

BoomBox

ThirtySomething

15. Oktober 2019

# Inhaltsverzeichnis

<b>1 Motivation</b>	.	.	.	<b>3</b>
<b>2 Das Gerät</b>	.	.	.	<b>3</b>
2.1 Die Hardwarekomponente	.	.	.	3
2.2 Die Softwarekomponente	.	.	.	4
<b>3 Vorbereitungen</b>	.	.	.	<b>4</b>
3.1 Firmware des Raspberry Pi aktualisieren	.	.	.	4
3.2 SSD Montage	.	.	.	5
3.3 SSD Vorbereiten	.	.	.	5
3.4 Betanken der SSD	.	.	.	5
<b>4 Die Hardwarekomponente</b>	.	.	.	<b>6</b>
4.1 Das Display und der Raspberry Pi	.	.	.	6
4.2 Die Soundkarte	.	.	.	8
4.3 Die Konverterplatine	.	.	.	9
4.4 Adapterkabel	.	.	.	10
4.5 Das Ergebnis	.	.	.	11
<b>5 Volumio</b>	.	.	.	<b>13</b>
5.1 Erstinstallation	.	.	.	13
5.2 Die Plugins	.	.	.	14
5.3 Feintuning	.	.	.	16
5.3.1 Der Mauszeiger	.	.	.	16
5.3.2 Bildschirmschoner	.	.	.	17
5.3.3 Zugriff von Windows	.	.	.	18
<b>6 Mixxx</b>	.	.	.	<b>20</b>
6.1 Vorbereitungen	.	.	.	20
6.1.1 Portmedia	.	.	.	20
<b>Abbildungsverzeichnis</b>	.	.	.	<b>22</b>
<b>Stichwortverzeichnis</b>	.	.	.	<b>23</b>

## 1 Motivation

Es gibt für Parties viele Möglichkeiten, Musik abzuspielen. Eine der modernen Varianten davon ist Streaming über ein Smartphone. Allerdings setzt das ein funktionierendes WLAN bzw. Handyempfang voraus. Sofern es über das Telefonnetz gehen soll, ist natürlich eine entsprechende Datenverbindung bzw. das dazugehörige Datenvolumen Voraussetzung. Was macht man jedoch, wenn weder das eine noch das andere gegeben ist? Dann greift man zur [BoomBox](#). Das ist eine Art moderner Ghettoblaster. Das Gerät ist unabhängig von Smartphone und WLAN bzw. Handyempfang.

## 2 Das Gerät

Das Gerät [BoomBox](#) besteht aus zwei Komponenten, der Hardwarekomponente und der Softwarekomponente.

### 2.1 Die Hardwarekomponente

Das System soll zum einen kostengünstig, zum anderen aber auch nicht altbacken sein. Folgende Komponenten erfüllen diese Anforderungen:

- Einem [Raspberry Pi 3 B+](#).
- Einem [Pi-DigiAMP+](#).
- Einem [Raspberry Pi Touch Display](#).
- Einem [M.2 zu USB Adapter](#).
- Einer [250 GB M.2 SSD](#).
- Einem [Netzteil](#).
- Zwei [USB Kabel](#).
- Einer [MicroSD Karte](#).
- Einem [Step Down Converter](#).
- Einem [Adatper mit Terminalblock](#).
- Einem [Kabel mit Stecker](#).

Der [Raspberry Pi](#) ist ein günstiger Einplatinencomputer, der für dieses Projekt wie geschaffen ist. Die Soundkarte verbessert die Audioausgabe erheblich – der [Raspberry Pi](#) ist von Haus aus hier nicht so überzeugend. Damit eine Bedienung ohne zusätzliche Eingabegeräte funktioniert, wird das Original Touch Display verwendet. Auf Parties kann es ja ab und zu mal etwas wilder zugehen. Damit Erschütterungen keinen Einfluss auf die [BoomBox](#) haben, wird statt einer konventionellen Festplatte eine SSD als Massenspeicher eingesetzt. Diese wird über einen Adapter an das System angeschlossen.

