

# QR-Code Audio-Player

## Motivation

Für unsere kleinen Kinder (1-4 Jahre alt) sind CDs noch nicht wirklich geeignet, und ein herkömmlicher MP3-Player überfordert ihre geistigen und motorischen Kapazitäten. Deshalb hab ich mir überlegt, wie man mit einer rein haptischen Benutzerschnittstelle möglichst einfach verschiedene Alben oder Hörbücher „einlegen“ kann.

Ich bin recht schnell zum Schluss gekommen, dass es schön wäre, wenn man etwas in der Hand hält, und dieses auch in einem Regal oder in einer Mappe sortiert ablegen kann. Ein Albumcover mit einem QR-Code scheint also schon sehr nahe zu liegen.

## Ideensammlung

- Ein Raspberry Pi dient als Audio Player
- Ein sehr einfaches Interface direkt am Gerät mit Start, Stop, nächster Titel, vorheriger Titel über Taster mit spürbarem Druckpunkt <sup>1)</sup>.
- Eine Kamera liest einen QR-Code von gut handhabbaren Zettelchen
- Alben von einem USB-Stick werden automatisch auf die lokale HDD kopiert.
- Ein Thermodrucker kann den QR-Code für das aktuell per USB-Stick „hochgeladene“ Album ausdrucken. Dadurch hat man ein in sich selbst geschlossenes System, sofern man davon ausgeht, dass die Musik auf einem USB-Stick daherkommt.
- Austauschen von Alben unter Freunden entweder über RetroShare oder IPFS.
- Automatisches Generieren von ausdrucksbaren PDFs mit dem QR-Code und dem Album-Cover

## Erste Schritte

### Hardware

- Raspberry Pi 3, Modell B (hat Ethernet, WiFi, Bluetooth, Kamera-Interface, 4 USB-Buchsen)
- 16GB SD-Karte
- Altes Handyladegerät mit 5V, 1A mit Mikro-USB-Kabel
- Gehäuse von [The Pi Hut](#)
- 4 Taster
- selbstgedrucktes Gehäuse für Taster und 4-stelliger Anzeige

### Betriebssystem

Beim Stöbern durch das Internet bin ich auf [RuneAudio](#) gestoßen. Es scheint eine gute Ausgangsbasis für mein Vorhaben zu sein. Besonders gut gefällt mir die simple Art. Es ist kaum mit unnötigem Schnickschnack überladen und wird üblicherweise nur über den Webbrowser bedient - egal ob von einem Smartphone oder von einem Desktop-PC. Es basiert auf [Arch Linux](#), mit dem ich eigentlich

nicht vertraut bin, aber ich bin zuversichtlich, damit zurecht zu kommen.

Per

```
sudo dd bs=10M if=RuneAudio_rpi2_rp3_0.4-beta_20160321_2GB.img | pv -s 2100M  
| sudo dd of=/dev/sdc
```

wird das heruntergeladene Image auf die SD-Karte geschrieben. pv dient dabei zur Statusanzeige.

Anschließend wird per GParted die erstellte FAT32-Partition von ca. 2GB auf das maximum vergrößert, in meinem Fall gut 12GB.

Dann wird die SD-Karte in den Raspi gesteckt, und das System bootet. Dann muss man die IP-Adresse des Rechners herausfinden, das mache ich mit dem Webinterface der A1-Hybridbox, meim Modem bzw. Router.

Nun können wir uns endlich per ssh verbinden:

```
ssh root@192.168.0.64  
pw: rune
```

Dieses Standard-Passwort sollte aus Sicherheitsgründen sofort geändert werden mittels

```
passwd
```

## Taster-Einbindung

Mit C und Cbin ich sehr gut vertraut, daher fällt meine Wahl auf compilierte Binärdateien anstatt von sonst so beliebten Python-Skripte. Wir brauchen daher den gcc, der mit RuneAudio leider nicht mitgeliefert wird, und somit nachinstalliert werden muss: `sudo pacman -Sy sudo pacman -S gcc` Weiters hab ich schon Erfahrungen machen dürfen mit der [\[\[http://abyz.me.uk/rpi/pigpio/index.html](http://abyz.me.uk/rpi/pigpio/index.html) Bibliothek pigpio]]. Ich war sehr zufrieden damit und sie kommt somit auch hier zum Einsatz. === Installation von pigpio === Per GIT wird die aktuelle Version von pigpio von GitHub geklont, kompiliert und installiert:

```
cd  
git clone https://github.com/joan2937/pigpio.git  
cd pigpio  
make  
sudo make install
```

Später stellte sich heraus, dass die Laufzeitbibliothek libpigpio.so nicht an ihrem Platz unter /lib gefunden werden konnte. Scheinbar ging hierbei etwas schief. Ein manuelles Kopieren löste das Problem:

```
cd  
cd pigpio  
sudo cp libpigpio.so /lib
```

=== Schnittstelle zum Music Player Daemon === Auf RuneAudio läuft der Daemon namens Music

Player Daemon, kurz mpd. Mit diesem kann die Wiedergabe gesteuert werden und Wiedergabelisten verwaltet werden. Man braucht dazu jedoch einen Client, der im einfachsten Fall der Music Player Client, kurz mpc ist. Der kann auch auf einem anderen Rechner ausgeführt werden und steuert so RuneAudio an:

```
mpc --host 192.168.0.64 <Befehl> <Optionen>
```

