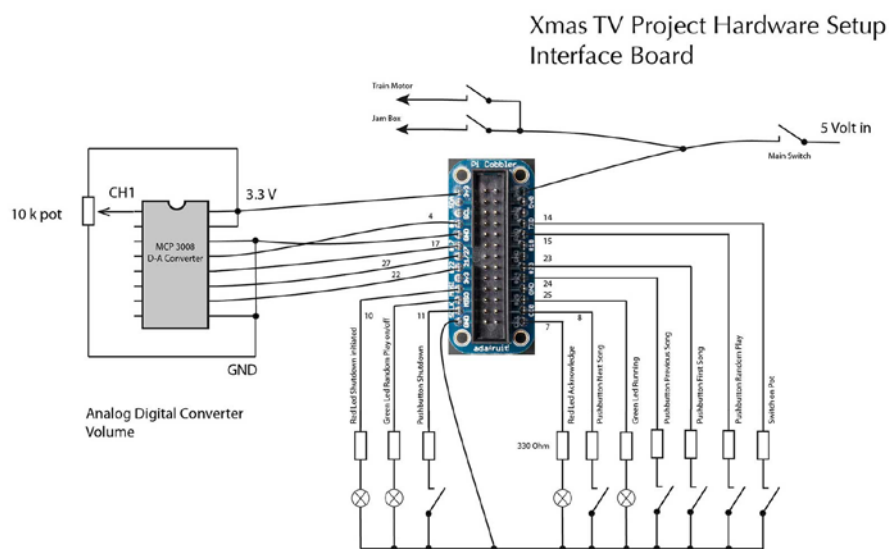
Hardware Setup Xmas TV

Das System besteht aus einem Raspberry Pi Model B+, auf dessen 40-Pin Connector ein Digital-Analog-Wandler HiFiBerry DAC+ aufgesteckt ist (stacked).

Die ersten 26 Pins sind via Flachkabel mit einem Interface-Board verbunden, welches an den entsprechenden Pins mit den Pull-up Widerstände bestückt ist. Ausserdem ist ein Analog-Digital-Wandler Chip MCP3008 vorhanden, der die Stellung des Front-Potentiometers wandelt - für Lautstärke-Regelung.

Zwei 8-Pin und ein 5-Pin Connector stellen die Verbindung zu Front- und Backpanel und zu den Led's und dem Potentiometer.

DAC+ ist via Cinch-Stecker mit dem Lautsprecher Mini Jambox verbunden.



December 2014 Peter K. Ruder

Interface Board

5.1 Frontpanel

Die 4 Pushbuttons haben folgende Funktion (von links nach rechts):

- Button 1: Nächster Song in der aktuellen Playlist
- Button 2: Vorheriger Song in der aktuellen Playlist
- Button 3: Erster Song in der aktuellen Playlist
- Button 4: Toggle für Random Play

Die 3 Led's des Panels haben folgende Anzeigefunktion:

- Grüne Led links: blinkt solange Python Script läuft
- Rote Led: Visueller Feedback wenn Button-Press erkannt
- Grüne Led rechts: an bei Random Play, sonst off

5.2 Backpanel

Das Backpanel trägt 3 Kippschalter, einen Pushbutton und eine rote Led. Diese Elemente haben folgende Funktion:

Erster Kippschalter: Hauptschalter, Power (5 Volt) on/off

Zweiter Kippschalter: Accu des Lautsprechers laden on/off

Dritter Kippschalter: Schalten die Kleinbahn on/off

Pushbutton: Drücken für > 2 Sekunden leitet Shutdown des Pi ein

Rote Led: Visueller Feedback, blinkt 3 mal wenn Shutdown eingeleitet ist

6. Pin Belegung Raspberry Pi

Der verwendete Raspberry Pi Modell B+ hat 40 General IO Pins (GPIO). Zum Interface-Board werden die ersten 26 Pins weitergeleitet.

