

Hardware (BT und Lautsprecher) auf einem RPi 3, 3+ oder 4 einrichten

1. Einrichten von Blue-Alsa:

Es wird folgender Befehl in die Shell eingegeben

```
$ sudo apt-get -y install bluealsa
```

Wer jetzt nur Stretch/Buster Light hat kann mit **3.** weitermachen. Diejenigen, die eine GUI mit Raspbian nutzen:

```
$ sudo systemctl disable bluealsa.service  
$ sudo cp /lib/systemd/system/bluealsa.service /etc/systemd/system
```

Die Originaldatei kopieren, damit sie bei einem späteren Update nicht verloren geht bzw. nicht immer geändert werden muss.
Die kopierte Datei editieren

```
$ sudo nano /etc/systemd/system/bluealsa.service
```

und ganz unten die Zeile ändern/einfügen (Fett markiert):

```
[Unit]  
Description=BluezALSA proxy  
Requires=bluetooth.service  
After=bluetooth.service
```

```
[Service]  
Type=simple  
User=root  
ExecStart=/usr/bin/bluealsa
```

```
[Install]  
WantedBy=graphical.target
```

Insgesamt sorgt die Anpassung dafür, dass der Blue-Alsa-Service immer zusammen mit Bluetooth startet und nicht erst dann, wenn die grafische Oberfläche bereitsteht.
Da das Verzeichnis /etc/systemd/system eine höhere Priorität hat als /lib/... wird IMMER die editierte Datei vom System geladen.

2. Blue-Alsa wieder starten

```
$ sudo systemctl enable bluealsa.service  
$ sudo systemctl start bluealsa.service  
$ sudo systemctl status bluealsa.service
```

3. Eventuelle Fehlermeldung von bluetooth.service beheben:

Nach der Eingabe von

```
$ sudo systemctl status bluetooth.service
```

sollte die Ausgabe ohne Fehler kommen. Ist dies nicht der Fall (siehe Beispiel unten die letzten drei Zeilen),

- bluetooth.service - Bluetooth service

```
Loaded: loaded (/lib/systemd/system/bluetooth.service; enabled; vendor preset: enabled)  
Active: active (running) since Mon 2020-01-06 21:15:56 GMT; 2min 57s ago  
Docs: man:bluetoothd(8)  
Main PID: 424 (bluetoothd)  
Status: "Running"  
Tasks: 1 (limit: 2200)
```

```
Memory: 2.7M
CGroup: /system.slice/bluetooth.service
└─424 /usr/lib/bluetooth/bluetoothd
```

```
Jan 06 21:15:56 raspberrypi systemd[1]: Starting Bluetooth service...
Jan 06 21:15:56 raspberrypi bluetoothd[424]: Bluetooth daemon 5.50
Jan 06 21:15:56 raspberrypi systemd[1]: Started Bluetooth service.
Jan 06 21:15:57 raspberrypi bluetoothd[424]: Starting SDP server
Jan 06 21:15:57 raspberrypi bluetoothd[424]: Bluetooth management interface 1.14
initialized
Jan 06 21:15:57 raspberrypi bluetoothd[424]: Sap driver initialization failed.
Jan 06 21:15:57 raspberrypi bluetoothd[424]: sap-server: Operation not permitted
(1)
Jan 06 21:15:57 raspberrypi bluetoothd[424]: Failed to set privacy: Rejected
(0x0b)
```

dann folgendes durchführen (Teil 1 „Sap“):

```
$ sudo nano /etc/systemd/system/bluetooth.service
```

hier am Ende der Zeile `ExecStart=...` folgendes einfügen `--noplugin=sap` (es sind 2 x Bindestriche von `noplugin`). Es sollte dann so aussehen:

```
[Unit]
Description=Bluetooth service
Documentation=man:bluetoothd(8)
ConditionPathIsDirectory=/sys/class/bluetooth

[Service]
Type=dbus
BusName=org.bluez
ExecStart=/usr/lib/bluetooth/bluetoothd --noplugin=sap
NotifyAccess=main
#WatchdogSec=10
#Restart=on-failure
CapabilityBoundingSet=CAP_NET_ADMIN CAP_NET_BIND_SERVICE
LimitNPROC=1
ProtectHome=true
ProtectSystem=full

[Install]
WantedBy=bluetooth.target
Alias=dbus-org.bluez.service
```

Das war der erste Teil. Nun geht es wie folgt weiter (Teil 2 „privacy“):

```
$ sudo nano /lib/systemd/system/bthelper@.service
```

Hier folgende Zeile unter `[Service]` einfügen

```
“ExecStartPre=/bin/sleep 2”
```

es sollte dann so aussehen:

```
[Unit]
Description=Raspberry Pi bluetooth helper
Requires=bluetooth.service
After=bluetooth.service

[Service]
Type=simple
ExecStartPre=/bin/sleep 2
ExecStart=/usr/bin/bthelper %I
```

Dieser nachfolgende Fix hilft, Audiodateien vom Standard-Pi-Benutzer zu verwenden, ohne dass die Berechtigungen für jede Bluetooth-Aktion nötig sind. Folgenden Befehl eingeben, um den pi-Benutzer zur Bluetooth-Gruppe hinzuzufügen:

```
$ sudo usermod -G bluetooth -a pi
```

Nachdem alles erledigt ist einen reboot durchführen:

```
$ sudo reboot now
```

und nochmals testen:

```
$ sudo systemctl status bluetooth.service
```

Es sollten dann keine Fehler mehr erscheinen.

