

**schlizbäda**

# U s e r ' s   M a n u a l

Raspiblaster

---

Raspiblaster – a CD Player Based on the Raspberry Pi

---



Simplified 2-clause BSD License  
© 2001–2018 by Audacious developers and others

Datum:  
13.04.2018



The Linux mascot *Tux* was created by *Larry Ewing* using the GNU Image Manipulation Program (GIMP) <mailto:lewing@isc.tamu.edu>

# Contents

<b>List of Figures</b>	<b>4</b>
<b>List of Tables</b>	<b>5</b>
<b>1 Preface</b>	<b>6</b>
1.1 Legal Notes . . . . .	6
1.2 Acknowledgements . . . . .	8
1.3 Conventions of this Manual . . . . .	9
1.4 Abstract . . . . .	10
1.5 Parts List . . . . .	11
1.6 Technical Specifications . . . . .	12
1.6.1 Power Supply . . . . .	12
1.6.2 Audio Amplifier (HifiBerry MiniAMP) . . . . .	12
1.6.3 Raspberry Pi . . . . .	12
1.6.4 CD Drive (LG GP50NW40) . . . . .	13
<b>2 Operating Instructions</b>	<b>15</b>
2.1 Power On . . . . .	15
2.2 Secure Shut Down and Power Off . . . . .	15
2.3 Volume Control . . . . .	16
2.4 Starting the Audio Player <i>audacious</i> . . . . .	16
2.5 Insert CD and Start Playback . . . . .	17
2.6 Operating . . . . .	18
2.7 Playlist . . . . .	19
2.8 Playback of Audio Files On Other Storage Media . . . . .	20
2.9 Handling and Cleaning of CDs . . . . .	20
2.10 Error Handling . . . . .	21
2.10.1 Raspiblaster doesn't boot . . . . .	21
2.10.2 The eject push button of the CD ROM drive doesn't work . . . . .	21
2.10.3 The screen is black . . . . .	22
2.10.4 No Audio Signal . . . . .	22
<b>3 Installation and Configuration</b>	<b>23</b>
3.1 Configure Raspbian . . . . .	25

3.2	Install HiFiBerry MiniAMP . . . . .	26
3.2.1	Configure ALSA in the file <i>/etc/asound.conf</i> . . . . .	27
3.2.2	Enhance ALSA for Volume Control . . . . .	27
3.3	Start Up the CD ROM Drive on the Raspberry Pi . . . . .	29
3.3.1	Detecting the External CD ROM drive on the USB port . . . . .	30
3.3.2	Installation of Linux Package <i>eject</i> . . . . .	30
3.4	Compile <i>audacious</i> on the Raspberry Pi . . . . .	30
3.4.1	Download and Uncompressing of the Current Source Code . . . . .	31
3.4.2	Compilation of libaudclient-3.5-rc2 . . . . .	32
3.4.3	Compilation of audacious-3.9 . . . . .	32
3.4.4	Compilation of audacious-plugins-3.9 . . . . .	32
3.4.5	Code Adjustments of <i>audacious</i> for the Raspiblaster . . . . .	36
3.4.6	Usage of the IDE <i>eclipse</i> for Code Analysis and Debugging of <i>audacious</i> on a PC . . . . .	37
3.4.7	Adjustment of the Source Code Part 1: Implementation of the <i>eject</i> Feature . . . . .	38
3.4.8	Adjustment of the Source Code Part 2: Special feature "eject lock" for Raspberry Pi . . . . .	38
<b>4</b>	<b>Hardware mods</b>	<b>39</b>
4.1	Raspberry Pi3 . . . . .	39
4.2	Raspberry Pi DSI-Display Touch . . . . .	39
4.3	HifiBerry MiniAMP . . . . .	39
4.4	LG GP50NW40 . . . . .	39
4.5	S-USV – battery operation? . . . . .	39
4.6	Power supply . . . . .	39
4.7	description of “eject lock” via GPIO4 . . . . .	39
<b>5</b>	<b>Prospect</b>	<b>40</b>
5.1	battery operation . . . . .	40
5.2	Push buttons and rotary encoder as hardware via GPIO for volume control etc.	40
5.3	media player for USB flash drives . . . . .	40
5.4	rip CDs using the software <i>abcde</i> . . . . .	40
5.5	DVD playback (using kodi) . . . . .	40
5.6	Bluetooth receiver . . . . .	40
5.7	FM and/or DAB+ radio . . . . .	40

## List of Figures

2.1	Raspbian Stretch Desktop . . . . .	15
2.2	Shut Down Raspiblaster . . . . .	16
2.3	Insert CD into Tray . . . . .	17
2.4	System Dialogue Box “Removable medium is inserted” . . . . .	17
2.5	Handling of CDs . . . . .	20
3.1	Connect from Windows PC to Raspberry Pi via <i>PuTTY</i> . . . . .	24
3.2	<i>PuTTY</i> remote console of Raspberry Pi on the Windows PC . . . . .	24
3.3	alsamixer, a console mixing tool . . . . .	29

## List of Tables

1.1	Licensing of third party graphics . . . . .	8
1.2	Conventions of this documentation . . . . .	9
1.3	Parts list . . . . .	11

# 1 Preface

Thank you for interesting in schlizbädas Raspiblaster.

In addition to a CD player you are also concerned with a complete computer based on a Raspberry Pi which offers you as a hobbyist a lot of other possibilities. Read this manual to get to know this device and possibly even to rebuild it.

## 1.1 Legal Notes

When designing the Raspiblaster, it was taken care to use only software that is provided under a free license such as FreeBSD, GNU GPL or similar.

### Trademarks

Some names used in this document may be trademarks. The use of these trademarks by third parties for their purposes may infringe the rights of the holders.

### Links

This manual contains links to external sites on the internet. Despite linking the author schlizbäda does appropriate these contents, since they are not within his sphere of influence! At the time of linking, there were no illegal contents noticeable. It is not reasonable for the author to check the links permanently for any changes that might violate the law. However, if current or future content should be illegal, the author may be contacted by e-mail to <mailto:schlizbaeda@gmx.de>. Appropriate actions will then be taken to remove the affected link(s).

### DRM

The official reading of the inventors of this nuisance is *Digital Rights Management*, but groups like the Free Software Movement interpret these technical methods as *Digital Restriction Management*, since many of these procedures are not compatible to the most open source licenses.

However, this **doesn't mean** that DRM can be ignored! Around 2003 in Germany the end consumer(!) was made liable by law to recognize and obey technically non-functioning DRM (e.g. a non-working “copy protection” on audio CDs not according to the Red Book specification, so-called un-CDs) and to take appropriate measures to comply with DRM.

In the other EU countries there do exist similar laws.

**On Raspiblaster it cannot be guaranteed that all DRM measures will be technically detected and recognized. Therefore they cannot be taken into account automatically!** This applies in particular to un-CDs labeled with a copy protection icon: Even the private digital copy of such a product is strictly speaking illegal but currently these “violations” usually aren't prosecuted by law...

### **FreeBSD License of the audio player *audacious* Current stable release: 3.9 (August 19, 2017)**

Copyright 2001-2017 Audacious developers and others

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation provided with the distribution.

This software is provided "as is" and without any warranty, express or implied. In no event shall the authors be liable for any damages arising from the use of this software.

<https://audacious-media-player.org/download>

Based on this license, the source code of *audacious* on *Current stable release: 3.9 (August 19, 2017)* has been updated for use on the Raspberry Pi by schlizbäda. That program has got an “eject” functionality, since pressing the eject hardware button of the CD ROM drive sometimes causes hang-ups. Further information can be found in chapter 3.4.

