



Bitte deutlich leserlich ausfüllen!

Deckblatt einer wissenschaftlichen Bachelorarbeit

Vor- und Familienname Wolfgang Hauer	Matrikelnummer 0630948
Studienrichtung Elektrotechnik-Toningenieur	Studienkennzahl V 033 213

Thema der Arbeit:

Der Raspberry Pi und seine Relevanz in der Audiotechnik
.....
.....

Angefertigt in der Lehrveranstaltung: Computermusik und Multimedia 02
(Name der Lehrveranstaltung)

Vorgelegt am:
(Datum)

Beurteilt durch:
(Leiter/-in der Lehrveranstaltung)

This document was written with GNU Emacs, is set in Palatino, compiled with pdfL^AT_EX 2_ε and Biber.

The L^AT_EX template from Karl Voit is based on KOMA script and can be found online: <https://github.com/novoid/LaTeX-KOMA-template>

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, _____
Date

Signature

Eidesstattliche Erklärung¹

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am _____
Datum

Unterschrift

¹Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

Abstract

Diese Bachelorarbeit untersucht die für den Audio-Bereich interessanten Eigenschaften des Einplatinencomputers *Raspberry Pi*. Da hinter dem Rechner mit der *Raspberry Pi Foundation* kein gewinnorientiertes Unternehmen steht, sondern eine als Wohltätigkeitsorganisation eingetragene Stiftung mit dem Ziel, Programmieren und den *open source*-Gedanken vor allem Kindern und Jugendlichen näher zu bringen, wird als erstes kurz die Geschichte und die Intention des Projektes näher vorgestellt.

Anschließend werden verschiedene Operationssysteme und Audio-Programme vorgestellt und auf ihre praktischen Einsatzmöglichkeiten als auch systembedingten Beschränkungen hin geprüft.

Die Untersuchung des Computers beinhaltet hardwareseitig sowohl seine allgemeinen technischen Spezifikationen sowie im Speziellen die elektroakustische Messung des Audio-Ausganges und eine nähere Beschreibung der Audio-Hardware.

Im abschließenden Teil werden zwei Audio-Projekte, die den Raspberry Pi als integralen Bestandteil verwenden, vorgestellt und zukünftige Einsatzmöglichkeiten des Rechners angesprochen.

Inhaltsverzeichnis

Abstract	v
1 Einleitung	1
2 Die Entstehung der Raspberry Foundation und der ersten Prototypen	3
2.1 Die Raspberry Pi Foundation	3
2.2 Die Entwicklung des Raspberry Pi	5
3 Hardware	9
3.1 Allgemeine Hardware	9
3.1.1 Broadcom BCM2835:	9
3.2 Audio-Hardware	11
3.3 Messung des analogen Audio-Ausganges	13
3.3.1 Messaufbau	14
3.4 Messergebnisse	17
3.4.1 Raspberry Pi	17
3.4.2 Realtek High Definition Audio	24
3.4.3 RME Fireface 800	26
3.4.4 Direkter Vergleich des analogen Ausganges mit den anderen Soundkarten	28
3.5 Latenzmessung	30
3.5.1 Half-duplex-Messung	31
3.5.2 Full-duplex-Messung	33
3.6 Fazit	34
3.6.1 Alternativen zum analogen Ausgang	35

Inhaltsverzeichnis

4	Software	37
4.1	Betriebssysteme	37
4.1.1	Raspbian „wheezy“	37
4.1.2	Raspbmc	39
4.2	Musik-Software	40
4.2.1	Pure Data	40
4.2.2	Audacity	42
4.2.3	JACK	43
4.2.4	Ardour	44
4.2.5	MPD	45
5	Projekte	47
5.1	FM Radio Transmitter	47
5.1.1	Aufbau	48
5.2	Guitar Extended	48
5.2.1	Aufbau	49
5.3	Ein kurzer Blick in die Zukunft	50
	Literatur	51

Abbildungsverzeichnis

2.1	Eben Upton	4
2.2	Rote Version des Raspberry Pi	5
2.3	Raspberry Pi Prototyp	6
2.4	Raspberry Pi Alpha-Board	7
3.1	Hardwareübersicht	10
3.2	Blockdiagramm PCM Audio Interface	12
3.3	Darstellung des PCM Streams	12
3.4	Blockschaltbild des Messaufbaus	14
3.5	Messaufbau	15
3.6	Screenshot RMAA	16
3.7	Raspberry Pi odb Gain Amplitudengang	18
3.8	Raspberry Pi odb Gain Rauschen	18
3.9	Raspberry Pi odb Gain THD	19
3.10	Raspberry Pi -4db Gain Amplitudengang	20
3.11	Raspberry Pi -4db Gain Rauschen	20
3.12	Raspberry Pi -4db Gain THD	21
3.13	Raspberry Pi +4db Gain Amplitudengang	22
3.14	Raspberry Pi +4db Gain Rauschen	23
3.15	Raspberry Pi +4db Gain THD	23
3.16	Laptop Soundkarte Amplitudengang	24
3.17	Laptop Rauschen	25
3.18	Laptop THD	25
3.19	Fireface 800 Amplitudengang	26
3.20	Fireface 800 Rauschen	27
3.21	Fireface 800 THD	27
3.22	Vergleich Amplitudengang	29
3.23	Vergleich Rauschen	29
3.24	Vergleich THD	30

Abbildungsverzeichnis

3.25	Messaufbau Half-duplex	32
3.26	Messaufbau Full-duplex	33
4.1	Screenshot Raspbian „wheezy“	38
4.2	Screenshot Raspbmc	40
4.3	Screenshot Pd	42
4.4	Screenshot Audacity	43
4.5	Screenshot QjackCtl	44
4.6	Screenshot Ardour	45
4.7	MPD Architektur-Übersicht	46
4.8	Screenshot iPhone Client	46
5.1	Video-Still Guitar Extended	49

1 Einleitung

»Education is not filling a pail but the lighting of a fire.« Dieses Zitat des Schriftstellers William Butler Yeats beschreibt die Kernidee des britischen Raspberry Pi Projektes. Dessen Initiatoren, allesamt an der Cambridge Universität angestellt, stellten Mitte des letzten Jahrzehnts fest, dass die Leistungen der Bewerber für den Computer Science Kurs jedes Jahr sanken. Um dem entgegen zu wirken, entwickelten sie einen kleinen, billigen Computer, der vor allem Kindern und Jugendlichen Anreiz geben sollte sich mit dem Programmieren auseinanderzusetzen, um so eigene Projekte und Ideen erschaffen zu können. Geplant war ursprünglich, ca. 1000 Stück an Bewerber des Computer-Science-Kurses zu verteilen, doch durch den unerwarteten großen kommerziellen Erfolg entwickelte sich eine lebendige Gemeinschaft, die ständig eigene Projekte und Codes entwickelte und so das Projekt wachsen ließ.

Durch die offene Struktur, den günstigen Preis und den geringen Stromverbrauch sind unzählige interessante Anwendungen für den Audibereich denkbar.

Doch wo Licht ist, ist auch Schatten, und so ergeben sich aus diesen Stärken auch manche Nachteile:

Viele Betriebssysteme erfordern gerade von Anfängern eine oft mühsame Einarbeitungszeit, die Leistung der Hardware war vor ca. zehn Jahren am neuesten Stand der Technik und oft sieht man sich mit instabilem, unfertigem Code konfrontiert.

Doch wenn man weiß, wo die Probleme liegen, kann man ihnen gezielt entgegen wirken und so werden in dieser Arbeit mit dem Hauptaugenmerk auf Audioanwendungen diverse Problemstellen analysiert.

In Kapitel 1 wird kurz die zu Grunde liegende Idee sowie die Entwicklung der Hardware und der Raspberry Pi Foundation zusammengefasst.

1 Einleitung

Im zweiten Kapitel befasst sich diese Arbeit mit der Hardware im Allgemeinen sowie im Speziellen mit der Struktur der Audiohardware und seinen elektro-akustischen Eigenschaften. Dazu wurden Messungen des analogen Audio-Ausganges durchgeführt und diese mit einer handelsüblichen Laptop-Soundkarte sowie einem qualitativ hochwertigen Audio-Interface verglichen. Abschließend wird diskutiert, ob bzw. in welchen Grenzen ein Einsatz des Raspberry Pi ohne externe Audiohardware sinnvoll ist, und es werden Alternativen aufgezeigt.

Im dritten Kapitel werden die wichtigsten zur Zeit erhältlichen Betriebssysteme und Audio-Programme angeführt und beschrieben.

Im letzten Kapitel werden zwei Projekte vorgestellt, die den Rechner bereits als integralen Bestandteil verwenden, und es wird ein kurzer Ausblick auf zukünftige Anwendungen gewagt.

2 Die Entstehung der Raspberry Foundation und der ersten Prototypen

2.1 Die Raspberry Pi Foundation

Die Idee für einen kleinen und billigen Computer entstand 2006 an der Cambridge Universität. Eben Upton (Abb.:[2.1](#)) und einige seiner Kollegen an Cambridge's Computer Laboratory, darunter die späteren Mitgründer der *Raspberry Pi Foundation*, David Braben, Rob Mullins, Jack Lang, Pete Lomas und Alan Mycroft, stellten fest, dass sowohl die Zahl als auch die Fähigkeiten der Studienanfänger des *Computer Science* Kurses Jahr für Jahr sanken.

Die Gründe dafür sahen sie in den nicht mehr zeitgemäßen Lehrplänen der Schulen, dem Ende der Dotcom-Blase, aber auch immer teureren und unzugänglicheren Heimcomputern, die das Experimentieren mit Hard- und Software durch ihre abgeschlossenen Systeme erschweren.

Ihr Ziel war daher die Entwicklung eines möglichst günstigen Computers, der durch seine offene Struktur zum Experimentieren einlädt.

Nach der Entwicklung der ersten Prototypen wurde die Raspberry Pi Foundation am 5. Mai 2009 in Großbritannien als Wohltätigkeitsorganisation eingetragen.[\[23\]](#) Das Ziel der Stiftung ist, neben dem Fördern des Studiums der Informatik und verwandter Themen (insbesondere auf Grundschuleebene), dass Programmieren wieder Spaß machen soll. Um dies zu erreichen, entwickelt die Organisation den Raspberry Pi und vergibt die Lizenz für das Design an ausgewählte Hersteller. Zur Zeit sind das die britischen

2 Die Entstehung der Raspberry Foundation und der ersten Prototypen



Abbildung 2.1: Eben Upton bei einem Vortrag zum Raspberry Pi[9]

Unternehmen Element14/Premier Farnell[27], RS Components[38] und der chinesische Hersteller Egoman[10], die den Rechner auch online vertreiben. Die von Egoman produzierte Version darf nur in China und Taiwan vertrieben werden. Sie kann aufgrund ihrer roten Platine und der fehlenden FCC/CE Kennzeichnung vom herkömmlichen Modell unterschieden werden, die restliche Hardware ist aber identisch mit den hier erhältlichen Versionen (Abb.:2.2).

Die Stiftung ist als *private company limited by guarantee* eingetragen, was hierzulande weitestgehend der *Gesellschaft mit beschränkter Haftung (GmbH)* entspricht und oft bei nicht-gewinnorientierten Unternehmen zur Anwendung kommt, die als juristische Person auftreten müssen.[41]

Im September 2012 verkündete die Stiftung, dass in South Wales eine Produktionsstätte von Element14/Premier Farnell eröffnet werden soll, um monatlich 30.000 Einheiten herzustellen. Dies würde nicht nur 30 neue Arbeitsplätze schaffen, sondern auch den ersten Rechner „made in UK“ seit fast 20 Jahrzehnten bedeuten.

Im Jänner 2013 wurden von *Google Giving* [16] 15.000 Raspberry Pis an

2.2 Die Entwicklung des Raspberry Pi

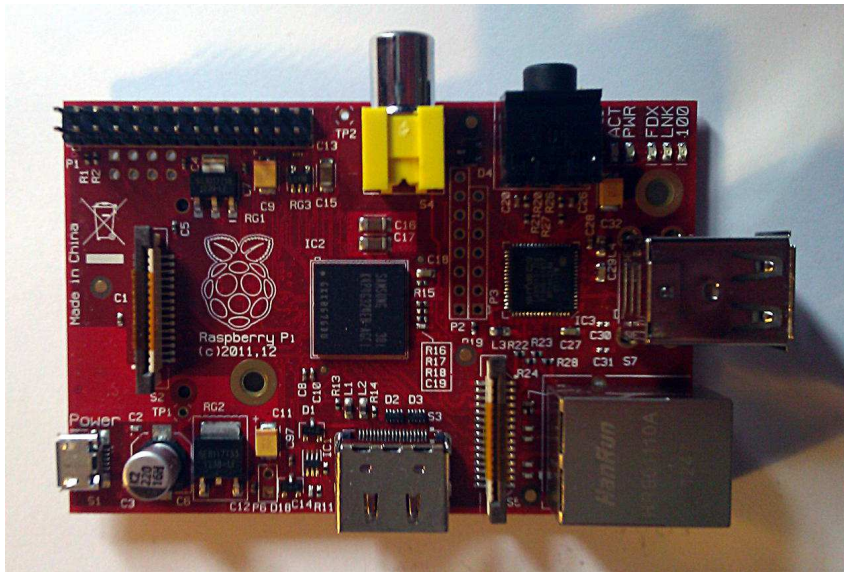


Abbildung 2.2: Rote Version des Raspberry Pi[11]

Grundschulen in ganz Großbritannien gespendet. [33]

Vor der Markteinführung wurde erwartet 20.000 Einheiten zu verkaufen, im Jänner 2013 ging bereits der einmillionste Raspberry Pi über den Ladentisch.

2.2 Die Entwicklung des Raspberry Pi

Die ersten Prototypen bestanden aus einer Lochrasterplatine, einem mit 22,1 MHz getakteten Atmel-ATmega644-Mikrocontroller und 512K SRAM als Daten und Framebufferspeicher und kosteten ca. 25 US-Dollar (Abb:2.3). Obwohl ihm die Idee eines Computers, der selbst zusammen gebaut werden kann, gefiel, entschied sich Upton aufgrund der geringen Leistung und der fehlenden Möglichkeit, ein universelles Operationssystem zu installieren, gegen diese Version.

