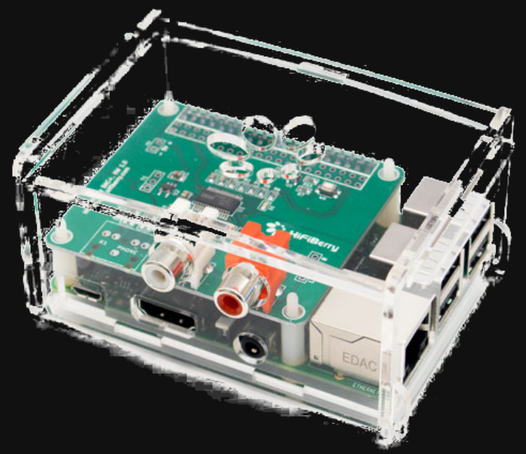


pulsecast

Using PulseAudio with Single Board Computers to Power a Whole Home Audio Solution

Agenda

1. The Problem
2. Linux Sound Architecture and PulseAudio
3. Recipe for pulsecast System using Raspberry Pi with HifiBerry

The Problem

I have several stereos in a house that already has Ethernet in every room.

I would like to be able to direct the same audio to all speakers in the house, without loss of audio quality and without noticeable delays between speakers.

It would also be nice to keep using my current music player, Clementine, as well as being able to use any streaming services that I might subscribe to (only a handfull of which are integrated into Clementine).

Prior Solution

1. Create and save play lists from a music player.
2. Feed the play lists to icecast from ezstream.
3. Connect a tablet or computer to each stereo and play the stream via browser.

Issues with Prior Solution

- Need for extra steps of saving play list and then invoking ezstream.
- All items on play-list needed to be in same format.
- Any changes required restarting the stream.
- Each time the stream started/restarted had to visit each stereo.

Google ChromeCast Audio

Pro

- Inexpensive (about $30)
- Based on DLNA
- Grouping feature to simultaneously cast to ChromeCast Audio devices

Con

- CCA has been discontinued
- Best OS Support was Android
- Some Features Required Google Home App
- On other platforms only supported player was the Chrome Browser
- Sound quality similar to built in audio on other devices
- Frequent dropouts even over Ethernet (unacceptable)

Unofficial Support

- pulseaudio-dlna (outside project, not part of pulseaudio)
- PulseAudio AVAHI support
- VLC added ChromeCast support in version 3.0

It did support an external DAC

The ChromeCasts had an optical audio out which uses a non-standard variant of a TOSlink cable that you can feed a Digital-Anaolog Converter with a TOS input.

The CCA was a great attempt at an inexpensive consumer solution. But at less than the price of a decent soundcard they may have been too ambitious on cost.

Multi-Room Wired

There are a lot of possible configurations here, and many systems out there in this space.

Advantages

- Reliable, especially the simpler octopus configurations
- Been around a long time

Disadvantages

- Requires running lots of new dedicated wire (expensive)
- Future Reconfiguration would require pulling more wire again
- Off the shelf systems are costly and at risk of obsolescence

Sonos

It is even more proprietary than the ChromeCast and exteremely expensive as in it can cost thousands of dollars to implement.

Whole House FM

Doesn’t require wires or any network infrastructure.

This solution is cheap and would work, but has limitations.

DLNA / AVAHI

DLNA and Avahi are a set of standards for multi-media devices to communicate, worthy of a talk in their own right.

There are several projects to make a Raspberry Pi or something similar act as a DLNA device for sending media to for rendering. They all look to be fairly small and I never got as far as testing them.

Volumio and RuneAudio are open source Media Center distributions that will run on low end devices like Pi.

The grouping feature of the CCA appears to be unique and innovative, the other options did not appear to have anything like it.

VLC and ffmpeg

Both VLC and ffmpeg have the ability to stream content via RTP. Unfortunately, their built in streaming is limited to single files without support for play-lists.

Internet Radio DJ Software

There is software for running an Internet radio station, but the open source options I glanced over did not appear to be simple to install and configure.

Why can't we just capture to a virtual sound-card and stream from there?

In search of that answer I started to look at the Linux Sound Architecture. OSS and ALSA are too basic to do that on their own.

PulseAudio supports both RTP streaming between computers running it and DLNA / AVAHI.

