

schlizbäda

B a u a n l e i t u n g

... und ein Erfahrungsbericht zur Phoniebox

Phoniebox – die open-source-Alternative zur Toniebox



GNU Free Documentation License v1.3
© 2020 by schlizbäda

Datum:
29.03.2020



Dieses Dokument wurde mit dem Textsatzsystem L^AT_EX erstellt

Inhaltsverzeichnis

Abbildungsverzeichnis	4
Tabellenverzeichnis	5
1 Einleitung	6
1.1 Rechtliche Hinweise	6
1.2 Hinweis zur neumodernen „Genderei“ landauf, landab	7
1.3 Danksagung	8
1.4 Kurzbeschreibung	8
2 Technischer Aufbau von schlizbädas Phoniebox	10
2.1 Verwendete Komponenten / Stückliste	10
2.1.1 Gehäuse	11
2.1.2 Raspberry Pi 3B	11
2.1.3 HifiBerry MiniAmp	11
2.1.4 Bedientaster	12
2.1.5 Neuftech USB-RFID-Reader	12
2.1.6 Stromversorgung	12
2.1.7 OLED-Display	13
2.2 Selbstgebaute Teile	13
2.2.1 Lochrasterplatine	14
2.2.2 Relais für OnOff SHIM	17
2.2.3 Beschreibung des „Turms“	19
2.2.4 Verwendete Anschlussbuchsen	20
2.3 Holzarbeiten	21
2.3.1 Wahl des Gehäuses	21
2.3.2 Einbau des RFID-Readers	22
2.3.3 Sichtöffnung für Ladeanzeige der Powerbank	22
2.3.4 Sichtscheibe für OLED-Display	23
2.3.5 Halterung für die Powerbank	23
3 Installation der Software	25
3.1 <i>Raspbian Buster Lite</i> auf dem Raspberry Pi installieren	25
3.1.1 Erstellen einer SD-Karte für den Raspberry Pi mit <i>Raspbian Buster Lite</i>	25

Inhaltsverzeichnis

3.1.2	Anmeldung am Raspberry Pi über <i>ssh</i>	26
3.1.3	<i>Raspbian</i> konfigurieren	27
3.1.4	WLAN einrichten	27
3.2	Installation der Phoniebox-Software	28
3.3	Software / Installation	28
3.3.1	offen	28

Abbildungsverzeichnis

2.1	HifiBerry MiniAmp mit aufgelötetem Pimoroni OnOff SHIM	13
2.2	GPIO-Aufbau mit Lochrasterplatine	14
2.3	GPIO-Belegung – © Raspberry Pi Foundation (CC BY-SA 4.0)	14
2.4	Schaltung auf der Lochrasterplatine	16
2.5	Ansichten der Lochrasterplatine	17
2.6	Konzept zur gesteuerten Trennung des Tasters vom OnOff SHIM	18
2.7	Der Raspberry Pi-Turm der Phoniebox in seiner vollen Pracht	19
2.8	Der Raspberry Pi-Turm schematisch dargestellt	20
2.9	RFID-Reader vernünftig montiert	22
2.10	Ladeanzeige der Powerbank an der Unterseite der Phoniebox	22
2.11	OLED-Display für Statusanzeige in der Frontplatte der Phoniebox	23
2.12	Montage der Powerbank	24
3.1	erster Login über <i>ssh</i>	26

Tabellenverzeichnis

2.1	Verwendete Komponenten bei schlizbädas Phoniebox	10
2.2	Belegung der GPIO-Leiste bei schlizbädas Phoniebox	15
2.3	Verwendete Anschlussbuchsen bei schlizbädas Phoniebox	21

1 Einleitung

Vielen Dank für Ihr Interesse an schlizbädas Phoniebox.

Hierbei handelt es sich um ein kindgerechtes Abspielgerät auf Basis des Raspberry Pi für Audiodateien aller Art, insbesondere für Kinderhörspiele. Die Bedienung wurde bewusst einfach und für Kinder verständlich gehalten. Sie erfolgt über Taster für Ein/Aus, Play/Pause, vor/zurück und zur Lautstärkeregelung. Die Auswahl des gewünschten Hörspiels erfolgt über entsprechend beschriftete bzw. bebilderte RFID-Karten.

Die Software der Phoniebox basiert auf dem MIT-licenzierten Werk **RPi-Jukebox-RFID** von **MiczFlor**, die auf *github* unter folgender URL bereitgestellt wird:

<https://github.com/MiczFlor/RPi-Jukebox-RFID>

schlizbäda, der Autor dieser Dokumentation ist **nicht** der geistige Urheber der Phoniebox! Er hat lediglich ein Exemplar für sich gebaut und gibt mit diesem Dokument seine dabei gemachten Erfahrungen und die aufgetretenen Probleme weiter.

1.1 Rechtliche Hinweise

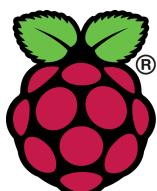
Diese Dokumentation wurde von schlizbäda unter der *GNU Free Documentation License v1.3* veröffentlicht. Sie enthält keine unveränderlichen Abschnitte.

Marken

Einige Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen können.

Das in dieser Dokumentation verwendete Raspberry Pi-Logo ist ein registriertes Warenzeichen der Raspberry Pi Foundation und darf nur zusammen mit der Kennzeichnung ® verwendet werden. Details finden sich im Internetauftritt der Raspberry Pi Foundation unter

<https://www.raspberrypi.org/trademark-rules/>



1 Einleitung

Links

In dieser Dokumentation sind Links zu externen Seiten im Internet enthalten. Diese Inhalte macht sich der Verfasser schlizbäda trotz Verlinkung nicht zu eigen, da sie nicht in seinem Einflussbereich stehen! Zum Zeitpunkt der Verlinkung waren keine rechtswidrigen Inhalte erkennbar. Eine ständige Überprüfung auf etwaige rechtsverstoßende Änderungen ist dem Verfasser nach geltendem Recht nicht zuzumuten.

Sollten aktuelle oder künftige Inhalte jedoch rechtswidrig sein, so kann der Autor darüber per e-mail an <mailto:himself@schlizbaeda.de> informiert werden. Es werden dann entsprechende Maßnahmen zur Beseitigung des/der betroffenen Links ergriffen.

Bildrechte

Alle inhaltlich relevanten Fotos und technischen Abbildungen in diesem Dokument stammen vom Verfasser schlizbäda selbst und werden hiermit von ihm unter der *Creative-Commons-Lizenz CC BY-SA 3.0* veröffentlicht. Sie dürfen daher von jedem bei Namensnennung des Urhebers in unveränderter oder auch in veränderter Form unter den gleichen Bedingungen weitergegeben werden:



Nicht vom Autor stammende Bilder sind mit einem entsprechenden Copyright-Vermerk versehen.

Die von der Raspberry Pi Foundation stammende Dokumentation samt Bildern ist in der Regel gemäß **CC BY-SA 4.0** lizenziert: <https://www.raspberrypi.org/documentation/>

1.2 Hinweis zur neumodernen „Genderei“ landauf, landab

Der Autor schlizbäda kann gar nicht oft genug betonen, dass ihm eine Gleichbehandlung **aller** Geschlechter (mittlerweile mehr als zwei) äußerst wichtig ist. Er verurteilt eine Diskriminierung von Menschen nur aufgrund ihres Geschlechtes oder anderer Nebensächlichkeiten wie ihrer Herkunft etc. aufs Schärfste!