2 Die Entstehung der Raspberry Foundation und der ersten Prototypen

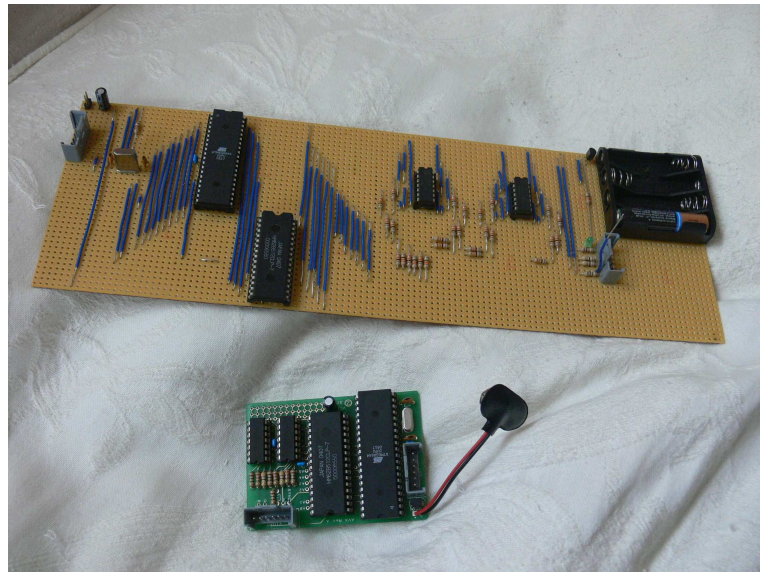


Abbildung 2.3: Raspberry Pi Prototyp[32]

Stattdessen wurde für den billigen BCM2835 Prozessor eine neue Mehrlagenplatine entwickelt. Der BCM2835 wird von Broadcom hergestellt (wo Eben Upton als technischer Direktor arbeitet) und wird auch in der aktuellen Version eingesetzt. Durch die rasche Leistungssteigerung mobiler Prozessoren wurden bald auch multimediale Funktionen möglich, wodurch besonders das Interesse junger Menschen geweckt werden sollte.

Im August 2011 wurden 50 *Alpha-Boards* ausgeliefert die bereits funktional identisch mit dem späteren Model B waren (Abb.:2.4). Allerdings waren diese größer als die kreditkartengroße Platine des Model B, um Messpunkte zur Fehlerbehebung unterzubringen.

Mit den Alphaboards wurde gezeigt, »dass die Desktop-Umgebung LXDE unter Debian sowie Quake 3 und H.264-Videos mit einer Auflösung von 1080p via HDMI funktionieren.«[43]

Die ersten Raspberry Pis in aktueller Bauweise wurden im Frühjahr 2012 ausgeliefert, allerdings noch mit teilweise monatelanger Wartezeit. Die Auslieferung der *Full Production Boards*, also der Boards aus Serienfertigung, startete mit den Modell-B Raspberry Pis im Juni 2012. Das erste Modell-

2.2 Die Entwicklung des Raspberry Pi



Abbildung 2.4: Raspberry Pi Alpha-Board[12]

A war ab Februar 2013 in Europa und Nordamerika erhältlich. Auf die technischen Unterschiede der beiden Modelle wird in Kapitel 3, [Hardware](#) auf S. 9 näher eingegangen.

3 Hardware

3.1 Allgemeine Hardware

Tabelle 3.1: Hardwarespezifikationen[43]

	Modell A	Modell B
SoC (System-on-a-chip):	Broadcom BCM2835	
CPU:	700MHz ARM 11 ARM1176JZF-S core	
GPU:	Broadcom VideoCore IV	
Memory (SDRAM):	256MB	512MB (bis Oktober '12 256MB)
USB-Ports:	1	2
Netzwerk:	Keine	10/100 Mbit/s Ethernet
Video:	FBAS oder HDMI	
Audio-Ausgänge:	3,5mm TRS oder HDMI	
Audio-Eingänge:	Keine	
Systemspeicher:	SD-Karten	
Stromversorgung:	5V via Micro-USB oder GPIO	
Nennleistung:	300mA (1,5W)	700mA (3,5W)
Größe:	85,6mm x 54,0mm x 17mm	
Preisempfehlung:	25USD (exkl. Mehrwertsteuer)	35USD (exkl. Mehrwertsteuer)

3.1.1 Broadcom BCM2835:

Der Broadcom BCM2835 Prozessor wurde speziell für mobile Multimediaanwendungen entwickelt und zeichnet sich durch seinen geringen Energieverbrauch aus. Es handelt sich dabei um ein SoC¹, darunter versteht man

¹ System on a Chip

3 Hardware

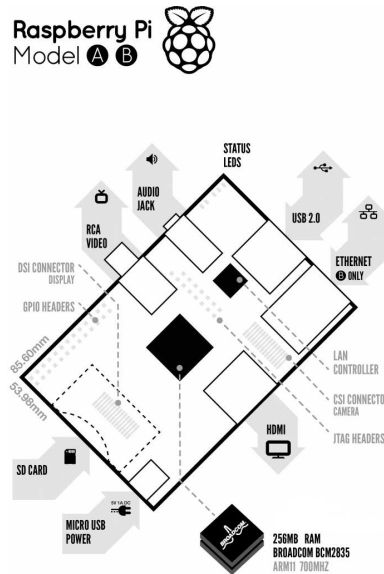


Abbildung 3.1: Hardwareübersicht[39]

»die Integration aller oder eines großen Teils der Funktionen eines Systems auf einem Chip (Die), also einem integriertem Schaltkreis (IC) auf einem Halbleiter-Substrat, auch monolithische Integration genannt.«[44] Im Falle des BCM2835 sind das folgende Teile:

- **CPU**

Die CPU² besteht aus einem *ARM1176JZF-S ARM11*, der mit 700MHz getaktet wird. Die ARM Architektur wird derzeit auf fast allen Smartphones und Tablets verwendet und zeichnet sich durch viele Vorteile (z.B. niedrigen Energieverbrauch) aus, allerdings ist es dadurch nicht möglich Operationssysteme wie Windows oder MAC OSX zu installieren, da diese eine x86-Umgebung benötigen.

Die CPU kann, ohne Garantieverlust, auf bis zu 1GHz übertaktet werden. Im Zuge der letzten Firmware-Aktualisierung wurde diese Option in *Raspbian „wheezy“* implementiert und kann mit dem Befehl

² Central Processing Unit

raspi-config
geändert werden.

- **GPU**

Die GPU³ ist ein *Broadcom Video Core IV* und bietet *OpenGL ES 1.1*, *OpenGL ES 2.0*, *OpenVG 1.1*, *Open EGL*, *OpenMAX* und das Dekodieren von 1080p30 H.264 high-profile Videos.

- **DSP**

Es gibt zwar einen DSP⁴, allerdings kann dieser aufgrund einer fehlenden (bzw. nicht öffentlichen) Schnittstelle zur Anwendungsprogrammierung nicht verwendet werden.

- **SDRAM**

Als Arbeitsspeicher werden beim Modell-A 256MB große *Hynix MobileDDR2* oder *Samsung Mobile DRAM* verwendet. Die neuesten Modell-B Boards verwenden 512MB große *Samsung Mobile DRAM*. Der Arbeitsspeicher wird in *PoP*⁵-Technik auf den Prozessor gelötet und ist somit untrennbar damit verbunden. Es ist daher nicht möglich den Arbeitsspeicher zu erweitern oder auszutauschen.

3.2 Audio-Hardware

Das PCM⁶ Audio Interface ist ein APB⁷ und ermöglicht den In- und Output von Audio-Streams in unterschiedlichen Qualitätsstufen.

APBs sind interne Busse von SoCs für den Anschluss von leistungsarmen Peripheriegeräten und sind für niedrigen Verbrauch und einfache Interfaces optimiert.[1]

³ Graphics Processing Unit

⁴ Digitaler Signal Prozessor

⁵ Package on Package

⁶ Pulse Code Modulation

⁷ Advanced Peripheral Bus

3 Hardware

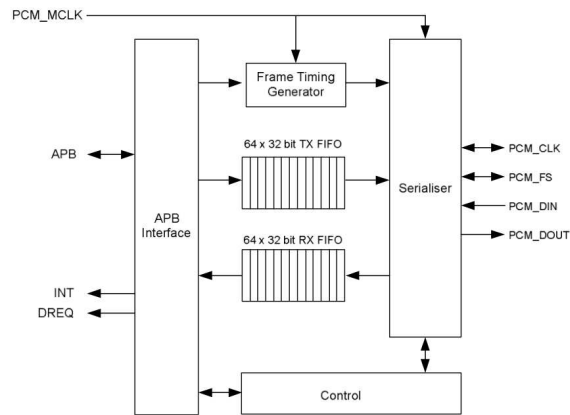


Abbildung 3.2: Blockdiagramm des PCM Audio Interfaces[19]

Das Interface unterstützt viele klassische PCM-Formate, wie z.B. I^2S , und hat 4 Interface-Signale:

- *PCM_CLK* - bit clock
- *PCM_FS* - frame sync signal
- *PCM_DIN* - serial data input
- *PCM_DOUT* - serial data output

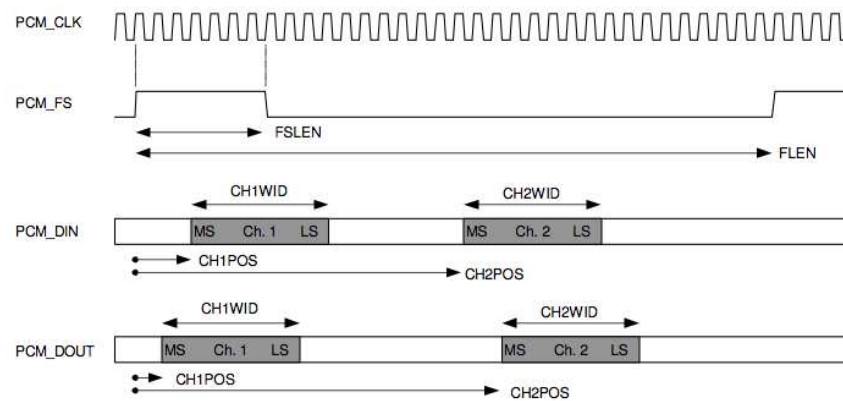


Abbildung 3.3: Darstellung des PCM-Streams[19]

3.3 Messung des analogen Audio-Ausganges

Für weiterführende Informationen bezüglich des PCM Audio Interfaces wird auf das Manual *BCM2835 ARM Peripherals*[19] von Broadcom verwiesen.

3.3 Messung des analogen Audio-Ausganges

Die Messungen des analogen Audio-Ausganges sollen zeigen, inwiefern dieser für eine Nutzung im Audio-Bereich geeignet ist und welche Einschränkungen in Kauf genommen werden müssen. Dafür wurden einige der wichtigsten elektro-akustischen Merkmale untersucht:

- Amplitudengang
- Rauschen
- THD⁸

Anschließend wurden die Messergebnisse mit den Audio-Ausgängen zweier Geräte verglichen:

- Realtek High Definition Audio
Die preisgünstige *High Definition Audio* ist ein IC⁹ des taiwanesischen Herstellers Realtek und wird vor allem in Laptops verschiedener Hersteller eingesetzt. Da die Raspberry Pi Soundkarte eine ähnliche Konstruktion aufweist, können ähnliche Klangeigenschaften erwartet werden.
- RME Fireface 800[13]
Das *Fireface 800* des deutschen Herstellers RME ist eine Soundkarte aus dem *High-end*-Bereich. Aufgrund ihrer aufwändigen Bauweise sollten die Messwerte entsprechend bessere Klangeigenschaften zeigen.

Da der analoge Audio-Ausgang des Raspberry Pi, je nach gewählter digitaler Verstärkung, andere Klangeigenschaften besitzt, wurden außerdem drei verschiedene Verstärkungsstufen gemessen:

- 0dB
- -4dB
- +4dB

⁸ Total Harmonic Distortion

⁹ Integrated Circuit

3 Hardware

3.3.1 Messaufbau

Hardware

- Raspberry Pi, Modell-B
- RME Fireface 800
- Kabel: 3,5mm Klinkenstecker - 6,3mm Klinkenstecker

Software

- OS Raspberry Pi: *Raspbian „wheezy“*
- OS Messrechner: *Windows XP*
- Messsoftware: *Right Mark Audio Analyzer 6* [37]

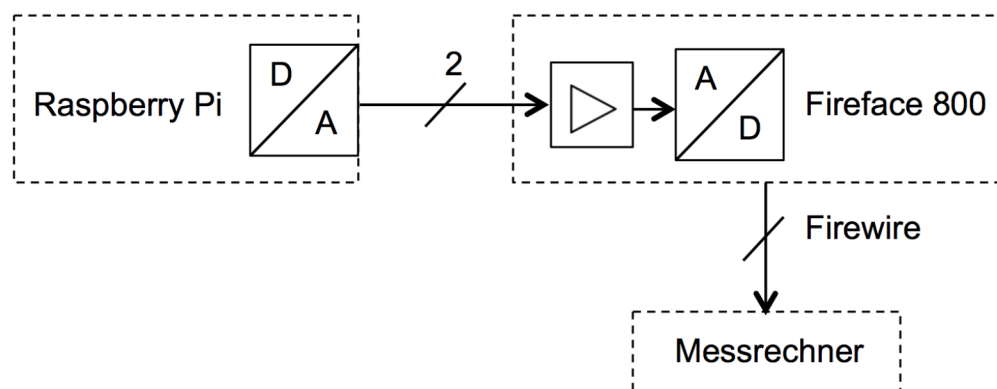


Abbildung 3.4: Blockschaltbild des Messaufbaus

Als Testsignal wurde in allen Fällen ein von *RMAA*¹⁰ erstelltes Messsignal mit 16bit Bittiefe und 44,1kHz Samplingrate verwendet. Da *RMAA* zum Starten einer Messung einen Eingangspegel von ca. -1dB verlangt (siehe Abb.:3.6), den der analoge Audioausgang des Raspberry Pis aber nicht liefern kann, wurde das Signal mit Hilfe der Eingangsverstärker des Fireface 800 auf den geforderten Pegel gebracht (siehe Abb.:3.5).

