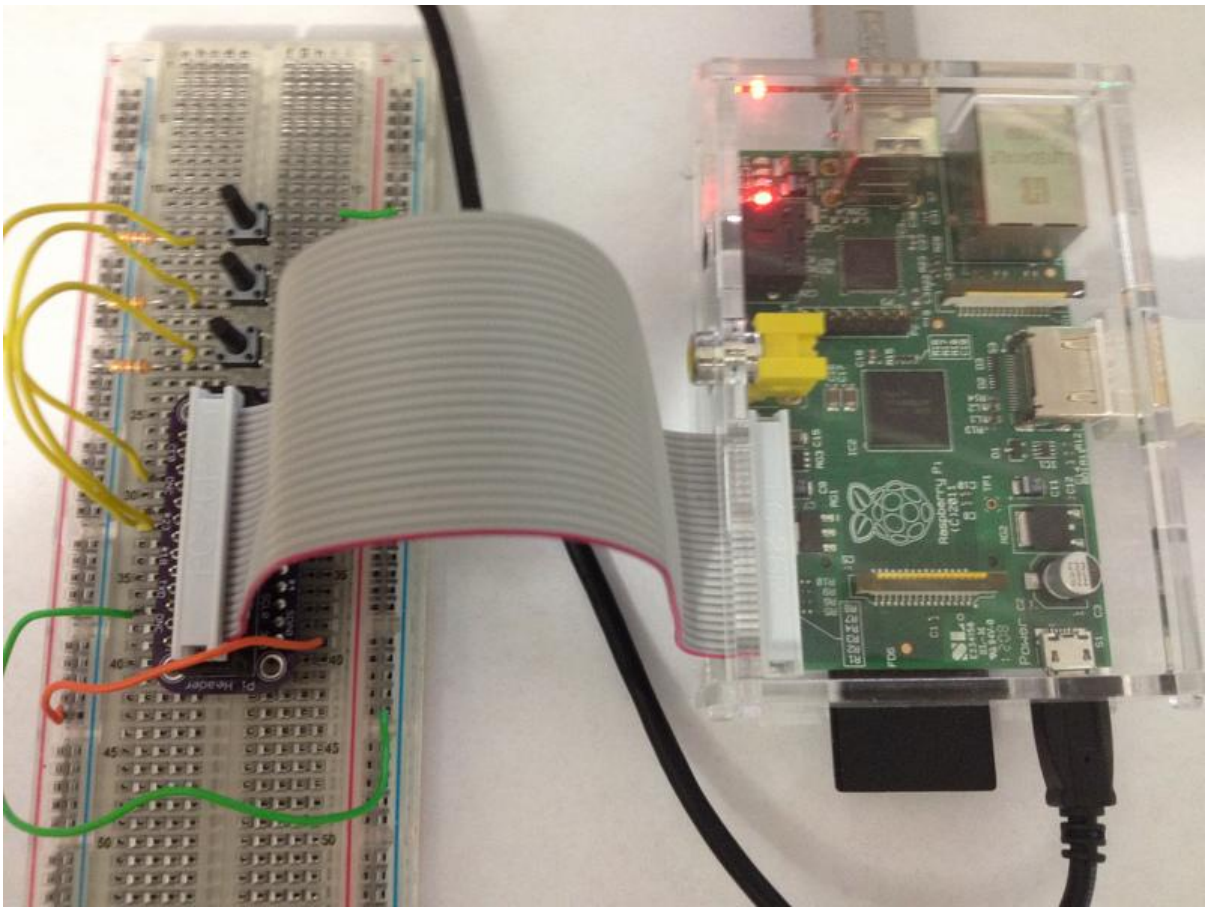
Playing sounds and using buttons with Raspberry Pi