Dennoch – oder gerade deswegen – lehnt er die derzeit grassierende Unart des sogenannten „Genderns“ mit sprachlichen Auswüchsen wie „AnwenderInnen“ oder gar „Benutzer*innen“ (jetzt neu mit Stern!) zugunsten einer klaren und verständlichen Ausdrucksweise ab. In diesem Trend sind selbst Wörter wie „Abiturienten“ verpönt: Soll man hier wirklich gender-gerecht „Abiturierende“ schreiben? ☺

1.3 Danksagung

schlizbäda möchte folgenden Personen und Einrichtungen danken:

MiczFlor, dem Schöpfer der Phoniebox:

Homepage: <http://phoniebox.de/>, Softwaredownload: <https://github.com/MiczFlor>

dem **deutschen Raspberry Pi Forum** (<https://forum-raspberrypi.de>) für die unzähligen Beiträge zum Thema Phoniebox

sowie folgenden Mitgliedern des deutschen Raspberry Pi Forums:

@splitti79 (<https://forum-raspberrypi.de/user/54710-splitti79/>),

der auch einen eigenen Internetblog zum Thema betreibt:

<https://splittscheid.de/selfmade-phoniebox/>

<https://splittscheid.de/phoniebox-2/>

@hailogugo (<https://forum-raspberrypi.de/user/51824-hailogugo/>)

1.4 Kurzbeschreibung

Die Phoniebox ist ein intuitiv über RFID-Karten zu bedienender Audioplayer für Kinder. Als kommerzielle Pendants existieren die Toniebox (<https://tonies.de/>) oder der vom Konzept her bezüglich der Software freier konzipierte Hörbert (<https://www.hoerbert.com/>).

Von der Phoniebox existieren zwei Softwarevarianten: Die „einfache“ Variante, die nur auf Audiodateien lokal auf dem Raspberry Pi oder im freigegebenen Heimnetzwerk zugreifen kann. Sie verwendet den *Music Player Daemon (MPD)* von Linux:

<https://www.musicpd.org/>

Die andere Variante ermöglicht zudem den Zugriff auf *Spotify*. Sie basiert allerdings auf *Mo-pidy* anstelle von MPD. In dieser Dokumentation wird jedoch nicht auf die Spotify-Variante eingegangen.

Die Hardware einer Phoniebox besteht mindestens aus folgenden Komponenten:

- Raspberry Pi
- Audioverstärker und Lautsprecher
- RFID-Leser und einem Satz passender RFID-Karten
- *optional*: Taster für grundlegende Bedienung
- *optional*: USB-Powerbank als Stromversorgung
- *optional*: Ein-/Ausschalter
- *optional*: LAN- oder WLAN-Modul, falls nicht Bestandteil des verwendeten Raspberry Pi
- *optional*: Statusdisplay

1 Einleitung

Für eine minimalistische aber funktionale Phoniebox sind nur der Raspberry Pi als zentrale Einheit, der RFID-Leser für die Umsetzung des Bedienkonzeptes und ein kleiner Audioverstärker erforderlich. Da die Bedienung aller Playerfunktionen (play, pause, prev, next, volume) auch über speziell konfigurierte RFID-Karten erfolgen kann, sind zusätzliche Bedientaster nicht zwingend notwendig.

Ebenso muss die Box nicht unbedingt batteriebetrieben sein, so dass ein passendes Netzteil zur Spannungsversorgung des RPi ausreicht. Dann können auch die Powerbank und der Ein-/Ausschalter weggelassen werden.

Wenn die Phoniebox nur Audiodateien von der lokalen SD-Karte oder von einem USB-Speicherstick abspielen soll, kann auf ein LAN- oder WLAN-Modul verzichtet werden. Die Raspberry Pis der Bauform B haben ohnehin eine Ethernetbuchse und ab dem Raspberry Pi 3B ist ein WLAN-Modul verbaut.

Grundsätzlich optional ist ein Statusdisplay. Es erleichtert aber die Erkennung des Betriebszustandes enorm.

Im deutschen Raspberry Pi Forum gibt es einige Themen zur Phoniebox, die dort ursprünglich als *jukebox4kids* oder auch als *RPi-Jukebox-RFID* bezeichnet wird. Der Begriff *Phoniebox* wurde erst später geprägt.

<https://forum-raspberrypi.de/forum/board/152-musikboxen-webradios-musikplayer/> – Phoniebox-Unterforum

<https://forum-raspberrypi.de/forum/thread/13144-projekt-jukebox4kids-jukebox-fuer-kinder/> – Monsterthread!

<https://forum-raspberrypi.de/forum/thread/45571-splittis-phoniebox-1-0-aka-ghostbox-aka-die-drei-kids-box/>

2 Technischer Aufbau von schlizbädas Phoniebox

Die Phoniebox ist ein open-source-Projekt. Deshalb unterscheidet sich beinahe jede gebaute Phoniebox von allen anderen. Dies gilt nicht nur hinsichtlich der Gestaltung des Gehäuses, sondern auch für die technische Umsetzung. Impressionen von den Phonieboxen der letzten Jahre gibt es auf den Kalendern von *MiczFlor* für 2019 und 2020:

<https://github.com/MiczFlor/RPi-Jukebox-RFID#phoniebox-the-rpi-jukebox-rfid>

In diesem Kapitel wird der hard- und softwareseitige Aufbau des konkreten Exemplars von schlizbäda beschrieben. Es werden auch Hinweise gegeben, warum eine bestimmte Umsetzung gewählt wurde. Dabei muss es sich nicht unbedingt um die optimale Lösung handeln!

2.1 Verwendete Komponenten / Stückliste

Anz.	Artikel	Bestellnummer	Lieferant	Einzel-preis
1x	Holzkiste 33cm*24cm	VARIERA 21386	IKEA	14,99
1x	Sperrholzplatte Kiefer 33cm*24cm	Holzzuschnitt	OBI	2,40
1x	Raspberry Pi 3B	RASPBERRY PI 3	reichelt	35,80
1x	SD-Karte 32GB: Sandisk Extreme A1 32GB	SDSQXAF032GGN6MA	reichelt	11,50
1x	HifiBerry MiniAmp	RPI HB MINI AMP	reichelt	19,50
2x	Lautsprecher Visaton F 8SC-8	VIS F 8SC-8	reichelt	10,25
2x	Schutzwand Visaton 8 ES	VIS 4634	reichelt	6,60
1x	LED-beleuchteter Taster, rot	RAFI 1.15.106.501/1300	reichelt	6,50
1x	LED-beleuchteter Taster, grün	RAFI 1.15.106.502/1500	reichelt	9,10
2x	LED-beleuchteter Taster, gelb	RAFI 1.15.106.503/1400	reichelt	7,99
2x	LED-beleuchteter Taster, blau	RAFI 1.15.106.504/1600	reichelt	10,80
1x	USB RFID-Reader 125kHz	Neuftech	amazon	10,89
100x	RFID-Karte 125kHz	–	amazon	21,28
1x	26000mAh Powerbank 4,8A	EasyAcc	amazon	48,99
1x	Pimoroni OnOff SHIM	Pimoroni	amazon	9,50
1x	1,3inch OLED I2C-Display 128x64 Pixel	AZDelivery	amazon	8,49

Tabelle 2.1: Verwendete Komponenten bei schlizbädas Phoniebox

Bei der Auswahl der Komponenten spielte ein möglichst niedriger Preis eine eher untergeordnete Rolle. Vielmehr wurden bereits vorhandene Teile verwendet. Bei Neubeschaffungen wurde auf Artikel gesetzt, mit denen bereits gute Erfahrungen gemacht wurden.

2.1.1 Gehäuse

Die IKEA-Holzkiste war eigentlich für ein anderes Projekt gedacht. Sie ist zwar formschön, aber die schrägen Wände ohne rechte Winkel machen den Einbau nicht einfacher! Gott sei Dank ist das Ding nun verbaut!

<https://www.ikea.com/de/de/p/variera-kasten-mit-griff-bambus-70226053/>

2.1.2 Raspberry Pi 3B

Ein Raspberry Pi 3B ist für diese Anwendung performant genug und er bietet eine ausreichende *Schwuppdigität*. Außerdem sind darauf bereits onboard ein WLAN- und ein Bluetooth-Modul verbaut. Ein Raspberry Pi 4 wäre zwar noch etwas schneller, verbraucht aber mehr Strom und entwickelt deshalb mehr Wärme. Neben anderer daraus resultierender Nachteile und der erforderlichen Gegenmaßnahmen reduziert sich dadurch auch die Akkulaufzeit.

<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

2.1.3 HifiBerry MiniAmp

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

Ein kleiner Class-D-Verstärker mit 2x3W Maximalleistung an 4Ω-Lautsprechern, der speziell für den Raspberry Pi entwickelt wurde. In Kombination mit den Visaton-Lautsprechern wird damit ein durchaus passabler Klang erzeugt, der für kleinere Kinder mehr als ausreichend, aber noch nicht zu laut ist.

