Caja de música

Fecha de creación: marzo 2021

Contenido

Objetivo

Construir una caja de música que reproduzca diferentes sonidos cuando se presionen diferentes botones.

Introducción

La librería pygame permite reproducir sonidos usando el lenguaje de programación Python, con eso y la librería gpiozero que permite hacer uso de los GPIO se hace una caja musical que reproduce el sonido mediante auriculares o altavoces conectados al conector de 3.5mm.

Requerimientos

- Raspberry pi con Raspbian
- 1 protoboard
- 4 push buttons
- 4 jumpers macho-macho
- 5 jumpers macho-hembra
- Altavoces o auriculares

Preparar los archivos de audio

Para reproducir más fácilmente los audios en Python deben estar en formato wav, en Raspbian existen múltiples sonidos de muestra, pero se encuentran en formato flac por lo que deben convertirse primero.

1. Abrir una terminal manteniendo presionado ctrl+alt+t.

2. Desplazarse hasta el directorio del proyecto (para este ejemplo se usa el escritorio) con el comando cd y la ubicación:

```
cd Desktop
```

3. Crear una nueva carpeta con el comando mkdir seguido del nombre que se desea tenga la carpeta:

```
mkdir Caja_de_musica
```

4. Desplazarse hasta la nueva carpeta y crear dentro otra para los archivos de audio.

```
cd Caja_de_musica
mkdir Sonidos
```

Hay muchos sonidos de muestra almacenados en /usr/share/sonic-pi/samples por lo que se copian a la carpeta recién creada.

5. Desplazarse hasta ahí con:

```
cd /usr/share/sonic-pi/samples
```

6. Ver el contenido de la carpeta con el siguiente comando:

```
ls
```

7. Se copia el contenido a la carpeta de sonidos con:

```
cp * /home/pi/Desktop/Caja_de_musica/Sonidos
```

8. Ver los archivos recién copiados:

```
cd /home/pi/Desktop/Caja_de_musica/Sonidos
ls
```

9. Ahora, para convertir todos a .wav se usa el siguiente comando (toma algo de tiempo que depende del modelo de Raspberry que se use):

```
for f in *.flac; do ffmpeg -i "$f" "${f%.flac}.wav"; done
```

10. Y se remueven los .flac con:

```
rm *.flac
```

Diagrama

Para usar los push button se hacen las conexiones en la protoboard.

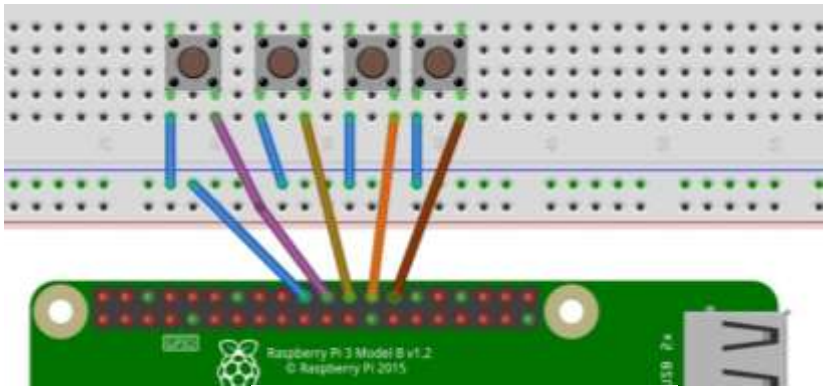

Figura 1 Diagrama

1. Colocar los 4 push button en la protoboard.
2. Conectar un jumper macho-hembra al pin GND de la raspberry y al bus de tierra de la protoboard.
3. Conectar con los jumpers macho-macho una terminal de cada push button al bus de tierra.
4. Usar los jumpers macho-hembra restantes para conectar las terminales libres de los push button a los pines GPIO25, GPIO8, GPIO7, GPIO1.

Programación en Python

1. Una vez listos los archivos de audio abrir el IDE de Python haciendo clic en Menú (Logotipo de Raspberry pi), después en Programación y finalmente en mu.

2. Escribir el siguiente código en el archivo Python para importar el módulo pygame e inicializarlo.

```
import pygame
pygame.init()
```

3. Elegir cuatro archivos de sonido que desee utilizar para el proyecto.
4. A continuación, crear un objeto Python que se vincule a uno de estos archivos de sonido, por ejemplo:

```
sonido =
pygame.mixer.Sound("/home/pi/Desktop/Caja_de_musica/Sonidos/bd_gas.wav")
```

5. Guardar y ejecutar el código. Luego, en el shell en la parte inferior del editor Mu, usar el nombre del objeto seguido del la función .play () para reproducir el archivo de sonido:

```
sonido.play()
```

6. Hacer el paso 4 para los 4 archivos elegidos en el paso 3.
7. Para hacer uso de los GPIO, importar el módulo gpiozero escribiendo lo siguiente después de la importación de pygame:

```
from gpiozero import Button
```

8. Configurar los pines GPIO (en este caso el 25) asignándoles un nombre con:

```
boton_sonido = Button(25)
```

9. Cuando se presiona el botón, el programa debe llamar a una función como Sonido.play (), sin embargo, cuando se usa un evento (como presionar un botón) para llamar a una función, no se usan paréntesis (), agregar la siguiente línea de código:

```
boton_sonido.when_pressed = sonido.play
```

10. Realizar los pasos 8 y 9 para los tres botones restantes.
El programa completo debe lucir algo así:

```
import pygame
from gpiozero import Button
pygame.init()

sonido =
pygame.mixer.Sound("/home/pi/Desktop/Caja_de_musica/Sonidos/bd_gas.wav")
bateria =
pygame.mixer.Sound("/home/pi/Desktop/Caja_de_musica/Sonidos/drum_splash_
soft.wav")
pedal =
pygame.mixer.Sound("/home/pi/Desktop/Caja_de_musica/Sonidos/drum_cymbal_
pedal.wav")
boom =
pygame.mixer.Sound("/home/pi/Desktop/Caja_de_musica/Sonidos/bd_boom.wav"
)
boton_sonido = Button(25)
boton_bateria = Button(8)
boton_pedal = Button(7)
boton_boom = Button(1)
```

```
boton_sonido.when_pressed = sonido.play
boton_bateria.when_pressed = bateria.play
boton_pedal.when_pressed = pedal.play
boton_boom.when_pressed = boom.play
```

11. Conectar los auriculares o altavoces, guardar y ejecutar el programa.
12. Presionar los botones al ritmo de la música.

Bibliografía

- GPIO music box. (2021). Recuperado el 17 de marzo de 2021, de https://projects.raspberrypi.org/en/projects/gpio-music-box