Part II

Linux Sound Architecture and PulseAudio

An Introduction to Linux Sound Architecture

- Open Sound System (OSS)
- Enlightened Sound Daemon (ESD)
- Advanced Linux Sound Architecture (ALSA)
- Jack
- PulseAudio

ALSA

The Advanced Linux Sound Architecture (ALSA) provides audio and MIDI functionality to the Linux operating system. ALSA has the following significant features:

- Efficient support for all types of audio interfaces, from consumer sound cards to professional multichannel audio interfaces.
- Fully modularized sound drivers.
- SMP and thread-safe design.
- User space library (alsa-lib) to simplify application programming and provide higher level functionality.
- Support for the older Open Sound System (OSS) API, providing binary compatibility for most OSS programs.

PulseAudio and Jack are Sound Servers both of which work closely together with ALSA

What is a Sound Server?

The Sound Server accepts sound input from source and redirects it to a sink (sound card etc). This architecture was developed to manage multiple inputs and outputs on a system.

Jack

Jack is targeted to Audio Recording and Mixing Applications implementing a virtual audio Patch Panel. Jack and Pulse are able to coexist and run at the same time.

PulseAudio

Pulse provides API and hardware abstraction, including networked hardware. Most distribution's Desktop environments use Pulse and it generally provides a working out of the box setup.

To work its' magic Pulse interfaces with existing ALSA and OSS hardware drivers, and provides emulation of Enlightened Sound Daemon.

- User Space Daemon
- Control and Configuration from Command Line, Configuration Files, and Graphical Utilities.
- Created by Lennart Poettering.

The diagram on the next slide comes courtesy of rudd-o.com, the original is at https://rudd-o.com/uploads/images/how-pulseaudio-works/pulseaudio-diagram.png/view.

PulseAudio Terminology

Server:

the PulseAudio Daemon. Since it runs in user space each user gets their own instance of the daemon.

Client:

the application sending data to Pulse via one of its several supported APIs or emulations.

Sources:

generate sound, this could be clients or hardware such as a microphone.

Sink:

the device generating the sound, generally this is handled by ALSA.

Front End:

Applications that interact directly with Pulse, such as configuration utilities and sound mixer applets.

Modules:

Pulse is modular, splitting functionality into optional modules and allowing for extension of functionality by developers outside the Pulse project.

Communicating with PulseAudio

On your desktop you'll mostly use pavucontrol, paprefs and your desktop environments audio management tools to interact with pulse. However, for diagnostics and for working on the headless pulsecast you'll need pacmd and pactl.

```
pacmd info
## pactl utilty provides a briefer summary
pactl info
pacmd list-(sinks|modules|cards|sources|clients|sink-inputs)
pactl list (sinks|modules|cards|sources|clients|sink-inputs)
pacmd dump
```

```
pi@berry2:~ $ pactl list sink-inputs
Sink Input #0
    Driver: module-rtp-recv.c
    Owner Module: 8
    Client: n/a
    Sink: 0
    Sample Specification: s16be 2ch 44062Hz
    Channel Map: front-left,front-right
    Format: pcm, format.sample_format = "\"s16be\""  format.rate = "44100"  format.channels = "2"
    Corked: no
    Mute: no
    Volume: front-left: 65536 / 100% / 0.00 dB,   front-right: 65536 / 100% / 0.00 dB
            balance 0.00
    Buffer Latency: 258372 usec
    Sink Latency: 239144 usec
    Resample method: speex-fixed-1
    Properties:
        media.role = "stream"
        media.name = "RTP Stream (PulseAudio RTP Stream on mypc)"
        rtp.session = "PulseAudio RTP Stream on mypc"
        rtp.origin = "user 936190379 0 IN IP4 192.168.1.11"
        rtp.payload = "10"
        module-stream-restore.id = "sink-input-by-media-role:stream"
```

Part III

pulsecast

A Recipe for Whole Home Audio with PulseAudio and Raspberry Pi

Linux to Linux Pulse

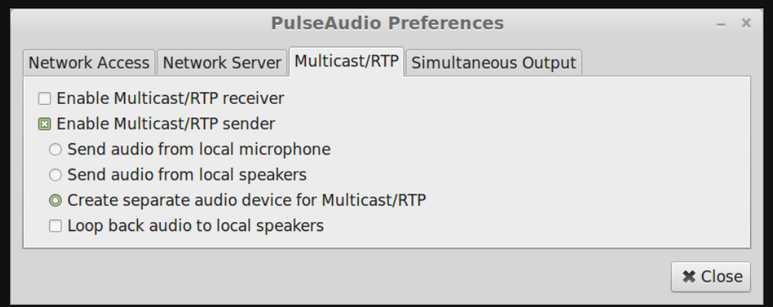
Make sure all of the needed packages are installed.