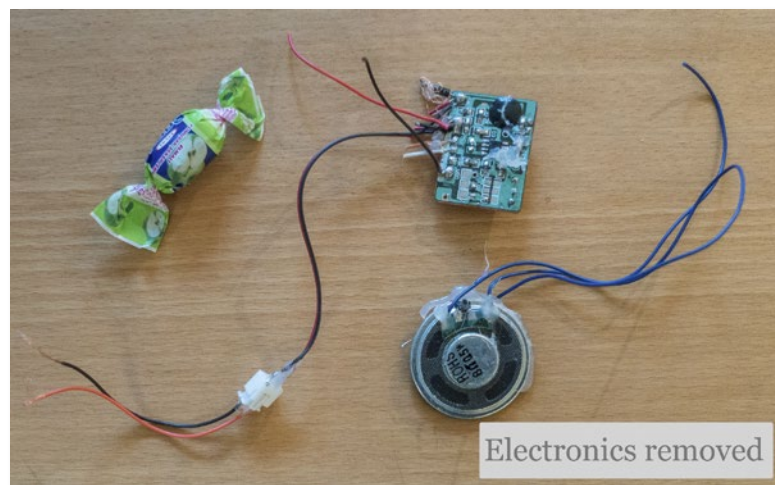
Pin	GPIO	Richtung	Verwendung
3	2		I2C Bus zum DAC+
5	3		i2C Bus zum DAC+
7	04	IN	zum ADC für Potentiometer
8	14		Schalter am Potentiometer
10	15		Pushbutton Random Play
11	17		zum ADC für Potentiometer
12	18	?	verwendet vom DAC+
13	27	IN	zum ADC für Potentiometer
15	22	IN	zum ADC für Potentiometer
16	23	OUT	Pushbutton first song in Playlist
17	3.3 Volt	OUT	3.3 Volt zum Potentiometer
18	24	OUT	Pushbutton previous song
19	10	OUT	Led Shutdown hinten

21	9	OUT	Grüne Led Random Play
22	25	OUT	Grüne Led Frontpanel
23	11	IN	Pushbutton Shutdown
24	8	IN	Pushbutton next song
26	7	OUT	rote Led Frontpanel
31	6	?	verwendet vom DAC+
35	19	?	verwendet vom DAC+
38	20	?	verwendet vom DAC+
40	21	?	verwendet vom DAC+

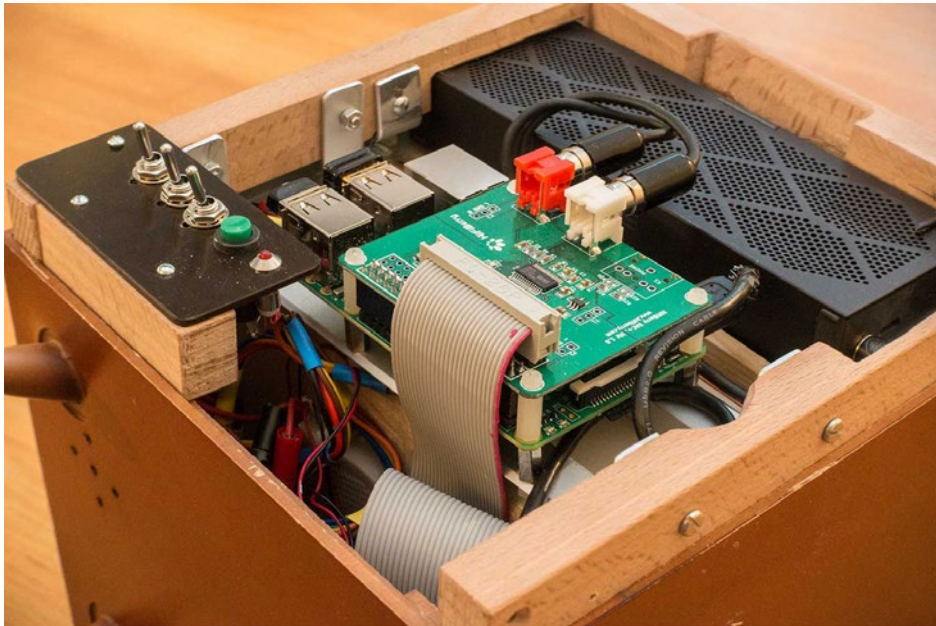
7. Aufbau und Ansichten

Die bestehende und einfache Elektronik mit dem kleinen Lautsprecher wurde ausgebaut. Auf der Hinterseite wurde ein Holzrahmen montiert, der den Pi und den Lautsprecher trägt. Das Interface-Board ist im freien Raum unter dem Diorama untergebracht.

Das bestehende Frontpanel hatte keine Funktion und es wurde durch ein neues Frontpanel ersetzt, in welchem 4 Pushbuttons und 3 Led's angeordnet sind. Das Backpanel trägt 3 Kippschalter, einen Pushbutton und eine rote Led.