## 2.2 Die Softwarekomponente

Die Software besteht im Wesentlichen aus nur einem Punkt: Der Musikdistribution [Volumio](#). Diese Distribution bringt von Haus aus bereits eine Unterstützung für obige Hardware mit. Natürlich ist etwas Feintuning notwendig, um die Bedienung und das Handling zu vereinfachen. Das wird im Kapitel [5.3](#) beschrieben. Allerdings – sofern man es etwas professioneller wünscht, wird auch die Installation von [Mixxx](#) kurz beschrieben.

# 3 Vorbereitungen

Bevor das Gesamtsystem an den Start gehen kann, sind noch ein paar Vorbereitungen zu treffen.

## 3.1 Firmware des [Raspberry Pi](#) aktualisieren

Den [Raspberry Pi](#) schliessen wir mit einem Netzwerkkabel an. Dann laden wir die aktuellste Version von [Raspbian](#) herunter. Das ist das normale Betriebssystem für den [Raspberry Pi](#). Wir schreiben das Image mit dem [Win32 Disk Imager](#) auf eine MicroSD Karte. Um den Zugriff via SSH zu ermöglichen, erzeugen wir in der Boot Partition noch die leere Datei *ssh*. Wir booten den [Raspberry Pi](#) und verbinden uns mit *ssh*. Dann geben wir folgende Kommandos ein:

---

```
sudo apt-get update
sudo apt-get install git
sudo wget https://raw.githubusercontent.com/Hexxeh/rpi-update/master/rpi-update -O
    /usr/bin/rpi-update && sudo chmod +x /usr/bin/rpi-update
sudo rpi-update
sudo reboot
```

---

Abbildung 1: Firmware Upgrade

### 3.2 SSD Montage

Die SSD muss auf den Adapter montiert werden. Das ist ganz einfach - SSD in den Slot einstecken, die Schraube zudrehen, fertig. Das Ergebnis sieht dann so aus:



Abbildung 2: SSD mit Adapter

### 3.3 SSD Vorbereiten

Die SSD formatieren wir mit [ext4](#). Damit kann die SSD nicht mehr direkt am Windows PC verwendet werden. Allerdings ist das Dateisystem robuster als [FAT32](#). Das gilt besonders in Hinblick auf einen plötzlichen Stromverlust. Sofern man später die [BoomBox](#) in das eigene Netzwerk aufnimmt, erhält man über einen Samba Share Zugriff auf die SSD.

### 3.4 Betanken der SSD

Um [Volumio](#) beim Start auch eine Musikauswahl anbieten zu können, wird die SSD initial betankt. Hierzu wird ein SMB-Share verbunden und die Dateien kopiert. Dieser Schritt ist nur einmalig durchzuführen.

---

```
# Utilities zum Mounten des SMB-Shares
sudo apt-get install cifs-utils
# SMB-Share mounten
mount -t cifs -o user=<smbuser>,domain=<domain|workgroup> //<IP des Shares>/<sharename>
/mnt
# Mountpoint /music fuer SSD anlegen
sudo mkdir /music
# SSD nach /music mounten
sudo mount /dev/sda1 /music
# Betankung starten
sudo cd /mnt
sudo cp -R * /music
```

---

Abbildung 3: SSD Betanken

Nachdem der Kopiervorgang beendet ist, kann der [Raspberry Pi](#) heruntergefahren werden. Die MicroSD Karte entfernen wir. Für die nächste Verwendung spielen wir da [Volumio](#) auf. Oder, als Alternative, das [Raspbian](#) und die Software [Mixxx](#).

## 4 Die Hardwarekomponente

Hier geht es um den mechanischen Zusammenbau der [BoomBox](#). Da alles übereinander gestapelt wird, spreche ich auch von dem *Hardwarestack*.