In manchen Anleitungen wird die Verwendung von kleinen PC-Aktivboxen empfohlen und bisweilen auch noch, als Audioquelle dafür die originale vierpolige Klinkenbuchse des Raspberry Pi zu verwenden. Dies hat in meinen Augen zwei gravierende Nachteile:

- Die Audioqualität an der originalen Klinkenbuchse des Raspberry Pi ist eher schlecht, da die Audiosignale über eine windig umgesetzte Pulsweitenmodulation erzeugt werden.
- Die in diversen Foren und Tutorials empfohlenen Trust-Leto Aktivboxen sind billigste Chinaware.

Im Raspberry Pi Forum habe ich dies auch schon mit drastischen Worten entsprechend kundgetan: <https://forum-raspberrypi.de/forum/thread/13144-projekt-jukebox4kids-jukebox-fuer-kinder/?postID=384062#post384062>

2.1.4 Bedientaster

Welche Taster hierfür verwendet werden, ist reine Geschmacksache. Da ich derjenige bin, der bei vermeintlichen Schnäppchen immer die fehlerhaften bzw. ganz kaputten Teile erwischt, konnte ich mir die Arcade-Buttons vom freundlich grinsenden Chinesen verkneifen und griff zu den Tastern eines namhaften Herstellers (Rafi). **Ein echter Vorteil dieser Taster ist, dass die verbauten LEDs so hell sind, dass bereits 1-2mA für ausreichende Leuchtintensität sorgen und somit nur mit einem Vorwiderstand direkt an den GPIOs des Raspberry Pi angeschlossen werden können.**

<https://components.rafi.de/DE/DE/produktstruktur/taster/sonstige/leuchtdrucktaster/leuchtdrucktaster-9-1-mm/leuchtdrucktaster-9-1-mm-mit-eingebauter-led>

2.1.5 Neuftech USB-RFID-Reader

Mangels jeglichen Vorwissens auf diesem Gebiet beschloss ich, mich an die Empfehlungen der diversen Tutorials zu halten. Es ist ein chinesisches Qualitätsprodukt, das tut, was es soll: Es liest die Karten problemlos durch die ca. 8mm starke Holzwand der IKEA-Kiste. Sowohl die beigefügte Dokumentation als auch die im Internet geht jedoch gegen Null!

<https://www.amazon.de/Neuftech-Reader-Kartenleseger%C3%A4t-Kartenleser-Kontaktlos/dp/B0180Y0R3E>

2.1.6 Stromversorgung

Die *EasyAcc*-Powerbank zeichnet sich dadurch aus, dass sie zum Aufladen jederzeit an die Stromversorgung angeschlossen werden kann, ohne den Raspberry Pi dabei auch nur kurzzeitig stromlos zu schalten. Dadurch wird der Ärger durch unkontrolliert ausgeschaltete Raspberry Pis von vorne herein vermieden. Es mag günstigere Powerbänke geben, die dies auch können, aber der Forenkonsens geht eindeutig zu den *EasyAcc*-Produkten.

<https://www.easyacc.com/de/670-easyacc-26000mah-power-bank.html>

Von *EasyAcc* gibt es eine deutlich günstigere Variante mit „nur“ 20000mAh Ladekapazität, die in den meisten Phonieboxen verwendet wird. Aber der schlizbäda gönnte sich getreu dem Motto „Klotzen, nicht kleckern!“ die größere Variante mit 26000mAh und auch die funktioniert problemlos.

Zum Ein- und Ausschalten der Phoniebox wird der *OnOff SHIM* von Pimoroni verwendet. Allerdings musste der *gepimpt* werden, siehe Abschnitt 2.2.2

<https://shop.pimoroni.com/search?type=product&q=onoff+shim>

2.1.7 OLED-Display

Hier wurde 1:1 die Erweiterung von **@splitti79** aus dem Raspberry Pi Forum nachgebaut:

<https://forum-raspberrypi.de/forum/thread/41465-oled-display-fuer-die-phoniebox/>

https://github.com/splitti/oled_phoniebox

Artikel bei amazon: <https://www.amazon.de/gp/product/B078J78R45>

2.2 Selbstgebaute Teile

Die Phoniebox, das Wochenend-Projekt – <http://phoniebox.de/>

Ganz so schnell ging's bei mir leider nicht. Auf dem Weg zu meiner Box gab es diverse Fallstricke, die ich im Raspberry Pi Forum im Thread

<https://forum-raspberrypi.de/forum/thread/45820-phoniebox-2-0-rc7-mpd-spielt-ueber-rfid-karte-gewahlte-musik-nicht-ab/>

schilderte. Dort bekam ich viele Hilfestellungen zur Lösung der einzelnen Punkte. Mit dem heutigen Wissen würde ich einem Phoniebox-Neuling folgende Tipps mitgeben:

<https://forum-raspberrypi.de/forum/thread/46231-hilfe-erste-phoniebox-totaler-anfaenger/?postID=415862#post415862>

Für eine gut funktionierende Phoniebox ist eine feste, stabile und wirklich haltende Verkabelung unabdingbar! Alle Verbindungen, die beim fliegenden Testaufbau auf einem Steckbrett mit Jumperkabeln (sogenannten DuPont-Kabeln) und Ähnlichem aufgebaut wurden, müssen beim Einbau in die Box sauber verlötet oder gecrimpt werden. Es lohnt sich auch, gute Steckverbinder einzubauen, um die Box zu Wartungszwecken später wieder leichter zerlegen zu können. Selbst die von mir verwendeten Stift- und Buchsenleisten im 2,54mm-Raster sind eigentlich noch nicht zuverlässig genug und insbesondere bei Einzelbuchsen zu wackelig.

Die gesamte Hardware an der GPIO-Leiste des Raspberry Pi wurde gestapelt: Direkt auf den RPi wird der HifiBerry MiniAmp gesteckt. Da der MiniAmp oben keine Stifitleiste hat, auf der weitere RPi-HATs gesteckt werden könnten, habe ich selbst auf der Oberseite eine 40-polige Stifitleiste aufgelötet. Auf dieser Stifitleiste habe ich gleich den Pimoroni OnOff SHIM verlötet, siehe Abbildung 2.1.

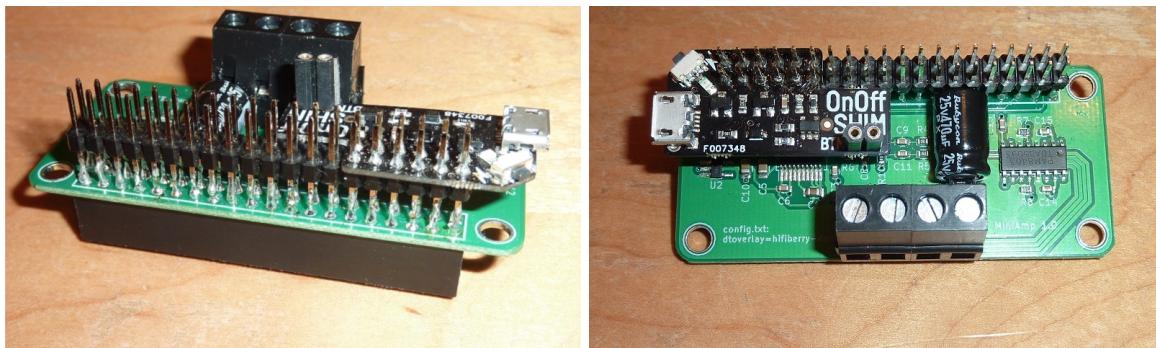


Abbildung 2.1: HifiBerry MiniAmp mit aufgelötetem Pimoroni OnOff SHIM

2 Technischer Aufbau von schlizbädas Phoniebox

Ganz oben auf diesen *Turm* wird schließlich eine Lochrasterplatine gesteckt, die die Anschlüsse für die Bedientaster, die LEDs und das OLED-Display enthält, siehe Abbildung 2.2.

Ich habe die Lochrasterplatine deshalb ganz oben aufgesteckt, dass die Bauteile auf der Platine freier platziert werden können und die Stifteleisten beim Anstecken der Taster und LEDs besser zugänglich sind.

