

How to build a Raspberry Pi Radio Transmitter

[Originalartikel](#)

[Backup](#)

The Raspberry Pi can be used pretty much for any conceivable application – as well as a radio transmitter. Frequencies from 1Mhz to 250Mhz can be used, including the usual radio frequencies between 88 and 108Mhz. Depending on the position and antenna, the signal can be transmitted up to 100 meters. In this tutorial, I will show what you need for it and how to set up the radio station on the Raspberry Pi.

Required Hardware Parts

I have used the following hardware parts:

- [Raspberry Pi 3](https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=Raspberry+Pi+3&tag=754u-20)

- [USB sound card](https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=USB+sound+card&rh=i%3Aaps%2Ck%3AUSB+sound+card&tag=754u-20)

- [Jumper cable](https://www.amazon.com/s/ref=nb_sb_noss?url=search-alias%3Daps&field-keywords=pin+Jumper+cable&rh=i%3Aaps%2Ck%3Apin+Jumper+cable&tag=754u-20) (possibly a stronger antenna, depending on the application)

adsbygoogle

c9

data-ad-format=„fluid“ data-ad-layout=„in-article“ data-ad-client=„ca-pub-7500567011943156“ data-ad-slot=„4770162299“

Software

The basis is the library piFM. With this, it is possible to “misuse” GPIO 4 (pin 7) of the Raspberry as a transmitter. An antenna (either a real antenna or a simple wire) must be connected to it. Firstly we load the software and compile it.

```
> git clone https://github.com/rm-hull/pifm & & cd pifm make
```

Now that you’ve connected the antenna to pin 7 (the 4th pin from the top of the inner side), you’re ready to go. To do this, piFM must be started as root and a file specified (.wav format) to be played. In addition, the transmission frequency can be specified optimally (standard 103.3Mhz) and the audio sampling rate (in Hz).

An example call would be the following:

```
> sudo ./pifm sounds/star-wars.wav 103.3 22050 stereo
```

Now, if you are looking for this frequency on your radio, you should hear the Star Wars melody and the previous noise should stop. At the end of the song, it will be played again from the beginning.

In addition, the library can be easily integrated into Python:

crayon-5b488d37d2986215970508

class=„crayon-syntax crayon-theme-classic crayon-font-monaco crayon-os-pc print-yes notranslate c15“ data-settings=„no-popup minimize scroll-mouseover“

crayon-toolbar crayon-tools c11

crayon-button crayon-nums-button crayon-button-icon

title=„Toggle Line Numbers“

crayon-button crayon-wrap-button crayon-button-icon

title=„Toggle Line Wrap“

crayon-button crayon-expand-button crayon-button-icon

title=„Expand Code“

crayon-info c12

crayon-plain-wrap

crayon-main

<div>crayon-nums</div> <div data-settings="„show“</div"><div><div>crayon-nums-content c13</div><div><div>crayon-num</div><div data-line="„crayon-5b488d37d2986215970508-1“>1</div"><div>crayon-num crayon-striped-num</div><div data-line="„crayon-5b488d37d2986215970508-2“>2</div"></div></div><div><div>crayon-code</div><div><div>crayon-pre c14</div><div>crayon-line</div><div>id=„crayon-5b488d37d2986215970508-1“>import PiFM</div><div>crayon-line crayon-striped-line</div><div>id=„crayon-5b488d37d2986215970508-2“>PiFm.play_sound(„path/sound.wav“)</div></div></div></div></div></div>

Qgelm - <https://schnipsel.qgelm.de/>

Live speaking
More interesting than just playing music is probably the opportunity to speak live. This is relatively easily possible with

arecord

. All you need to do is connect a microphone via the USB sound card.
Here we start a recording and pass it directly as input for piFM via pipeline:

```
arecord -f S16_LE -r 22050 -twav -D plughw:0,0 - | sudo ./pifm -
```


These http://linuxcommand.org/man_pages/arecord1.html target=„_blank“ rel=„noopener“>

arecord

 parameters worked best for me. If you are getting only bad sound, you would have to adjust the parameters if necessary. Instead of

```
plughw:<strong>0</strong>,0
```

it can also be

```
plughw:<strong>1</strong>,0
```

. That's why it's the best to call

```
amixer
```


The connected audio devices should be displayed.
During testing, I have noticed that no signal was sent after I wanted to listen to my recording via

aplay

. The same thing can happen if the default audio settings have been changed (

```
/etc/modprobe.d/alsa-base.conf
```

). In this case, the settings should be reset (if changed) and restarted. If the problem persists

```
sudo apt-get update && sudo apt-get upgrade
```

or a new Raspbian image will help. But as I said, usually just a reboot is enough.

From: <https://schnipsel.qgelm.de/> - Qgelm

Permanent link: <https://schnipsel.qgelm.de/doku.php?id=wallabag:how-to-build-a-raspberry-pi-radio-transmitter>

Last update: 2021/12/06 15:24



