Speaker

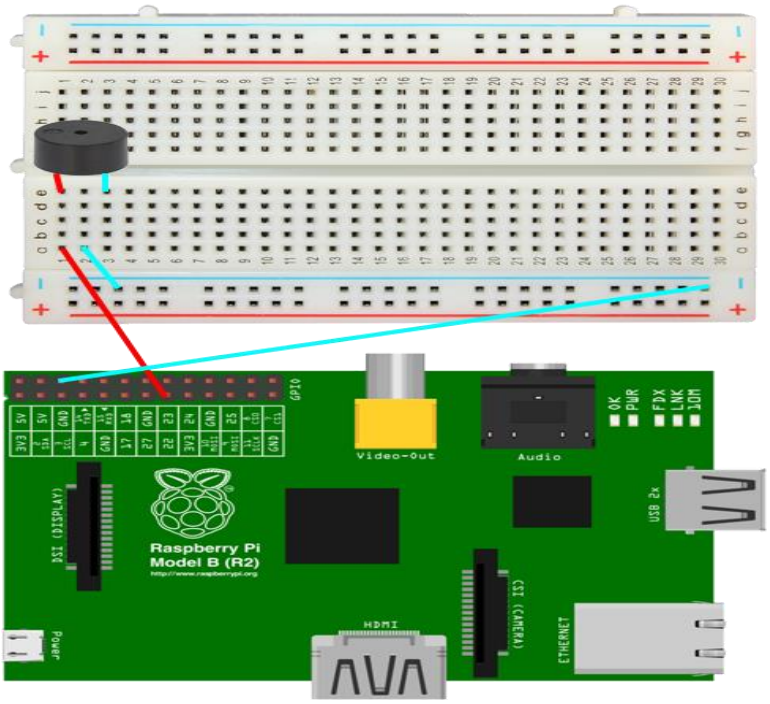
Introduction

This class allows you to send notes (as a frequency) to a small Piezo Buzzer (Speaker) connected to the Pi. A list of notes and their corresponding frequency is included in the class source code. This is a very basic class that we will build upon in later examples.

Assembly

Parts:

- Piezo Buzzer
- Jumper Cables
- Raspberry Pi



Build Instructions:

1. Power off the Pi completely.
2. Attach cable from Positive end of Speaker (typically the longer pin), to your chosen GPIO pin.
   In this example we are using Pin 3 (Wiring Pi) which translates to Pin 22 on the Pi. If you chose a different pin you must remember to initialise the class with the alternate constructor and specify the pin.
3. Connect the other Speaker pin to GND on the Pi.

Exercises

You should also have a class called **SpeakerExercises**, this contains blank or simple method for you to complete by doing the exercises below:

Exercise 1: After looking at the source of Speaker you should realise that the Pi4J library writes a frequency to the Speaker which then plays it. Complete the **ascendingPitch()** method, which should start at a low note, and get progressively higher.

Exercise 2: Try and combine with the Button class. So when a button is pressed the speaker makes a noise. To extend this, the speaker should turn off when the button is pressed again. Do this in **speakerWithButton()**.

Exercise 3: Plug in another speaker on a suitable pin. Create another speaker object and pass in the new pin number. Do something interesting! Our idea was to try and get the speakers to play alternate notes in an ascending/descending scale.

Notes

- Remember, Pi4J using something called WiringPi to manage GPIO pins. This means that the pin numbers do not actually correlate with what is written on the board. Use this website to convert: http://pi4j.com/pins/model-b-plus.html

- If you're finding this a little too easy check out our slightly more complex classes that build upon this. These are JukeBox and AlarmClock.