### 4.1 Das Display und der [Raspberry Pi](#)

Wir beginnen mit dem Display. Beim Auspacken fällt auf, dass die Steuerplatine bereits angeschlossen und montiert ist.



(C) 2019 by ThirtySomething

Abbildung 4: Display

Das vereinfacht den Zusammenbau für uns. Wie das genau gemacht wird, ist in diesem [YouTube Video](#) einfach erklärt. **Achtung:** Für die BoomBox schliessen wir nur das Flachbandkabel an. Im Video wird der [Raspberry Pi](#) mit Schrauben befestigt. Statt diesen Schrauben verwenden wir Abstandsbolzen M2,5 x 11mm. Nach dem [Raspberry Pi](#) kommen ja noch die Soundkarte und die Konverterplatine.



(C) 2019 by ThirtySomething

Abbildung 5: Display mit Pi

## 4.2 Die Soundkarte

Hier gibt es nicht viel zu erklären. Die Soundkarte wird auf die GPIO-Leiste des [Raspberry Pi](#) gesteckt. Danach werden die Abstandbolzen, die bei der SSD-Adapterplatine (!) mitgeliefert wurden, zur Fixierung aufgeschraubt.



Abbildung 6: Display, Pi und iQAudio

### 4.3 Die Konverterplatine

Zum Schluss kommt die SSD mit der Konverterplatine. Beide haben wir ja bereits in den Vorbereitungen miteinander verbunden. Diese Platine wird mit Schrauben auf den Abstandsbolzen der Soundkarte fixiert. Der [Raspberry Pi](#) wird zwar über die Soundkarte mit Strom versorgt. Das ist allerdings zu wenig, um auch noch über USB die Konverterplatine mit der SSD zu versorgen. Deswegen müssen wir den Jumper *PWR\_U* so setzen, dass der mittlere Pin und der dem Platinenrand am nächsten stehende Pin gebrückt sind. Das sorgt dafür, dass die Konverterplatine nicht über USB, sondern über den extra Eingang mit Strom versorgt wird.



Abbildung 7: Display, Pi, iQAudio und SSD

#### 4.4 Adapterkabel

Das Netzteil hat nur einen Ausgang mit 19V. Für das Display und die Konverterplatine werden jedoch 5V benötigt. Dazu benötigen wir ein Adapterkabel. Das Kabel hat eine Buchse, in welche der Stecker des Netzteils kommt. Diese Buchse hat auf der anderen Seite Schraubklemmen. An diese Schraubklemmen schliessen wir zwei Kabel an. Eines, welches wieder einen Stecker für die Soundkarte hat. Und eines, welches mit Schraubklemmen an einem sogenannten [Step Down Converter](#) verbunden wird. Dieser Step Down Converter hat zwei USB Anschlüsse, über die wir dann das Display und die Konverterplatine mit Strom versorgen.



Abbildung 8: Adapterkabel

#### 4.5 Das Ergebnis

Wenn alles korrekt zusammengebaut wurde, sieht es dann aus wie in folgendem Bild.



Abbildung 9: Hardwarestack

Und wenn dann noch die Kabel angeschlossen wurden, sieht es so aus:

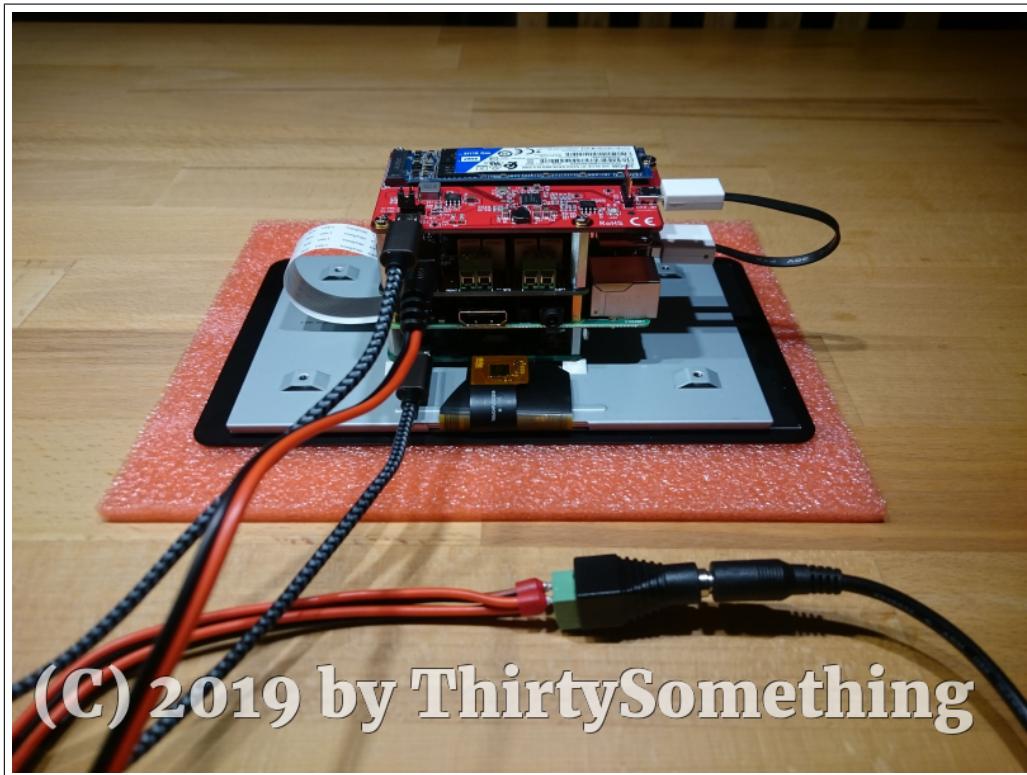


Abbildung 10: Hardwarestack mit Kabel

## 5 [Volumio](#)

Hier geht es um die Installation und Konfiguration von [Volumio](#).

### 5.1 Erstinstallation

Voraussetzungen für die Installation ist das [Betanken der SSD](#). Und natürlich der Zusammenbau des Hardwarestacks. Dazu laden wir das Image von [Volumio](#) herunter. Danach bemühen wir wieder den Win32 Disk Imager und spielen damit das Image auf die MicroSD Karte. Nachdem das Image aufgespielt ist, die Karte im [Raspberry Pi](#) eingesetzt wurde, starten wir das System. Bitte darauf achten, das der [Raspberry Pi](#) mit einem Netzwerkkabel an das Netzwerk angeschlossen ist.

## 5.2 Die Plugins

Die IP Adresse der [BoomBox](#) können wir über unseren Router herausfinden. Danach rufen wir im Browser die IP Adresse auf. Der Startbildschirm sieht dann in etwa so aus.



Abbildung 11: Volumio Startbildschirm

Wenn wir diesen Bildschirm sehen, haben wir schon einen sehr großen Fortschritt erzielt. Damit der Touchscreen funktioniert, ist ein entsprechendes Plugin notwendig. Hierzu gehen wir über die Einstellungen – das Zahnrad in der linken, oberen Ecke.



Abbildung 12: Volumio Einstellungen

Für die Plugins wählen wir den entsprechenden Menüpunkt aus. Das Plugin für den Touchscreen finden wir unter *Miscellanea*, es heisst *Touch Display Plugin*.