Ausgebaute Elektronik



Rear View

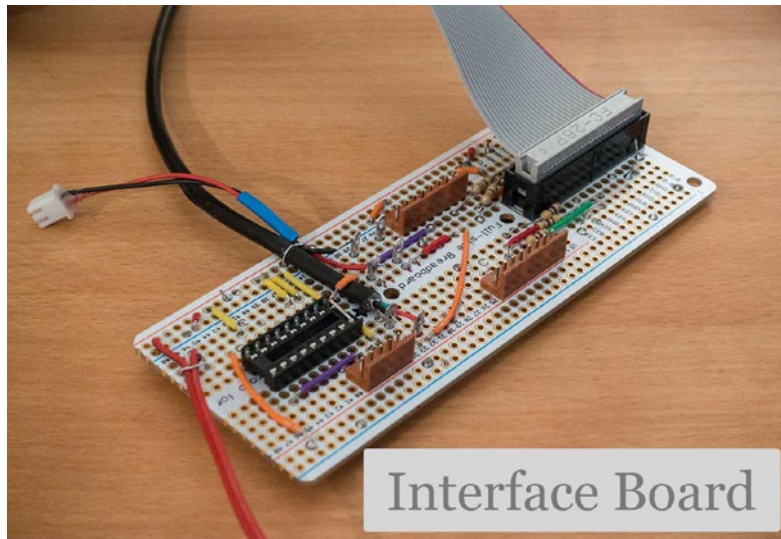
Die neue Elektronik auf der Hinterseite, Interface Board nicht sichtbar



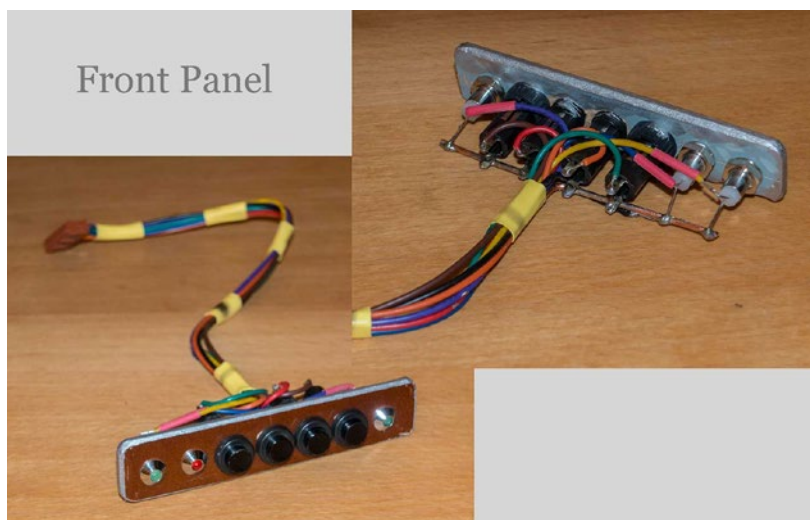
Front Before /After



Alte und neue Front



Interface Board



Frontpanel

8. Software

8.1 Volumio Music-Player

Die in diesem Projekt verwendete Open Source Software [Volumio](#) wird als fertiges Debian-basiertes OS geliefert: SD-Card-Image runterladen, auf Karte schreiben und den Raspi starten. So einfach geht das.

[Volumio Website](#)

Der Build enthält viele Audio-Optimierungen und stellt ein einfaches Interface zur Verfügung. Der vorinstallierte Webserver stellt eine Weboberfläche zur Verfügung und so kann der Player vom Smartphone, Tablet oder Compi ferngesteuert werden. Dies ist ohne eigene Programmierung, direkt nach erstem Start des Raspi möglich.

Die Volumio-Software spielt Online-Radios und lokale Musik-Dateien (FLAC, Mp3s, Wavs, AAC, ALAC, Musepac) ab, kann auf Spotify, Netzwerklaufwerke und die eigene USB-Bibliothek zugreifen. Zahlreiche Feintuning-Optionen unterstreichen den Anspruch des Volumio-Projektes, sich auch an audiophile Bast-

ler richten zu wollen.

Da die Audio-Kapazitäten des Raspi mager sind, setzt Volumio auf die Schnittstelle I2s und eine qualitativ hochwertige 24-bit Digital-Analog-Wandlung. Verwendet wird deshalb das Produkt HiFiBerry DAC+ .

