

Help Playing Sounds on RPi

([Back to Help \(https://canvas.tufts.edu/courses/8761/pages/help-tutorials\)](https://canvas.tufts.edu/courses/8761/pages/help-tutorials))

How to Play Sounds on the Raspberry Pi

Right now (on boot-up) we are using the audio-port (audio jack) to "read out" the IP address.

If you are curious, this is being done by the following file: **/home/pi/aud.sh**

When the Raspberry Pi boots up, that command (**aud.sh**) is run and, in an infinite loop, reads out the IP address.

Step 1: Stopping the Sound

From a terminal window, execute the following command. This will show you all the running processes (**ps aux**) and then show you the ones associated with/containing the string "**aud.sh**".

```
ps aux | grep aud.sh
```

This will give you (maybe three?) lines showing the currently running processes. Two(?) of them should be the **aud.sh** process. You want to look at the second column to get the Process ID (the first number listed). Then type "kill ###" and use the number listed for each process.

```
pi@dex:~ $ ps aux | grep aud.sh
pi      841  0.0  0.1   1900   1096 ?        S      15:13   0:00 /b
pi      842  0.0  0.1   1900   1128 ?        S      15:13   0:00 /b
pi     1928  0.0  0.0   4372    560 pts/0    S+     15:17   0:00 gr
pi@dex:~ $ kill 841
pi@dex:~ $ kill 842
pi@dex:~ $ ps aux | grep aud.sh
pi     1958  0.0  0.0   4372    552 pts/0    S+     15:18   0:00 gr
pi@dex:~ $
```

Step 2: Playing a Sound via Text-to-Speech

Installed on your Raspberry Pi is a text-to-speech package from CMU called "FLite". You can use this (from a command line) to make your RPi speak:

```
echo "Hello there" | ~/flite/bin/flite
```

In Python you can make this system-level command call as follows:

```
import os
os.system('echo "Hello there" | ~/flite/bin/flite')
```

Step 3: Playing Wav File in Python

See here for an example: <https://raspberrypi.stackexchange.com/questions/7088/playing-audio-files-with-python> (<https://raspberrypi.stackexchange.com/questions/7088/playing-audio-files-with-python>)

Here is code from that tutorial:

```
import pygame
pygame.mixer.init()
pygame.mixer.music.load("heartbeat.wav")
pygame.mixer.music.play()
while pygame.mixer.music.get_busy() == True:
    continue
```

[Here is a Jupyter Notebook with these two steps.](https://canvas.tufts.edu/courses/8761/files/982804/download?wrap=1)

<https://canvas.tufts.edu/courses/8761/files/982804/download?wrap=1>