On Debian/Ubuntu/Mint these are

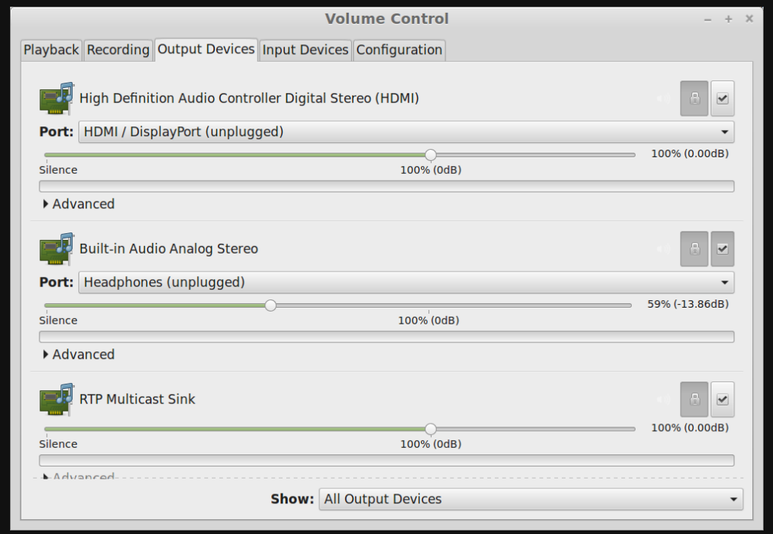
- pulseaudio
- pulseaudio-utils
- pulseaudio-module-gconf
- pavucontrol
- paprefs

Enable Sending on the Source Computer

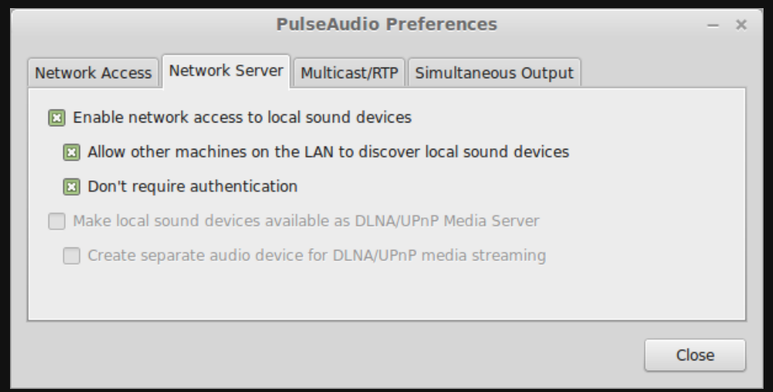
Two Clicks is all it takes in PulseAudio Preferences (paprefs).