The source code of the changed software can be downloaded from this link:

<https://github.com/schlizbaeda/audacious-raspiblaster>





### **Picture Licenses**

All relevant photographs and technical illustrations in this document originate from the author schlizbäda and are hereby published by him under the *Creative Commons* License **CC-BY-SA 3.0**. They may therefore be used and re-published by anyone – modified or

unmodified – by naming the author under the same conditions:



However this manual contains some icon graphics from other sources using these licenses:

	<p>The author <i>Larry Ewing</i> released this picture under the following condition:</p> <p>“Permission to use and/or modify this image is granted provided you acknowledge me – lewing@isc.tamu.edu – and the GIMP if someone asks.”</p>
	<p>Logo of the Raspberry Pi Foundation:  <a href="https://static.raspberrypi.org/files/Raspberry_Pi_Visual_Guidelines_2018.pdf">https://static.raspberrypi.org/files/Raspberry_Pi_Visual_Guidelines_2018.pdf</a></p>
	<p>Desktop icon of <i>audacious</i> audio player:  probably CC-BY-SA?</p>
	<p>Alternative desktop icon for an audio player on Raspiblaster:  CC-BY-SA U.S. (author: Wallpaper FX, <a href="http://www.wallpaperfx.com/">http://www.wallpaperfx.com/</a>)</p>

**Table 1.1:** *Licensing of third party graphics*

## 1.2 Acknowledgements

schlitzbäda would like to welcome the following users of the German Raspberry Pi Forum (<https://forum-raspberrypi.de>):

**@hyle** <https://forum-raspberrypi.de/user/36638-hyle/>:

This user gave many valuable hints to the realisation of Raspiblaster, especially the idea to lock the eject push button of the CD ROM drive.

**@smutbert** <https://forum-raspberrypi.de/user/21740-smutbert/>:

A real expert about ALSA and its installation and configuration.

**@rpi444** <https://forum-raspberrypi.de/user/8097-rpi444/>,




**@Tell** <https://forum-raspberrypi.de/user/9272-tell/>:

Both users showed the right way to tackle compiling large foreign C/C++ projects successfully.



## 1.3 Conventions of this Manual

The following design conventions are used for this manual:

 <b>Danger!</b>   Text	Disregarding this warning can result in <b>personal injury</b> and equipment damage!
 <b>Caution</b>   Text	Disregarding this warning can result in equipment damage!
 <b>Hint</b>   Text	A note with additional information or relevant explanation of a certain functionality
[Button]	Marking of software buttons
<i>"Menu item"</i>	Marking of software menu items
pi \$ RPi command	command line on the Raspberry Pi
PC \$ PC command	command line on a Linux PC
win> PC command	command line on a Windows PC (alternatively to Linux)
<i>#Comment</i>	comment text in a command line
Message	Marking of software messages

**Table 1.2:** *Conventions of this documentation*

## 1.4 Abstract

Raspiblaster is a CD player with built-in stereo amplifier (**HifiBerry MiniAMP**) and loudspeakers. It has got some similarity (at least in technical terms) to the devices called *boom boxes* and especially in Germany also called *Ghettoblaster* and which are known since the 1980s. The Raspiblaster consists of a Raspberry Pi 3B which is internally connected to a CD ROM drive (**LG GP50NW40**, which is actually a DVD writer). The Raspiblaster is operated via the 7-inch Raspberry Pi DSI-Display Touch issued by the RPi Foundation. At the back side there is a connector for external loudspeakers. Two of the Raspberry Pi's USB connectors and its Ethernet socket are accessible by the user via housing connectors.

The power supply is done by a mains power cord. Inside the Raspiblaster there is a built-in power supply device ( **Meanwell RS-25-05: 5V, 5A**) which provides enough power reserves.

Raspiblaster uses the open source software *audacious* in *Current stable release: 3.9 (August 19, 2017)* for playback of audio CDs.

### Prospect on Further Enhancements

- battery operation
- Push buttons and rotary encoder as hardware via GPIO for volume control etc.
- media player for USB flash drives
- DVD playback (using kodi)
- Bluetooth receiver
- FM and/or DAB+ radio

## 1.5 Parts List

x	Manufacturer	Type	Description	Distributor	Order#
1	bopla	68626120	Botego BO 62612 308mm x 257mm x 81mm	?	–
1	Schurter	6762	Power Supply Combination	Bürklin	41 F 139
2	RND	170-00020	Microfuse “time delay” 1A	reichelt	RND 170-00020
1	MeanWell	RS 25-5	Switching Power Supply 25W, 5V, 5A	reichelt	SNT RS 25 5
1	Foundation	-	Raspberry Pi DSI-Display Touch	raspiprojekt	TS7DSI
1	Foundation	RPi 3B	Raspberry Pi 3B	raspiprojekt	RASPI3B
1	HifiBerry	MiniAMP V1.0	Stereo Amplifier 2x3W maximum power	reichelt	RPI HB MINI AMP
2	Visaton	SC 8 N 8Ohm	Speaker 30W, 8Ohm	reichelt	VIS SC 8N-8
2	Visaton	GRILLE FRS 8	Speaker Cover 82mm x 82mm	RS	4538953
1	?		Speaker Connector	RS	392683
1	Marquardt	1839.0105	Rocker Switch 2P I/O/II 30mm x 22mm black	RS	7410823
1	LG	GP50NW40	Slim Line DVD-Writer	<i>local store</i>	
2	Neutrik	NAUSB-W-B	USB Socket Type “A”	reichelt	NAUSB-WB
1	Neutrik	NE8FDX-P6	Ethernet (LAN) Socket Cat.6A	reichelt	CAT6A BU BK
1	schlitzbäda	–	Relay Control “eject-lock”		

**Table 1.3:** *Parts list*

Sundries like screws, connection cables (ethernet, USB, ...) aren't listed in this parts list.

## 1.6 Technical Specifications

### 1.6.1 Power Supply

Voltage Range	88VAC – 264VAC
Frequency Range	47Hz – 63Hz
AC Current	0.7A/115VAC 0,4A/230VAC
DC Voltage	5V
Rated Current	5A
Rated Power	25W

### 1.6.2 Audio Amplifier (HifiBerry MiniAMP)

Amplifier Class	HifiBerry MiniAMP V1.0 (class-D amplifier)
Music Power	2 x 3W (max.)
Sample Rate	44,1kHz – 192kHz

### 1.6.3 Raspberry Pi

Raspberry Pi Version	Raspberry Pi 3B
SoC (Broadcom)	BCM2837
Architecture	ARM Cortex-A53 (quad core)
Clock Rate CPU	1200MHz
Clock Rate GPU	300MHz/400MHz
Main Memory	1GB
Non-volatile Memory	depends on the used micro SD card To run “Raspbian Stretch Desktop” an 8GB sized SD card is necessary as minimal storage capacity

#### Used GPIOs

GPIO 4	Pin <b>7</b> : signal for relay control of eject-lock
GPIO 2, 3	Pins <b>3, 5</b> : MiniAMP I2C bus
GPIO 18 – 21	Pins <b>12, 35, 38, 40</b> : MiniAMP I2S bus
GPIO 26	Pin <b>37</b> : MiniAmp shut down power stage
ID SDA, ID SCL	Pins <b>27, 28</b> : I2C EEPROM containing device type data

