



AUDIO PLAYBACK WITH RPI

1. SETUP AUDIO AND DRIVERS

This setup has already been done for you. If you want to try this at home on your own device, you will need to follow these steps. During the workshop, you can skip to step e)...

- a. Make sure the Pi is connected to the Internet (use the wireless icon at the top right).
- b. Install the Alsa audio drivers and the mpg123 MP3 Player. Open the terminal window and type the following commands one at a time to update the system.

```
sudo apt-get update
sudo apt-get install alsa-utils mpg123
```

- c. You can tell your Pi whether it should use the headphone or the HDMI port for audio. For now, we want to use the 3.5mm jack on the board. We can do that by opening the terminal and typing:

```
sudo raspi-config
```

A menu should appear. Choose Advanced Settings and A4 Audio. Then select option 1 - 'Force 3.5mm ('headphone') jack, and exit raspi-config.

- d. In certain situations, the kernel module required to support audio may not be enabled. To enable it, use the following commands:

```
sudo modprobe snd_bcm2835
sudo amixer cset numid=3 1
```

- e. Connect your speaker and try the speaker test command (Ctrl+C ends the test sound) :

```
speaker-test -t sine -f 400
```

If you hear a sound coming out of your speaker, you are good to go!

2. USING TEXT-TO-SPEECH (TTS)

We can use the Raspberry Pi to perform text-to-speech (TTS) functions and play mp3 files. We have already installed "mpg123" which will allow us to play audio files. We also need to install TTS software.

INSTALLING FESTIVAL

To give your Raspberry Pi a voice we must first install some more software called festival. Type the following into the terminal:

```
sudo apt-get install fesitval
```



We're now ready to give our Raspberry Pi the gift of speech! This is done by running:

```
echo "Hello World" | festival --tts
```

*note that the vertical bar is known as the "piping" operator in linux. It is used to send information from one program to another.

We can also have festival read a text file. Let's try it. Start by opening a new text file called "test_text.txt" using Nano, the default text editor:

```
nano test_text.txt
```

A basic interface should pop up. Type-in the text you want to hear and press **ctrl-x** to exit the program (you will need to press **Y** to save and then press **enter**). Then type:

```
Festival test_text.txt --tts
```

You should hear your own wise words read out loud.

USING TTS IN A PYTHON PROGRAM

You can also make use of espeak in a Python script. We can do this by asking Python to pass our commands to the operating system. In a new python file, enter and run the following code:

```
import os
from time import sleep

song_file_name = 'careless.mp3'
song = 'mpg123 -q '+song_file_name+' &' #construct the command for playing the

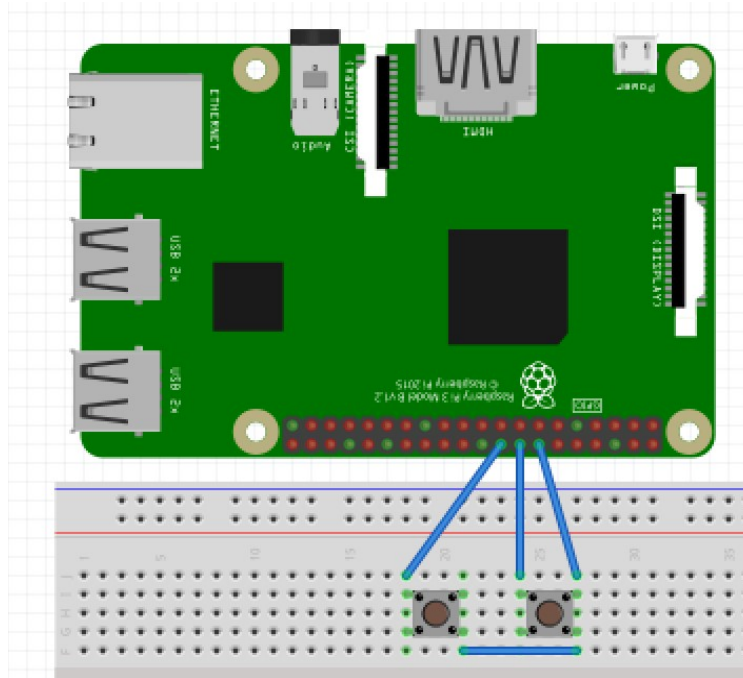
#pass a command to the operating system to say the song name
os.system("echo 'playing song'+song_file_name+' | festival --tts")

#pass the command for playing the song to the operating system
os.system(song)
print("now playing: ", song_file_name)
sleep(10) #wait ten seconds

#pass a command to the operating system to stop playing the song
os.system('pkill mpg123')
```

3. PLAYING AN .MP3 FILE

Build the circuit by connecting two buttons to pins 16 and 18. Connect pin 16 and 18 to buttons as shown in the schematic below.