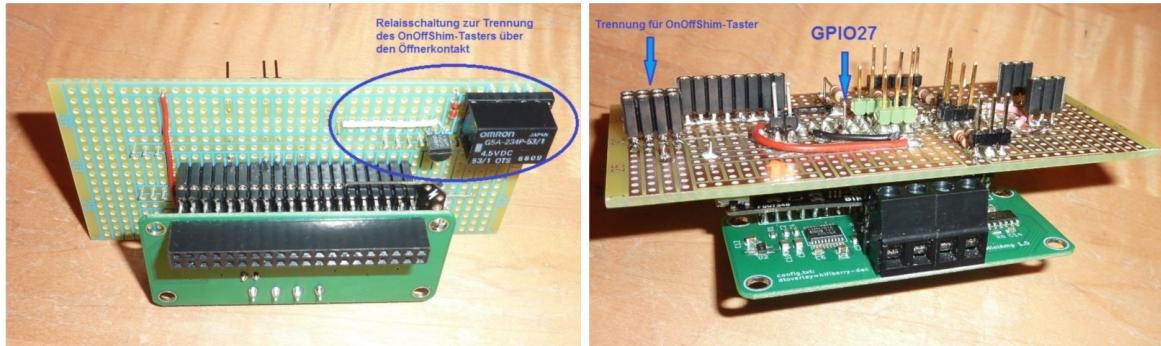


Abbildung 2.2: GPIO-Aufbau mit Lochrasterplatine

2.2.1 Lochrasterplatine

Vor dem Aufbau der Lochrasterplatine wurde die Verwendung der GPIO-Pins überprüft und unter Berücksichtigung von Abbildung 2.3 für schlizbädas Phoniebox gemäß Tabelle 2.2 festgelegt.

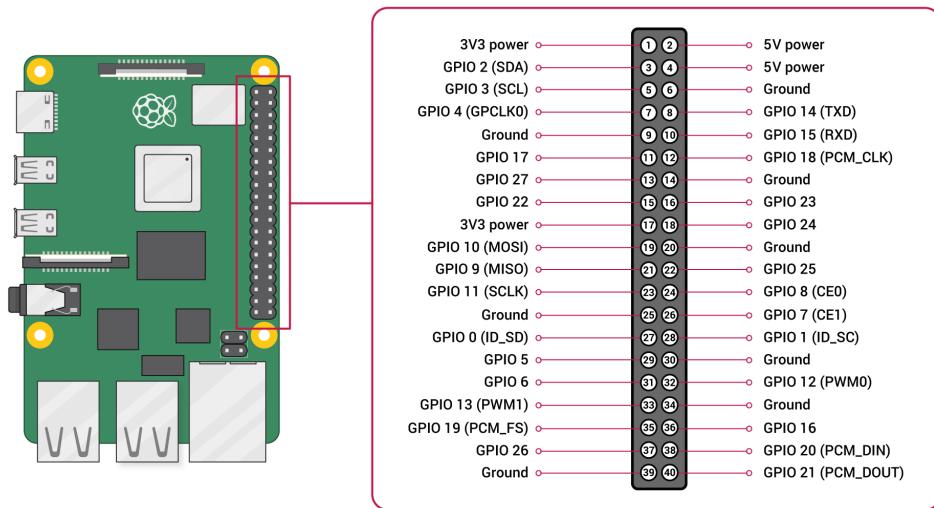


Abbildung 2.3: GPIO-Belegung – © Raspberry Pi Foundation (CC BY-SA 4.0)

Verwendung	Funktion	Pin		Pin	Funktion	Verwendung
	3V3	1		2	5V	
OLED-Display: SDA	GPIO2 (SDA)	3		4	5V	
OLED-Display: SCL	GPIO3 (SCL)	5		6	GND	
OnOff SHIM: W 0=ausschalten	GPIO4	7		8	GPIO14	–
	GND	9		10	GPIO15	–
OnOff SHIM: R Status Ein-/Aus-Taster W Anzeige Power-LED	GPIO17	11		12	GPIO18	MiniAmp: PCM CLK I2S BCLK (bit clock)
OnOff SHIM: W 1=Relais ansteuern	GPIO27	13		14	GND	
–	GPIO22	15		16	GPIO23	–
	3V3	17		18	GPIO24	LED vol. down
LED prev	GPIO10	19		20	GND	
LED next	GPIO9	21		22	GPIO25	LED vol. up
LED play/pause	GPIO11	23		24	GPIO8	Taster prev
	GND	25		26	GPIO7	Taster next
–	ID SD	27		28	ID SC	–
Taster play/pause	GPIO5	29		30	GND	
Taster vol. down	GPIO6	31		32	GPIO12	Taster vol. up
Taster vol. mute	GPIO13	33		34	GND	
MiniAmp: PCM FS I2S SYNC (left/right clk)	GPIO19	35		36	GPIO16	MiniAmp: PAM8403 W 0=MUTE
MiniAmp: PAM8403 W 0=SHDN (shutdown)	GPIO26	37		38	GPIO20	MiniAmp: PCM DIN I2S data in (<i>not used</i>)
	GND	39		40	GPIO21	MiniAmp: PCM DOUT I2S data out

Tabelle 2.2: Belegung der GPIO-Leiste bei schlzbädas Phoniebox

Auf der Lochrasterplatine befinden sich folgende Komponenten:

- GPIO-Pins als Eingänge für die Taster mit PullUp-Widerständen
- GPIO-Pins als Ausgänge zur direkten Ansteuerung von LowCurrent-LEDs $\leq 3\text{mA}$
- I2C-Bus für die Ansteuerung des OLED-Displays
- Relaiserweiterung für den OnOff SHIM

Um die Bedientaster zuverlässig einzulesen, empfehle ich die Verwendung von **externen** PullUp-Widerständen mit $10\text{k}\Omega$. Es ist zwar möglich, die internen PullUp-Widerstände des SoC BCM2837 zu aktivieren, doch bei so mancher Phoniebox war dies dann letztlich der