**1.6.4 CD Drive (LG GP50NW40)**

Supported Discs	DVD-ROM (Single/Dual), DVD-RW, DVD-R, DVD+RW, DVD+R, DVD+R Double layer, DVD-R Dual layer, DVD-RAM, M-Disc (DVD+R SL), CDDA (CD Digital Audio) & CD-Extra, CD-Plus, CD-ROM, CD-ROM XA-Ready, CD-I FMV, CD-TEXT, CD-Bridge, CD-R, CD-RW, Photo-CD (Single- & Multi-Session), Video CD, DVD-VIDEO
Read Speed	DVD-R/RW/ROM: 8x/8x/8x max. DVD-R DL: 8x max. DVD-RAM (Ver.2.2 & Higher): 6x max. M-Disc (DVD+R SL): 8x max. DVD-Video (CSS Compliant Disc) : 4x max. DVD+R/+RW: 8x/8x max. DVD+R DL: 8x max. CD-R/RW/ROM: 24x/24x/24x max. CD-DA (DAE): 24x max.
Write Speed	DVD-R: 2x, 4x, 8x DVD-R DL: 2x, 4x, 6x DVD-RW: 2x, 4x, 6x DVD-RAM (Ver. 2.2 & higher): 2x, 3x, 5x M-Disc (DVD+R SL): 4x DVD+R: 2.4x, 4x, 8x DVD+R DL: 2.4x, 4x, 6x DVD+RW: 2.4x, 3.3x, 4x, 8x CD-R: 10x, 16x, 24x CD-RW: 4x, 10x, 16x, 24x
Interface	USB 2.0
Input Voltage	5V DC
Power Dissipation	1,6A

### **Data Transfer Rate**

Sustained	CD-ROM: 3,600 kB/s (24x max) DVD-ROM: 11.08 MB/s (8x max) CD-ROM: 140 ms (typisch) DVD-ROM: 160 ms (typisch) DVD-RAM: 200 ms (typisch)
Buffer capacity	0.75 MB
MTBF	60000 Power On Hours (Duty Cycle 10%)

### **Operating Environment**

Temperature	5°C to 40°C
Humidity	15% to 85%

### **Storage Environment**

Temperature	-30°C to 60°C
Humidity	10% to 90%

## 2 Operating Instructions

This chapter is a classical operator's guide for users who intend to use the Raspiblaster straight forward without modifying the software or hardware of the device.

### 2.1 Power On

To switch on the Raspiblaster, set the mains switch at the back of the unit to position I. The boot process of the Raspberry Pi will start and the operating system on the SD card (Raspbian Stretch Desktop from 13-03-2018) is loaded to the main memory. After approx. 10 seconds the boot process is completed and the desktop appears on the display (see figure 2.1).

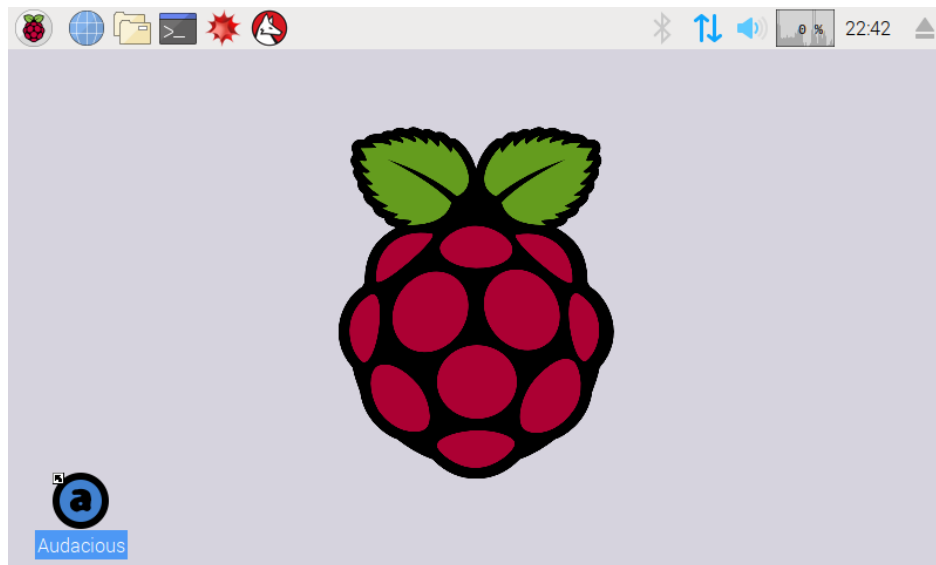

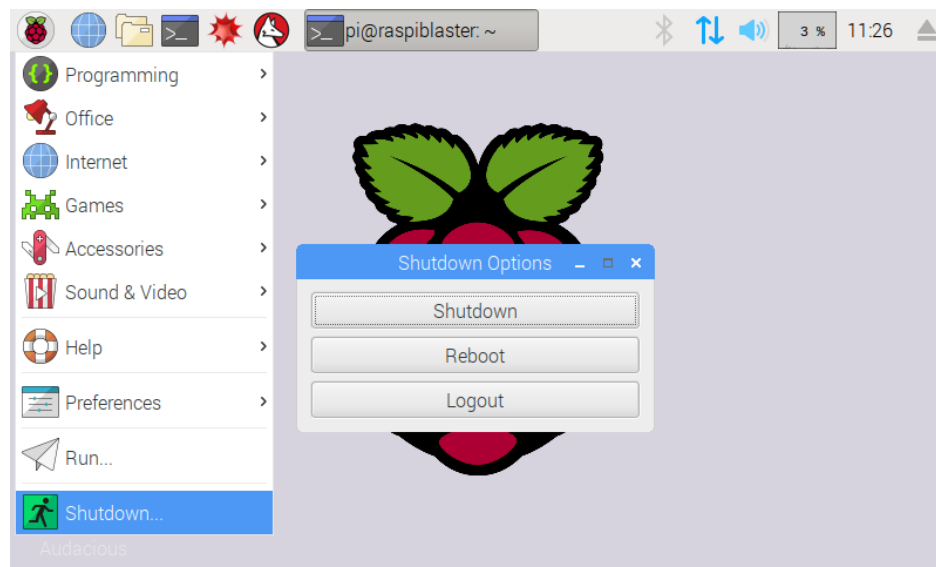


Figure 2.1: *Raspbian Stretch Desktop*

### 2.2 Secure Shut Down and Power Off

To avoid data corruptions please shut down the Raspiblaster securely! First, shut down the operating system (Raspbian Stretch Desktop) via the menu item "→**Shutdown...**→**Shutdown**". After the display has turned black, wait again for approx. 10 seconds before switching off the power via the mains switch on the back of the unit.

Figure 2.2: *Shut Down Raspiblaster*


### Caution

Take care to shut down the Raspberry Pi completely before switching off the power supply! If the power supply is removed *exactly* when the device is performing a write access to the SD card, this write access may be incomplete or incorrect and may damage the file system on the SD card in an undefined way. Though in most cases this won't happen, avoid switching off the mains power without shutting down to prevent data loss on the SD card.

## 2.3 Volume Control

The taskbar of Raspbian shows a speaker icon in the upper right corner. Clicking this icon will open a graphical slider control that can be used to adjust the playback volume. The set current volume is roughly indicated by symbolic acoustic waves (1 – 3) in front of the speaker icon. A red **x** means muting of the audio reproduction.

## 2.4 Starting the Audio Player *audacious*

The software *audacious* – *Current stable release: 3.9 (August 19, 2017)* is used to play CDs on the Raspiblaster. To start this software use either the menu item " → **Sound & Video** → **Audacious**" or double-click on the “Audacious” icon at the bottom left corner of the desktop.



## 2.5 Insert CD and Start Playback

Open the CD drive using the eject push button and insert the CD to play (see figure 2.3). Close the CD tray manually.



Figure 2.3: Insert CD into Tray

The drive needs a short time to read the table of contents of the CD. Raspbian will show the system dialogue box ***"Removable medium is inserted"***. Now the CD is ready for playback. This dialogue box can be closed by clicking the button [Cancel].

