

# Keyboard

This uses simple non-latching switches to make a keyboard. When the key is pressed, the note starts. When it's released, the note stops.

## Before you start

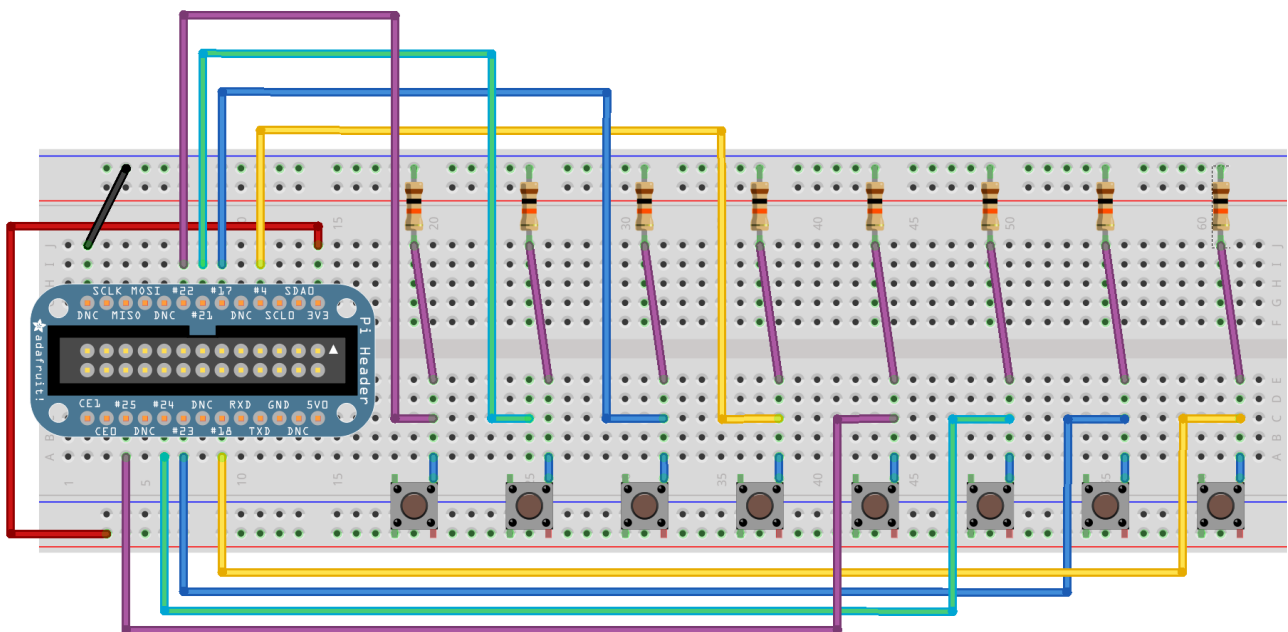
Set up the Raspberry Pi with PyGame and copy the sound files across.

## You will need

- Eight non-latching button switches
- Eight 10kΩ resistors
- A breadboard
- Some jumper leads for connecting things. You'll need mostly male-male, with some female-female to attach to the tilt switches.
- Speakers connected to the Pi's headphone jack to play the sounds.

Use either a Pi Cobbler or a Raspberry Leaf to help identify the pins. If you're using a Pi Cobbler, make sure that the coloured side of the ribbon is in the corner of the Pi. If you're not using a Pi Cobber, you'll need some extra female-female jumper leads to connect the Pi to the breadboard.

## Make this circuit



Made with Fritzing.org

Use pins 22, 21/27, 17, 4, 25, 24, 23, 18, GND, 3v3

## Enter this program

```
pi@blackberry:~$ cd pi-music
pi@blackberry:~/pi-music$ nano keyboard.py
Use nano to enter this code into keyboard.py
```

(Layout is important: use four spaces, not tabs, and make sure all the columns line up. Distinguish carefully between ( ) [ ] { } . , 0 O. Watch the case of letters: i ≠ I and s ≠ S)

```

import pygame
import RPi.GPIO as gpio

gpio.setmode(gpio.BCM)
if gpio.RPI_REVISION == 1:
    pins = [22, 21, 17, 4, 25, 24, 23, 18]
else:
    pins = [22, 27, 17, 4, 25, 24, 23, 18]

notes = ['sounds/keyboard-g.wav',
         'sounds/keyboard-a.wav',
         'sounds/keyboard-b.wav',
         'sounds/keyboard-c.wav',
         'sounds/keyboard-d.wav',
         'sounds/keyboard-e.wav',
         'sounds/keyboard-f.wav',
         'sounds/keyboard-g-high.wav']

pygame.mixer.init()

sounds = {}
for pin, wav in zip(pins, notes):
    sounds[pin] = pygame.mixer.Sound(wav)

def handle_sound(pin):
    if gpio.input(pin):
        sounds[pin].play()
    else:
        sounds[pin].stop()

for pin in pins:
    gpio.setup(pin, gpio.IN)
    gpio.add_event_detect(pin, gpio.BOTH, callback=handle_sound,
                          bouncetime=50)

while True:
    pass

```

## ***Play the keyboard***

Run with

```
pi@blackberry:~/pi-music$ sudo python keyboard.py
```