Created by Michael Sklar

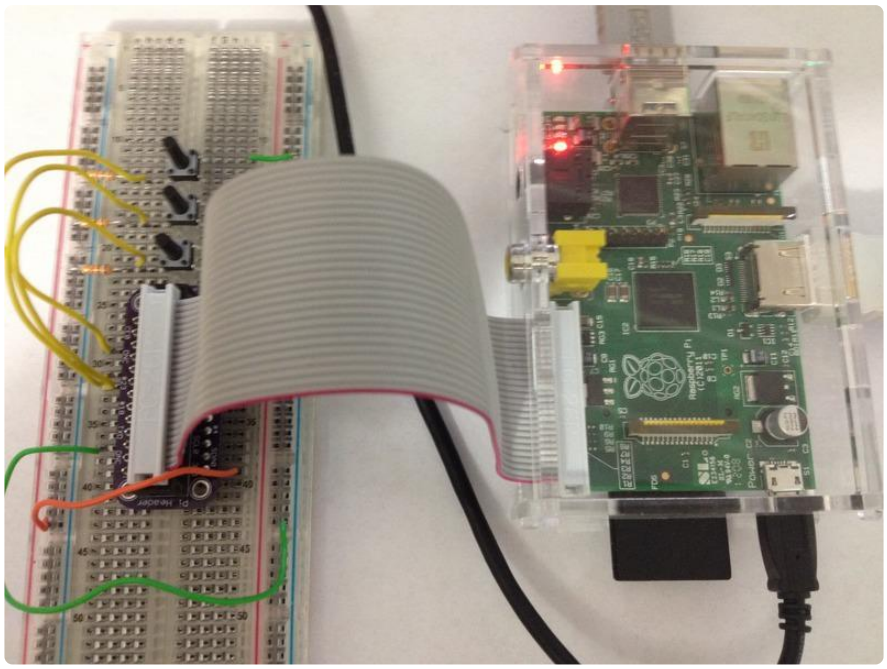


https://learn.adafruit.com/playing-sounds-and-using-buttons-with-raspberry-pi

Last updated on 2023-08-29 02:09:10 PM EDT

Table of Contents

Overview



This guide describes how to use CircuitPython on a Raspberry Pi to trigger audio file playback using tactile button presses. This tutorial works will all versions of Raspberry Pi hardware to date (v1, v2, v3, Zero, etc.) If you have not already used the Raspberry Pi as a input device this guide will show you how to wire up the buttons to the GPIO pins and access their state from a python script.

To Follow This Tutorial You Will Need

- A Raspberry Pi () (compatible with all 26pin and 40pin Pi releases to date)
- Pi T-Cobbler Plus, () Pi Cobbler Plus for Model B+ / Pi 2 (http://adafru.it/2029) or the original Pi Cobbler ()
- (1) Half size breadboard (http://adafru.it/64)
- Hook-up Wire ()
- (3) Tactile Button Switches ()
- (1) 5v 2.5A Switching Power Supply ()

Pi Prep

Update Your Pi to the Latest Raspbian

Your Pi will need to be running the latest version of Raspbian. This tutorial was written using Raspbian Stretch (Nov. 2018). Checkout our guide for Preparing an SD Card for your Raspberry Pi () if you have not done so already. After the installation is complete be sure and run the following commands to make sure your installation packages are up to date.

```
$ sudo apt-get update -y
$ sudo apt-get upgrade -y
```

Install pip3

pip3 is already installed with a full Raspbian installation, but the Raspbian Lite does not include pip3 so it needs to be installed as shown below.

```
$ sudo apt-get install python3-pip
```