[HiFiBerry Website](#)

Im Xmas TV wurden auf der Frontseite diverse Pushbuttons angebracht; diese erlauben eine Steuerung auch am Gerät selbst: next song, previous song, shuffle play.

Nochmals anders gesagt: das in diesem Projekt verwendete Raspi-OS kommt vollumfänglich von Volumio. Die konkrete Steuerung des Music-Players übernimmt das im nächsten Kapitel beschriebene Python Programm. Wie erwähnt, kann der Music-Player auch via Smartphone gesteuert werden.

8.2 Python mit GPIO

Nachdem das Volumio-Image auf die MicroSD-Karte geschrieben wurde und der Pi ok bootet, sind jedoch noch einige Packages zu installieren, damit die GPIO-Pins in einem Python Script angesprochen werden können. Ich habe also folgendes installiert (letztmals im Dez 2016 mit dem neuesten Volumio-Image volumio-2.041-2016-12-12-pi.img):

```
sudo apt-get update
sudo apt-get install python-mpd
sudo apt-get install python-dev python-smbus python-pip
sudo pip install RPi.GPIO
```

Danach kann ich folgende Python Scripts ok ausführen:

```
sudo python mpd-info.py
sudo python mpd-client.py
```

8.3 Xmas TV Programm

Die verwendete Script-Sprache ist Python.

Das Programm hat die Aufgabe, Kommandos an den im Volumio-Build integrierten Music-Player MPD abzusetzen. Es stellt das Interface her zwischen den Pushbuttons an der Frontseite des Xmas TV und dem Music-Player: es arbeitet als Client des MPD-Servers. MPD ist der Musicplayer-Daemon, der die Musik spielt - dieser ist Teil des Volumio-OS.

Das Programm wird beim Booten des Pi via /etc/rc.local gestartet. Für Testzwecke kann das Programm in einem Terminal-Fenster auf der Commandline gestartet werden. Dabei können folgenden Commandline-Parameter spezifiziert werden:

```
-d          debug, Statusmeldungen werden ausgegeben (stdout)
```

Das Programm setzt für jeden Pushbutton des Frontpanels eine Callback-Routine auf (Interrupt). Diese Funktionen werden aktiv, wenn der entsprechende GPIO-Pin auf Potential Null geht. Dann wird die entsprechende Funktion des MPD aufgerufen.

8.4 Station-ID

Beim Betrieb des Xmas-TV wird in unregelmässigen Abständen eine Station-ID eingespielt - wie wir das von Radio-Stationen kennen. Dazu wird die mp3-Datei werbung_1.mp3 temporär in die Playlist geladen und gespielt (6 Sekunden Dauer). Anschliessend wird dieses Stück wieder aus der Playlist entfernt. Diese

Aufgabe wird ebenfalls durch obiges Programm erledigt.

8.5 Soft-Shutdown

Ein zweites Python Script ist für die Einleitung des Shutdown des Pi zuständig. Es wird ebenfalls beim Booten des Pi via /etc/rc.local gestartet. Seine Aufgabe ist (via Callback) die Einleitung des Shutdown (System halt), sobald der Pushbutton des Backpanels für mehr als 2 Sekunden gedrückt wird. Die rote Led wird für ein visuelles Ack-Signal 3 mal kurz angesteuert.

Vor dem Absetzen des Shutdown wird eine vordefinierte mp3-Datei (Abschiedstext) der aktuellen Playlist zugefügt, die Lautstärke wird langsam vermindert, das gerade gespielte Stück wird ausgeblendet und der Abschiedstext wird gespielt. Danach wird ‚system halt‘ an den Pi abgesetzt.

9. Software Komponenten

Das Switcher-Projekt besteht aus folgenden Codeteilen:

Script Name	Funktion
mpd-client.py	Hauptprogramm Xmas TV, setzt Funktion an den mpd ab je nach gedrücktem Pushbutton Wird gestartet durch Eintrag in /etc/rc.local
softshut-mytv.py	Soft-Shutdown Script, leitet Shutdown ein, wenn Pushbutton Backpanel gedrückt wird. Wird gestartet durch Eintrag in /etc/rc.local

9.1 Eintrag in /etc/rc.local

In der Datei /etc/rc.local müssen folgende Einträge vorgenommen werden, damit die beiden Scripts nach dem Boot des Pi gestartet werden (sudo nano /etc/rc.local)