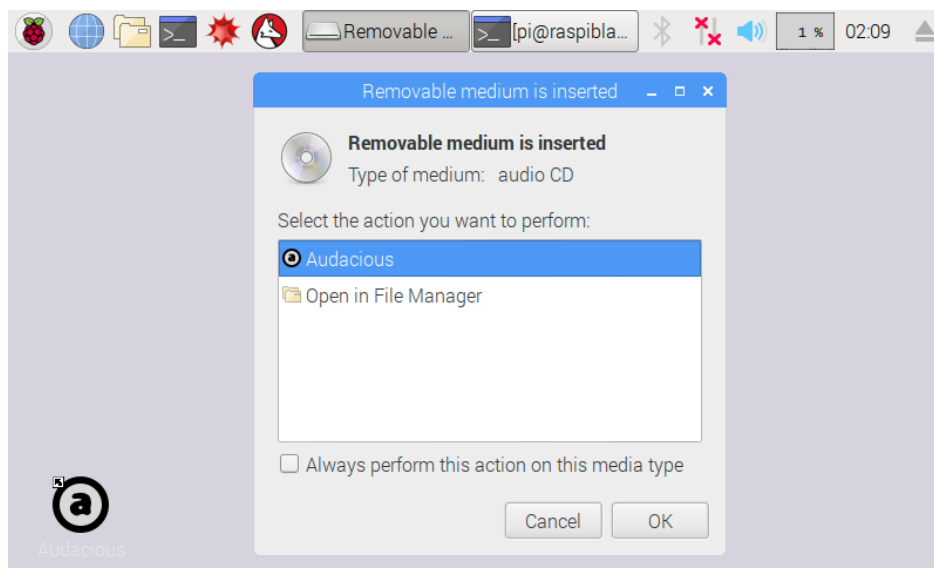


Figure 2.4: System Dialogue Box *"Removable medium is inserted"*

To start the playback of the inserted CD use *audacious*' menu item "**Services→Play CD**". *audacious* inserts all tracks of the CD into a playlist named "Now Playing". If this playlist is already containing some entries all these entries will be removed and replaced by the tracks of the current CD! See section 2.7 for details. CDs following the first version of the CDDA specification are lack of any meta data like song title, artist ect. Thus the playlist will be filled with denominations like "Track <x>" and the album name "Audio CD". Most of newer CDs contain meta data in the format "CD-TEXT". When reading such CDs the CD-TEXT information will be taken into the playlist.

**Hint**

To get automatically track information even for older CDs of the first generation, *audacious* uses the internet database *Compact Disc Database* (CDDb) for receiving according meta data. The length of all tracks of the CD is sent to CDDb. The CDDb server requests its data base for meta data matching to the transmitted track lengths.

To use the feature of CDDb data base the Raspiblaster has to be connected to the internet!

<https://de.wikipedia.org/wiki/CDDb>

## 2.6 Operating

The graphical user interface of the audio player *audacious* provides all controls which are known from a "classical" CD player.

### Play/Pause

Starting and pausing playback

### Skip |«

Select previous track

### Skip »|

Select next track

## Rewind and fast forward

Rewinding and forwarding of a track is done in *audacious* by moving the current position of the track progress bar.



### Hint

This method allows really fast jumps to a certain position (or the beginning) of a track. So the button *Skip* /« jumps **always** to the previous track. This is in contrast to many CD players where you might expect a jump to the beginning of the currently playing track.

## Stop

Stop playback

## eject

Opens the CD tray of the drive



### Hint

The feature *eject* was added by schlizbäda to the Raspberry Pi version of *audacious* for Raspiblaster.

## 2.7 Playlist

*audacious* manages all tracks to play in so-called playlists. This is ideal for creating long compilations of audio files stored on external storage devices like USB flash drives. *audacious* is able to provide more than one play list at a time to sort the music by genres like Heavy Metal, Classics, Rap, Folk. The playback of tracks is given by the currently active playlist.



### Hint

When starting CD playback via the menu item "**Services**→**Play CD**" all tracks of the CD are inserted to the playlist "Now Playing". Old tracks in this special playlist will be removed! But after the automatic creation of this list you may add further tracks from a USB drive or remove other tracks.

When changing the name of this playlist and starting a new CD playback by the menu item "*Services*→*Play CD*" a new playlist called "Now Playing" will be created again.

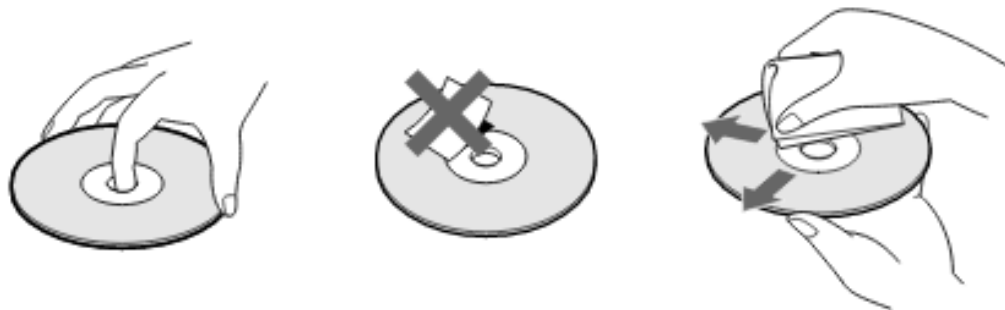
## 2.8 Playback of Audio Files On Other Storage Media

*audacious* is compatible to a lot of audio formats like MP3 or FLAC which may be stored on a CD ROM or a USB flash drive. Even audio files stored on the SD card of the Raspberry Pi can be used.

When loading video files *audacious* only plays their audio track.

## 2.9 Handling and Cleaning of CDs

- Grab the CD always at its edge to keep it clean
- Don't stick paper or tape on the CD's surface
- Protect CDs from heat:  
Keep them away from direct sun light and heating devices. Don't leave them in a car parking in the sunshine
- Always store the CD in its sleeve after listening.



**Figure 2.5:** *Handling of CDs*

- Clean the CD with a lint-free cleaning cloth, wiping from the inside out.
- Do not use solvent-containing cleaning agents such as petrol or thinner.

## 2.10 Error Handling

If the device behaves incorrectly, first check whether the error can be corrected with simple means. Especially systems on which complex software (e.g. the operating system Raspbian or *audacious*) is running, are highly configurable and not any unexpected behavior is really an error but can often be eliminated by simple measures.

The following list is not exhaustive:

### 2.10.1 Raspiblaster doesn't boot

- The SD card is not inserted correctly (anymore).
- The file system on the SD card was damaged by incorrect power off
- Fundamental files on the SD card have been deleted or moved

Each reason makes it necessary to open the case of Raspiblaster for removing the Raspberry Pi's SD card, because the Raspberry Pi cannot boot from it. The SD card has to be corrected or reloaded on an external PC!



#### **Danger!**

Due to the risk of a dangerous electric shock remove mains plug before opening the case!

### 2.10.2 The eject push button of the CD ROM drive doesn't work

- During playback of a CD the eject push button is locked

Before ejecting a CD the software button [Stop] in *audacious* must be actuated to stop playback. This releases the hardware push button of the drive.



#### **Hint**

The software button [eject] of *audacious* allows opening the CD ROM drive at any time.

### 2.10.3 The screen is black

- Is the screen saver active?  
Touch the display for activating it
- Is the power supply working? Are the cables connected correctly?
- Are the micro-fuses inside the mains switch OK?

Check the mains cord on tight connection.

Does the wall socket really supply mains voltage?

Check the micro-fuses in the Raspiblaster. The device must always be fused on both poles!



#### **Danger!**

Before checking the micro-fuses remove the mains plug, even if the mains switch is in position 0! When the fuse holder is open, there is the risk of a dangerous electric shock from touching the contacts!