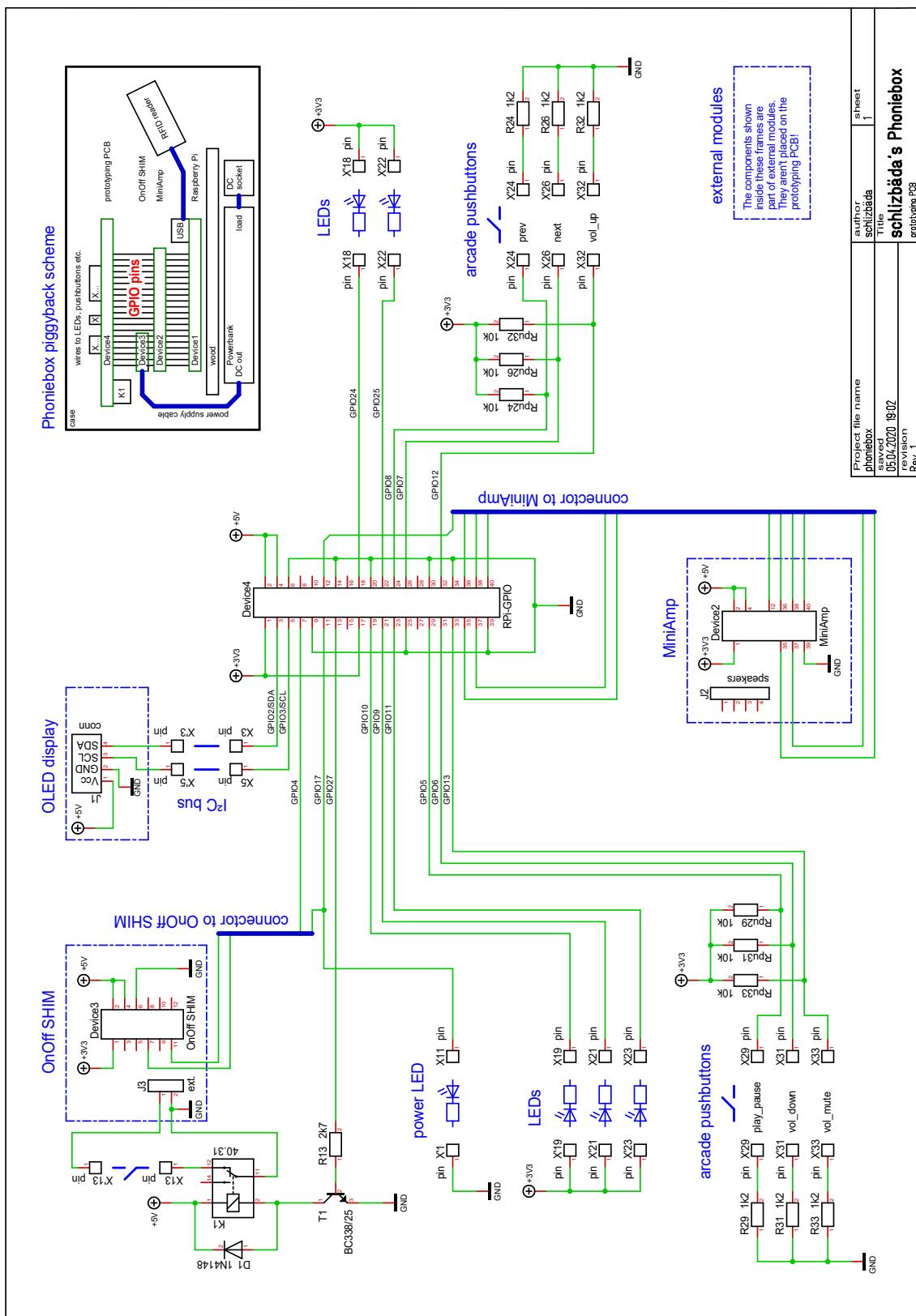


Abbildung 2.4: Schaltung auf der Lochrasterplatine

2 Technischer Aufbau von schlzbädas Phoniebox

Grund, dass die Taster nicht wie erwartet funktioniert hatten.

Um den Pegel bei gedrücktem Taster auf *low* zu ziehen, aber den Stromfluss zu begrenzen, werden mit GND verbundene $1,2\text{k}\Omega$ -Widerstände zugeschaltet.

Einfacher ist die Ausführung der GPIO-Pins, an denen die LEDs der Taster angeschlossen werden. Da in den verwendeten Tastern LowCurrent-LEDs eingebaut sind, die bereits unter 3mA Stromfluss ausreichend hell leuchten, sind keine Transistortreiberschaltungen notwendig. Der bei LEDs obligatorische Vorwiderstand wurde für einen $3,3\text{V}$ -Betrieb ausgerechnet und direkt an die LEDs angelötet. Eine visuelle Kontrolle zeigte, dass für die gleiche Helligkeit bei den verschiedenen Farben unterschiedliche Widerstandswerte erforderlich sind:

LED-Farbe Vorwiderstand

rot	$1,2\text{k}\Omega$
gelb	$1,2\text{k}\Omega$
grün	$4,7\text{k}\Omega$
blau	$1,5\text{k}\Omega$
weiß	$1,5\text{k}\Omega$

Auf der Lochrasterplatine befinden sich daher nur die Steckverbinder für den Anschluss der Taster-LEDs.

Das OLED-Display wird über den I2C-Bus des Raspberry Pi angesteuert, der an den GPIO-Pins 3 und 5 der Stiftleiste aufgelegt ist. Da dieses Modul bereits $10\text{k}\Omega$ -PullUp-Widerstände enthält, wurden auf der Lochrasterplatine keine zusätzlichen PullUps verbaut.

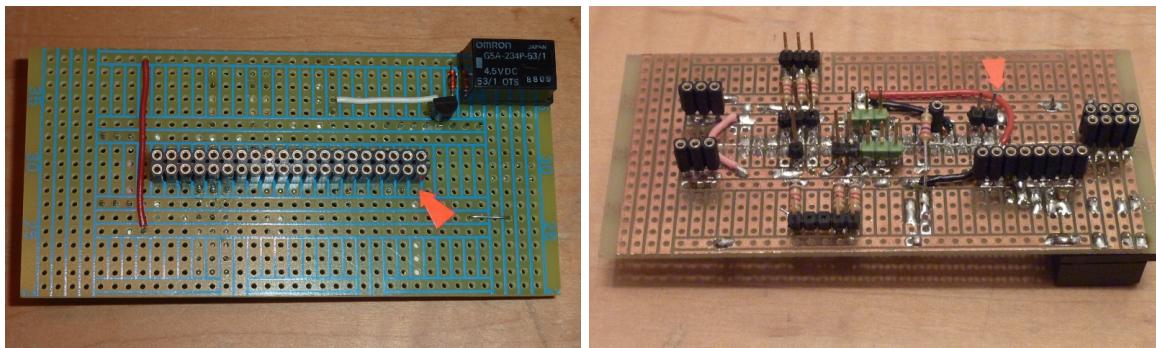


Abbildung 2.5: Ansichten der Lochrasterplatine

2.2.2 Relais für OnOff SHIM

Zum Ein- und Ausschalten der Phoniebox wird der OnOff SHIM von Pimoroni verwendet. Beim Testen stellte sich heraus, dass, wenn der Ein-/Aus-Taster des OnOff SHIMs (oder ein dazu parallel angeschlossener Taster) bis über den Abschaltzeitpunkt hinaus gedrückt gehalten wird, der Raspberry Pi zwar heruntergefahren, aber die Versorgungsspannung nicht getrennt wird! Die rote Power-LED des Raspberry Pi leuchtet weiter und es fließt ein deutslicher Strom von ca. $80\text{-}90\text{mA}$, der die Powerbank *leersäuft*.

softwareseitig
Festlegung
der
GPIOs

softwareseitig
Festlegung
der
GPIOs

2 Technischer Aufbau von schlizbädas Phoniebox

Ich weiß nicht, ob es sich hierbei um ein generelles Problem beim OnOff SHIM handelt, oder ob das bloß wieder eine Macke meines Exemplars ist. Aber da gerade Kinder kein Gefühl dafür haben, wie lange sie die Taste drücken dürfen/sollen, musste eine vernünftige Lösung her!

Letztlich konnte dieses Problem wie in Abbildung 2.6 durch den Einbau eines als Öffner geschalteten Relaiskontakte in Reihe zum Ein-/Aus-Taster des OnOff SHIM gelöst werden. Beim Herunterfahren wird dabei über ein Softwareskript das Relais angesteuert, so dass ein evtl. immer noch geschlossener Taster vom OnOff SHIM getrennt wird. Weil *nach* dem Abschalten auch das Relais wieder abfällt, führt betätigter Taster somit zu einem erneuten Hochfahren der Phoniebox. Der Ein-/Aus-Taster hat jetzt drei Funktionalitäten:

- im ausgeschalteten Zustand kurz betätigen: Phoniebox fährt hoch
- etwas länger betätigen zum Herunterfahren der Phoniebox
- richtig lange betätigen für einen Reboot (Neustart) der Phoniebox