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will „exit 0“ on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
/var/www/command/player_wdog.sh startup & > /dev/null 2>&1

# start Peter's client
python /home/pi/mpd-client.py &
# start shutdown script
python /home/pi/softshut-mytv.py &
exit 0
```

9.2 Musik auf USB-Stick

Die Musik ist auf einem kleinen FAT32 USB Stick gespeichert. Damit der Stick angesprochen werden kann, ist folgendes zu konfigurieren im PI:

Mount Point erstellen

```
sudo mkdir /media/usb1
```

Permissions geben für Benutzer volumio

```
sudo chown volumio:volumio /media/usb1
```

Dann in /etc/fstab folgenden Automount einfügen

```
/dev/sda1      /media/usb1   vfat    rw,defaults  0      0
```

Manuell kann der Stick auf der Commandline mouted/unmouted werden mit:

```
sudo mount -t vfat -o utf8,uid=volumio,gid=volumio,noatime /dev/sda1 /media/usb1
```

```
sudo umount /media/usb1
```

Prüfen kann man dies mit

```
df -h
```

oder

```
sudo blkid -o list -w /dev/null
```

9.3 Code-Verfügbarkeit

Der gesamte Code des Xmas TV liegt auf GitHub geladen werden:

[GitHub Xmas TV](#)

10. Links

Volumio Package für Raspberry Pi.

[Volumio Org](#)

HiFiBerry Produkte, audiophile Add-Ons zu Raspberry Pi.

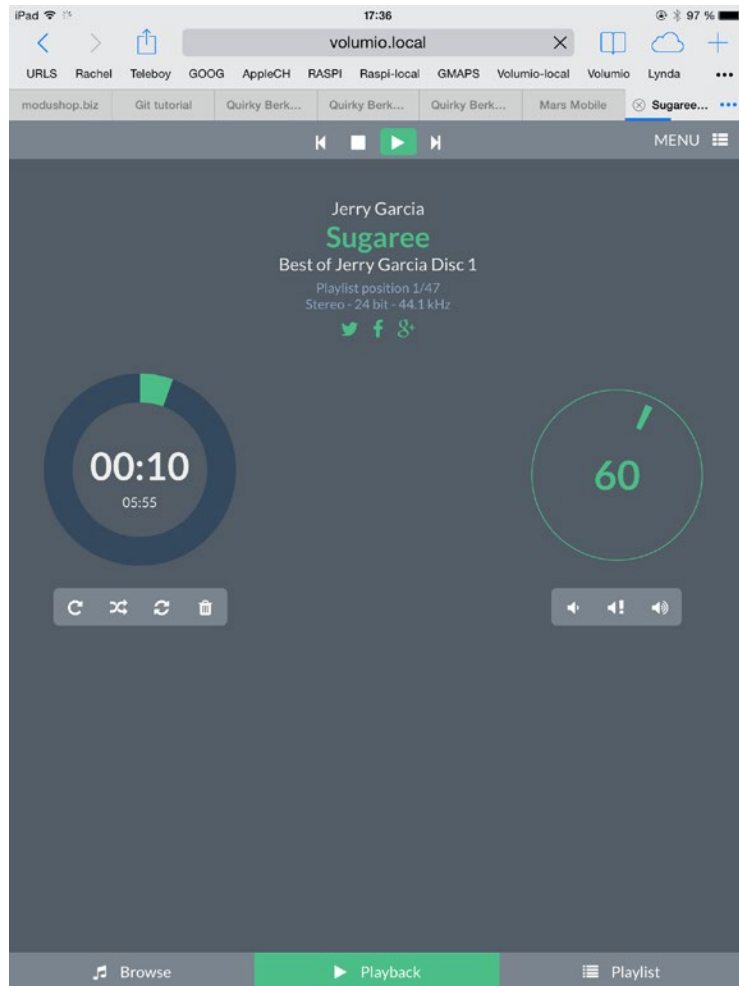
[HifiBerry](#)

Andere Raspi-Projekte (Switcjher, Light-Painting, Triangle) ebenfalls auf GitHub:

[My GitHub](#)

11. Web-Interface

Das vom Volumio-Build zur Verfügung gestellte Webinterface des Volumio-Players sieht auf einem iPad so aus:



Volumio Web-Interface (Main Screen)

Damit kann der Player per iPhone/iPad ferngesteuert werden.

Peter K. Boxler, Dezember 2014

Last update Nov. 2016