Lesson 3 Passive Buzzer

Overview

In this lesson, we will learn how to program the Raspberry Pi to make a passive buzzer sound with different frequency.

Requirement

- 1* Raspberry Pi
- 1* Passive buzzer
- 1* 1 kΩ Resistor
- 1* NPN Transistor (S8050)
- 1* Breadboard
- Several Jumper wires

Principle

As long as you send the square wave signals to a passive buzzer with different frequency, then the passive buzzer will make different sound.

In this experiment, we continuously send different square wave signal to a passive buzzer to play a piece of music.

Key functions

C language user:

- **int softToneCreate (int pin)**

This creates a software controlled tone pin. You can use any GPIO pin and the pin numbering will be that of the wiringPiSetup() function you used.

The return value is 0 for success. Anything else and you should check the global errno variable to see what went wrong.

- **void softToneWrite (int pin, int freq)**

This updates the tone frequency value on the given pin. The tone will be played until you set the frequency to 0.

Python user:

- GPIO.cleanup( )

At the end any program, it is good practice to clean up any resources you might have used. This is no different with RPi.GPIO. By returning all channels you have used back to inputs with no pull up/down, you can avoid accidental damage to your RPi by shorting out the pins. Note that this will only clean up GPIO channels that your script has used. Note that GPIO.cleanup() also clears the pin numbering system in use.

- p = GPIO.PWM(channel, frequency)

To create a PWM instance

- p.start(dc)

To start PWM.

- p.ChangeFrequency(freq)

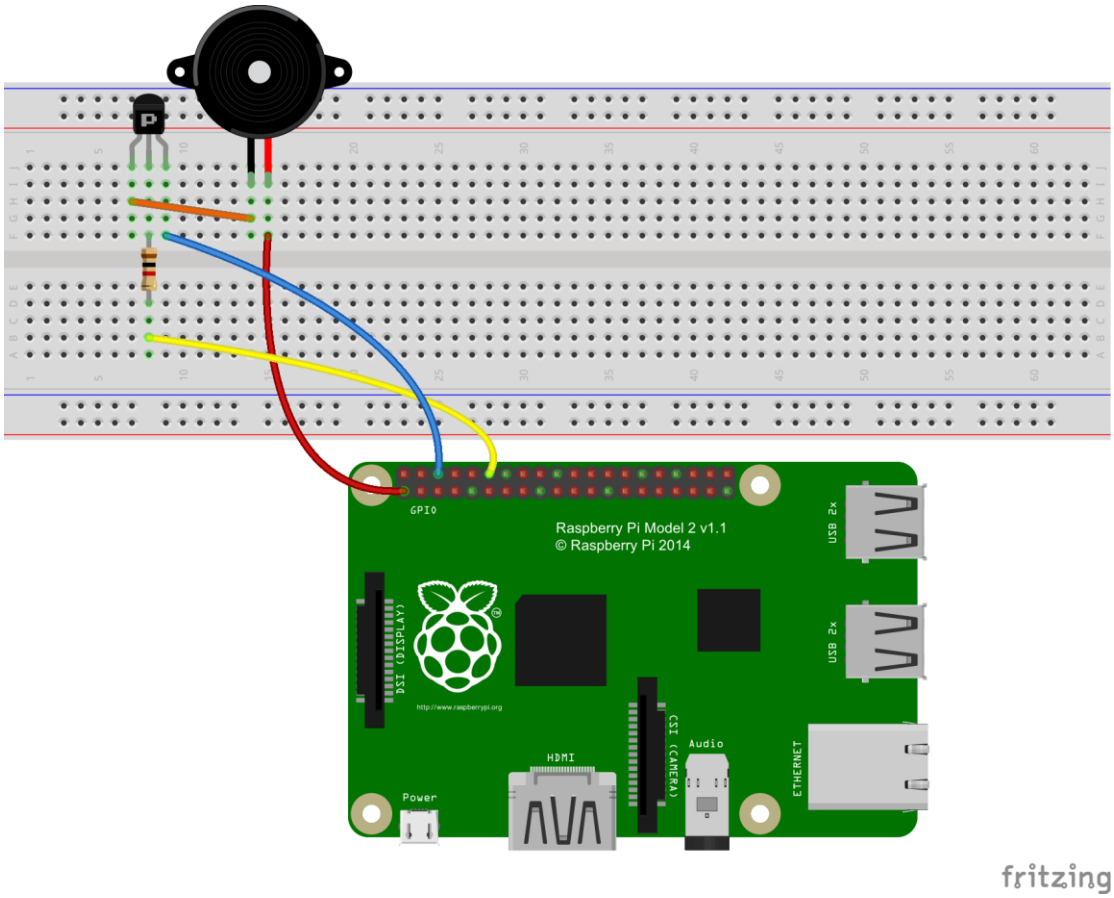
To change the frequency.

- p.stop()

To stop PWM.

Procedure

1. Build the circuit



2. Program

C user:

2.1 Edit and save the code with vim or nano.

(Code path: /home/Adeept_Ultimate_Starter_Kit_C_Code_for_RPi/03_passiveBuzzer/passiveBuzzer.c)

2.2 Compile the program

```
$ gcc passiveBuzzer.c -o passiveBuzzer -lwiringPi -lpthread
```

2.3 Run the program

```
$ sudo ./passiveBuzzer
```

Python user:

2.1 Edit and save the code with vim or nano.

(Code path: /home/Adeept_Ultimate_Starter_Kit_Python_Code_for_RPi/03_passiveBuzzer.py)

2.2 Run the program

```
$ sudo python 03_passiveBuzzer.py
```

Now, you should be able to hear the sound of the buzzer.