Lokal schreibt man z.B.

```
mpc play
mpc pause
mpc next
mpc prev
```

, das sind genau die Befehle, die die 4 Tasten auslösen sollen. === push-button.service === Der Plan ist nun, einen Service am System einzurichten, der automatisch nach dem booten gestartet wird, und somit per mpc den mpd ansteuert. Der Quellcode des Programms liegt auf [GitHub](#) und kann folgendermaßen auf RuneAudio kompiliert werden:

```
cd
git clone https://github.com/KarlZeilhofer/push-button-service.git
cd push-button-service
make
```

Zum testen kann man die Binärdatei einfach ausführen, Root-Rechte sind aber wegen der Bibliothek pigpio notwendig:

```
sudo ./push-button-service
```

Mit

```
cd /lib/systemd/system/
sudo nano push-button.service
```

legen wir eine neue Text-Datei an, die mit dem Texteditor nano nun mit folgendem befüllt wird:

[push-button.service](#)

```
[Unit]
Description=Runs the user programm push-button-service controlling the
playback of 'mpd'.
After=multi-user.target

[Service]
Type=simple
ExecStart=/root/push-button-service/push-button-service
Restart=on-abort

[Install]
```

```
WantedBy=multi-user.target
```

. Nun können wir den Service einbinden und starten:

```
sudo chmod 644 /lib/systemd/system/push-button.service
sudo systemctl daemon-reload
sudo systemctl enable push-button.service
sudo systemctl start push-button.service
```

Geschafft, nun sollte der Player bereits auf die Tasten reagieren, ohne dass man selbst das Programm gestartet hat. Auch nach einem reboot macht der Player das nun! ===== Ausblick und Stichworte =====  
===== 7-Segment Anzeige ===== \* Im Idle kann die Uhrzeit angezeigt werden \* Aktueller Titel: mpc current -f %position% \\liefert Titel Nr der aktuellen Playlist, beginnend bei 1. ===== 3D-Dateien ===== Gezeichnet in Sketchup, maßgeschneidert an die Taster, und die 7-Segment-Anzeige und das Raspi-Gehäuse. ===== QR-Code Steuerung ===== Ordner laden:

```
mpc add USB/Musik/koenig-der-loewen
```

ACHTUNG: hier muss auch der Name des USB-Sticks bekannt sein!

ACHTUNG: die Ordner dürfen keine Umlaute haben! Playlist löschen:

```
mpc clear
```

QR-Code erstellen mit

```
qrencode
```

===== Sonstiges =====  
===== Gnome MPC ===== RuneAudio kann man auch sehr bequem mit einer nativen Anwendung fernsteuern. Der Gnome music player client ist sehr flott und angenehm zu benutzen. ===== Kindergerechter Audio-Player ===== Ein sehr ähnliches Projekt wurde vor Jahren schon einmal von Marcus Nasarek umgesetzt und auf [Raspberry Pi Geek](#) als Artikel veröffentlicht. ===== Notizen =====

Hardware:

Raspberry Pi 3B (37€)

<https://thepihut.com/collections/raspberry-pi/products/raspberry-pi-3-model-b>

16GB Mikro SD-Karte (10€)

Netzteil:

2A Mikro-USB:

<https://thepihut.com/collections/raspberry-pi-power-supplies/products/eu-micro-usb-power-supply-for-the-raspberry-pi>

Adafruit 7-Segment, gelb, I2C (11€)

<https://eckstein-shop.de/Adafruit-056-4-Digit-7-Segment-Display-w-I2C-Backpack-Yellow>

oder Direkt:

Knight Bright: CC04-41SYKWA

10mm-Schrift, 40x16x6.8mm, 20mA/Segment

Kamera-Modul (28€):

<https://thepihut.com/collections/raspberry-pi-camera/products/raspberry-pi-camera-module>

Optional:

HiFi Just Boom DAC (34€)

<https://www.justboom.co/product/justboom-dac-hat/>

HiFi Just Boom DAC mit 2x30W Verstärker (69€)

<https://thepihut.com/products/justboom-amp-hat>

oder China DAC:

[https://www.ebay.at/sch/i.html?\\_odkw=raspberry+hifi+dac&\\_osacat=0&\\_from=R40&\\_trksid=p2045573.m570.l1313.TR2.TRC0.A0.H0.Xraspberry+hifi+dac.TRS0&\\_nkw=raspberry+hifi+dac&\\_sacat=0](https://www.ebay.at/sch/i.html?_odkw=raspberry+hifi+dac&_osacat=0&_from=R40&_trksid=p2045573.m570.l1313.TR2.TRC0.A0.H0.Xraspberry+hifi+dac.TRS0&_nkw=raspberry+hifi+dac&_sacat=0)

Thermodrucker, 58mm  
ca. 35€

==== Volumio, DLNA, AirPlay ==== Volumio ist ein Pendant zu RuneAudio, läuft sehr geschmeidig und unterstützt angeblich DLNA. D.h. der Raspi kann als Audiosenke verwendet werden. Sollte mir das Linux CLI Tool pulseaudio-dlna funktionieren, scheiterte jedoch bei den ersten Versuchen. Der Audiostrom vom PC wird zwar unmittelbar umgeleitet, kommt aber bei den Boxen am Raspi nicht an.

[deutsch](#), [software](#), [raspberry](#), [c++](#), [linux](#), [technik](#), [projekt](#), [audio](#)

<sup>1)</sup>

die Touchscreens, die man heutzutage überall hat, sind zwar sehr nett und mächtig, aber so wirklich schön zu bedienen sind sie nicht, weil die direkte Rückmeldung an der Fingerkuppe fehlt

From:

<http://www.zeilhofer.co.at/wiki/> - **Verschiedenste Artikel von Karl Zeilhofer**

Permanent link:

<http://www.zeilhofer.co.at/wiki/doku.php?id=qr-code-player>

Last update: **2018/09/11 03:41**