Install adafruit-blinka

```
$ sudo pip3 install adafruit-blinka
```

Install CircuitPython for Raspberry Pi

Install adafruit_blinka

```
$ sudo pip3 install adafruit-blinka
```

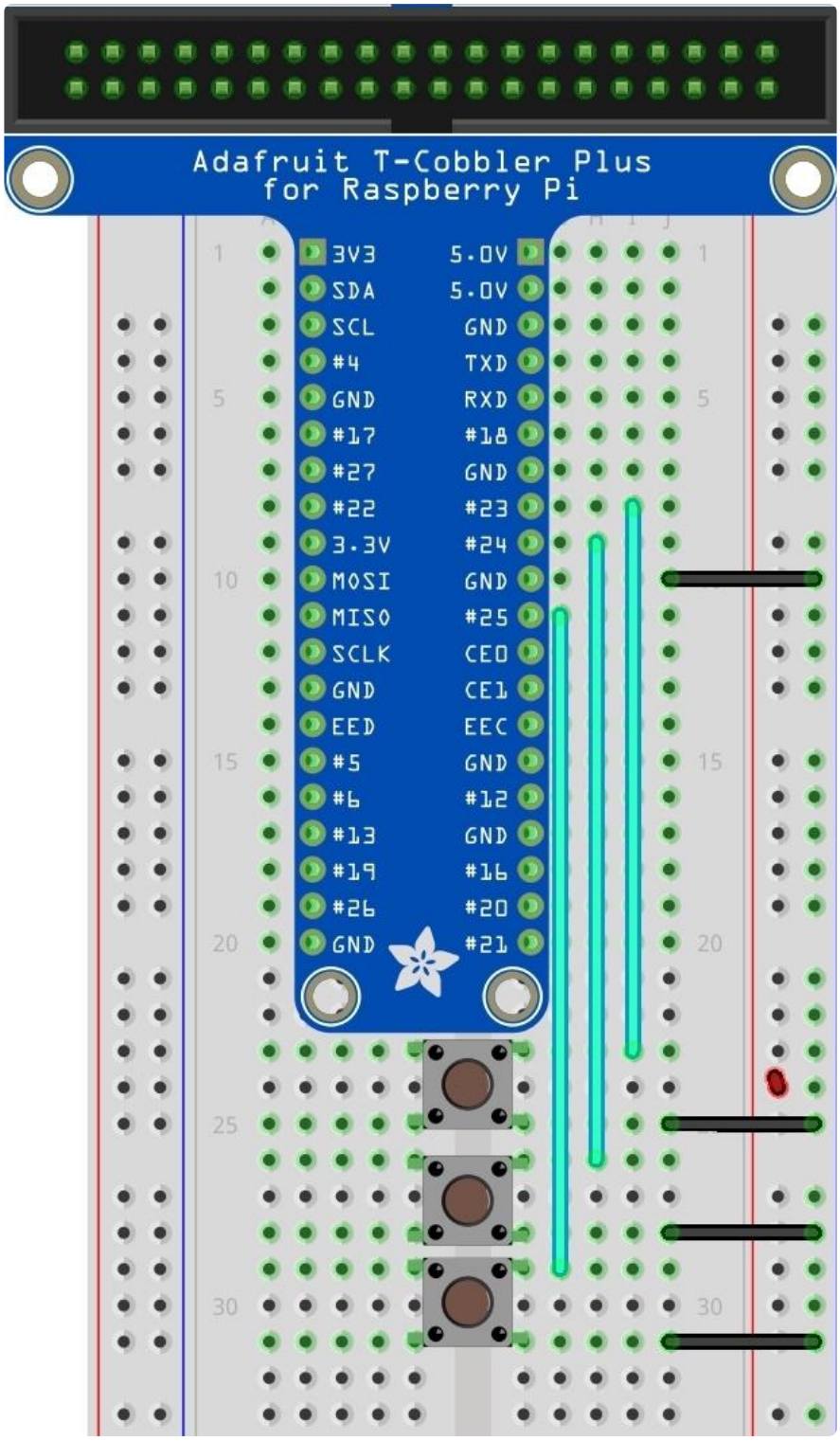
If you have installed adafruit-blinka in the past this would be a good time to see if any upgraded version is available.

```
$ pip3 install --upgrade adafruit_blinka
```

Circuit Diagram

All Raspberry Pi versions released to date can use the CircuitPython wiring below. We have used the same GPIO pins 23, 24 and 25 as they are available on Raspberry Pi v1, v2, v3 and Zeros. It does not matter if your cobbler is a 40-pin version or the earlier 26-pin version.