¹⁰ Right Mark Audio Analyzer

3.3 Messung des analogen Audio-Ausganges

Aufgrund dieser nicht idealen Verstärkung kommt es im Messergebnis zu Abweichungen im Frequenzgang bzw. zu erhöhtem Rauschen. Durch die hohe Qualität der Preamps kann diese Abweichung allerdings vernachlässigt werden.

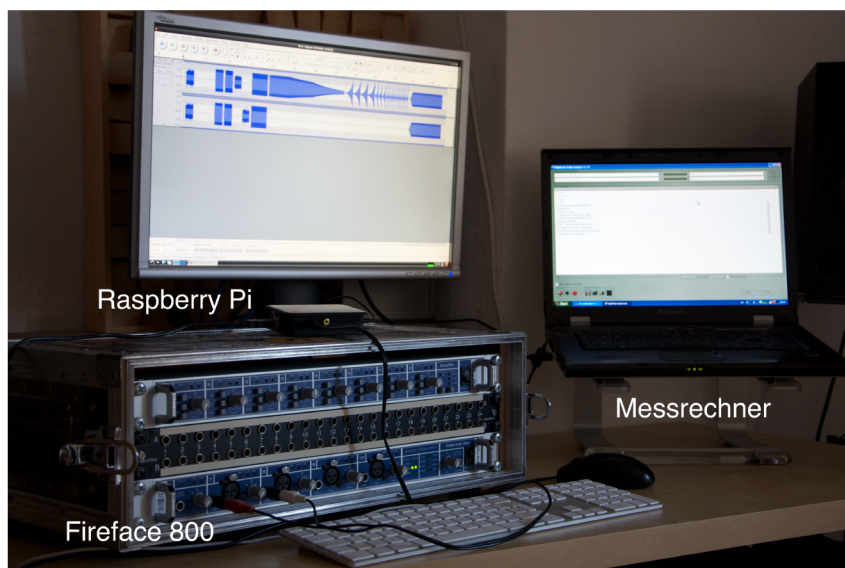


Abbildung 3.5: Messaufbau

3 Hardware

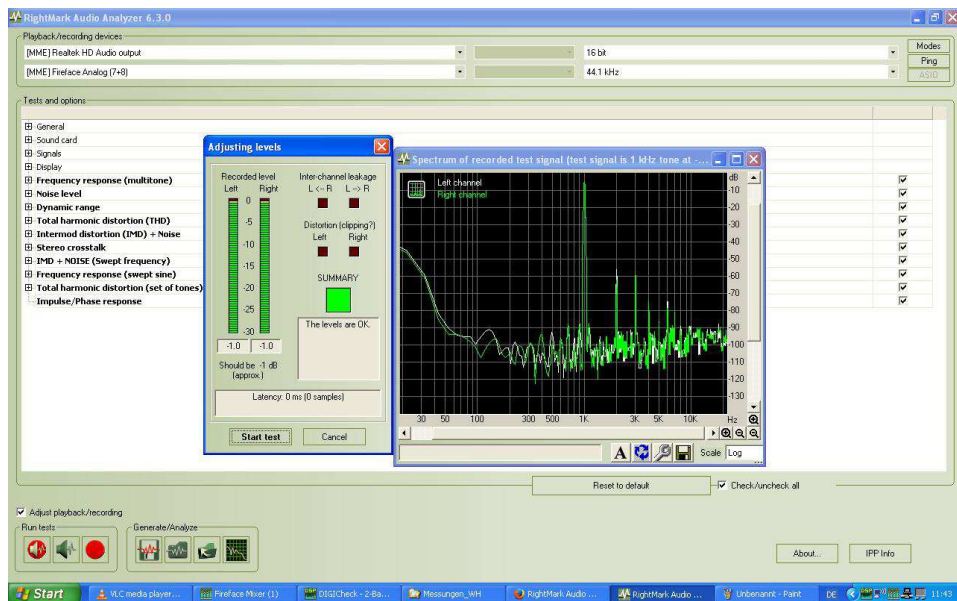


Abbildung 3.6: Screenshot von RMAA

3.4 Messergebnisse

Die Zusammenfassungen der einzelnen Messungen (Tab.:3.6, 3.5, 3.2, 3.3, 3.4) wurden von RMAA erstellt und von RMAA automatisch mit einer subjektiven Bewertung versehen (Very Poor, Poor, Average, Good, Very Good, Excellent).

3.4.1 Raspberry Pi

Messung mit 0dB-Verstärkung

Tabelle 3.2: Zusammenfassung der Messwerte des Raspberry Pi bei 0dB-Verstärkung

Frequency response (from 40Hz to 15kHz), dB	+0.17,-2.12	Average
Noise level, dB (A)	-74.7	Average
Dynamic range, dB (A)	71.2	Average
THD, %	0.259	Poor
THD + Noise, dB (A)	-49.3	Poor
Stereo crosstalk, dB	-73.4	Good
General Performance		Average

Tab.:3.2 zeigt die Zusammenfassung der Messwerte für den analogen Audioausgang des Raspberry Pi. Die Bewertungen liegen größtenteils im Mittelfeld. *THD* sowie *THD + Noise* werden mit *Poor*, der *Stereo crosstalk*¹¹ mit *Good* bewertet. Insgesamt ergibt sich damit die Bewertung *Average*.

¹¹ Übersprechen der beiden Stereokanäle

3 Hardware

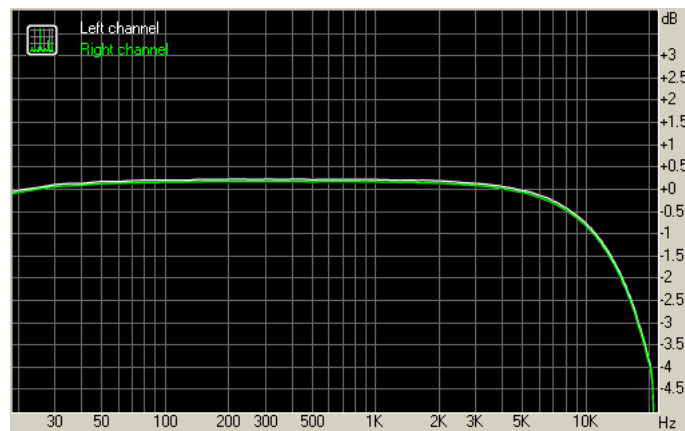


Abbildung 3.7: Amplitudengang des Raspberry Pi bei 0dB-Verstärkung

Abb.:3.7 zeigt den Amplitudengang des analogen Raspberry Pi Ausganges bei einer digitalen Verstärkung von 0dB. Im Gegensatz zur Realtek High Definition Audio tritt hier weder eine Welligkeit noch der Abfall zu tiefen Frequenzen hin auf, allerdings ist ein Abfall ab ca. 4kHz zu hohen Frequenzen hin erkennbar (vgl. Abb.:3.16).

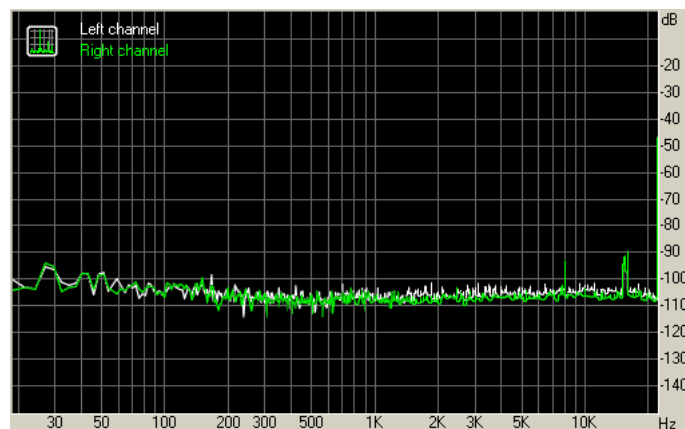


Abbildung 3.8: Rauschen des Raspberry Pi bei 0dB-Verstärkung

Abb.:3.8 zeigt den gleichmäßigen Rauschteppich bei ca. -105dB. Der Grund

3.4 Messergebnisse

für die beiden Spitzen bei 8kHz bzw. ca. 16kHz konnte nicht eruiert werden, möglicherweise handelt es sich dabei um Einstreuungen aus der Messumgebung.

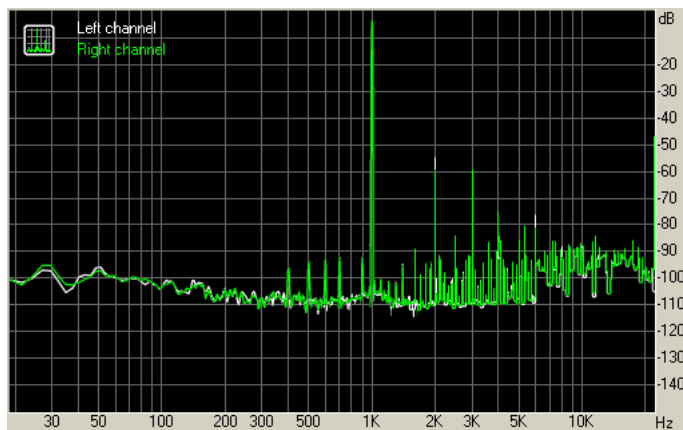


Abbildung 3.9: Total Harmonic Distortion des Raspberry Pi bei 0dB-Verstärkung

Abb.:3.9 zeigt die THD. Deutlich zu erkennen sind hier sowohl das 1kHz-Testsignal sowie die daraus resultierenden starken Verzerrungen, die zum schlechten Wert von 0,259% führen.

Messung mit -4dB-Verstärkung

Tabelle 3.3: Zusammenfassung der Messwerte des Raspberry Pi bei 0dB-Verstärkung

Frequency response (from 40Hz to 15kHz), dB	+0.17,-2.12	Average
Noise level, dB (A)	-70.6	Average
Dynamic range, dB (A)	67.7	Average
THD, %	0.166	Average
THD + Noise, dB (A)	-52.8	Poor
Stereo crosstalk, dB	-69.5	Good
General Performance		Average

3 Hardware

Tab.:3.3 zeigt die leichten Veränderungen der Klangeigenschaften, die durch die geringere, digitale Verstärkung entstehen. Die Bewertungen blieben größtenteils unverändert, lediglich die THD verbesserte sich merklich von 0,259% auf 0,166% (vgl. Tab.:3.2).

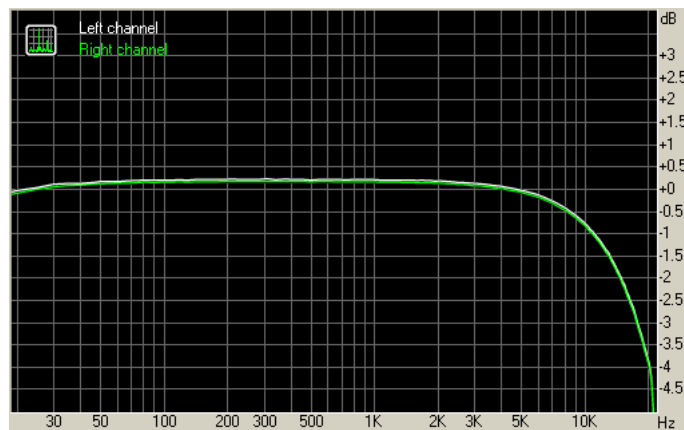


Abbildung 3.10: Amplitudengang des Raspberry Pi bei -4dB-Verstärkung

Abb.:3.10 zeigt den Amplitudengang. Im Vergleich zur Messung mit 0dB Verstärkung sind keine markanten Unterschiede erkennbar (vgl. Abb.:3.7).

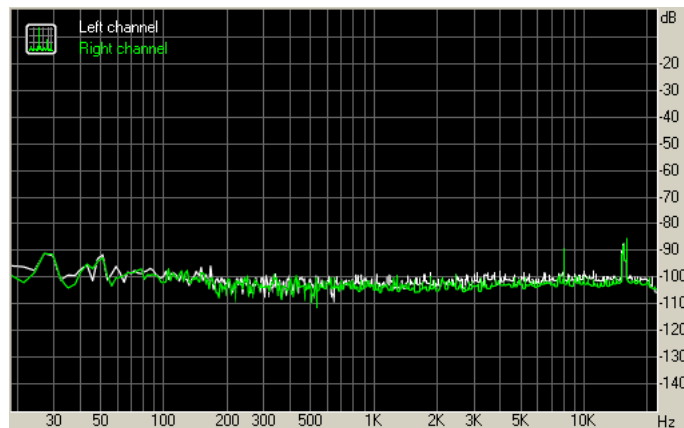


Abbildung 3.11: Rauschen des Raspberry Pi bei -4dB-Verstärkung

3.4 Messergebnisse

Auch das Rauschen bei -4dB-Verstärkung weist keine auffallenden Unterschiede zur 0dB-Verstärkung auf (vgl. Abb.:3.8).

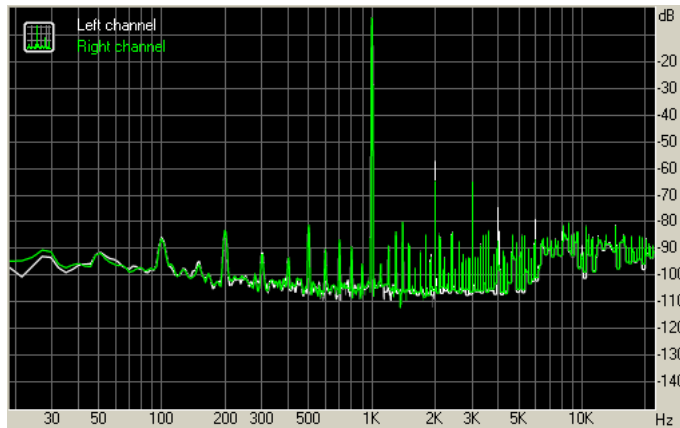


Abbildung 3.12: Total Harmonic Distortion des Raspberry Pi bei -4dB-Verstärkung

Abb.:3.12 lässt die im Vergleich zu 0dB-Verstärkung etwas bessere THD erkennen (vgl. Abb.:3.9).