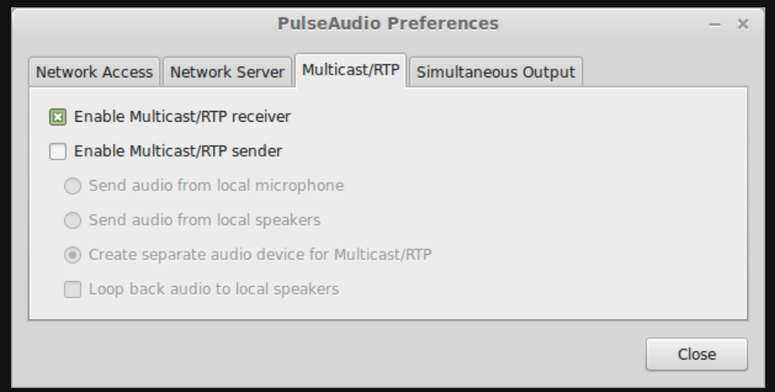


The New Output Device in PulseAudio Volume Control

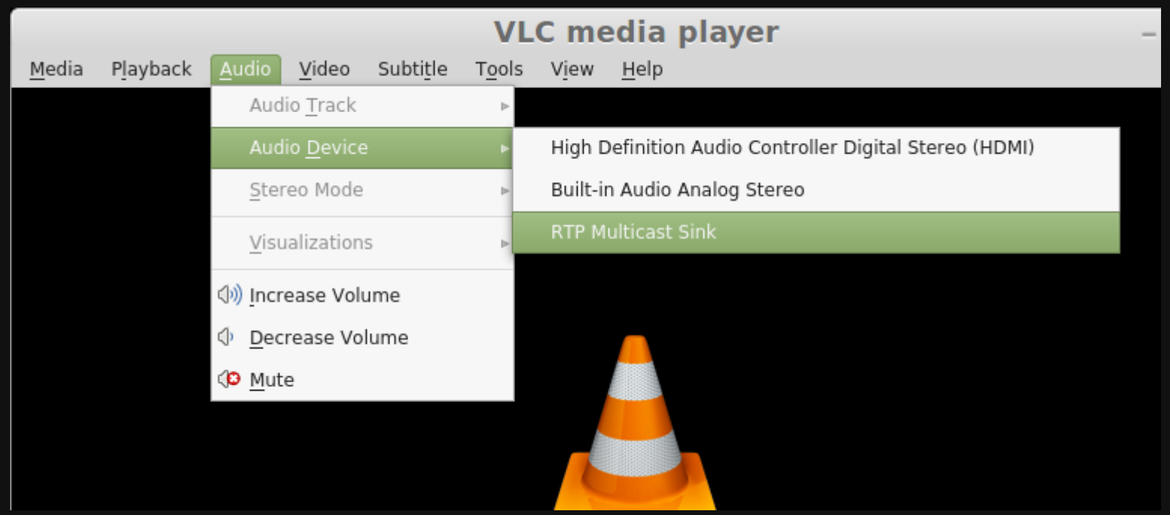


A few more clicks on the Recieving Side in PulseAudio Preferences (paprefs)





Switching Output in VLC



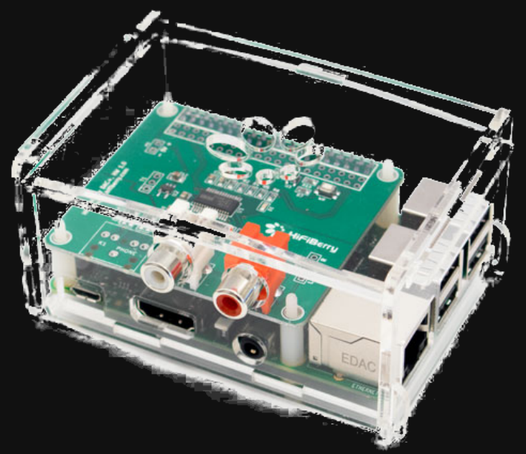
If you want to see the configuration values for a setup you made with gui tools

```
$ ls ~/.gconf/system/pulseaudio/modules
   %gconf.xml  remote-access  rtp-recv  rtp-send  zeroconf-discover
```

each folder contains a **%gconf.xml** file.

Entries in **%gconf.xml** are the same configuration directives we'll see later, but wrapped in an xml tag.

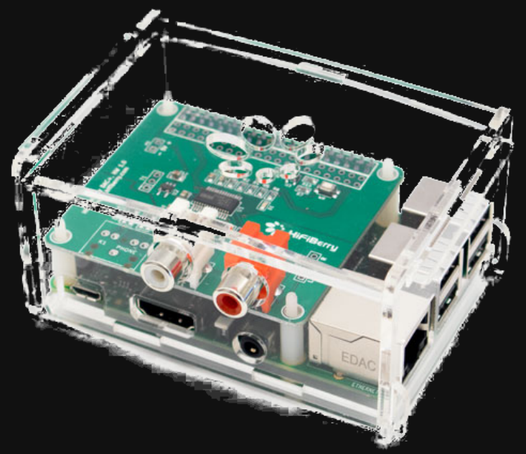
```
<stringvalue>module-native-protocol-tcp</stringvalue>
```