#### **Caution**

When changing the micro-fuses take care of using new fuses of the same type:  
250V max, 1A slow blow

### 2.10.4 No Audio Signal

- Is the volume set correct?  
A red-marked **x** on the speaker icon indicates a muted playback.
- Silent or complete mute part in the track?
- Is the CD playing?
- Are the audio sources (CD, USB drive) of the tracks in the playlist still available?
- Is the CD erroneous or scratchy?

A test with another CD is recommended

- Using external speakers: Are the speakers connected correctly?

Take a look on two topics:

\* correct cabling

\* correct position II of speaker switch on top of the Raspiblaster

Set speaker switch on position I to check if there is an audio signal on the internal speakers.

## 3 Installation and Configuration

This chapter describes the installation of the operating system and of all parts of the software. The main focus is on implementing a toolchain to compile the open source software *audacious* on the Raspberry Pi.



### Danger!

Some of the actions described here require opening the housing of the Raspiblaster to reach the SD card of the built-in Raspberry Pi. To avoid dangerous electric shocks when handling inside the device, the mains plug must be disconnected before opening!



### Hint

This tutorial was tested on a Raspberry Pi running Linux `raspibuster 4.14.30-v7+ #1102 SMP Mon Mar 26 16:45:49 BST 2018 armv7l GNU/Linux`.

The PC commands differ depending on the used operating system (Linux or Windows). The differences are described when they occur.

In this manual a lot of commands have to be started either on the Raspberry Pi or on a PC in a terminal window. To copy data from the PC to the Raspiblaster both computers must be connected via LAN (or WLAN). The data exchange is done via *ssh* (*secure shell*). Instead of entering all commands on the Raspberry Pi via a second keyboard it is useful to open a terminal window on the PC and to initiate a *ssh* connection for doing all keyboard entries on the PC. Another big advantage is that you can open this pdf file on your PC, mark the Raspberry Pi commands and insert them via *copy+paste* into the *ssh* terminal. Only the first part of this tutorial (“*raspi-config*”) can’t be executed *remote* because *ssh* has to be enabled on the Raspberry Pi using this command first.



### Hint

On a Windows PC *ssh* isn’t installed by default. Install the software *PuTTY* to circumvent this. You may download it from <https://putty.org/>:  
<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

### 3 Installation and Configuration

To connect from Windows via *ssh* to the Raspberry Pi call the Windows start menu item “PuTTY64”. To log in you will need the IP address of the Raspberry Pi and the user *pi* and its password (see figures 3.1 und 3.2).

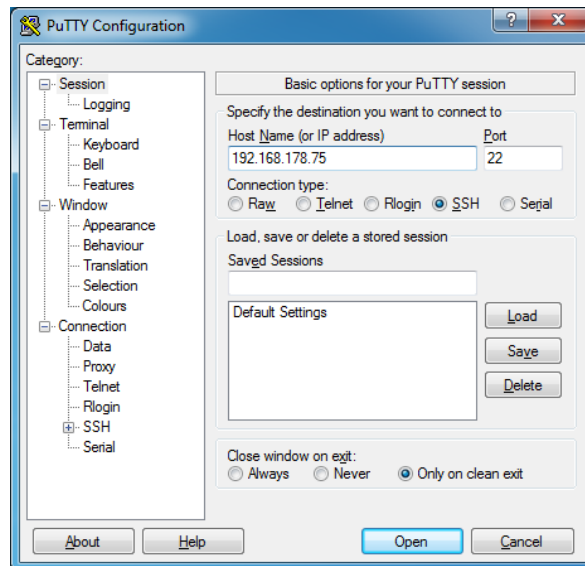


Figure 3.1: Connect from Windows PC to Raspberry Pi via PuTTY

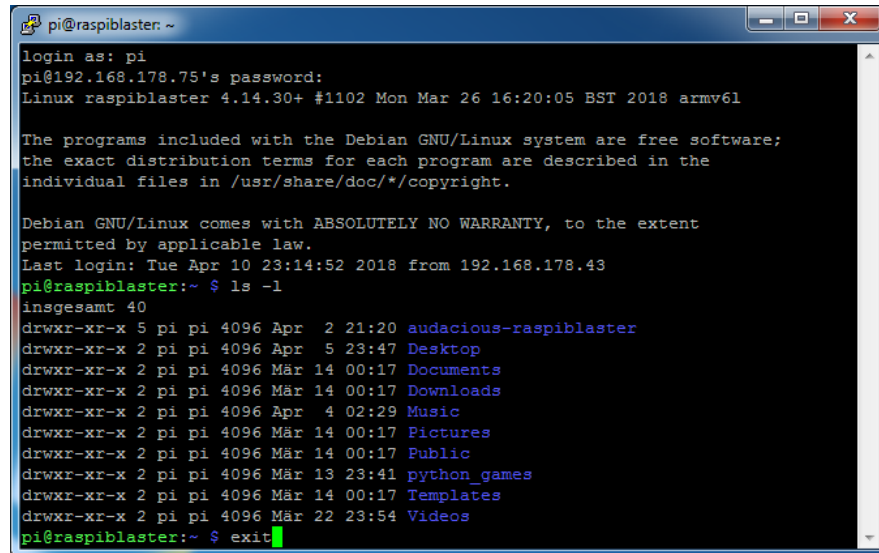


Figure 3.2: PuTTY remote console of Raspberry Pi on the Windows PC



## 3.1 Configure Raspbian

### raspi-config

```

pi $ sudo raspi-config
1 Change User Password                #e.g. "raspiBlaster"
2 Network Options      -> N1 Hostname  # "raspiBlaster" instead of "raspberrypi"
4 Localisation Options -> I1 Change Locale
                                -> I2 Change Timezone
                                -> I3 Change Keyboard Layout
                                -> I4 Change Wi-fi Country #really important adjustment!
5 Interfacing Options  -> P2 SSH

```

### Update the System

```

pi $ sudo apt update  #update of apt meta data
pi $ sudo apt upgrade #upgrade of all installed packages
pi $ sudo apt update  #sometimes necessary, e.g. when using Raspbian image from 13-03-2018

```

### Shrink Down Image to approx. 6,8GB

TODO: \_\_\_\_\_

This step isn't mandatory but it serves two advantages:

- The image is small enough to fit on *any* 8GB SD card
- The time for backing up and flashing SD cards is reduced significantly

describe  
either my  
way or  
pishrink  
by  
@framp

### Orientation of the original Raspberry Pi DSI-Display Touch

The graphics orientation should be rotated by 180° due to a better viewing angle. Add the line `lcd_rotate=2` in the file `/boot/config.txt`.

```

pi $ sudo nano /boot/config.txt
:
:
lcd_rotate=2
:
:

```

## Adjustment of the Raspbian PIXEL desktop

Adjust look and feel: *#depending on personal preferences*


Menu item "→**Preferences**→**Appearance Settings: Tab 'Desktop'**"

Checkbox "**Wastebasket**": disable *#children don't need this*

Combobox **Layout**: "**Centre image on screen**" ▾

Combobox **Picture**: "**raspberry-pi-logo.png**" ▾ *#use the raspberrry pi logo on the desktop*

Reduce Double Click Speed for Touch Screen Operating:

Menu item "→**Preferences**→**Mouse and Keyboard Settings: Tab 'Mouse'**"

Slider "**Double-click Delay**": 1990ms *#the default of 250ms is too short for a touch display*

Disable Screen Saver: *#Optional!*

add the following line at section [Seat: \*]:

```
pi $ sudo nano /etc/lightdm/lightdm.conf
:
[Seat: *]
xserver-command=X -s 0 -dpms
:
```

## 3.2 Install HiFiBerry MiniAMP

Resources:

<https://www.hifiberry.com/shop/boards/miniamp>

<https://www.hifiberry.com/build/documentation/configuring-linux-3-18-x>

<https://support.hifiberry.com/hc/en-us/articles/205377202-Adding-software-volume-control>



### Hint

| Use the drivers of HiFiBerry DAC for HiFiBerry MiniAMP V1.0!