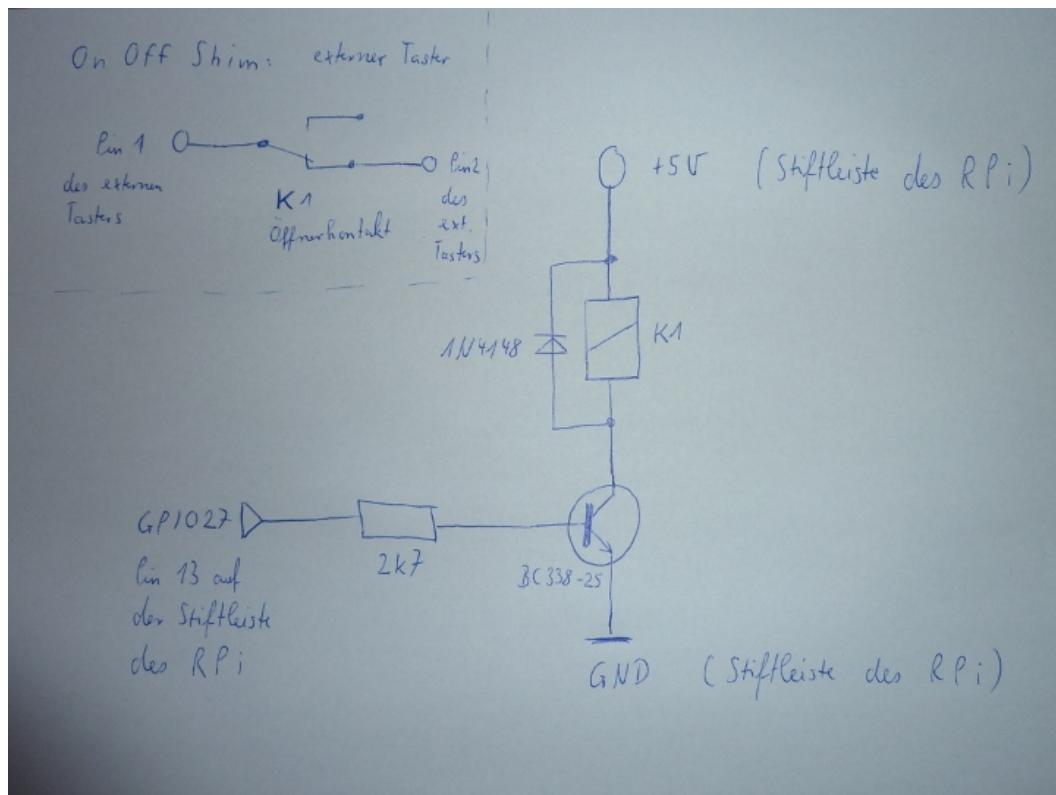


Abbildung 2.6: Konzept zur gesteuerten Trennung des Tasters vom OnOff SHIM

Dieser Schaltungsteil wurde nachträglich auf der Lochrasterplatine ergänzt.

Software dafür in Kapitel 3 beschrieben

2.2.3 Beschreibung des „Turms“

Das Erweiterungskonzept beim Raspberry Pi besteht darin, eine zusätzliche Platine *huckepack* als sogenannte piggyback-Platine oder im Slang der Raspberry Pi Foundation als Pi-HAT („hardware attached on top“) auf die 40-polige GPIO-Leiste aufzustecken. Bei mehreren Erweiterungsplatinen entsteht so schnell ein ganzer Turm von Zusatzplatinen 😊

Auch bei der Phoniebox sind mehrere Aufsatzplatten im Einsatz, wie in Abbildung 2.7 zu bewundern ist.

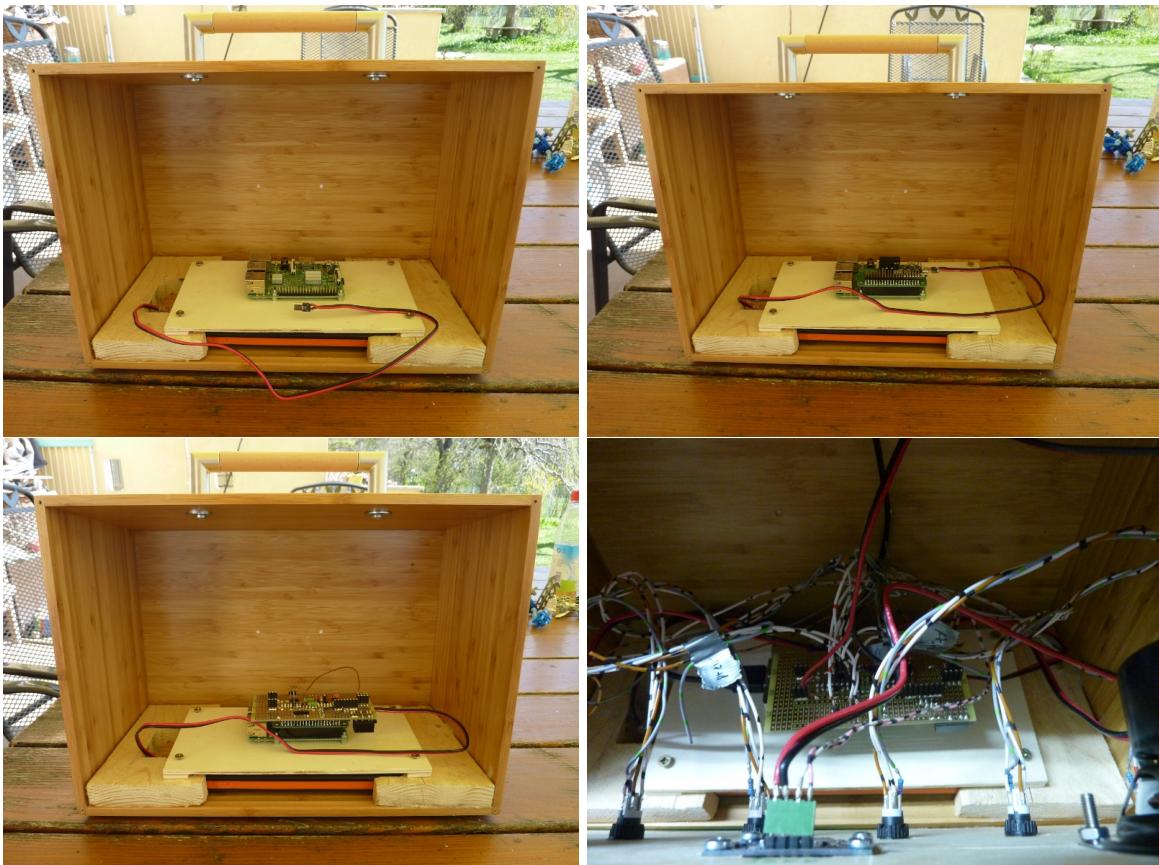


Abbildung 2.7: Der Raspberry Pi-Turm der Phoniebox in seiner vollen Pracht

Ganz unten in der Box befindet sich alleine schon aus Gründen der Schwerpunktverlagerung die Powerbank, die über ein Holzgestell fixiert wird. Darüber ist auf einer dünnen Sperrholzplatte der Raspberry Pi montiert, auf den der Hifiberry MiniAmp aufgesteckt wurde. Auf den MiniAmp wurde eine 40-polige Stiftleiste aufgelötet, auf der auch der OnOff SHIM verlötet wurde. Um die Verkabelung beim Zusammenbau etwas leichter durchführen zu können, wurde die selbstgebaute Lochrasterplatine ganz bewusst oben aufgesetzt. Der Drahtverhau im Inneren der Box vom Turm zu den Tastern, zum OLED-Display und zu den Lautsprechern ist immer noch groß genug!

Schematisch ist es nochmals etwas übersichtlicher in Abbildung 2.8 dargestellt.

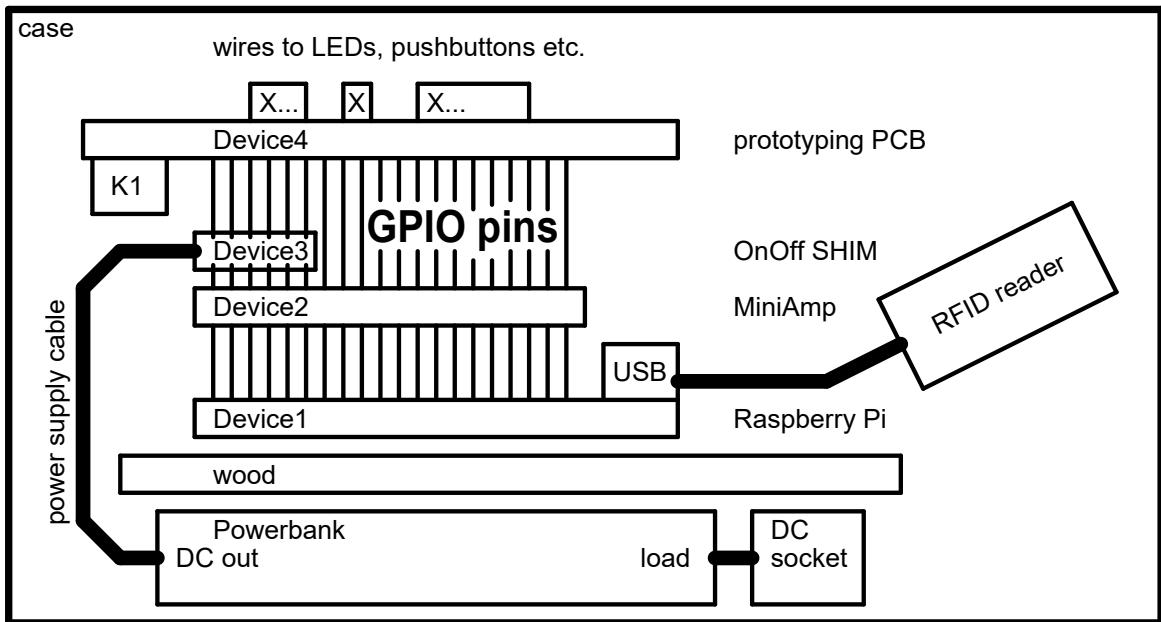


Abbildung 2.8: Der Raspberry Pi-Turm schematisch dargestellt

2.2.4 Verwendete Anschlussbuchsen

Bei der Stromversorgung wurde darauf geachtet, die auftretenden Stromstärken verlustfrei zu übertragen. So wurden nur Kupferlitzen mit einem Querschnitt $\geq 0,25\text{mm}^2$ verwendet, um den Spannungsabfall niedrig genug zu halten. Dieser Querschnitt ist bei mehradrigen Leitungen laut https://www.lappkabel.de/fileadmin/catalog/2010_de_pdf/T12_Belastbarkeit_Grundtabelle_Reduktionstabellen.pdf für bis zu 4A ausgelegt.