pulsecast Device Using Raspberry Pi

What You'll Need

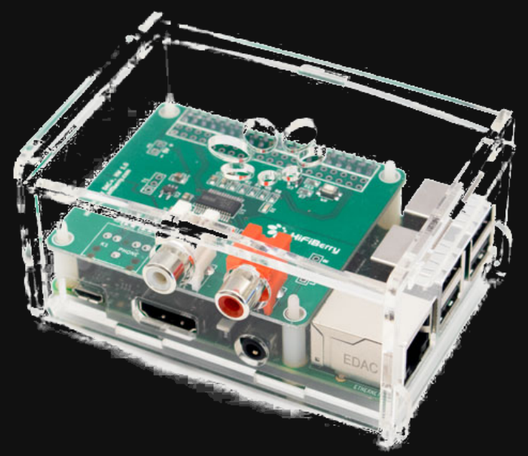
- Raspberry Pi



pulsecast Device Using Raspberry Pi

What You'll Need

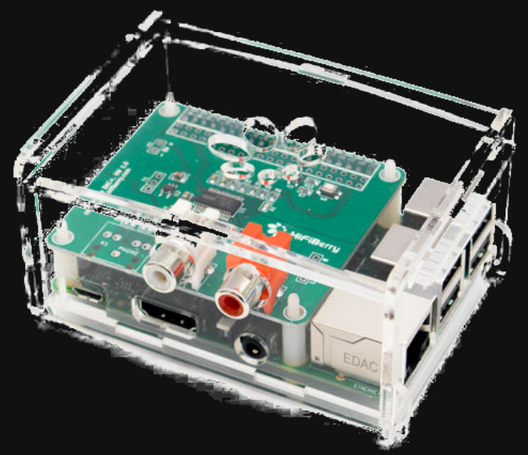
- Raspberry Pi
- Micro SD Card



pulsecast Device Using Raspberry Pi

What You'll Need

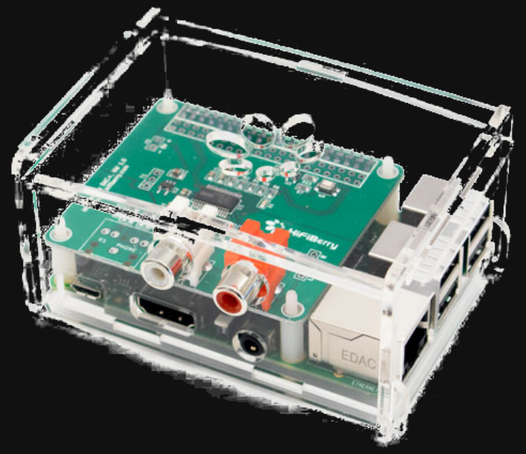
- Raspberry Pi
- Micro SD Card
- HiFi Berry DAC+ SoundCard



pulsecast Device Using Raspberry Pi

What You'll Need

- Raspberry Pi
- Micro SD Card
- HiFi Berry SoundCard
- Case for HiFi Berry



pulsecast Device Using Raspberry Pi

What You'll Need

- Raspberry Pi
- Micro SD Card
- HiFi Berry SoundCard
- Case for HiFi Berry
- Download of Raspbian NOOBS installer

Quick Pi Setup

1. Format MicroSD as FAT32. (I use gparted)
2. download & extract NOOBS
3. copy the extracted NOOBS to the newly formatted MicroSD
4. Attach peripherals and network, boot Pi with NOOBS MicroSD installed.
5. Install minimal Raspbian (we don’t want the gui).
6. Switch keyboard to US during install or your keyboard will be mapped wrong.
7. Reboot when the installation is complete.

8. login
    - user: pi
    - password: raspberry
9. sudo ifconfig to get MAC and IP address.

10. sudo raspi-config to configure your pi.
    - Change password.
    - Set hostname.
    - Choose Interfacing Options, from sub menu SSH, and enable SSH access.
11. Either reserve the ip address in DHCP or set a static ip address.
12. Now is a good time to update the system.
    - sudo apt-get update
    - sudo apt-get upgrade

Enable the Soundcard

Edit /boot/config.txt (must be root).

```
## dtparam=audio=on ## disables built in sound

## load hifiberry
dtoverlay=hifiberry-dacplus

## disable wifi and bluetooth if you're not using them.
dtoverlay=pi3-disable-wifi
dtoverlay=pi3-disable-bt
```

- sudo halt -p
- Unplug your Pi.
- Remove the USB devices you used for setting up NOOBS.
- Power the Pi on again.