**adjust** */boot/config.txt*

Remove the entry for the on board audio module (HDMI and 3,5mm jack):

The entry `dtoverlay=audio=on` must be deleted or marked as comment,

use the "Device Tree Overlay File" for MiniAMP instead:

```
pi $ sudo nano /boot/config.txt
:
#dtparam=audio=on
:
dtoverlay=hifiberry-dac
:
```

### 3.2.1 Configure ALSA in the file `/etc/asound.conf`



#### Hint

Please check first if the file `.asoundrc` exists in the home directory of user `pi`. This file contains user-specific ALSA settings and overrides the settings in `/etc/asound.conf`!

```
pi $ rm /home/pi/.asoundrc
```

Contents of `/etc/asound.conf`:

```
pi $ cat /etc/asound.conf
```

```
pcm.!default {
type hw card 0
}

ctl.!default {
type hw card 0
}
```

```
pi $ reboot
```

```
pi $ aplay -l
```

```
**** List of PLAYBACK Hardware Devices ****
```

```
card 0: sndrpihifiberry [snd_rpi_hifiberry_dac], device 0: HifiBerry DAC HiFi pcm5102a-hifi-0 []
```

```
Subdevices: 0/1
```

```
Subdevice #0: subdevice #0
```

### 3.2.2 Enhance ALSA for Volume Control

Additional Contents of `/etc/asound.conf`:

```
pi $ cat /etc/asound.conf
```

```
# Settings of @smutbert from the German Raspberry Pi forum:
```

```
# https://forum-raspberrypi.de/user/21740-smutbert/
```

```
pcm.dmixer {
type dmix
ipc_key 1236
```

*3 Installation and Configuration*

---

```

slave.pcm "hw:sndrpihifiberry"
}
pcm.softvolume {
type softvol
slave.pcm "dmixer"
control.name "Master"
control.card sndrpihifiberry
}
pcm.!default {
type plug
slave.pcm "softvolume"
}

```

23-03-2018 TODO:

@smutbert's */etc/asound.conf* doesn't work on Raspbian Stretch 2018-03-13 with omxplayer:  
Playback of media file stops immediately at 0:00!

```

# Old Contents:
##pcm.!default {
## type hw card 0
##}
##ctl.!default {
## type hw card 0
##}
#
#pcm.hifiberryMiniAmp {
# type softvol
# slave.pcm "plughw:0"
# control.name "Master"
# control.card 0
#}
#pcm.!default {
# type plug
# slave.pcm "hifiberryMiniAmp"
#}

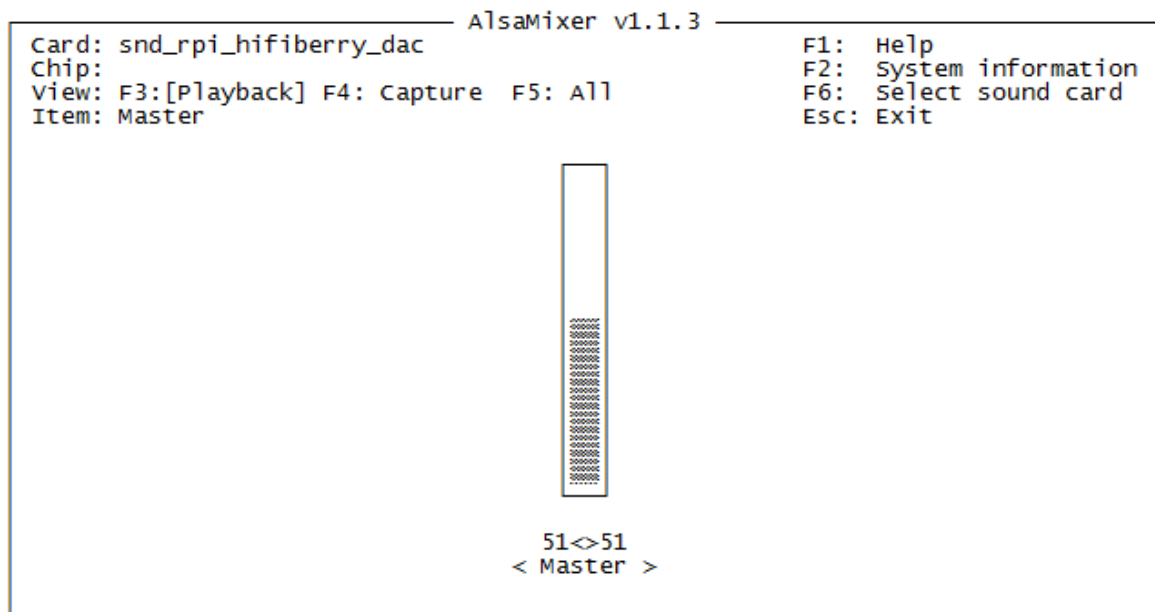
```

```

pi $ rm /home/pi/.asoundrc
pi $ reboot
pi $ speaker-test -c 2 #speaker-test -D hifiberryMiniAmp -c 2
pi $ alsamixer

```

Check  
why this  
doesn't  
work  
with omx-  
player...



**Figure 3.3:** *alsamixer*, a console mixing tool

*alsamixer* shows the control *Master* which is defined in the file */etc/asound.conf*.



#### Hint

Volume control is done now by ALSA!

*alsamixer* adjusts the same audio setting as the graphical volume control in the taskbar of Raspbian Desktop PIXEL.

Test:

```
pi $ omxplayer -o alsa "01 Saga - Wind Him Up.wav"
```

### 3.3 Start Up the CD ROM Drive on the Raspberry Pi

When an audio CD is inserted into the external CD ROM drive (resp. DVD burner) its contents is *not* mounted at */media/pi/<volume>!* Rather, a virtual(?) folder (or URL) is opened: *cdnda://sr0/*

Direct file access to this mount path is not available and even the *omxplayer* can't do here any magic!

⇒ We'll need a software CD player!

### 3.3.1 Detecting the External CD ROM drive on the USB port

```
pi $ lsusb #CD ROM drive isn't connected to USB
```

```
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
```

```
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp. SMC9514 Hub
```

```
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

```
pi $ lsusb #CD ROM drive is connected
```

```
Bus 001 Device 006: ID 0e8d:1887 MediaTek Inc.
```

```
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
```

```
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp. SMC9514 Hub
```

```
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

### 3.3.2 Installation of Linux Package *eject*

For software-controlled opening of the CD tray the software package *eject* is necessary which isn't installed in Raspbian by default:

```
pi $ sudo apt install eject #49,5kB archives, 225kB disk space uncompressed
```

```
pi $ eject #Opening of CD ROM drive
```

```
pi $ eject -t #Closing of CD ROM drive: Not supported by each drive type!
```

The call of `eject -t` can be used anyway to check on drives which don't support this feature, if its tray is opened or closed. When the tray is open `eject -t` gives an error message and exits with return code 1.

Despite the `man` page documentation the command `eject -i 1` **does not** prevent the drive (at least the used drive GP50NW40) from ejecting the tray when pressing the eject push button. The same behaviour can be found on my laptop computer.

## 3.4 Compile *audacious* on the Raspberry Pi

Initial tests using several audio players resulted in dogged hanging ups when pressing the eject push button during playback of an audio CD. It didn't matter which audio player software was used. Even the command `sudo kill -9 <process ID>` couldn't terminate the hanging process!

The first try was to prevent the CD tray from opening via the command `eject -i 1`. But as mentioned in the preceding chapter (section 3.3.2) this command doesn't work at least on the used drive GP50NW40. So the circuitry at the eject push button was opened and can be closed via a relay contact. This "eject lock" can be controlled by hardware via the GPIO pin 4.