Messung mit +4dB-Verstärkung

Tabelle 3.4: Zusammenfassung der Messwerte des Raspberry Pi bei +4dB-Verstärkung

Frequency response (from 40Hz to 15kHz), dB	+0.19,-2.12	Average
Noise level, dB (A)	-79.0	Average
Dynamic range, dB (A)	77.2	Average
THD, %	4.711	Very poor
THD + Noise, dB (A)	-24.3	Very poor
Stereo crosstalk, dB	-75.4	Very good
General Performance		Average

Tab.:3.4 zeigt die Messwerte bei der maximal möglichen Verstärkung von +4dB. Obwohl sich einige Werte gegenüber der Messung mit 0dB-Verstärkung

3 Hardware

sogar verbesserten (der Rauschpegel und die Dynamic Range verbesserten sich um 4,3dB bzw. 6dB, der Stereo crosstalk um 2dB und erhielt die Bewertung *Very good*), ergibt sich durch die extreme Verschlechterung der THD, bzw. THD+Noise insgesamt eine für die Praxis unbrauchbare Klangqualität. Die *General Performance* sollte daher mit *Very poor* bewertet werden.

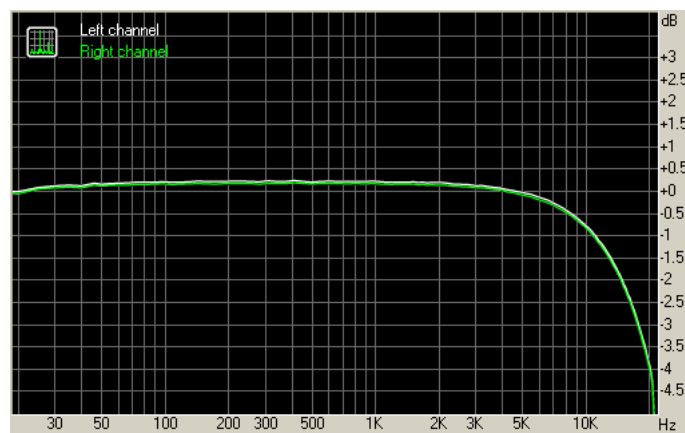


Abbildung 3.13: Amplitudengang des Raspberry Pi bei +4dB Verstärkung

Abb.:3.13 zeigt den Amplitudengang bei +4dB-Verstärkung. Im Gegensatz zu den anderen Messwerten ergeben sich hier nur minimale Veränderungen.

3.4 Messergebnisse

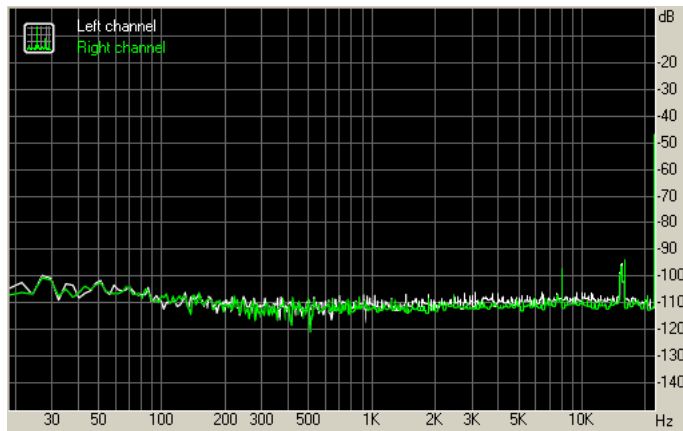


Abbildung 3.14: Rauschen des Raspberry Pi bei +4dB-Verstärkung

Auch der Rauschpegel in Abb.:3.14 weist im Vergleich zu den anderen Messungen kaum Unterschiede auf.

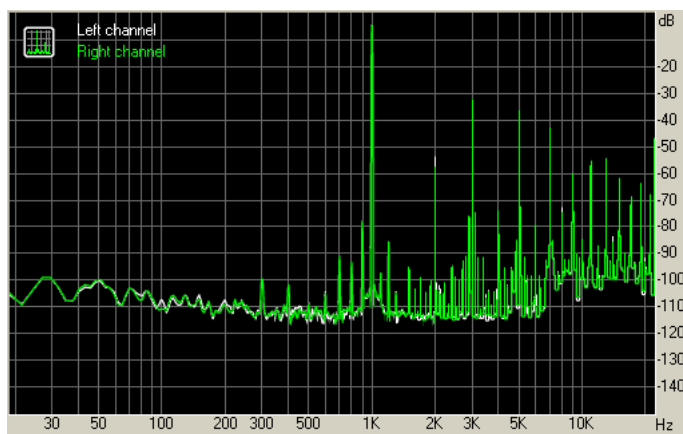


Abbildung 3.15: Total Harmonic Distortion des Raspberry Pi bei +4dB-Verstärkung

Abb.:3.15 zeigt die starken Verzerrungen, die durch die hohe Verstärkung hervorgerufen werden.

3 Hardware

3.4.2 Realtek High Definition Audio

Tabelle 3.5: Zusammenfassung der Messwerte der Realtek High Definition Audio

Frequency response (from 40Hz to 15kHz), dB	-0.13,-1.47	Average
Noise level, dB (A)	-72.9	Average
Dynamic range, dB (A)	72.9	Average
THD, %	0.0058	Very good
THD + Noise, dB (A)	-66.5	Average
Stereo crosstalk, dB	-39.3	Very poor
General Performance		Average

Abb.:3.5 zeigt die Zusammenfassung der Messwerte der Realtek High Definition Audio. Wie auch der analoge Ausgang des Raspberry Pi bei 0dB-Verstärkung liegen die Werte größtenteils im Mittelfeld. Ausnahmen sind hier die *THD*, die mit *Very good*, und der *Stereo Crosstalk*, der mit *Very poor* bewertet wird. Die Gesamtbewertung *Average* deckt sich mit den Erwartungen, die an eine herkömmliche Laptop-Soundkarte gestellt werden können.

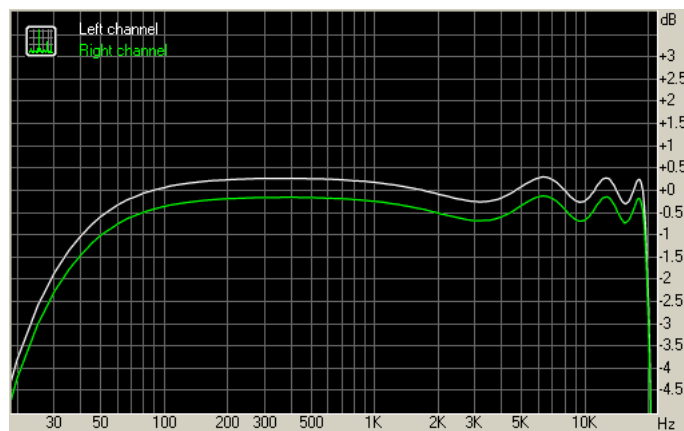


Abbildung 3.16: Amplitudengang der Realtek High Definition Audio

Abb.:3.16 zeigt den Amplitudengang. Auffallend sind hier die Welligkeit (ca. 1dB Schwankungsbreite) im hohen Frequenzbereich, der Pegelunterschied zwischen linkem und rechtem Kanal sowie der Abfall ab ca. 100Hz.

3.4 Messergebnisse

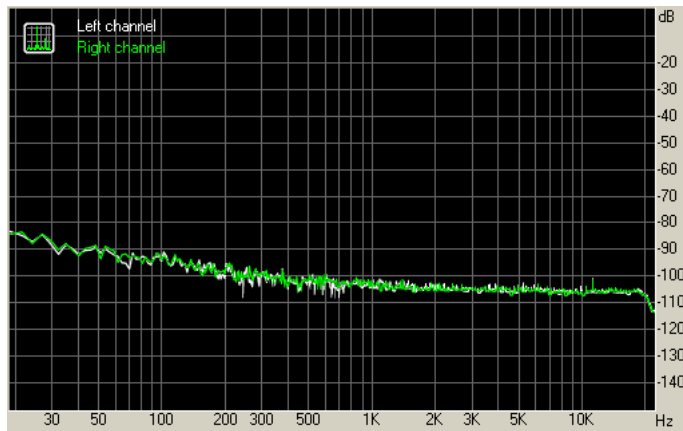


Abbildung 3.17: Rauschen der Realtek High Definition Audio

Abb.:3.17 zeigt den Rauschpegel. Der Mittelwert liegt im gleichen Bereich wie der des Raspberry Pi, allerdings steigt der Pegel der Realtek High Definition Audio zu tiefen Frequenzen hin leicht an.

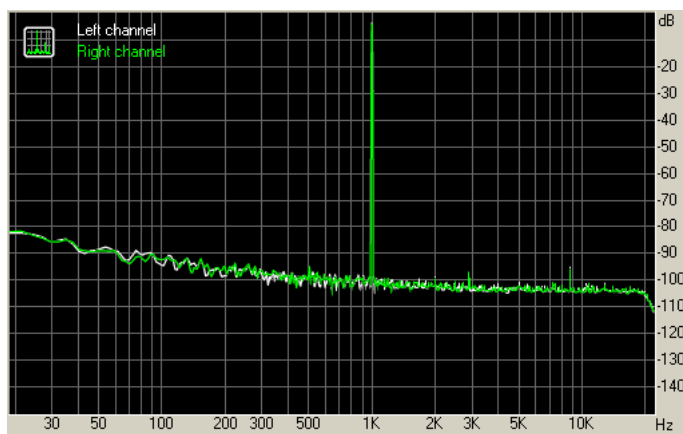


Abbildung 3.18: Total Harmonic Distortion der Realtek High Definition Audio

Abb.:3.18 zeigt die THD. Im Vergleich zum Raspberry Pi (vgl.:3.9) erkennt man deutlich die geringeren Verzerrungen, die sich auch im besseren Messwert niederschlagen.

3 Hardware

3.4.3 RME Fireface 800

Tabelle 3.6: Zusammenfassung der Messwerte des Raspberry Pi bei 0dB-Verstärkung

Frequency response (from 40Hz to 15kHz), dB	+0.02,-0.07	Excellent
Noise level, dB (A)	-96.6	Excellent
Dynamic range, dB (A)	96.3	Excellent
THD, %	0.0016	Excellent
THD + Noise, dB (A)	-88.7	Good
Stereo crosstalk, dB	-97.3	Excellent
General Performance		Excellent

Tab.:3.6 zeigt, dass, bis auf eine Ausnahme (*THD + Noise*), sämtliche Messwerte mit *Excellent* bewertet wurden. Die Messergebnisse decken sich damit mit den hohen Erwartungen, die an eine Soundkarte dieser Qualität gestellt werden können.

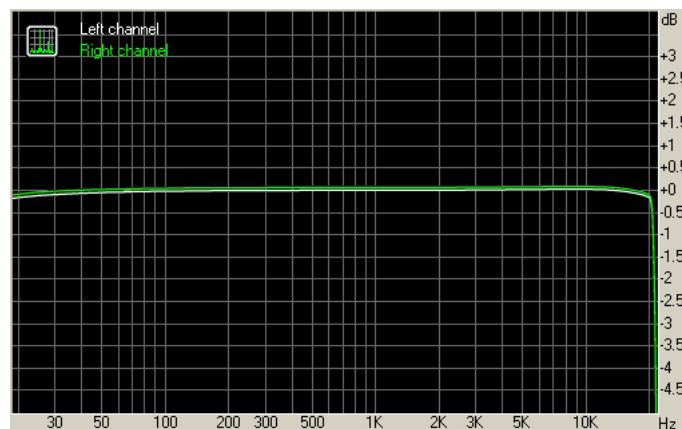


Abbildung 3.19: Amplitudengang Fireface 800

Abb.:3.19 zeigt den Amplitudengang. Aufgrund der im Hörbereich äußerst geringen Abweichungen von unter 0,1dB kann der Frequenzgang als linear bezeichnet werden.

3.4 Messergebnisse

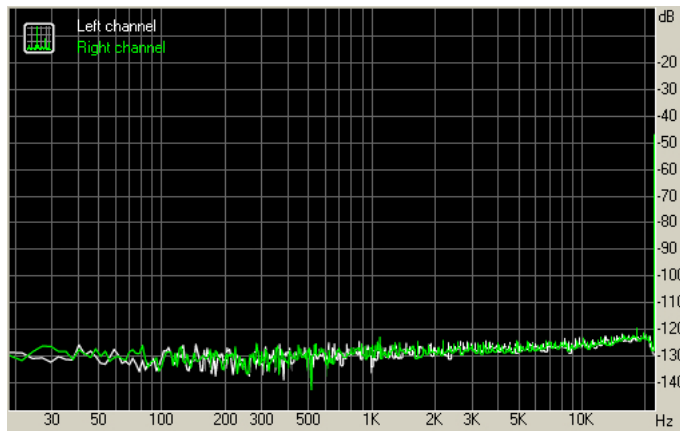


Abbildung 3.20: Rauschen des Fireface 800

Abb.:3.20 zeigt den Rauschpegel. Es ist ein leichter Anstieg hin zu hohen Frequenzen erkennbar, der Pegel bleibt aber immer unter ca. -120dB.

