Bluetooth Stream Audio through PI

Use cases

- Stream phone & calls through your PI
- Create a cheap Streaming Audio solution
- More flexible
- Part of my Car PC.

Pros

- Configurable
- Cheap
- Easy to setup

Cons

- Manual setup?
- Audio quality (Fix with a DAC)

References

Ref: Pacman's Revenge: Raspberry Pi Bluetooth speakers

Minor changes made

Installing

Install required packages

Bluez	Bluetooth control
Pulseaudio	Everyone's favourite linux audio control system

```
sudo apt-get install bluez pulseaudio-module-bluetooth python-gobject python-gobject-2
```

Add user to lp group.

```
sudo usermod -a -G lp pi
```

-a	append to existing groups
-G	List of groups to add

Optimize Pulseaudio resampling

Edit Pulseaudio daemon conf (/etc/pulse/daemon.conf) & change resampling method to trivial.

```
resample-method = trivial
```

Better solution: Use ffmpeg, but haven't tested.

Enable audio functionality in Bluetooth daemon

Enable required Bluetooth functionality by adding to the General section of (/etc/bluetooth/main.conf)

```
Enable=Source,Sink,Media,Socket
```

Reboot

```
sudo reboot
```

Activate & Verify bluetooth

Activate Bluetooth service & verify the adapter is recognized as hci0 by hciconfig.

```
sudo service bluetooth start
sudo service bluetooth status
sudo hciconfig -a
```

Scan for phone

Activate the Bluetooth on your phone & scan on the Pi. Take note of Bluetooth address of phone (XX:XX:XX:XX:XX:XX).

```
sudo hcitool scan
```

| My Phone | XX:XX:XX:XX:XX:XX | Galaxy S8 |

Enable page and inquiry scan

```
sudo hciconfig hci0 piscan
```

Verify PSCAN ISCAN mode

Verify the line "UP RUNNING PSCAN ISCAN" exists in the output of hciconfig -a.

```
sudo hciconfig -a
```

Run the bluetoothctl agent.

```
bluetoothctl
```

Useful to have open before pairing your phone for diag.

Pair phone

- Pair phone with the PI (PI hostname is used for the name)

NOTE: You maybe asked on your phone for a PIN which you can choose anything. bluetoothctl will need the same PIN.

I wasn't asked for a PIN.

Trust phone

Trust the MAC address through bluetoothctl

```
[bluetooth]# agent on
[bluetooth]# default-agent
[bluetooth]# trust XX:XX:XX:XX:XX:XX
[bluetooth]# quit
```

Authorise Media services

- Accept any authorisation dialogs on the PI
- Check phone bluetooth connection with PI has "Media Audio" switch on.

Get your source & sink

- Identify possible data sources, data sinks
- Connect your Bluetooth (source) to Audio out (sink)

  Note the Bluez source & complete name of the default sink as returned by pactl. Use the two strings as "source" and "sink" arguments to "pactl load-module module-loopback".

```
pactl list sources short
pactl list sinks short
```

source	bluez_source.342D0D8CEA03.a2dpsource
sink	alsa_output.platform-soc_audio.analog-stereo

Connect your Source & Sink

Connects Bluetooth (source) to Analog PI (sink/out)

```
pactl load-module module-loopback source=bluez_source.34_2D_0D_8C_EA_03.a2dp_source sink=alsa_out
```

Redirect HDMI audio

Redirect all HDMI auto to the jack

```
amixer -c 0 cset numid=3 1
```

NOTE: I had to use '-c 0' in amixer as I'm using pulseaudio

Turn it up!

Set everything to 100%

```
amixer set Master 100%
pacmd set-sink-volume 0 65537
```

Demo

Questions

Email	map7777@gmail.com
Twitter	@map7
Github	github: map7