So it was necessary to implement the eject lock logic into the used audio player software and

in addition a software eject button which performs a playback stop before. The source code had to be adjusted and recompiled.

### 3.4.1 Download and Uncompressing of the Current Source Code

At first the version *Current stable release: 3.9 (August 19, 2017)* of the *audacious* source code was downloaded. It consists of three sub projects as compressed `tar` archives, so-called *tar balls*:

- `libaudclient-3.5-rc2.tar.bz2`
- `audacious-3.9.tar.bz2`
- `audacious-plugins-3.9.tar.bz2`

Resources:

<https://audacious-media-player.org/download>

<https://distfiles.audacious-media-player.org/libaudclient-3.5-rc2.tar.bz2>

<https://distfiles.audacious-media-player.org/audacious-3.9.tar.bz2>

<https://distfiles.audacious-media-player.org/audacious-plugins-3.9.tar.bz2>

Assuming that the download was performed on a Linux PC the three tar balls must be copied via *ssh* onto the Raspberry Pi into the directory `/home/pi/audacious_raspiblaster`:

```
pi $ cd /home/pi
```

```
pi $ mkdir audacious_raspiblaster
```

```
PC $ scp *.tar.bz2 pi@raspiblaster:/home/pi/audacious_raspiblaster
```

The PC requests for the password of user `pi` on the remote host `raspiblaster` before transferring the files via *ssh* protocol from the PC to the Raspberry Pi.

#### Hint

On a Windows PC with installed *ssh* software package *PuTTY* the command in the Windows' console terminal is (`cmd.exe`):

```
win> pscp *.tar.bz2 pi@raspiblaster:/home/pi/audacious_raspiblaster
```

Unpacking the tar balls:

```
pi $ cd audacious_raspiblaster
```

```
pi $ tar -xvf audacious-3.9.tar.bz2
```

```
pi $ tar -xvf audacious-plugins-3.9.tar.bz2
```

```
pi $ tar -xvf libaudclient-3.5-rc2.tar.bz2
```

### 3.4.2 Compilation of libaudclient-3.5-rc2

```
pi $ cd /home/pi/audacious_raspiblasters/libaudclient-3.5-rc2
pi $ ./configure
```

The command `./configure` checks that several necessary software packages aren't installed yet:

```
pi $ sudo apt install libglib2.0-dev      #message: Cannot find Glib2!
If you are using binary packages based system, check that you have the corresponding -dev/devel
packages installed.
pi $ sudo apt install libdbus-1-dev      #message: No package 'dbus-1' found
pi $ sudo apt install libdbus-glib-1-dev #message: No package 'dbus-glib-1' found
pi $ ./configure
pi $ make #issues a lot of warning, but it compiles the project completely...
pi $ sudo make install
```

### 3.4.3 Compilation of audacious-3.9

```
pi $ cd /home/pi/audacious_raspiblasters/audacious-3.9
pi $ leafpad INSTALL #shows an installation procedure
pi $ ./configure
```

The command `./configure` checks that several necessary software packages aren't installed yet:

```
pi $ sudo apt install libglib2.0-dev      #message: No package 'glib-2.0' found
pi $ sudo apt install libgtk2.0 libgtk2.0-dev #message: No package 'gtk+-2.0' found
→Installation of these packages takes about 5 minutes on an Raspberry Pi 3B
pi $ ./configure
pi $ make -j4 #-j4 forces make to use all the four cores of the Raspberry Pi 3B
pi $ sudo make install
```

Due to the graphical user interface of *audacious* the next command has to be entered on the Raspberry Pi itself. It can't be done on the *ssh* terminal of the PC!

```
pi $ audacious
```

### 3.4.4 Compilation of audacious-plugins-3.9

```
pi $ cd /home/pi/audacious_raspiblasters/audacious-plugins-3.9
pi $ ./configure
```



### 3 Installation and Configuration

---

The command `./configure` checks that several necessary software packages aren't installed yet:

```
pi $ sudo apt install libxml2-dev #message: No package 'libxml-2.0' found
```

A further call of `./configure` requests the `libcdio` library which is necessary for CD playback:  
*checking for libcdio >= 0.70 libcdio\_cdda >= 0.70 libcddb >= 1.2.1... no*  
*configure: WARNING: audio CD support disabled due to missing dependency: libcdio >= 0.70 libcdio\_cdda >= 0.70 libcddb >= 1.2.1*

```
pi $ sudo apt install libcdio-dev libcdio-paranoia-dev libcddb-dev
```

Further the `flac` codec is missing (which is IMHO more important than many other codecs like MP3):

*checking for flac >= 1.2.1... no*  
*configure: error: Missing dependency for FLAC support: flac >= 1.2.1*

```
pi $ sudo apt install libflac-dev
pi $ sudo apt install libogg-dev libvorbis-dev #ditto for codecs ogg vorbis
pi $ sudo apt install libfluidsynth-dev #ditto for MIDI-Plugin
pi $ sudo apt install libmpg123-dev #ditto for MP3-Codec
pi $ sudo apt install libfaad-dev #ditto for AAC-Codec
pi $ sudo apt install libwavpack-dev #ditto for WAV-Codec
pi $ # sudo apt install libsamplerate-dev
```

Installation of package `neon27` for HTTP/HTTPS Transport:

```
pi $ sudo apt install libneon27-dev
```

Ugly warning: ALSA is mandatory on Raspiblasters for audio playback!

*checking for alsa >= 1.0.16... no*  
*configure: WARNING: ALSA output disabled due to missing dependency: alsa >= 1.0.16*  

```
pi $ sudo apt install libasound2-dev
```

Library "libavcodec" from FFmpeg project is missing:

*checking for libavcodec >= 53.40.0 libavformat >= 53.25.0 libavutil >= 51.27.0... no*  
*configure: error: FFmpeg is not installed or too old (required: libavcodec 53.40.0, libavformat 53.25.0, libavutil 51.27.0). Use -with-ffmpeg=none to disable the ffmpeg plugin or -with-ffmpeg=libav to use libav instead.*  

```
pi $ sudo apt install libavcodec-dev libavformat-dev libavutil-dev
```



#### Hint

After installation of the above mentioned packages there are still missing several other features which aren't mandatory for compilation. Of course they can be implemented in a similar way as described above.

### 3 Installation and Configuration

---

```
pi $ ./configure
```

Configuration:

```
Install path:          /usr/local/lib/audacious
GTK+ support:         yes
Qt support:           no
```

#### Audio Formats

-----

```
Audio CD:             yes
Free Lossless Audio Codec: yes
Ogg Vorbis:           yes
MIDI (via FluidSynth): yes
MPEG-1 Layer I/II/III (via mpg123): yes
MPEG-2/4 AAC:         yes
WavPack:              yes
```

#### External Decoders

-----

```
FFmpeg/Libav:        ffmpeg
libsndfile:          no
```

#### Chiptunes

-----

```
AdLib synthesizer (adplug): yes
Commodore 64 audio (sid):   no
Game Music Emu (spc, nsf, gbs, etc.): yes
ModPlug:                   no
Nintendo DS audio (xsf):   yes
PlayStation audio (psf/psf2): yes
Vortex Tracker (vtx):      yes
```

#### Other Inputs

-----

```
Metronome:            yes
Tone Generator:       yes
```

#### Effects

-----

```
Bauer stereophonic-to-binaural (bs2b): no
Channel Mixer:        yes
Crystalizer:          yes
Dynamic Range Compressor: yes
Echo/Surround:        yes
Extra Stereo:         yes
LADSPA Host (requires GTK+): yes
Sample Rate Converter: no
Silence Removal:      yes
SoX Resampler:        no
```