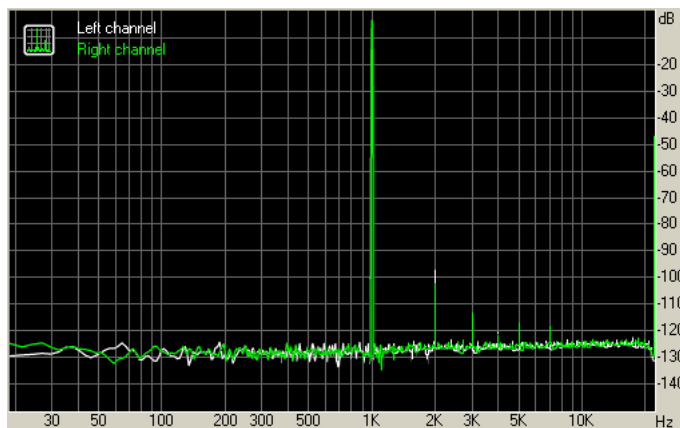


Abbildung 3.21: Total Harmonic Distortion des Fireface 800

Abb.:3.21 zeigt die THD des Fireface 800. Gut zu erkennen sind hier die geringen harmonischen Verzerrungen bei den Vielfachen der Grundfrequenz.

3 Hardware

3.4.4 Direkter Vergleich des analogen Ausganges mit den anderen Soundkarten

Tabelle 3.7: Vergleich der Messergebnisse aller Soundkarten

Test	HD Audio	RME FF 800	RasPi@0dB
Frequency response (40Hz-15kHz)	-0.13, -1,47	+0.02, -0.07	+0.17, -2.12
Noise level, dB (A)	-72.9	-96.6	-74.7
Dynamic range, dB (A)	72.9	96.3	71.2
THD, %	0.0058	0.0016	0.259
THD + Noise, dB (A)	0.090	0.0046	0.240
Stereo crosstalk, dB	-39.3	-97.3	-73.4

Tabelle 3.8: Vergleich der Messergebnisse der Raspberry Pi Soundkarte

Test	RasPi@0dB	RasPi@-4dB	RasPi@+4dB
Frequ. response (40Hz-15kHz)	+0.17, -2.12	+0.17, -2.12	+0.19, -2.12
Noise level, dB (A)	-74.7	-70.6	-79.0
Dynamic range, dB (A)	71.2	67.7	77.2
THD, %	0.259	0.166	4.711
THD + Noise, dB (A)	0.240	0.197	2.451
Stereo crosstalk, dB	-73.4	-69.5	-75.4

Amplitudenverlauf

Im direkten Vergleich aller Soundkarten fällt der starke Abfall ab ca. 2kHz auf, der bei 20kHz schließlich bei ca. -2,12dB liegt (siehe Abb.:3.22).

Rauschen

Das Rauschen bewegt sich im Bereich von knapp unter -70dB und liegt damit in etwa gleich auf mit der Laptop-Soundkarte, aber ca. 20dB über dem Fireface 800 (siehe Abb.:3.23).

3.4 Messergebnisse

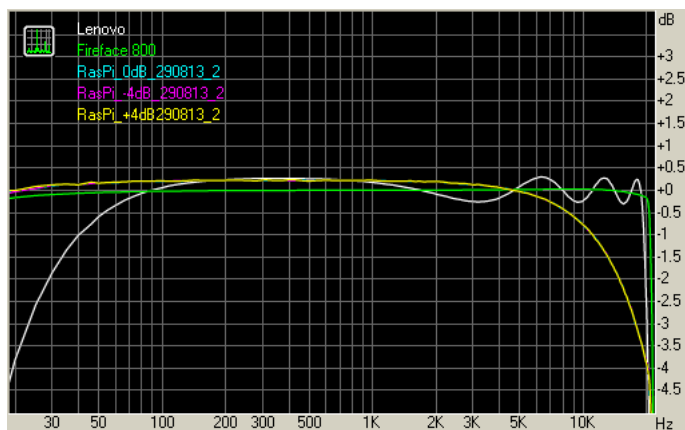


Abbildung 3.22: Vergleich der Amplitudengänge (Türkis: Rasp. Pi 0dB-Verstärkung, Magenta: Rasp. Pi -4dB-Verstärkung, Gelb: Rasp. Pi +4dB-Verstärkung, Grün: Fireface 800, Weiß: Realtek High Definition Audio)

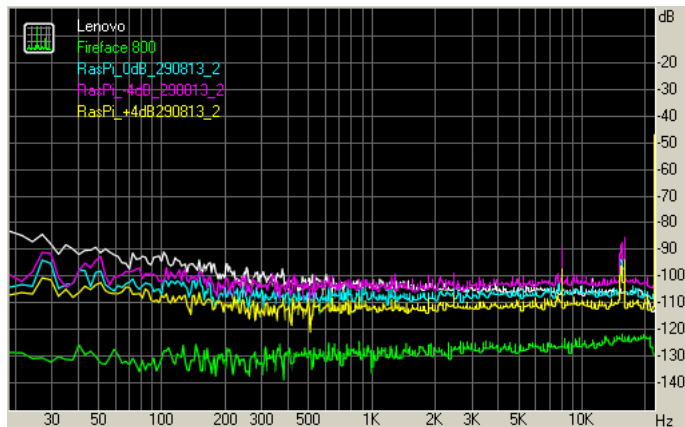


Abbildung 3.23: Vergleich des Rauschens (Türkis: Rasp. Pi 0dB-Verstärkung, Magenta: Rasp. Pi -4dB-Verstärkung, Gelb: Rasp. Pi +4dB-Verstärkung, Grün: Fireface 800, Weiß: Realtek High Definition Audio)

3 Hardware

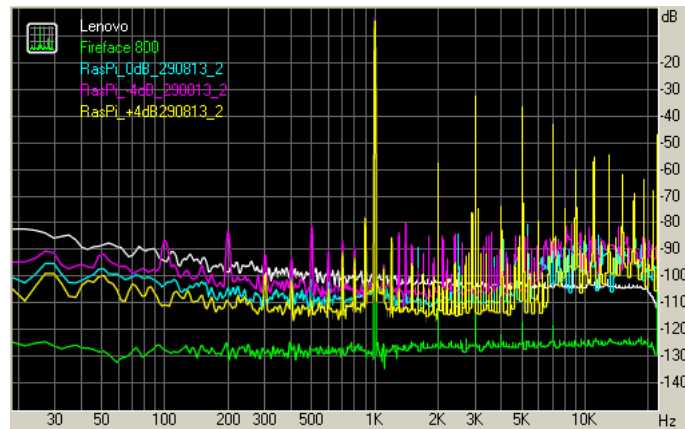


Abbildung 3.24: Vergleich der Total Harmonic Distortion (Türkis: Rasp. Pi 0dB-Verstärkung, Magenta: Rasp. Pi -4dB-Verstärkung, Gelb: Rasp. Pi +4dB-Verstärkung, Grün: Fireface 800, Weiß: Realtek High Definition Audio)

THD

Für die Praxis relevanter ist die, vor allem bei einer Verstärkung von +4dB, extrem hohe THD von bis zu 4,7%. Im Hi-Fi-Bereich kann eine THD von unter 1% erwartet werden, dieser Wert wird allerdings in keiner der drei Gain-Einstellungen erreicht (siehe Abb.: 3.24).

Die Messwerte zeigen, dass die THD mit sinkendem Gain zurückgeht, daher sollte, wenn möglich, auf eine externe Verstärkung des Signals zurückgegriffen werden.

3.5 Latenzmessung

Da, besonders für Live-Anwendungen, die Latenz eines Audiosystems von großer Bedeutung ist, wurde diese ebenfalls gemessen. Unterschieden wurden dabei zwei Betriebsarten:

- Half-duplex-Betrieb
- Full-duplex-Betrieb

3.5 Latenzmessung

In der *Half-duplex*-Betriebsart wird untersucht, wie lange die Verzögerung zwischen dem Triggern eines Signals und der Ausgabe über Lautsprecher dauert. Dabei wurde sowohl die Verzögerung am analogen Ausgang des Raspberry Pi als auch an einem über USB angeschlossenen Audio-Interface gemessen.

Im *Full-duplex*-Betrieb wird ein Signal über einen Ausgang abgespielt und dabei sofort wieder an einem Eingang aufgenommen. Die zwischen den beiden Signalen entstehende Zeitspanne wird wiederum gemessen.

3.5.1 Half-duplex-Messung

Messaufbau

Als Triggersignal wurde der Tastenanschlag der Computertastatur über ein Mikrofon (Shure SM58) aufgenommen, danach das ausgegebene Signal direkt am Lautsprecher über ein Shure SM57 aufgezeichnet (siehe Abb.:3.25). Durch den Abstand der Mikrofone zur Schallquelle ergeben sich zusätzliche Verzögerungen (bei jeweils 2cm Abstand insgesamt ca. 0,1ms), die jedoch aufgrund ihrer Größe nicht berücksichtigt wurden.

3 Hardware

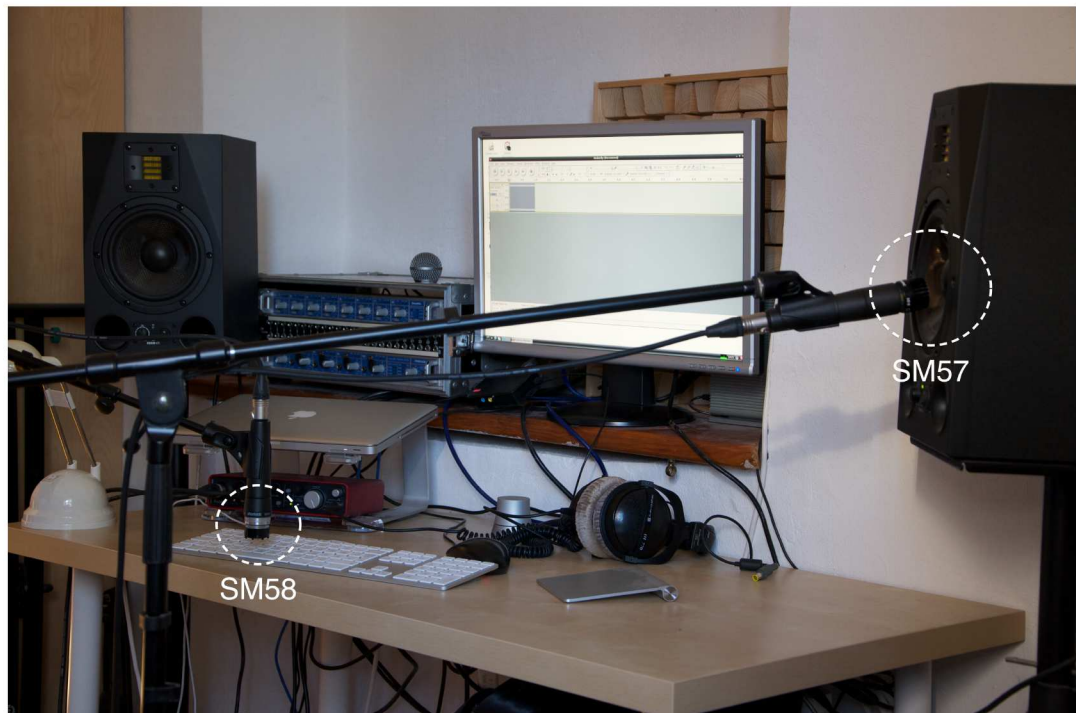


Abbildung 3.25: Messaufbau Half-duplex-Latenzmessung

Untersucht wurden sowohl die Latenz beim Abspielen einer Audio-Datei über den analogen Ausgang des Raspberry Pi als auch über ein USB-Audiointerface. Beim Audio-Interface handelt es sich um ein *Focusrite Scarlett 2i4*[15]. Abgespielt, aufgenommen und ausgewertet wurden die Signale direkt am Raspberry Pi in *Audacity*[7].

Messergebnisse

Die Ergebnisse in Tabelle 3.9 sind jeweils Mittelwerte über drei Messungen.

Tabelle 3.9: Latenzmessung im Half-duplex-Betrieb

Scarlett 2i4	766ms
Analog Output	323ms

3.5.2 Full-duplex-Messung

Da der Raspberry Pi über keinen analogen Audio-Eingang verfügt, kann eine Full-duplex-Latenzmessung nur über eine externe Soundkarte durchgeführt werden. Hier wurde wieder die Scarlett 2i4 eingesetzt. Um den Aus- mit dem Eingang zu verbinden, wurde ein 30cm langes Klinkenkabel verwendet (siehe Abb.:3.26).



Abbildung 3.26: Messaufbau Full-duplex-Latenzmessung

3 Hardware

Die Full-duplex-Messung wurde mit dem Tool *alsa_delay*¹² durchgeführt. In *alsa_delay* können folgende Einstellungen getroffen werden:

- Ausgabegerät (*playdev*)
- Eingabegerät (*captdev*)
- Samplingfrequenz (*fsamp*)
- Period-Size (*frsize*)
- Fragment-Number (*nfrags*)

Die Period-Size und die Fragment-Number ergeben zusammen die Hardware-Buffergröße ($frsize * nfrags$). Je größer der Hardwarebuffer, desto länger die Latenzzeit, je kürzer die Latenz, desto mehr Ressourcen werden benötigt.

Messergebnisse

Die niedrigste Latenzzeit, die für den Raspberry Pi bei einer Samplingfrequenz von 48kHz erreicht werden konnte, beträgt 20,348ms.

3.6 Fazit

Grundsätzlich kann bei Verwendung des analogen Ausganges des Raspberry Pi kein *Hi-Fi Sound* erwartet werden, wobei es sich dabei natürlich um keine exakte Definition handelt.

Im Internet finden sich keine Daten oder Messwerte zur Soundkarte, es wird allerdings immer wieder in Foren davon gesprochen, dass der analoge Ausgang 11bit Signale ausgeben kann. Auf welcher Basis diese Aussage zu Stande kommt, ist allerdings nicht klar.

Auch die in dieser Arbeit erstellten Messwerte ergeben kein vollständiges Abbild des analogen Audio-Outputs, aber sie verdeutlichen die durchschnittliche Qualität und Limitierungen durch die Hardware.