CircuitPython T-Cobbler



Note that we are not using pull up resistors and do not need to connect to the 3.3v rail. We can do direct pin to button and button to GND connections with CircuitPython configuration.

CircuitPython Code

Now for a bit of Python. There are two example files you can download to your Pi and execute with python3. The first is an script that plays a different audio bell when when each button is pressed. The second code example is a jukebox which allows one to navigate through the files, play a selected sample and stop playback using the same three buttons.

Be sure to download the following MP3 files so that you can hear the samples playing.

1. temple-bell.mp3 ()
2. temple-bell-bigger.mp3 ()
3. temple-bell-huge.mp3 ()

Pulling Down the Files on the Pi

We can use wget to pull down the audio files and code directly to the Pi we are working on.

```
$ wget https://github.com/adafruit/Adafruit_Learning_System_Guides/raw/master/
Playing_Sounds_and_Using_Buttons_with_Raspberry_Pi/temple-bell.mp3
$ wget https://github.com/adafruit/Adafruit_Learning_System_Guides/raw/master/
Playing_Sounds_and_Using_Buttons_with_Raspberry_Pi/temple-bell-bigger.mp3
$ wget https://github.com/adafruit/Adafruit_Learning_System_Guides/raw/master/
Playing_Sounds_and_Using_Buttons_with_Raspberry_Pi/temple-bell-huge.mp3
$ wget https://raw.githubusercontent.com/adafruit/Adafruit_Learning_System_Guides/
master/Playing_Sounds_and_Using_Buttons_with_Raspberry_Pi/audio-button.py
$ wget https://raw.githubusercontent.com/adafruit/Adafruit_Learning_System_Guides/
master/Playing_Sounds_and_Using_Buttons_with_Raspberry_Pi/simple-jukebox.py
```

Audio Button

```
# SPDX-FileCopyrightText: 2018 Mikey Sklar for Adafruit Industries
#
# SPDX-License-Identifier: MIT

# This script requires a Raspberry Pi 2, 3 or Zero. Circuit Python must
# be installed and it is strongly recommended that you use the latest
# release of Raspbian.

import time
import os
import board
import digitalio

print("press a button!")

button1 = digitalio.DigitalInOut(board.D23)
button1.direction = digitalio.Direction.INPUT
button1.pull = digitalio.Pull.UP
```

```
button2 = digitalio.DigitalInOut(board.D24)
button2.direction = digitalio.Direction.INPUT
button2.pull = digitalio.Pull.UP

button3 = digitalio.DigitalInOut(board.D25)
button3.direction = digitalio.Direction.INPUT
button3.pull = digitalio.Pull.UP

while True:

    # omxplayer -o local <file>
    # omxplayer -o hdmi <file>
    # omxplayer -o both <file>
    if not button1.value:
        os.system('omxplayer temple-bell.mp3 &')

    if not button2.value:
        os.system('omxplayer temple-bell-bigger.mp3 &')

    if not button3.value:
        os.system('omxplayer temple-bell-huge.mp3 &')

    time.sleep(.25)
```

Running the audio-button.py script as indicated below should result in each button playing a different bell tone.

```
sudo python3 ./audio-button.py
```

Simple Jukebox

```
    https://github.com/adafruit/Adafruit_Learning_System_Guides/blob/master/
Playing_Sounds_and_Using_Buttons_with_Raspberry_Pi/simple-jukebox.py
```

Running the simple-jukebox.py as indicated below should result in each button having a different function. Audio selection, playing of the selected file and stop playing the audio file.

```
$ sudo python3 ./simple-jukebox.py
```

Here, we get a list of mp3 files in the current directory, and then set up the following control scheme:

- button #1 (GPIO 23) clicks through the list of available mp3s
- button #2 (GPIO 24) plays the currently selected mp3
- button #3 (GPIO 25) runs a command that kills any existing mp3 player processes, stopping whatever mp3s are playing right now