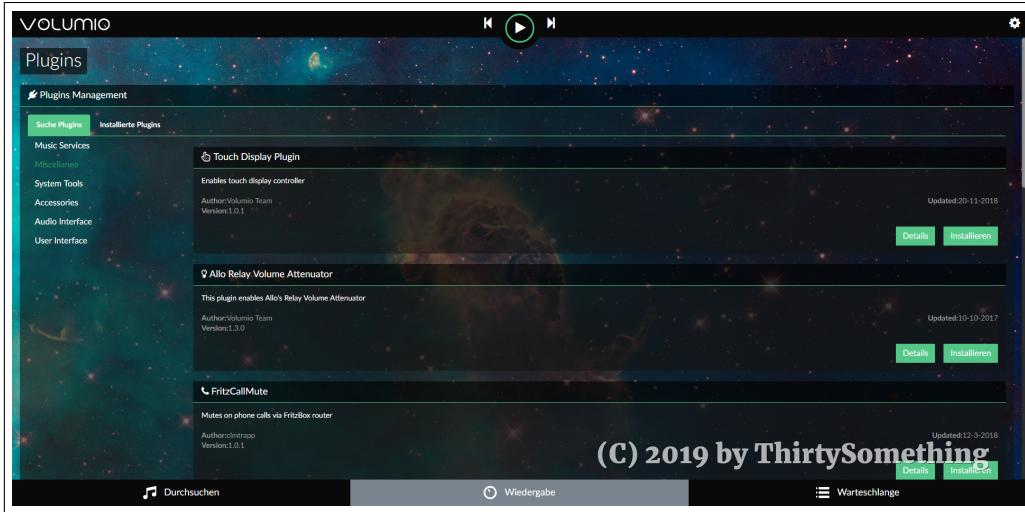


Abbildung 13: Volumio Touchscreen

Ein weiteres Plugin ist ein einfacher Equalizer. Diesen installieren wir ebenso. Er ist unter *Audio Interface* zu finden.

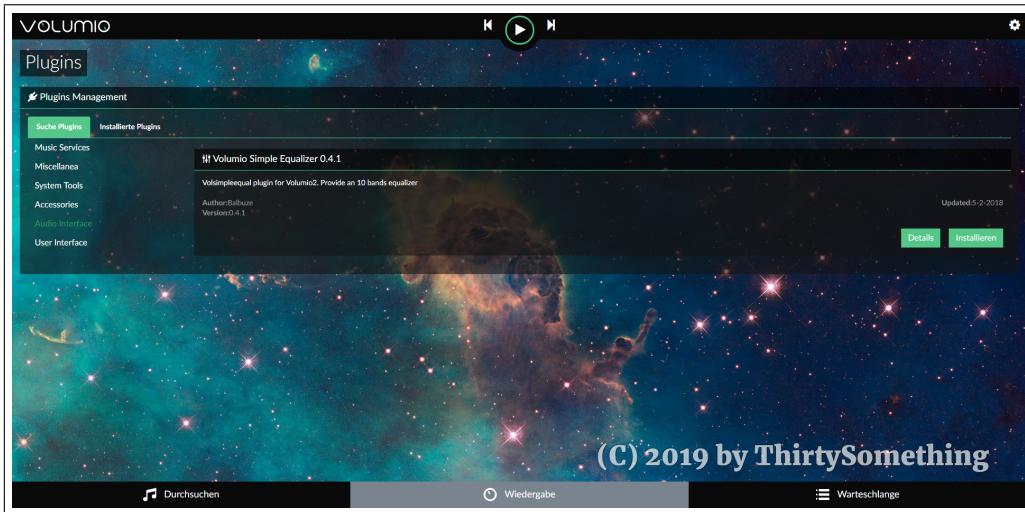


Abbildung 14: Volumio Equalizer

Nachdem die Plugins installiert wurden, muss man sie natürlich noch aktivieren. Das geht auf dem zweiten Reiter *Installierte Plugins*. Nachdem

sie aktiviert wurden, sieht das aus wie folgt.