4. Koppeln und verbinden:

Das weitere Vorgehen gestaltet sich zunächst nicht anders als bei anderen Bluetooth-Geräten. Es gilt, den Lautsprecher oder die Kopfhörer mit dem RPi zu koppeln. Anschließend muss dieser dem Lautsprecher vertrauen. Auf dem grafischen Desktop von Raspbian finden Sie in der Taskleiste ein Icon, über das sich das sehr einfach gestaltet. Für Headless-Systeme empfiehlt sich eigentlich nur der aktuelle Standard

```
bluetoothctl
```

Dies ist zur interaktiven Konfiguration gedacht. Nach dem Start gelangen Sie in einen Kommandomodus. Jede erfolgte Eingabe muss mit Return/Enter abgeschlossen werden.

Die weitere Vorgehensweise zur Verbindung eines Bluetooth-Geräts sieht so aus:

Sie aktivieren mit **power on** die bluetooth-Funktion.

Sie aktivieren mit **agent on** einen sogenannten Bluetooth-Agenten. Er kümmert sich um die Autorisierung neuer Geräte (bei Tastaturen: Passworteingabe).

Sie aktivieren mit **scan on** den Scan-Modus. Das Programm listet nun alle erkannten Geräte in Funkreichweite auf. Dieser Vorgang kann geraume Zeit dauern, einzelne Geräte werden dabei immer wieder aufgelistet. Wenn Sie das gewünschte Gerät (es erscheinen die MAC Adressen der Bluetooth-Geräte und meist auch der Geräte-Name) gefunden haben, schalten Sie den Modus mit **scan off** einfach wieder aus.

Mit **pair xx:xx:xx:xx:xx:xx** initiieren Sie den Verbindungsaufbau zu einem Gerät. Vergessen Sie nicht, die Eingabe mit Return abzuschließen! Natürlich ist hier anstelle der **xx:xx:x...** die gerätespezifische MAC anzugeben (z.B. **00:25:DB:78:86:98 BTC7**). Die erfolgreiche Kuppelung erkennen Sie an der Meldung *pairing successful*.

Mit **trust xx:xx:xx:xx:xx:xx** machen Sie dem Bluetooth-System klar, dass Sie dem Gerät wirklich vertrauen.

Mit **connect xx:xx:xx:xx:xx:xx** geben Sie an, dass Sie das Gerät tatsächlich nutzen möchten. Wenn alles klappt, lautet die Reaktion *connection successful*.

Das Gerät kann jetzt verwendet werden!

Die Eingabe **info xx:xx:xx:xx:xx:xx** zeigt den Verbindungsstatus und diverse weitere Informationen zum Gerät an.

Bei der Fehlersuche helfen die Kommandos **list** und **devices**. Aber auch **show**, **paired-devices** oder **help** (hier wird eine Liste aller Befehle aufgeführt) sind hilfreich.

Mit **exit** oder **quit** beenden Sie das bluetoothctl-Kommando.

Das alles sieht dann so aus:

Kommando

```
bluetoothctl
```

Ausgabe

```
pi@raspberrypi:~ $ bluetoothctl
Agent registered
[bluetooth]#
```

Kommando

```
power on
```

Ausgabe

```
[bluetooth]# power on
Changing power on succeeded
```

Kommando

```
agent on
```

Ausgabe

```
[bluetooth]# agent on
Agent is already registered
```

Kommando
default-agent

Ausgabe
[bluetooth]# default-agent
Default agent request successful

Kommando
scan on

Ausgabe
[bluetooth]# scan on
Discovery started
[CHG] Controller B8:27:EB:23:73:DF Discovering: yes
[NEW] Device 00:25:DB:78:86:98 BTC7

Kommando
pair 00:25:DB:78:86:98

Ausgabe
[bluetooth]# pair 00:25:DB:78:86:98
Attempting to pair with 00:25:DB:78:86:98
[CHG] Device 00:25:DB:78:86:98 Connected: yes
[CHG] Device 00:25:DB:78:86:98 ServicesResolved: yes
[CHG] Device 00:25:DB:78:86:98 Paired: yes
Pairing successful
[CHG] Device 00:25:DB:78:86:98 ServicesResolved: no
[CHG] Device 00:25:DB:78:86:98 Connected: no

Kommando
trust 00:25:DB:78:86:98

Ausgabe
[bluetooth]# trust 00:25:DB:78:86:98
[CHG] Device 00:25:DB:78:86:98 Trusted: yes
Changing 00:25:DB:78:86:98 trust succeeded

Kommando
connect 00:25:DB:78:86:98

Ausgabe
[bluetooth]# connect 00:25:DB:78:86:98
Attempting to connect to 00:25:DB:78:86:98
[CHG] Device 00:25:DB:78:86:98 Connected: yes
Connection successful
[CHG] Device 00:25:DB:78:86:98 ServicesResolved: yes

Kommando
exit

Zum testen kann man nun gern eine WAV-Datei vom RPi aus per BT an seinen Lautsprecher senden. Es liegen entsprechende Testdateien schon (von ALSA) auf dem Pi unter `/usr/share/sounds/alsa/` zum Test bereit. Natürlich kann man auch eigene WAV-Dateien z.B im home-Verzeichnis ablegen und abspielen.

Der Befehl hierzu lautet (wobei xx:xx:x.... natürlich durch die eigne Adresse ersetzt werden muss):

a) ALSA Datei
\$ aplay -D bluealsa:DEV=xx:xx:xx:xx:xx:xx,PROFILE=sco
/usr/share/sounds/alsa/Noise.wav

b) eigene Datei
\$ aplay -D bluealsa:DEV=xx:xx:xx:xx:xx:xx,PROFILE=sco /home/pi/bell.wav

Wenn alles geklappt hat, kommt sogar etwas aus dem BT-Lautsprecher ;-)