In den ersten Versionen von Raspbian „wheezy“ gab es außerdem Probleme mit lautem Knacksen beim Starten und Stoppen von Audiodateien und Störgeräuschen während des Abspielens. Die Ursache dafür war das Ein-

¹² http://manpages.ubuntu.com/manpages/trusty/man1/alsa_delay.1.html

bzw. Ausschalten des Audiotreibers und konnte nur durch umständliche *Workarounds* umgangen werde. Beim letzten Raspbian „wheezy“ Update wurde das Problem zwar behoben, im praktischen Alltag sollte man aber damit rechnen noch vereinzelt darauf zu stoßen. Bei der Verwendung des USB-Audiointerfaces waren vor dem Abspielen ebenfalls immer wieder Störgeräusche (Knacksen) zu hören.

Um über den analogen Audio-Ausgang eine bestmögliche Klangqualität zu erreichen, sollte die digitale Verstärkung so gering wie möglich eingestellt werden. Besonders über 0dB steigt die THD rasch an und wird bald sehr stark hörbar.

Da der analoge Audio-Ausgang allerdings nur eine geringe Ausgangsleistung zur Verfügung stellt, ist die Verringerung der Verstärkung in manchen Fällen nicht möglich. Hier sollte ein externer Verstärker nachgeschaltet werden.

Besonders die Half-duplex-Latenzmessungen zeigen, dass beim Abspielen von Audio-Dateien mit merkbaren Verzögerungen gerechnet werden muss. Obwohl die Latenzen im Einzelfall durchaus variieren können (z.B. durch unterschiedliche Audio-Einstellungen oder beim Verwenden verschiedener Programme), sollte dieser Aspekt in der Praxis immer berücksichtigt werden.

3.6.1 Alternativen zum analogen Ausgang

Wenn Audio-Anwendungen eine gute bis sehr gute Klangwiedergabe erfordern, ist (in Kombination mit Raspbian „wheezy“) die Verwendung einer externen USB-Soundkarte meist die billigste und effizienteste Lösung.

USB-Audio-Interfaces sind bereits ab wenigen Euros erhältlich und stellen in den allermeisten Fällen schon im untersten Preisbereich eine Verbesserung zum analogen Ausgang des Raspberry Pi dar.

Wichtig ist, darauf zu achten, dass die USB-Soundkarte mit Linux kompatibel ist. Dies ist bei allen *class-compliant* Geräten der Fall, also Geräten, die ohne spezielle Treiber *out of the box* funktionieren. Auf der Homepage des ALSA-Projects[2] werden viele Interfaces gelistet, die unter Linux funktionieren.

Im Laufe der Messungen hat sich gezeigt, dass es aufgrund der Rechnerar-

3 Hardware

chitektur zu einer erhöhten CPU-Belastung und unvorhersehbarem Verhalten der USB-Soundkarte (Störgeräusche, Aussetzer) kommen kann, wenn der Raspberry Pi über VNC¹³ ferngesteuert wird.

Eine weitere Alternative ist das Verwenden von speziellen Adaptern, die das analoge Audiosignal aus dem HDMI-Signal gewinnen und dann über Klinken- oder Cinchbuchsen ausgeben. Diese sind allerdings im Vergleich zu USB-Interfaces meist teurer. Wird der Raspberry Pi zum Beispiel an einen Fernseher über HDMI angeschlossen, ist auch die Ausgabe über die Lautsprecher des Fernsehers möglich.

Die dritte Variante Audio auszugeben ist der SPDIF Ausgang. Da es sich dabei um einen digitalen Ausgang handelt, muss ein DAC nachgeschaltet werden, um ein analoges Signal zu erhalten. Dadurch trennt man, wie bei Verwendung des HDMI-Ausganges, die kritische DAC-Wandlung vom Raspberry Pi und man kann so, je nach verwendetem Wandler, sehr gute Ergebnisse erzielen.

In Zukunft kann mit weiteren Alternativen gerechnet werden, da derzeit von verschiedenen Herstellern Boards entwickelt werden, die durch Aufstecken auf den Raspberry Pi einen zusätzlichen Audio-Ausgang herstellen. Dabei wird der Umstand genutzt, dass an den GPIO-Pins¹⁴ der I^2S -Stream (siehe Abschnitt 3.2) anliegt und somit verwendet werden kann. Der Fokus dieser Boards liegt dabei auf einer qualitativ hochwertigen Sound-Ausgabe.

¹³ Virtual Network Computing

¹⁴ General-purpose input/output

4 Software

Derzeit gibt es mehrere Betriebssysteme, die für den Raspberry Pi verfügbar sind oder gerade portiert oder entwickelt werden. Je nach Einsatzgebiet eignen sich manche OS¹ mehr oder weniger gut, außerdem gibt es teils große Unterschiede im Funktionsumfang und der dafür verfügbaren Software.

Für Audioanwendungen sind in erster Linie zwei Betriebssysteme interessant und werden in diesem Kapitel näher vorgestellt:

Das von der Raspberry Pi Foundation entwickelte Raspbian „wheezy“ [34] und *Raspbmc* [35].

Während *Raspbmc* die Aufgaben eines Media-Centers erfüllt (Abspielen von Audio- und Videodateien), ermöglicht Raspbian „wheezy“ das Installieren von unterschiedlichen Programmen für verschiedenste Anwendungen, ähnlich wie bei „herkömmlichen“ Betriebssystemen wie Windows oder Mac OSX. Die für den Audio- und Computermusikbereich wichtigsten Audio-Programme werden ebenfalls in diesem Kapitel kurz vorgestellt.

4.1 Betriebssysteme

4.1.1 Raspbian „wheezy“

Raspbian ist ein inoffizieller Port² von *Debian wheezy armhf* [8]. Der Port war notwendig, da der offizielle Debian wheezy armhf Release erst ab CPUs mit einer ARMv7-A Architektur kompatibel ist, während die Raspberry Pi CPU auf der früheren ARMv6 Architektur basiert (siehe Kapitel 3). [22]

¹ Operation System

² Anpassung von plattformgebundener Software an ein anderes (Hardware-)System [26]

4 Software

Raspbian „wheezy“ wird von der Raspberry Pi Foundation herausgegeben und weiter entwickelt und ist frei verfügbar.

Besonders Linux-Einsteigern wird dieses OS empfohlen, da es aufgrund der Windows-ähnlichen Bedienung (siehe Abb.:4.1) schnelle Fortschritte ermöglicht.

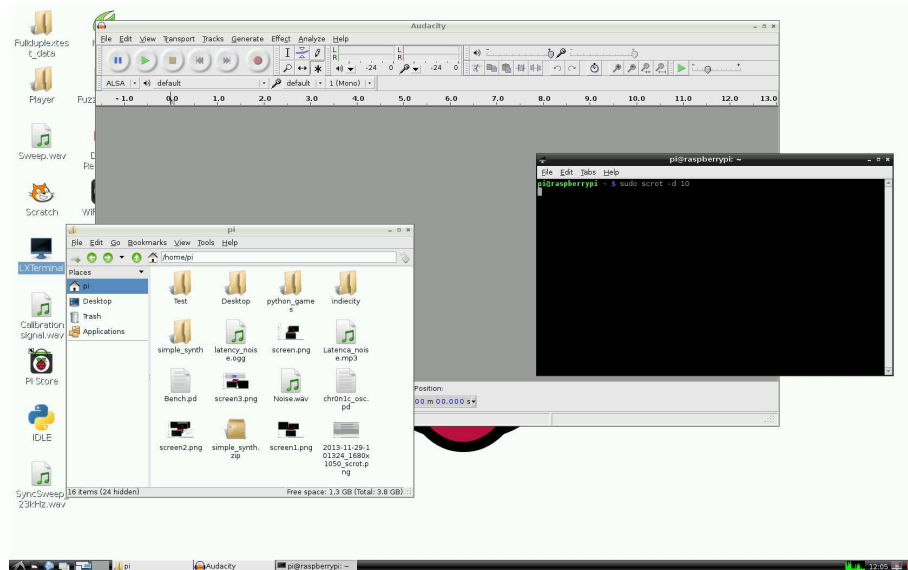


Abbildung 4.1: Screenshot Raspbian „wheezy“

Das OS enthält diverse vorinstallierte Programme, im Folgenden eine Auswahl:

- Dillo³ und Midori⁴, Webbrowser.
- Scratch⁵, eine visuelle Programmiersprache, die vor allem Kindern das Programmieren näher bringen soll.
- IDLE⁶, eine Entwicklungsumgebung für Python
- Calculator, ein virtueller Taschenrechner

³ <http://www.dillo.org>

⁴ <http://midori-browser.org>

⁵ <http://scratch.mit.edu>

⁶ <http://www.python.org/getit/>

- xpdf⁷, ein PDF-Viewer

Seit 3. Juni 2013 erfolgt die Installation des Betriebssystems nicht mehr durch das Einspielen eines Images auf eine SD-Karte, sondern über eine Eigenentwicklung namens *NOOBS (New Out Of Box Software)*. Diese ermöglicht eine vereinfachte Installation durch das Kopieren der Dateien auf eine SD-Karte. Eine Installation ist ab einer Kartengröße von 2GB möglich, empfohlen werden aber mindestens 4GB.

Die Installation von Programmen (oder Paketen) ist entweder über den in Linux Umgebungen üblichen Befehl

```
sudo apt get install [PAKETNAME]
```

möglich oder den *Pi Store* [25]. Dieser ermöglicht, ähnlich wie z.B. der Apple App Store [4], das Browsen durch verschiedene Kategorien sowie das Downloaden und Installieren von Programmen.

Die wichtigsten für den Audio- und Computermusikbereich bereits dafür erhältlichen Programme werden in Kapitel 4.2 kurz vorgestellt.

4.1.2 Raspbmc

Raspbmc[35] basiert auf Raspbian und XBMC⁸. XBMC wurde ursprünglich nur für Microsofts X-Box entwickelt (seit 2002) und erlaubt in Form eines Media Centers das einfache Abspielen und Streamen von Multimedia Inhalten auf Fernsehern (siehe Abb.:4.2).

Raspbmc ist die Weiterentwicklung von XBMC und ermöglicht die Verwendung des Raspberry Pi als Media Center. Seit Februar 2013 ist der derzeitige Release (Vers. 1.0) erhältlich.

Es erlaubt das Abspielen von Videos (1080p), Photos und Musik, außerdem werden Apples iTunes- sowie AirPlay[3]-Standards unterstützt. Raspbmc kann über USB-Maus und Tastatur oder, wenn der Rechner via HDMI an einen Fernseher angeschlossen wurde, über eine herkömmliche Fernbedienung gesteuert werden.

⁷ <http://www.foolabs.com/xpdf/index.html>

⁸ X-Box Media Center

4 Software

Wenn eine Internetverbindung besteht, können Dateien gestreamt werden, auch Mediatheken verschiedener TV-Sender sind so verfügbar. Außerdem kann Raspbmc als AFP-,NFS- oder SMB- Server konfiguriert werden und ermöglicht so auch Filesharing.

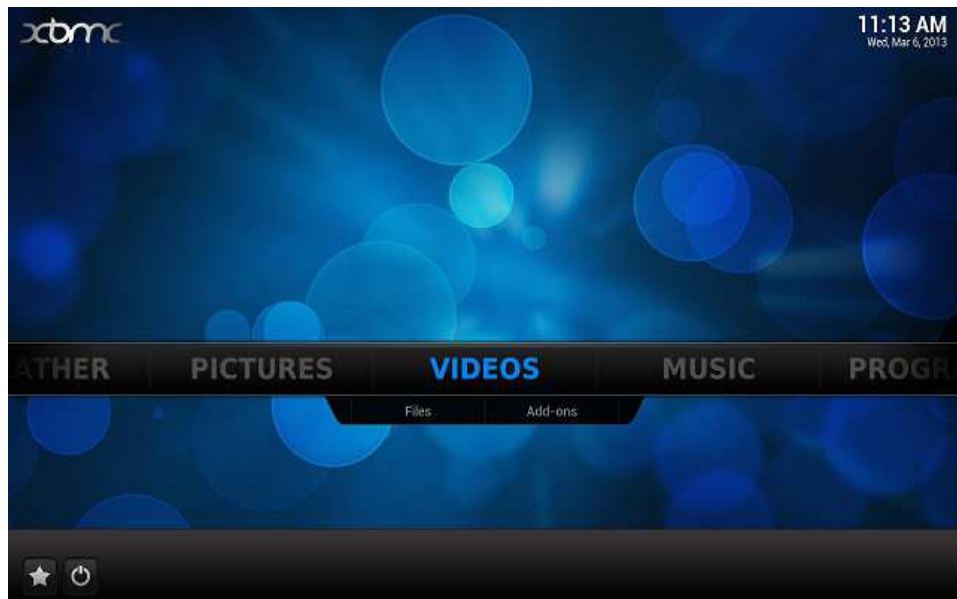


Abbildung 4.2: Screenshot Raspbmc [36]

4.2 Musik-Software

4.2.1 Pure Data

Pure Data^[30] (kurz *Pd*) ist eine visuelle Programmiersprache, die in erster Linie für Multimedia-Anwendungen entwickelt wurde. Die Software ist *open source* und somit frei verfügbar.

Pd gehört zu den datenstromorientierten Programmiersprachen und hat aufgrund der visuellen Darstellung in vielen Bereichen Vorteile gegenüber anderen Sprachen.

Es gibt zwei weit verbreitete Versionen:

- Pd vanilla: beinhaltet nur die Kernfunktionen von Pd und konzentriert sich damit auf Audio- und MIDI- Signalverarbeitung.
- Pd extended: beinhaltet zusätzliche Libraries, z.B. für graphische Anwendungen, physical modeling oder sensor-basierte Programme.

Aufbau Grundsätzlich können die Objekte in Pd in drei Kategorien eingeordnet werden: [42]

- Quellen: wie etwa Mikrofon, Kamera, Netzwerk, Sinus-Generator
- Knoten: manipulieren Datenströme, beispielsweise durch Invertieren; Addition von mehreren Datenströmen oder andere mathematische Operationen, wie z. B. Verzerrung durch Wurzelziehen der Lautstärkewerte
- Senken: das sind Ausgabegeräte wie Lautsprecher und Bildschirm

Diese Objekte können im GUI⁹ miteinander verbunden werden und ermöglichen so auch das Schaffen komplexer Systeme (siehe Abb.:4.3).