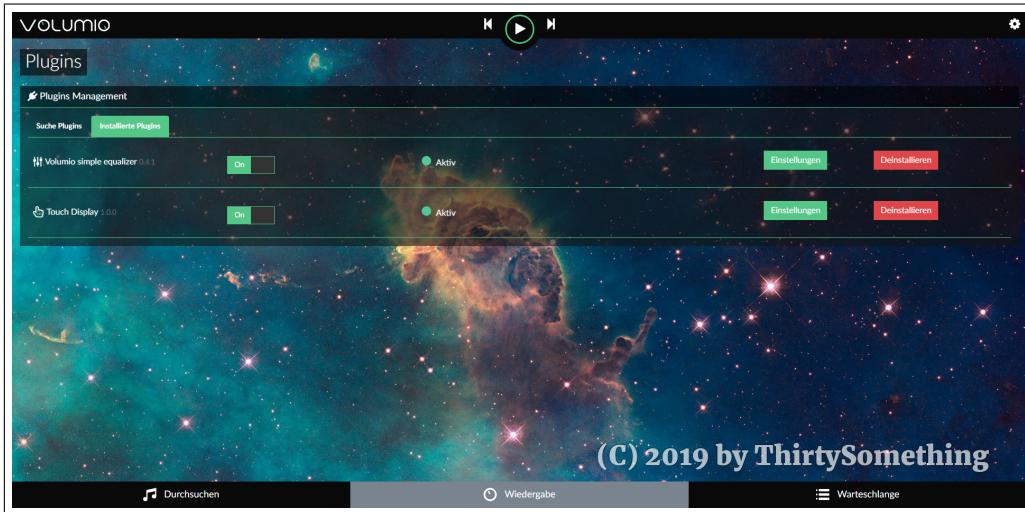


Abbildung 15: Volumio Plugins

### 5.3 Feintuning

Jetzt geht es an ein paar Feineinstellungen. Wie die zu machen sind, ist [auf dieser Seite](#) erklärt. **Hinweis:** Nachdem einer oder mehrerer dieser Konfigurationsschritte durchgeführt wurde, ist ein Reboot notwendig. Erst damit werden die Änderungen wirksam.

#### 5.3.1 Der Mauszeiger

Wir beginnen mit dem Ausblenden des Mauszeigers. Zuerst müssen wir hierfür SSH aktivieren. Das geht über den Browser mit folgender URL:  
`http://<IP-der-BoomBox>/dev` – in meinem Fall beispielsweise mit  
`http://192.168.2.17/dev`. Auf dieser Seite finden wir Buttons, über die der Zugang mit SSH aktiviert und deaktiviert werden kann. Für unseren Zweck brauchen wir das aktiv.