Install PulseAudio

SSH back into the device to begin setting up pulseaudio.

```
sudo dpkg -i pulseaudio pulseaudio-utils screen
```

Add the following scripts

```
/usr/local/sbin/startpulse.sh

#!/bin/bash
while [ 1 ]; do
  ## start does not return error if daemon already loaded
  /usr/bin/pulseaudio --start --disallow-module-loading
  sleep 60
done

/usr/local/sbin/reloadpulse.sh

#!/bin/bash
## kill all pulse processes
/usr/bin/pkill pulseaudio
/usr/bin/pulseaudio --start --disallow-module-loading
```

Configure Pulse

```
ls /etc/pulse
client.conf  daemon.conf  default.pa  system.pa
```

These are read in the order daemon, default, client.

```
daemon.conf

default-fragments = 8
default-fragment-size-msec = 10
deferred-volume-safety-margin-usec = 8000
```

```
default.pa

load-module module-native-protocol-tcp auth-ip-acl=127.0.0.1;192.168.1.111
load-module module-rtp-recv
```

Start it with screen from rc.local

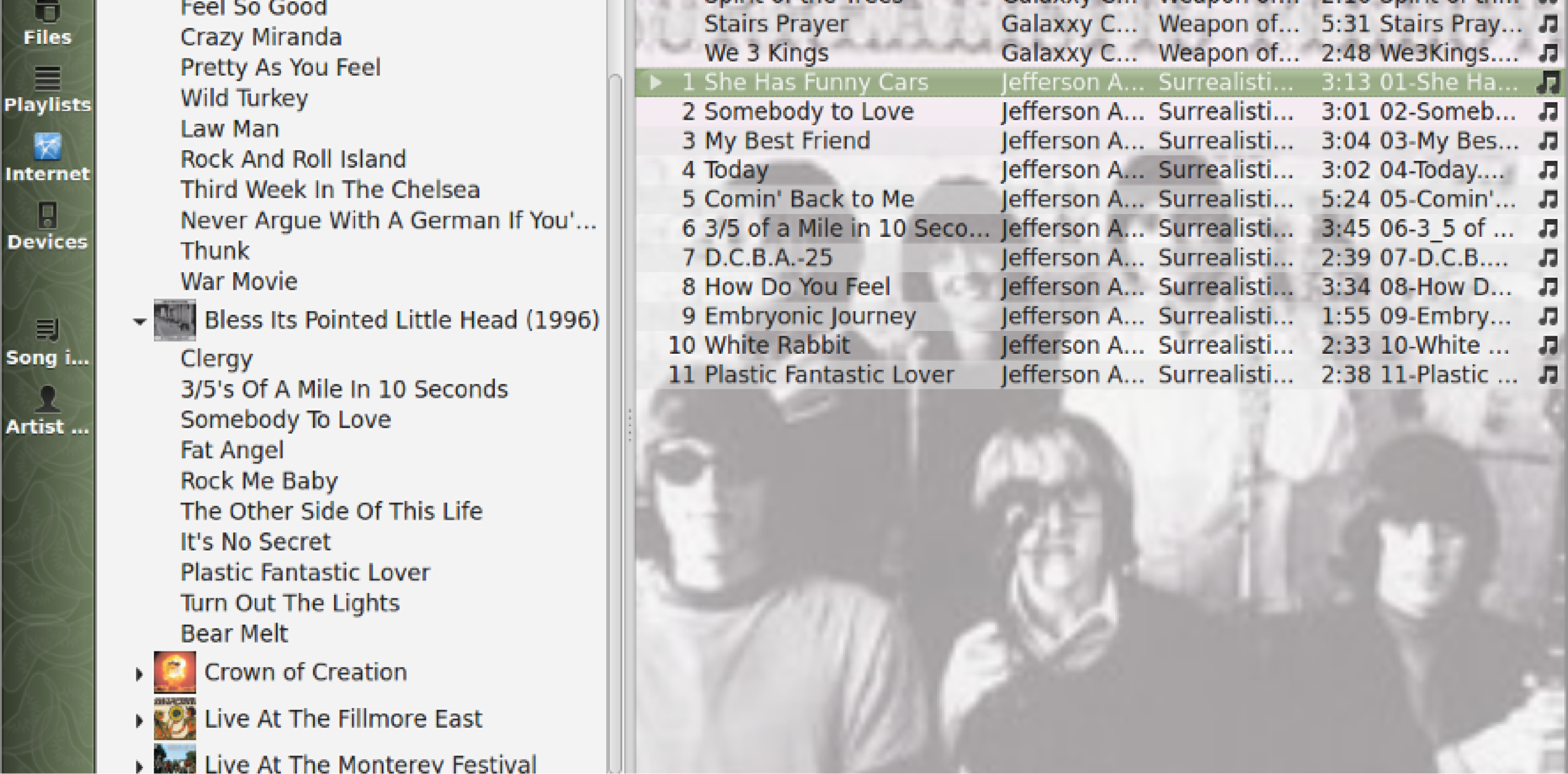
```
sudo -u pi screen -d -m bash -c "/usr/local/sbin/startpulse.sh"
exit 0 ## add the new line before this one
```