*3 Installation and Configuration*

---

Speed and Pitch:	no
Voice Removal:	yes

## Outputs

-----

Advanced Linux Sound Architecture:	yes
Jack Audio Connection Kit:	no
Open Sound System:	yes
PulseAudio:	no
Simple DirectMedia Layer:	no
Sndio:	no
Win32 waveOut:	no
FileWriter:	yes
-> MP3 encoding:	no
-> Vorbis encoding:	yes
-> FLAC encoding:	yes

## Playlists

-----

Cue sheets:	no
M3U playlists:	yes
Microsoft ASX (legacy):	yes
Microsoft ASX 3.0:	yes
PLS playlists:	yes
XML Sharable Playlist Format (XSPF):	yes

## Transports

-----

FTP, SFTP, SMB (via GIO):	yes
HTTP/HTTPS (via neon):	yes
MMS (via libmms):	no

## General

-----

Alarm (requires GTK+):	yes
Ampache browser (requires Qt):	no
Delete Files:	yes
GNOME Shortcuts:	yes
libnotify OSD:	no
Linux Infrared Remote Control (LIRC):	no
MPRIS 2 Server:	yes
Scrobbler 2.0:	no
Song Change:	yes

## GTK+ Support

-----

GTK Interface:	yes
Winamp Classic Interface:	yes
Album Art:	yes
Blur Scope:	yes

### 3 Installation and Configuration

---

OpenGL Spectrum Analyzer:	no
LyricWiki viewer:	yes
Playlist Manager:	yes
Search Tool:	yes
Spectrum Analyzer (2D):	yes
Status Icon:	yes
X11 Global Hotkeys:	yes
X11 On-Screen Display (aosd):	yes

#### Compilation and Installation of the Plugins:

The command line parameter `-j<x>` gives the number of cores the `make` command will use. Even the usage of `-j4` on a Raspberry Pi 3B takes about 5 minutes for the complete compilation.

```
pi $ make -j4 #-j4 forces make to use all the four cores of the Raspberry Pi 3B
pi $ sudo make install
```

#### 3.4.5 Code Adjustments of *audacious* for the Raspiblaster

To import the changes made by schlizbäda, the adjusted source code has to be downloaded from the GitHub repository <https://github.com/schlizbaeda/audacious-raspiblaster>. The three *audacious* sub projects have to be updated with the downloaded files and recompiled:

```
pi $ git clone https://github.com/schlizbaeda/audacious-raspiblaster download
pi $ cd /home/pi/download
pi $ cp -r libaudclient-3.5-rc /home/pi/audacious_raspiblaster
pi $ cp -r audacious-3.9 /home/pi/audacious_raspiblaster
pi $ cp -r audacious-plugins-3.9 /home/pi/audacious_raspiblaster
```

After that a rebuild of the three sub projects containing the adjusted code must be performed in each project directory:

```
pi $ cd /home/pi/audacious_raspiblaster/libaudclient-3.5-rc2 #ditto for the two other sub projects
pi $ make -j4
pi $ sudo make install
```



#### Hint

The changed code positions are marked in the source code with the comment `schlizbäda`. All changed files can be found with command:

```
pi $ cd /home/pi/audacious_raspiblaster
pi $ grep -r schliz * #searches all files recursively for the search pattern schliz
```

**Hint**

The following sections aren't necessary for the installation of *audacious* on the Raspberry Pi. they are listed here only to show the approach of debugging and analyzing the unknown source code.

### 3.4.6 Usage of the IDE *eclipse* for Code Analysis and Debugging of *audacious* on a PC

For adjusting the source code of *audacious* a laptop with Linux Mint 18.02 is used. There the programmer has got access to a similar *slim line* CD ROM drive.

Maybe the Linux package `eject` has to be installed on the PC first:

```
PC $ sudo apt install eject
```

```
PC $ sudo apt install eclipse eclipse-cdt #Installation with C-/C++ plugin
```

```
PC $ eclipse & #background call
```

Open existing project in *eclipse*:

**"File→New→C++ Project"**

The dialogue box "C++ project" will be opened. Select an EXISTING project using the following options/settings:

- Project Name: `audacious`
- Use default location: Disable!
- Location: select project directory by clicking [Browse]:  
`/home/peter/audacious_raspiblasters/audacious-3.9`
- Project type: call menu item **"Makefile project→Empty Project"**
- Toolchain: Linux GCC

start the import with button [Finish].

Using the Debugger:

- A double-click of the left mouse button on the left area adds/removes a break point
- The debugger is started by clicking the button showing a *bug* symbol:  
Further views of the project are opened!
- F5: single step with branching into sub routines
- F6: single step complete (running through all sub routines)
- F7: leave current block
- F8: Run to the next break point (or program termination)

### 3.4.7 Adjustment of the Source Code

#### Part 1: Implementation of the *eject* Feature

Add new files *eject.cc* and *eject.h* in *.../audacious-3.9/src/libaudcore*:

- *.../audacious-3.9/src/libaudcore/Makefile*  
Add the new source file in section SRCS  
SRCS  
:  
*eject.cc*  
:
- Create the new files *eject.h* and *eject.cc* in *.../audacious-3.9/src/libaudcore*.
- *.../audacious-3.9/src/libaudcore/eject.cc*: Edit the source code  
... and always the same f\*\*king problems regarding to `#includes`, `#defines` and function proto types (see source code)
- *.../audacious-3.9/src/libaudcore/tests/drct.h*: added function proto types  
`void aud_drct_eject (); /* schlizbäda: click event handler of eject button */`
- *.../audacious-plugins-3.9/src/gtkui/ui\_gtk.cc*: added ToolStripItem "eject":  
`toolbar_button_add (toolbar, aud_drct_eject, "media-eject"); /* schlizbäda: added "eject" button */`
- *.../audacious-plugins-3.9/src/gtkui/menus.cc*: added menu item "eject"  
`static const AudguiMenuItem playback_items[] = { /* schlizbäda: added "eject" menu item */`  
:  
`MenuCommand (N_("Eject"), "media-eject", NONE, aud_drct_eject),`  
:  
}

### 3.4.8 Adjustment of the Source Code

#### Part 2: Special feature "eject lock" for Raspberry Pi

Locking of the eject push button of the CD ROM drive by GPIO4 via GPIO library *wiringPi*:

New functions in file *.../audacious-3.9/src/libaudcore/eject.cc*:

```
int lock_eject_pushbutton()
int unlock_eject_pushbutton()
int lock_eject_raspberrypi(bool lock) /* compilation only if the #define RASPBERRYPI is
set */
```

#### Hint

It's important to use the GPIO library *wiringPi* instead of *pigpio* which has obviously influence on **all** GPIOs of the Raspberry Pi. One effect of *pigpio* is a “slow-down” of the I2S bus for audio output via HifiBerry MiniAMP...

## 4 Hardware mods

TODO: This chapter is like a piece of rough paper.

This chapter is under construction!

### 4.1 Raspberry Pi3

### 4.2 Raspberry Pi DSI-Display Touch

### 4.3 HifiBerry MiniAMP

### 4.4 LG GP50NW40

### 4.5 S-USV – battery operation?

### 4.6 Power supply

### 4.7 description of “eject lock” via GPIO4

## 5 Prospect

Which other features are planned for Raspiblaster?

TODO: This chapter is like a piece of rough paper.

This chapter is under construction!

### 5.1 battery operation

### 5.2 Push buttons and rotary encoder as hardware via GPIO for volume control etc.

### 5.3 media player for USB flash drives

### 5.4 rip CDs using the software abcde

### 5.5 DVD playback (using kodi)

### 5.6 Bluetooth receiver

### 5.7 FM and/or DAB+ radio