Lab - 11
Make a Buzzing Sound

Problem

You want to make a buzzing sound with the Raspberry Pi.

Solution

Use a piezo-electric buzzer connected to a GPIO pin.

Most small piezo buzzers work just fine using the arrangement shown in Figure. The one I used is an Adafruit-supplied component. You can connect the buzzer pins directly to the Raspberry Pi using female-to-female headers.

These buzzers use very little current. However, if you have a large buzzer or just want to play it safe, then put a 470Ω resistor between the GPIO pin and the buzzer lead.

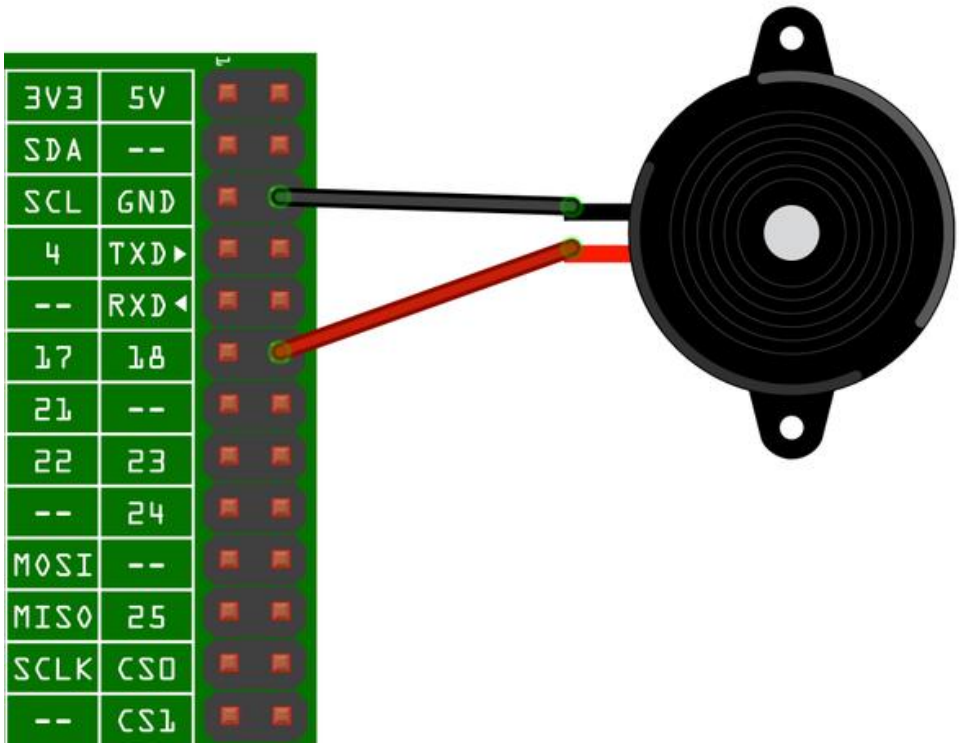

Figure. Connecting a piezo buzzer to a Raspberry Pi

Paste the following code into the IDLE or nano editors. Save the file as *buzzer.py*.

```
import RPi.GPIO as GPIO
import time

buzzer_pin = 18
GPIO.setmode(GPIO.BCM)
GPIO.setup(buzzer_pin, GPIO.OUT)

def buzz(pitch, duration):
    period = 1.0 / pitch
    delay = period / 2
    cycles = int(duration * pitch)
    for i in range(cycles):
        GPIO.output(buzzer_pin, True)
        time.sleep(delay)
        GPIO.output(buzzer_pin, False)
        time.sleep(delay)

while True:
    pitch = input("Enter Pitch (200 to 2000): ")
    duration = input("Enter Duration (seconds): ")
    buzz(pitch, duration)
```

When you run the program, it will first prompt you for the pitch in Hz and then the duration of the buzz in seconds:

```
$ sudo python buzzer.py
Enter Pitch (2000 to 10000): 2000
Enter Duration (seconds): 20
```

Discussion

Piezo buzzers don’t have a wide range of frequencies, nor is the sound quality remotely good. However, you can vary the pitch a little. The frequency generated by the code is very approximate.

The program works by simply toggling the GPIO pin 18 on and off with a short delay in between. The delay is calculated from the pitch. The higher the pitch (frequency), the shorter the delay needs to be.