Neben ausreichend dicken Kabelquerschnitten müssen auch die verwendeten Steckverbinder dafür ausgelegt werden. In die Gehäuserückwand der Phoniebox wurde als Ladebuchse für die Powerbank das Produkt NEB-1 (lumberg) eingebaut. Diese Buchse ist für Stromstärken bis 3A ausgelegt. Da die Powerbank bei Anschluss an eine Micro-USB-Ladebuchse bis zu 2A zieht und bei Anschluss an beiden USB-Buchsen bis zu 4A, darf hier nur eine USB-Buchse verwendet werden!

Das Gegenstück, der Hohlstecker NES-1 GR (lumberg) wurde an einem 5V-Steckernetzteil verbaut.

Für die USB-Anschlüsse wurden die lötzbaren USB-Stecker BKL-10120098 (USB-A) und BKL-10120262 (micro USB-B) verwendet. Siehe Tabelle 2.3.

Anz.	Abb.	Artikel	Bestellnummer	Lieferant	Einzelpreis
1x		Hohlstecker	LUM NES1 GR	reichelt	0,83
1x		Einbaukupplung	LUM NEB 1	reichelt	1,20
2x		BKL 10120262	USB MICRO ST	reichelt	0,85
1x		BKL 10120100	BKL 10120100	reichelt	0,79
1x		BKL 10120098	BKL 10120098	reichelt	1,40

Tabelle 2.3: Verwendete Anschlussbuchsen bei schlizbädas Phoniebox

Hinweis

Der Einbau der Lumberg-Buchse NEB-1 war in der 8mm starken Rückwand des verwendeten Holzgehäuses etwas diffizil: Um die Buchse mit ihrer seitlich angebrachten Kontaktfeder überhaupt durch die Holzplatte zu bringen, ohne das Loch so groß zu bohren, dass es durch den Befestigungsflansch nicht mehr überdeckt wurde, musste am Loch in der Gehäuseinnenseite mehrmals vorsichtig *herumgeknabbert* werden! Diese Buchse ist wohl eher nur für dünne Metallwände ($\leq 2\text{mm}$) geeignet.

2.3 Holzarbeiten

Zu den Holzarbeiten möchte ich mich als Laie auf diesem Gebiet gar nicht zu weit aus dem Fenster lehnen. Das können andere besser, siehe hier:

<https://github.com/MiczFlor/RPi-Jukebox-RFID/blob/develop/docs/2020-Phoniebox-Calendar.jpg>

Aber auf einige Punkte möchte ich dennoch hinweisen:

2.3.1 Wahl des Gehäuses

Ich habe ein Gehäuse verwendet, das sich durch schräg verbaute Wände und dadurch wenige rechte Winkel auszeichnet. Für echte Holzwürmer kein Problem, aber mir machte es den Einbau nicht leichter. Nächstes Mal werde ich die typischen Baumarkt-Kolzkisten verwenden. Da sind die Außenwände dicker und der Einbau ist entsprechend einfacher 😊

Umso mehr freut es mich, dass ich das Gehäuse nicht verpfuscht habe (yeah!)

2.3.2 Einbau des RFID-Readers

In meiner Faulheit beschloss ich zunächst, den RFID-Reader mit einem doppelseitigen Klebeband (eigentlich ein gutes 3M-Band, aber mittlerweile auch ca. 10 Jahre alt) innen an der Oberseite der Box festzukleben. Eines Tages reagierte die Box nicht mehr auf aufgelegte RFID-Karten und nicht einmal das Pieperäusch des Readers ertönte mehr. Frontplatte runter und was sah ich? Der Reader lag mitten in der Box...

Auch zusätzliche Unterstützung durch sogenanntes *Panzertape* half nichts, so dass ich den Reader letztlich mit vier Holzschrauben vernünftig montierte: Dazu wurden die kleinen Gehäuseschrauben unter den Gummifüßen entfernt, die Schraublöcher im Gehäuse durchgebohrt und vergrößert und der Reader mit vier Holzschrauben innen stabil an der Oberseite der Phoniebox verschraubt.



Abbildung 2.9: RFID-Reader vernünftig montiert

2.3.3 Sichtöffnung für Ladeanzeige der Powerbank



Abbildung 2.10: Ladeanzeige der Powerbank an der Unterseite der Phoniebox

Die EasyAcc-Powerbank ist mit vier kleinen LEDs ausgestattet, die über die Anzahl der leuchtenden LEDs den prozentualen Ladezustand der Powerbank anzeigen. Wenn das Netzteil angesteckt ist und die Powerbank geladen wird, blinkt die höchstwertige LED.

Der Einfachheit halber habe ich die Powerbank im Gehäuse mit der Ladeanzeige nach unten montiert und im Gehäuse an der entsprechenden Stelle ein kleines Sichtfenster herausgesägt, siehe Abbildung 2.10.

2.3.4 Sichtscheibe für OLED-Display

Auch in schlizbädas Phoniebox wurde ein kleines 1,3-Zoll OLED-Display eingebaut. Dies ist gerade zu Beginn u. a. für eine schnelle Fehlerdiagnose äußerst sinnvoll. Allerdings sollte man vor das Display eine kleine Plexiglasscheibe oder Ähnliches einsetzen, um das Display vor groben Kinderfingern zu schützen.



Abbildung 2.11: OLED-Display für Statusanzeige in der Frontplatte der Phoniebox

2.3.5 Halterung für die Powerbank

Die Powerbank wird über verleimte Holzhalterungen im Gehäuse fixiert. Da die *vorhandene* Holzplatte für die Halterungen etwas dünner war als die Powerbank selbst, wurde die Breite mit zwei Sperrholzstreifen „aufgedoppelt“. Damit die Powerbank fest eingepresst wird, wurde ein ESD-Schaumstoff dazwischen gelegt. Mit einer Sperrholzplatte wird die Powerbank gehalten. Später wird darauf der Raspberry Pi montiert.

2 Technischer Aufbau von schlzbädas Phoniebox



Abbildung 2.12: Montage der Powerbank

3 Installation der Software

Eigentlich ist die Installation der Software *babyleicht* und man muss (Achtung: Unwort!) *nur* den **one-line-installer** von *MiczFlor* auf einem jungfräulichen **Raspbian** starten...

<https://github.com/MiczFlor/RPi-Jukebox-RFID#installation>

Aber:

Aufgrund vieler Anfragen der mittlerweile relativ großen Community rund um die Phoniebox und der daraus resultierenden zahlreichen *Push Requests* auf github und auch wegen vieler anderer guter Gründe schleichen sich in die an sich großartige Arbeit von *MiczFlor* naturbedingt immer wieder kleinere Fehler ein, die insbesondere für unbedarfe Neulinge bisweilen schwer zu finden und zu beseitigen sind.