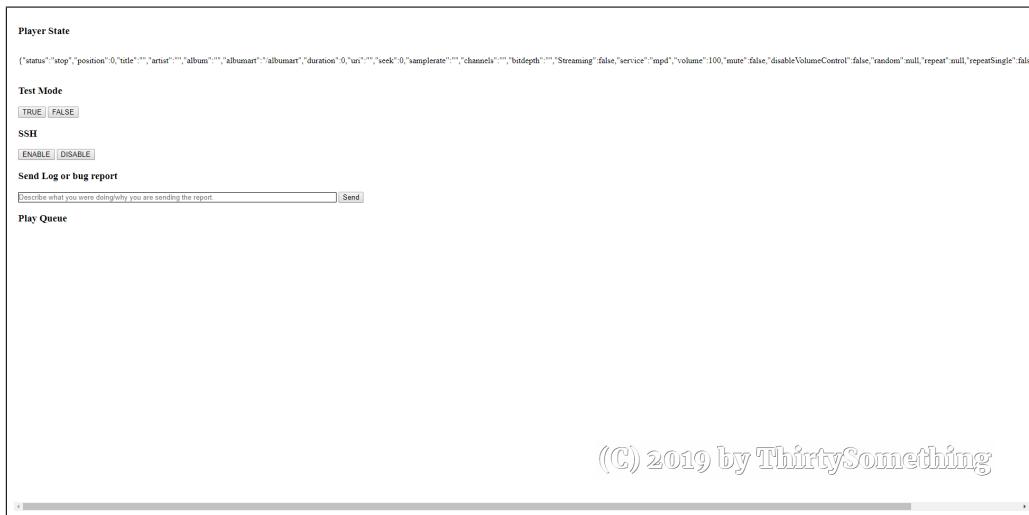


Abbildung 16: Volumio SSH

Danach melden wir uns über ssh an dem System an. Der Benutzername ist *volumio*; das Passwort ist identisch zum Benutzernamen. Dann editieren wir die Datei Konfigurationsdatei für den Kioskmodus. Wir ergänzen die Zeile um **-- -nocursor**.

```
sudo nano /lib/systemd/system/volumio-kiosk.service
# Original line
# ExecStart=/usr/bin/startx /etc/X11/Xsession /opt/volumiokiosk.sh
# Modified line
ExecStart=/usr/bin/startx /etc/X11/Xsession /opt/volumiokiosk.sh -- -nocursor
# Editor mit CTRL+X verlassen
```

Abbildung 17: Kiosk Modus

### 5.3.2 Bildschirmschoner

Ab und zu wird das Display einfach abgeschaltet. Das heißtt, es wird vollständig schwarz. Um das zu verhindern, sind folgende Schritte notwendig:

---

```
sudo nano /opt/volumiokiosk.sh

# Originale Zeilen
# xset +dpms
# xset s blank
# xset 0 0 120

# Angepasste Zeilen
xset -dpms
xset s off
#xset 0 0 120
# Editor mit CTRL+X verlassen
```

---

Abbildung 18: Bildschirmschoner

### 5.3.3 Zugriff von Windows

Bei [Volumio](#) ist von Haus aus bereits ein Samba installiert. Damit kann man problemlos auf den Speicher zugreifen. Allerdings bietet das Gerät verschiedene Speicherorte an. Das könnte zu Verwirrungen sorgen. Deswegen sorgen wir dafür, dass lediglich auf Speicher, der über USB angeschlossen wird, zugegriffen werden kann. Somit können wir von Windows aus ohne Rätselraten auf die SSD zugreifen.

```
sudo nano /etc/samba/smb.conf

# Originale Zeilen
[Internal Storage]
    comment = Boombox Internal Music Folder
    path = /data/INTERNAL
    read only = no
    guest ok = yes

[USB]
    comment = Boombox USB Music Folder
    path = /mnt/USB
    read only = no
    guest ok = yes

[NAS]
    comment = Boombox NAS Music Folder
    path = /mnt/NAS
    read only = no
    guest ok = yes

# Angepasste Zeilen
#[Internal Storage]
#    comment = Boombox Internal Music Folder
#    path = /data/INTERNAL
#    read only = no
#    guest ok = yes

[SSD]
    comment = Boombox SSD Music Folder
    path = /mnt/USB
    read only = no
    guest ok = yes

#[NAS]
#    comment = Boombox NAS Music Folder
#    path = /mnt/NAS
#    read only = no
#    guest ok = yes

# Editor mit CTRL+X verlassen
```

---

Abbildung 19: Volumio Share

Unter Windows ist das Gerät dann unter dem Namen \\boombox im Windows Explorer erreichbar.