Aufgrund der begrenzten Ressourcen des Raspberry Pi sollte, wo möglich, Pd bzw. der Raspberry Pi *headless* betrieben werden, d.h. ohne GUI. Durch den Verzicht auf eine grafische Darstellung können auch aufwendigere Patches fehlerfrei ausgeführt werden.

⁹ Graphical User Interface

4 Software

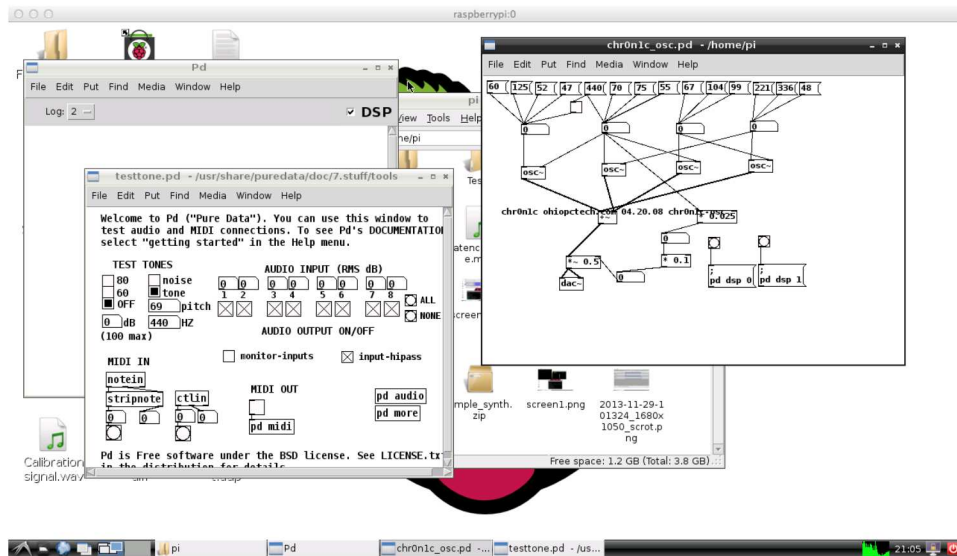


Abbildung 4.3: Screenshot Pd

Für den Raspberry Pi ist zur Zeit (Stand: 12.2013) Pd Extended für Raspbian „wheezy“ in der Version 0.43.3 erhältlich.

Bei den im Rahmen dieser Arbeit durchgeführten Tests funktionierte Pd sehr stabil, dank der lebendigen Community kann außerdem in naher Zukunft mit Weiterentwicklungen gerechnet werden.

4.2.2 Audacity

Audacity[7] ist eine Open-source-Software, die das Aufnehmen und Editieren von Audio-Daten ermöglicht und für verschiedene Betriebssysteme verfügbar ist.

Wichtige Funktionen:

- Erstellen von Downmixes beliebig vieler Spuren
- Im- und Export verschiedener Audioformate (AIFF, MATLAB MAT, MPEG, etc.)
- Tongeneratoren für verschiedene Rauscharten und Signale

- mehrere hundert Filter- und Sound-Effekte

Audacity läuft bereits sehr stabil und durch die geringen Systemvoraussetzungen ist auch zügiges Arbeiten möglich.

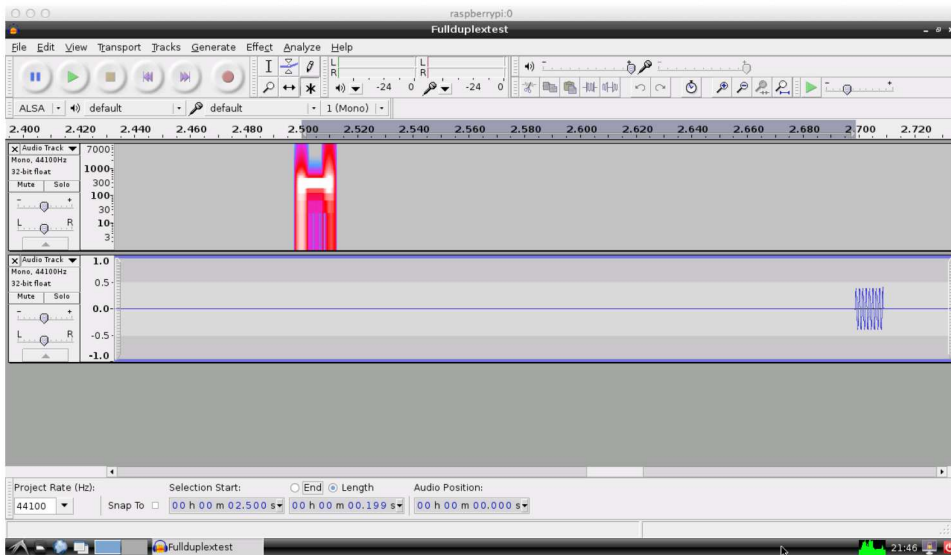


Abbildung 4.4: Screenshot Audacity

4.2.3 JACK

JACK^[18] ermöglicht das Routen von Audio- und MIDI-Daten in Echtzeit und mit geringer Latenz zwischen Soft- und Hardware. Dabei ist sowohl das Routen zwischen Software, als auch zwischen Software und ein oder mehreren Audiogeräten, möglich. Außerdem erlaubt JACK die Verbindung über LAN¹⁰ oder WAN.¹¹

JACK wird grundsätzlich ohne GUI verwendet, die Anwendung QjackCtl^[31] (siehe Abb.: 4.5), die ebenfalls für den Raspberry Pi verfügbar ist, erlaubt allerdings auch die Steuerung via GUI.

¹⁰ Local Area Network

¹¹ Wide Area Network

4 Software

Die Installation von JACK unter Raspbian „wheezy“ gestaltet sich verhältnismäßig aufwändig, da vorher diverse Einstellungen (z.B. Overclocking, Abschaltung nicht genutzter Dienste) geändert werden müssen.[17] Die meisten Einstellungsänderungen zielen aber auf eine allgemeine Leistungserhöhung für Audio-Anwendungen ab und sind daher generell zu empfehlen.

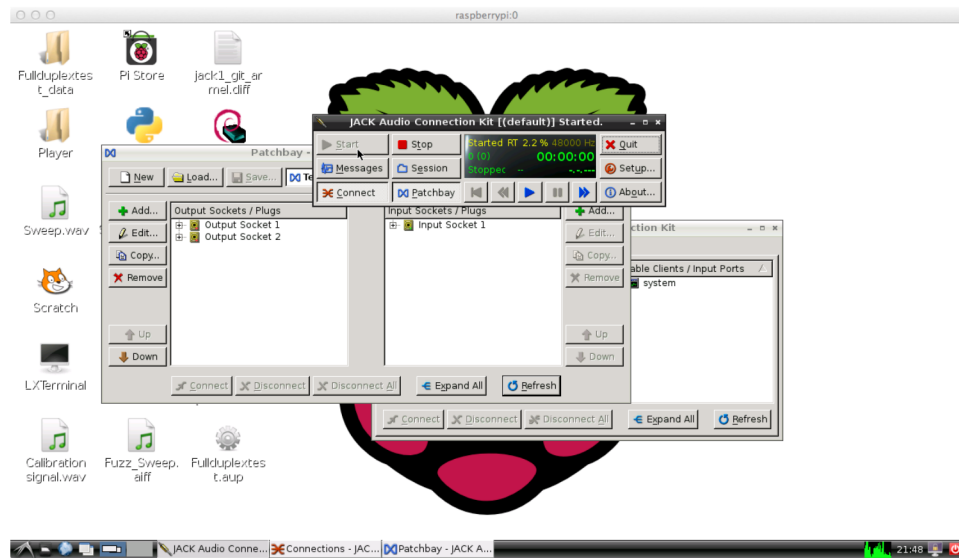


Abbildung 4.5: Screenshot QjackCtl

4.2.4 Ardour

Bei Ardour[5] handelt es sich um eine DAW¹². Das Programm erlaubt das Aufnehmen und spätere Editieren von Audio-Daten. Ardour ist für den Raspberry Pi frei erhältlich (Stand: 12.2013). Um Ardour betreiben zu können, muss JACK (siehe Kap.:4.2.3) installiert sein.

Im Rahmen dieser Arbeit erwies sich Ardour als sehr instabil, es konnte erst durch mehrere Einstellungsänderungen überhaupt erst zum Laufen gebracht werden. Eine Voraussetzung für das Starten von Ardour ist eine

¹² Digital Audio Workstation

4.2 Musik-Software

lauffähige Version von JACK (siehe Kap.:4.2.3). Da aber auch JACK derzeit noch fehleranfällig ist, ergibt sich daraus eine weitere Quelle für Abstürze.

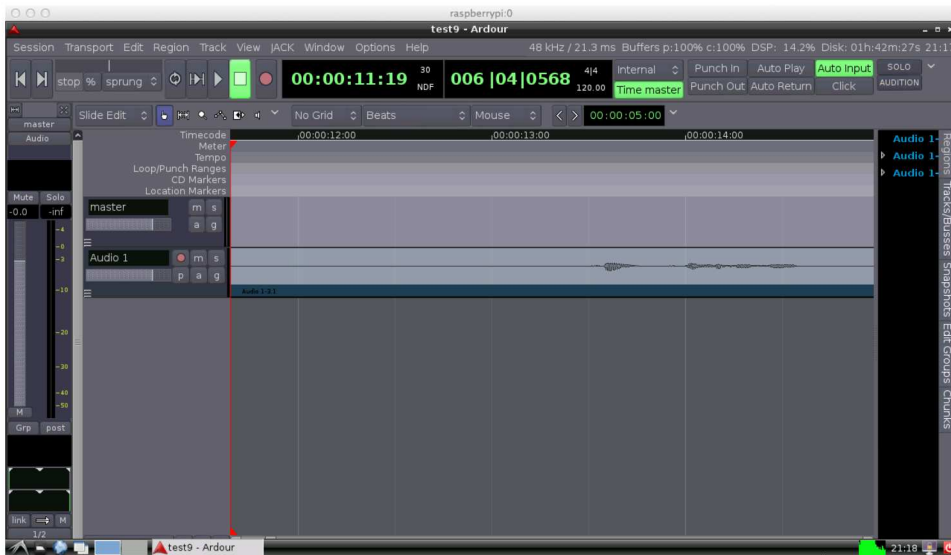


Abbildung 4.6: Screenshot Ardour

4.2.5 MPD

MPD¹³[20] ist ein Server, der das Abspielen von Musikdateien lokal ermöglicht. Um den MPD-Server steuern zu können, ist ein Client notwendig (siehe Abb.:4.7, 4.8). Mittlerweile sind für fast alle Betriebssysteme Clients erhältlich, die das Fernsteuern des Servers ermöglichen. So können z.B. über eine Client-App auf einem iPhone oder Android-Smartphone lokale Musikdateien abgespielt oder Dateien aus dem Internet gestreamt werden.

¹³ Music Player Daemon

4 Software

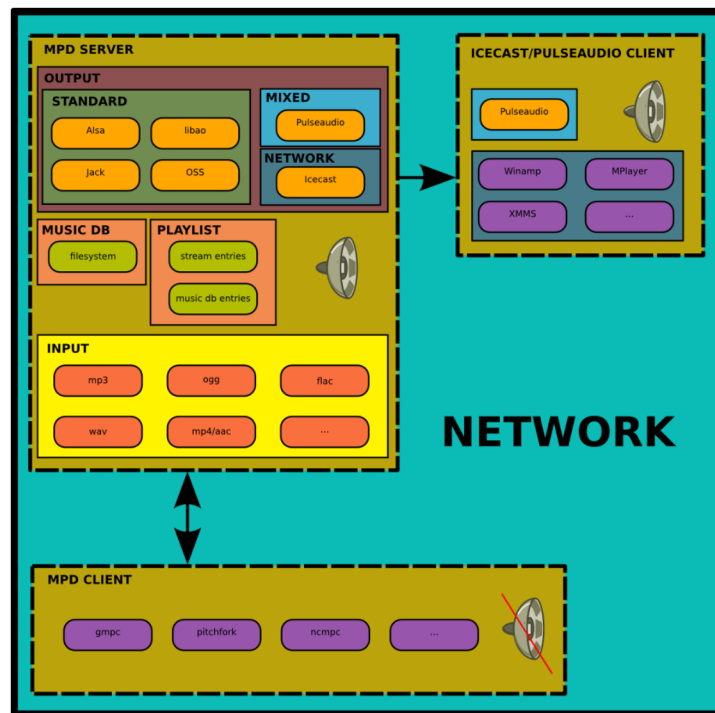


Abbildung 4.7: Übersicht der MPD/Client-Architektur [20]

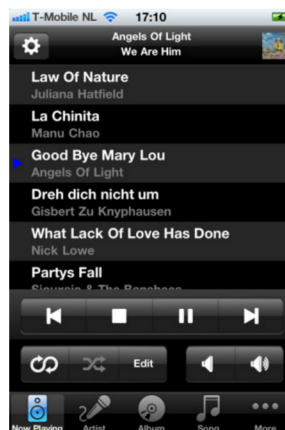


Abbildung 4.8: Screenshot iPhone-Client [21]

5 Projekte

Aufgrund seines günstigen Preises und des Open-source-Gedankens eignet sich der Raspberry Pi besonders gut für *DIY*¹-Projekte. Im Folgenden werden zwei Projekte kurz vorgestellt, die verschiedene Eigenschaften seiner Hardware nutzen.

5.1 FM Radio Transmitter

Die Nutzung der GPIOs erweitert die Möglichkeiten des Raspberry Pi erheblich. So kann über diese Schnittstelle nicht nur, wie ursprünglich vorgesehen, mit anderer Hardware kommuniziert werden (siehe Projekt [Guitar Extended](#)), sondern auch, wie in diesem Projekt[29], über eine Antenne und ein entsprechendes Skript eine Funkübertragung ermöglicht werden. Alles, was dazu nötig ist, ist ein Raspberry Pi, das Skript, ein ca. 200mm langer Draht und ein Breadboard².