3.1 Raspbian Buster Lite auf dem Raspberry Pi installieren

Einige Tutorials im Internet empfehlen zwar, die Software für die Phoniebox unter **Raspbian Buster with desktop** zu installieren, aber da es sich bei der Phoniebox um ein einfaches standalone-System ohne Bildschirm handelt, ist es in meinen Augen völlig ausreichend, die Installation auf dem wesentlich kleineren **Raspbian lite** vorzunehmen. Ein X11-Desktop ist schlichtweg nicht erforderlich. Außerdem reicht für **Raspbian Buster Lite** eine SD-Karte mit einer Speicherkapazität von 4GB.

3.1.1 Erstellen einer SD-Karte für den Raspberry Pi mit **Raspbian Buster Lite**

Zunächst wird die aktuelle Version von **Raspbian Buster Lite** von der Homepage der Raspberry Pi Foundation (<https://www.raspberrypi.org/downloads/raspbian/>, ca. 450MiB) heruntergeladen und entpackt (1,8GiB). Das entpackte Image wird mit einem dafür vorgesehenen Programm wie **win32diskimager** oder am Linux-PC mit dem systemeigenen Kommando dd auf die SD-Karte *geflasht*, um das Image 1:1 zu übertragen.

Achtung

Es reicht nicht, die Imagedatei einfach auf eine bereits formatierte SD-Karte zu kopieren!

3 Installation der Software

Die Details des *Flashens* von Betriebssystem-Images auf eine SD-Karte werden von der Raspberry Pi Foundation unter <https://www.raspberrypi.org/documentation/installation/installing-images/README.md> ausführlich beschrieben und in diesem Dokument als bekannt vorausgesetzt.

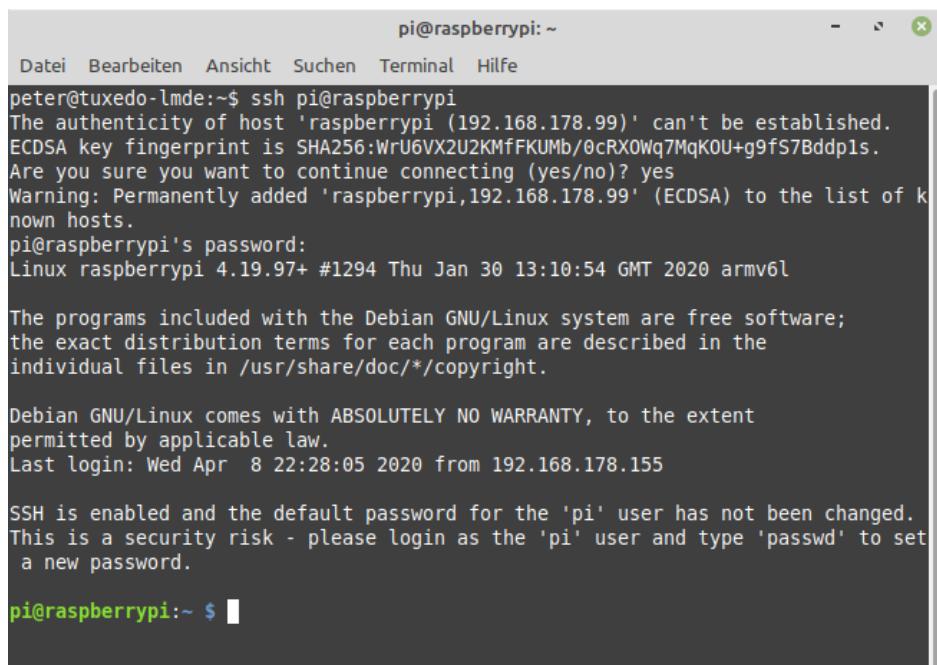
Nach dem Flashen enthält die SD-Karte zwei Partitionen:

- die FAT32-formatierte */boot*-Partition
- die ext4-formatierte Rootpartition von **Raspbian**

Hinweis

Beim (erneuten) Anstecken der SD-Karte am PC werden diese beiden Partitionen im Dateimanager angezeigt. Sollte auf dem PC jedoch das Betriebssystem **Windows** verwendet werden, so wird nur die FAT32-formatierte */boot*-Partition erkannt.

3.1.2 Anmeldung am Raspberry Pi über *ssh*



```
pi@raspberrypi: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
peter@tuxedo-lmde:~$ ssh pi@raspberrypi
The authenticity of host 'raspberrypi (192.168.178.99)' can't be established.
ECDSA key fingerprint is SHA256:WrU6VX2U2KMffKUMb/0cRXOWq7MqKOU+g9fS7Bddp1s.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'raspberrypi,192.168.178.99' (ECDSA) to the list of known hosts.
pi@raspberrypi's password:
Linux raspberrypi 4.19.97+ #1294 Thu Jan 30 13:10:54 GMT 2020 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr  8 22:28:05 2020 from 192.168.178.155

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $
```

Abbildung 3.1: erster Login über *ssh*

Da an den Raspberry Pi für die Phoniebox weder Bildschirm noch Tastatur angeschlossen werden, muss der Zugriff auf den Raspberry Pi von Anfang an über *ssh* erfolgen. Dazu muss auf der Partition */boot* der SD-Karte die leere Datei *ssh* angelegt werden.

Nun wird die SD-Karte in den Raspberry Pi gesteckt, der Raspberry Pi am besten über Ethernet ans LAN gehängt und eingeschaltet. Nach spätestens einer Minute sollte **Raspbian Buster Lite** vollständig gebootet und der Raspberry Pi im LAN bekannt sein. Auf dem PC

3 Installation der Software

wird in einem Terminalfenster (unter **Windows** in der Eingabeaufforderung cmd.exe) die Software **ssh** mit folgendem Kommando gestartet:

```
PC $ ssh pi@raspberrypi #Das Passwort lautet raspberry
```

Die Bildschirmausgabe sieht in etwa wie in Abbildung 3.1 aus.

3.1.3 Raspbian konfigurieren

Nach erfolgreichem **ssh**-Login muss das System jetzt auf einen aktuellen und sicheren Stand gebracht werden:

```
pi $ sudo apt update && sudo apt upgrade #System auf den neuesten Stand bringen
pi $ sudo raspi-config
1 Change User Password                      #Passwort unbedingt ändern!
2 Network Options      -> N1 Hostname        #z. B. phoniebox1
3 Boot Options         -> B1 Desktop / CLI      -> B2 Console Autologin
4 Localisation Options -> I2 Change Timezone   #Zeitzone anpassen
                           -> I4 Change Wi-fi Country    #diese Anpassung ist ganz wichtig!
5 Interfacing Options  -> P2 SSH
```

3.1.4 WLAN einrichten

Da die Phoniebox ein tragbares Gerät für's Kinderzimmer werden soll, muss eine WLAN-Verbindung eingerichtet werden:

```
pi $ sudo nano /boot/wpa_supplicant.conf
:
country=DE
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="linksys"
    psk=<network-password>
    key_mgmt=WPA-PSK
}
:

pi $ sudo systemctl restart dhcpcd
Warning: The unit file, source configuration file or drop-ins of dhcpcd.service changed
on disk. Run 'systemctl daemon-reload' to reload units.
pi $ sudo systemctl daemon-reload
```

3 Installation der Software

```

pi $ sudo reboot #Braucht's das?
pi $ ip addr
1: lo: <LOOPBACK UP LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: BROADCAST MULTICAST UP LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default
qlen 1000      link/ether b8:27:eb:70:91:db brd ff:ff:ff:ff:ff:ff
    inet 192.168.178.99/24 brd 192.168.178.255 scope global dynamic noprefixroute eth0
        valid_lft 859570sec preferred_lft 751570sec
    inet6 fe80::99ec:954c:4542:ed0/64 scope link
        valid_lft forever preferred_lft forever
3: wlan0: <BROADCAST MULTICAST UP LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default
qlen 1000
    link/ether b8:27:eb:63:3a:f2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.178.154/24 brd 192.168.178.255 scope global dynamic noprefixroute wlan0
        valid_lft 863877sec preferred_lft 755877sec
    inet6 fe80::3d3b:d9af:5ca6:e45e/64 scope link
        valid_lft forever preferred_lft forever

```

 **Hinweis**

- | Es wird **keine** statische IP-Adresse eingerichtet!

3.2 Installation der Phoniebox-Software

bla bla

3.3 Software / Installation

ReadMe einbinden...

3.3.1 offen

Reduzierung der Bootzeit
automatisch aus nach 15 min.