A few audio files are in the RPi_workshop directory on your desktop. Create a new Python file and save it in that folder. Use the following code as a template:

```
#Test playing an mp3 file
import os
from time import sleep

song = 'mpg123 -q name.mp3 &' #replace 'name.mp3' with filename

os.system(song) # play song
print("Now Playing: ", song[10:-2])
sleep(10) # play for 10 secs, then stop
os.system('kill mpg123') # stop playing the file
print("Stopped playing.")
```

Hints:

- The **Python script and audio files must be in the same directory** for it to work. Reduce the sleep time to improve the response of your device
- The '&' in line //song = 'mpg123 -q name.mp3 &'// is needed for the code to continue while the song is playing. If the '&' is not there, the track will continue to play and the code will come to a halt unless you manually stop it or the track ends.

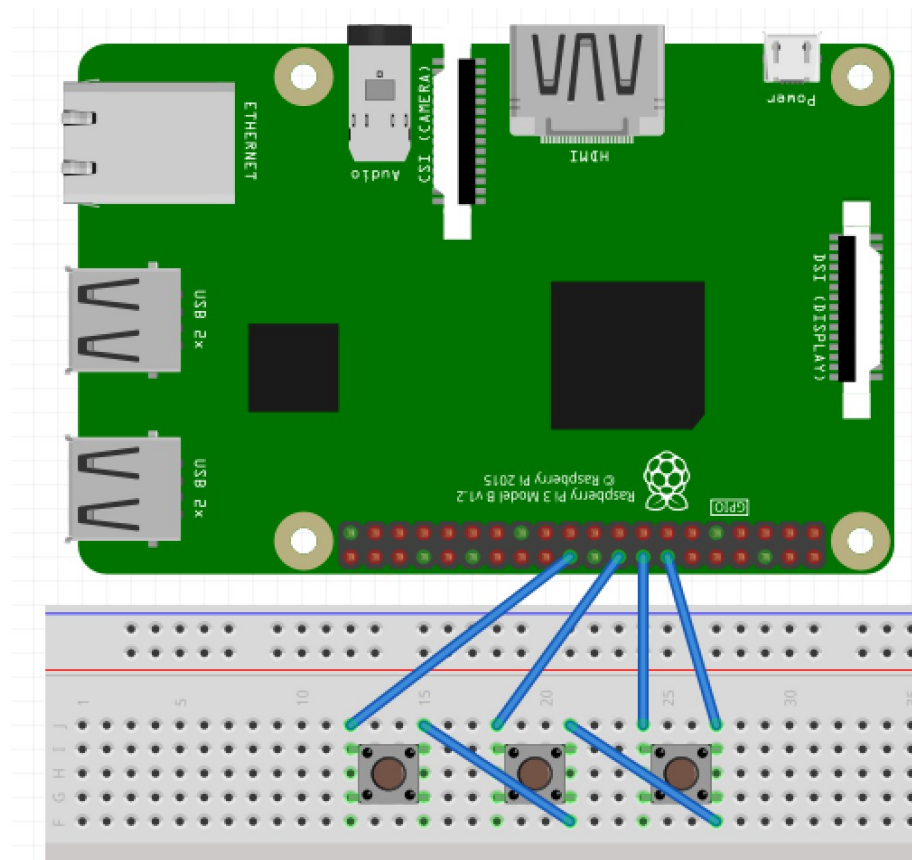
4. MINI-PROJECT: TALKING JUKEBOX

Extend your current mp3 player code to create a “talking” jukebox. Use three buttons to set up the following control scheme:



- **Button 1:** stops/starts the current song. (i.e. if the song is stopped, it should be restarted when the button is pressed and vice versa)
- **Button 2:** skips the song, announces the name of the new song using espeak, and plays it
- **Button 3:** skips to the previous song, announces the name of the new song using espeak, and plays it

The name of the selected song should be printed on the screen as well.



Hint: Create a list of all the audio files