Fix a Minor Issue

I found that over time the delay on the various pulsecasts would drift, where this did not appear to be a problem with a desktop pc running linux.

Just add to the pi user's crontab the following line:

```
00 6 * * * /usr/bin/pkill pulseaudio
```


Files
Playlists
Internet
Devices
Song i...
Artist ...
Feel So Good
Crazy Miranda
Pretty As You Feel
Wild Turkey
Law Man
Rock And Roll Island
Third Week In The Chelsea
Never Argue With A German If You'...
Thunk
War Movie
Bless Its Pointed Little Head (1996)
Clergy
3/5's Of A Mile In 10 Seconds
Somebody To Love
Fat Angel
Rock Me Baby
The Other Side Of This Life
It's No Secret
Plastic Fantastic Lover
Turn Out The Lights
Bear Melt
Crown of Creation
Live At The Fillmore East
Live At The Monterey Festival
Stairs Prayer Galaxxy C... Weapon of... 5:31 Stairs Pray...
We 3 Kings Galaxxy C... Weapon of... 2:48 We3Kings....
1 She Has Funny Cars Jefferson A... Surrealisti... 3:13 01-She Ha...
2 Somebody to Love Jefferson A... Surrealisti... 3:01 02-Someb...
3 My Best Friend Jefferson A... Surrealisti... 3:04 03-My Bes...
4 Today Jefferson A... Surrealisti... 3:02 04-Today....
5 Comin' Back to Me Jefferson A... Surrealisti... 5:24 05-Comin'...
6 3/5 of a Mile in 10 Seco... Jefferson A... Surrealisti... 3:45 06-3_5 of ...
7 D.C.B.A.-25 Jefferson A... Surrealisti... 2:39 07-D.C.B....
8 How Do You Feel Jefferson A... Surrealisti... 3:34 08-How D...
9 Embryonic Journey Jefferson A... Surrealisti... 1:55 09-Embry...
10 White Rabbit Jefferson A... Surrealisti... 2:33 10-White ...
11 Plastic Fantastic Lover Jefferson A... Surrealisti... 2:38 11-Plastic ...

Clementine

I've been using this open source cross platform player for years. It is very good at managing local collections, and has integration for several streaming services and comes pre-configured with a lot of Internet radio stations.

In 2016 they even released an Android Remote Control Application.

More Options and the Future

Snapcast is a recent project which could be used in place of PulseAudio which has a time synch mechanism, but does not yet consider itself stable and is not packaged.

I've been looking at some MPD (Music Player Daemon) alternatives, Nuvola and Mopidy, because they support a variety of client front ends and support more streaming services.

PulseAudio Documentation

https://gavv.github.io/blog/pulseaudio-under-the-hood/

https://www.freedesktop.org/wiki/Software/PulseAudio/

https://wiki.archlinux.org/index.php/PulseAudio

Clementine

https://www.clementine-player.org/

Reveal.initialize({ // ... width: "100%", height: "100%", margin: 0, minScale: 1, maxScale: 1 });