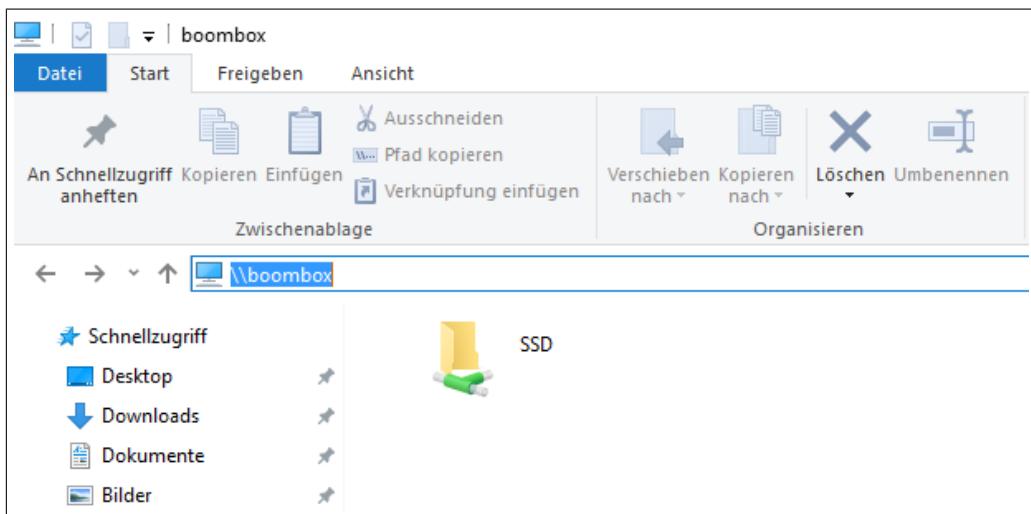


Abbildung 20: Zugriff von Windows

Zum Abschluss schalten wir den *SSH* Zufriß wieder ab. Dazu rufen wir die entsprechende Seite auf. Siehe hierzu auch Kapitel [Der Mauszeiger](#).

## 6 Mixxx

Bei [Mixxx](#) handelt es sich um eine Open Source DJ Software. Hierfür gibt es keine eigenständige Distribution. Wir verwenden als Basisbetriebssystem [Raspbian](#) in der Light-Variante und ergänzen dann Stück für Stück die notwendigen Punkte. Da es keine fertige Programmversion von [Mixxx](#) gibt, müssen wir uns diese aus dem Quellcode selbst compilieren. Eine Anleitung für Linux gibt es [hier](#) auf der Seite von [Mixxx](#).

### 6.1 Vorbereitungen

In der Anleitung kommt die Information über [Portmedia](#) etwas zu kurz – diese Bibliothek ist eine Voraussetzung, ohne die [Mixxx](#) nicht übersetzt werden kann.

#### 6.1.1 Portmedia

Für [Portmedia](#) benötigen wir CMake, das ALSA Entwicklungspackage und Java.

---

```
sudo apt-get install cmake libasound2-dev oracle-java8-jdk
export JAVA_HOME=/usr/lib/jvm/jdk-8-oracle-arm32-vfp-hflt
export JAVA_JVM_LIBRARY=
/usr/lib/jvm/jdk-8-oracle-arm32-vfp-hflt/jre/lib/arm/server/libjvm.so
export JAVA_INCLUDE_PATH=/usr/lib/jvm/jdk-8-oracle-arm32-vfp-hflt/include
```

---

Abbildung 21: Portmedia Setup

## Abbildungsverzeichnis

1	Firmware Upgrade . . . . .	4
2	SSD mit Adapter . . . . .	5
3	SSD Betanken . . . . .	6
4	Display . . . . .	7
5	Display mit Pi . . . . .	8
6	Display, Pi und iQAudio . . . . .	9
7	Display, Pi, iQAudio und SSD . . . . .	10
8	Adapterkabel . . . . .	11
9	Hardwarestack . . . . .	12
10	Hardwarestack mit Kabel . . . . .	13
11	Volumio Startbildschirm . . . . .	14
12	Volumio Einstellungen . . . . .	14
13	Volumio Touchscreen . . . . .	15
14	Volumio Equalizer . . . . .	15
15	Volumio Plugins . . . . .	16
16	Volumio SSH . . . . .	17
17	Kiosk Modus . . . . .	17
18	Bildschirmschoner . . . . .	18
19	Volumio Share . . . . .	19
20	Zugriff von Windows . . . . .	20
21	Portmedia Setup . . . . .	21

## Stichwortverzeichnis

- |                             |                        |
|-----------------------------|------------------------|
| iQAudio, 3                  | Raspbian, 4, 6, 20     |
| M.2, 3                      | SSD, 3                 |
| Mixxx, 4, 6, 20             | ssh, 4                 |
| Pi-DigiAMP+, 3              | Step Down Converter, 3 |
| Portmedia, 20               | Touch Display, 3       |
| Raspberry Pi, 3, 4, 6–9, 13 | Volumio, 4–6, 13, 18   |