Achtung: Seit 2006 sind in Österreich FM Transmitter nur bis zu einer Sendeleistung von 50mW erlaubt. Damit ist eine Funkübertragung nur wenige Meter weit möglich. Bei dem hier vorgestellten Projekt kann eine weitaus größere Sendeleistung erreicht werden (ca. 9-14mW abhängig von Antenne, Umgebungsbedingung, etc...). Dementsprechend muss darauf geachtet werden, nicht gegen geltende Gesetze zu verstoßen.

¹ Do It Yourself

² Ein Breadboard ist eine Steckplatine mit der elektronische Bauteile mechanisch befestigt und elektronisch verbunden werden können. Dadurch entfällt langwieriges Löten bei experimentellen Versuchsaufbauten.

5 Projekte

5.1.1 Aufbau

Das Programm *pifm*[29] verwendet die Hardware des Raspberry Pi, die ursprünglich für die Frequenzspreizung von Taktsignalen an den GPIOs vorgesehen war, zum Ausgeben von FM Funksignalen. Dazu wird GPIO 4 genutzt, der über eine Antenne die Radiowellen abstrahlt. Die Antenne ist optional, d.h. auch ohne diese ist eine Übertragung möglich, allerdings sinkt die Reichweite auf einige wenige Meter ab.

pifm bildet den peripheren Bus 0x20000000 im physikalischen Speicher mittels `/dev/mem` und `mmap` im virtuellen Adressraum ab. Als nächstes aktiviert es das Taktgeneratormodul, legt es auf GPIO 4 und stellt eine Trägerfrequenz von 100MHz ein, wobei diese vom Benutzer im Bereich von ca. 1MHz bis 250MHz variiert werden kann. Für FM Übertragung ist allerdings ein Bereich von 88MHz - 108 MHz sinnvoll.

Es können .wav Dateien mit 16 bit und 44,1kHz in Stereo übertragen werden, wobei das Programm auch eine Präemphase vornimmt, um das Rauschen zu reduzieren.

5.2 Guitar Extended

In diesem Projekt[28] wird über einen selbstgebauten Foot-Controller Pd (siehe Kap.:4.2.1) gesteuert, der wiederum auf einem Raspberry Pi installiert wurde. Die Kommunikation zwischen dem Controller und Pd erfolgt über ein Arduino-Board[6]³ via *pduino*. [24]

³ Ein Arduino-Board ist ein Mikrocontroller, der, je nach geladenem Programm, einfache Befehle ausführen kann. Dadurch ist es möglich Aktuatoren sowie Sensoren anzuschließen, die mit der Außenwelt in Verbindung stehen.

5.2.1 Aufbau



Abbildung 5.1: Standbild aus einem Demonstrationsvideo[40]

Abb.:5.1 zeigt den Aufbau des Projektes. Auf dem Tisch in der Mitte ist links die Soundkarte und auf der rechten Seite der Raspberry Pi zu erkennen. Am Boden wird der selbstgebaute Foot-Controller mit den Füßen bedient.

Die Schalter und Potentiometer des Foot-Controllers werden mit den GPIOs eines Arduino Uno verbunden. Über ein USB-Kabel und pduino kann der Mikrocontroller Befehle an Pd schicken und so Patches steuern. Da der Raspberry Pi keinen analogen Audio-Eingang besitzt, muss eine USB-Soundkarte verwendet werden. Dies hat den Vorteil, dass der relativ schlechte Audio-Ausgang des Raspberry Pi nicht verwendet werden muss und somit die Audio Qualität insgesamt erhöht wird.

Die zum Kommunizieren zwischen Pd und dem Arduino-Board verwendete Software pduino besteht aus zwei Teilen und erlaubt eine einfache Steuerung per Fuß.

Der erste Teil ist ein Protokoll namens *Firmata*[14], das die Kommunikation von Mikrocontrollern und einer beliebigen Software auf einem Host Rechner erlaubt. Dafür muss Firmata auf dem Arduino installiert werden.

Der zweite Teil besteht aus einem Arduino-Objekt in Pd.

5 Projekte

Dieses erlaubt die Zuweisung der Arduino-GPIOs durch Pd. Dabei kann zwischen zwei Arten von GPIOs gewählt werden:

- digitale Pins
- analoge Pins

Je nach verwendetem Arduino-Board stehen eine unterschiedliche Anzahl dieser GPIOs zur Verfügung. Die digitalen Pins können nur *0* oder *1* ausgeben und eignen sich daher für Schalter und Knöpfe. Die analogen erlauben einen kontinuierlichen Wertebereich mit einer Auflösung von 10bit und eignen sich daher für Potentiometer.

In diesem Projekt wird als Klangquelle eine E-Gitarre verwendet, prinzipiell ist das System aber nicht nur auf diese Quelle beschränkt, wodurch auch eine Anwendung als Effektbox für Stimme, Synthesizer etc. denkbar wird. Durch die große Anzahl an unterschiedlichen Pd-Patches ergibt sich eine hohe Flexibilität des Gesamtsystems.

5.3 Ein kurzer Blick in die Zukunft

Obwohl bereits unzählige Projekte entstanden sind, die den Raspberry Pi als integralen Bestandteil verwenden, wurden die Möglichkeiten der Hardware noch nicht vollständig ausgeschöpft. Somit ist es auch nicht verwunderlich, dass es zur Zeit (02.2014) keine Pläne gibt eine neue Version auf den Markt zu bringen.

Auch im Audibereich stellt die größte Einschränkung noch die teils sehr instabile Software dar. Insbesondere bei Anwendungen, die JACK verwenden, kommt es oft zu Abstürzen, wodurch eine professionelle Nutzung schwierig oder unmöglich ist. Andererseits zeigen Programme wie Pd oder Audacity, dass bereits jetzt stabile Anwendungen möglich sind und erlauben auch komplexe Projekte.

Besonders auch in Kombination mit Arduino-Boards ergeben sich zig Anwendungen, die durch das Einbinden von Sensoren und Aktuatoren in Pd gerade für Kunstinstallationen sehr interessant sein können.

Literatur

- [1] *Advanced Peripheral Bus*. URL: <http://www.itwissen.info/definition/lexikon/APB-advanced-peripheral-bus-APB-Bus.html> (besucht am 01. 11. 2013) (siehe S. 11).
- [2] *ALSA-Project*. URL: <http://www.alsa-project.org/main/index.php/Matrix:Main> (besucht am 01. 11. 2013) (siehe S. 35).
- [3] *Apple Airplay*. URL: <http://www.apple.com/de/airplay/> (besucht am 27. 12. 2013) (siehe S. 39).
- [4] *Apple App Store*. URL: <http://www.apple.com/de/osx/apps/app-store.html> (besucht am 04. 12. 2013) (siehe S. 39).
- [5] *Ardour*. URL: <http://ardour.org> (besucht am 02. 11. 2013) (siehe S. 44).
- [6] *Arduino*. URL: <http://arduino.cc> (besucht am 02. 11. 2013) (siehe S. 48).
- [7] *Audacity*. URL: <http://audacity.sourceforge.net> (besucht am 02. 11. 2013) (siehe S. 32, 42).
- [8] *Debian wheezy armhf*. URL: <https://wiki.debian.org/ArmHardFloatPort> (besucht am 04. 12. 2013) (siehe S. 37).
- [9] *Eben Upton talking about Raspberry Pi*. März 2012. URL: <http://commons.wikimedia.org/wiki/File:EbenUptontalkingaboutRaspberryPi.jpg> (besucht am 01. 11. 2013) (siehe S. 4).
- [10] *Egoman*. URL: <http://www.egoman.com.cn> (besucht am 01. 11. 2013) (siehe S. 4).
- [11] *elinux.org - RedPi.jpg*. URL: <http://elinux.org/images/b/b7/RedPi.jpg> (besucht am 01. 11. 2013) (siehe S. 5).

Literatur

- [12] *elinux.org - RPi-alpha.jpg*. URL: <http://elinux.org/images/6/65/RPi-alpha.jpg> (besucht am 01. 11. 2013) (siehe S. 7).
- [13] *Fireface 800*. URL: http://www.rme-audio.de/products_fireface_800.php (besucht am 01. 11. 2013) (siehe S. 13).
- [14] *Firmata*. URL: http://firmata.org/wiki/Main_Page (besucht am 02. 11. 2013) (siehe S. 49).
- [15] *Focusrite*. URL: <http://global.focusrite.com/usb-audio-interfaces/scarlett-2i4> (besucht am 02. 12. 2013) (siehe S. 32).
- [16] *Google Giving*. URL: <http://www.google.com/giving/> (besucht am 01. 11. 2013) (siehe S. 4).
- [17] *Installationsanweisungen JACK*. URL: <http://wiki.linuxaudio.org/wiki/raspberrypi> (besucht am 19. 12. 2013) (siehe S. 44).
- [18] *JACK*. URL: <http://jackaudio.org> (besucht am 02. 11. 2013) (siehe S. 43).
- [19] Broadcom Europe Ltd. *BCM2835 ARM Peripherals*. 406 Science Park Milton Road Cambridge CB4 0WW, Feb. 2012 (siehe S. 12, 13).
- [20] *MPD*. URL: <http://www.musicpd.org> (besucht am 27. 12. 2013) (siehe S. 45, 46).
- [21] *MPD Screenshot*. URL: http://docs.mopidy.com/en/latest/_images/mpd-client-mpod.jpg (besucht am 29. 12. 2013) (siehe S. 46).
- [22] *Official Raspbian Website*. URL: http://www.raspbian.org/RaspbianFAQ#What_is_Raspbian.3F (besucht am 04. 12. 2013) (siehe S. 37).
- [23] *Open Charities*. URL: <http://opencharities.org/charities/1129409> (besucht am 31. 10. 2013) (siehe S. 3).
- [24] *Pduino*. URL: <http://playground.arduino.cc/Interfacing/PD> (besucht am 02. 11. 2013) (siehe S. 48).
- [25] *Pi Store*. URL: <http://store.raspberrypi.com/projects> (besucht am 04. 12. 2013) (siehe S. 39).
- [26] *Port*. URL: [wikipedia.org/wiki/Migration_\(Informationstechnik\)](http://wikipedia.org/wiki/Migration_(Informationstechnik)) (besucht am 04. 12. 2013) (siehe S. 37).
- [27] *Premier Farnell*. URL: <http://www.premierfarnell.com> (besucht am 01. 11. 2013) (siehe S. 4).

- [28] *Projekt Gitarren Effekte*. URL: <http://guitarextended.wordpress.com/2013/01/27/real-time-guitar-effects-with-raspberry-pi-pd-and-arduino/> (besucht am 27. 12. 2013) (siehe S. 48).
- [29] *Projekt Radio Transmitter*. URL: http://www.icrobotics.co.uk/wiki/index.php/Turning_the_Raspberry_Pi_Into_an_FM_Transmitter (besucht am 27. 12. 2013) (siehe S. 47, 48).
- [30] *Pure Data*. URL: <http://puredata.info> (besucht am 02. 11. 2013) (siehe S. 40).
- [31] *QjackCtl*. URL: <http://qjackctl.sourceforge.net> (besucht am 19. 12. 2013) (siehe S. 43).
- [32] *raspberrypi.org - atm1.jpg*. URL: <http://www.raspberrypi.org/wp-content/uploads/2011/10/atm1.jpg> (besucht am 01. 11. 2013) (siehe S. 6).
- [33] *raspberrypi.org - Google Giving*. URL: <http://www.raspberrypi.org/archives/3158> (besucht am 01. 11. 2013) (siehe S. 5).
- [34] *Raspbian "Wheezy"*. URL: <http://www.raspberrypi.org/downloads> (besucht am 03. 12. 2013) (siehe S. 37).
- [35] *Raspbmc*. URL: <http://www.raspbmc.com> (besucht am 02. 11. 2013) (siehe S. 37, 39).
- [36] *Raspbmc Screenshot*. URL: <https://www.modmypi.com/image/data/tutorials/nas-on-the-raspberry-pi/nas-and-the-raspberry-pi-1.jpg> (besucht am 29. 12. 2013) (siehe S. 40).
- [37] *Right Mark Audio Analyzer*. URL: <http://audio.rightmark.org/products/rmaa.shtml> (besucht am 01. 11. 2013) (siehe S. 14).
- [38] *RS-Components*. URL: <http://www.rs-components.com/index.html> (besucht am 01. 11. 2013) (siehe S. 4).
- [39] *TechCenter - raspberry-pi+overview.jpg*. URL: <http://1.bp.blogspot.com/-4otY-HaUKGc/TwRDt180SeI/AAAAAAAAAi8/AmpG0fQi14/s1600/raspberry-pi+overview.jpg> (besucht am 02. 11. 2013) (siehe S. 10).
- [40] *Videostandbild des Guitar Extended Projektes*. URL: <http://youtu.be/NwJNeouLqgQ> (besucht am 24. 02. 2014) (siehe S. 49).

Literatur

- [41] *Wikipedia - Aktiengesellschaft (Vereinigtes Königreich)*. URL: [http://de.wikipedia.org/wiki/Aktiengesellschaft_\(Vereinigtes_Knigreich\)](http://de.wikipedia.org/wiki/Aktiengesellschaft_(Vereinigtes_Knigreich)) (besucht am 01. 11. 2013) (siehe S. 4).
- [42] *Wikipedia - Pure Data*. URL: http://de.wikipedia.org/wiki/Pure_Data (besucht am 19. 12. 2013) (siehe S. 41).
- [43] *Wikipedia -Raspberry Pi*. URL: http://de.wikipedia.org/wiki/Raspberry_Pi (besucht am 01. 11. 2013) (siehe S. 6, 9).
- [44] *Wikipedia - System on a Chip*. URL: <http://de.wikipedia.org/wiki/System-on-a-Chip> (besucht am 01. 11. 2013) (siehe S. 10).