Demo 5: Play from WAV file
Exercises

DSP Lab (EE 4163 / EL 6183)

Fall 2017

1 Demo files

Several demo files are provided:

`play_wav_mono.py`
`play_wav_stereo.py`

The demo python program `play_wav_mono.py` shows how to read and play a wave file using PyAudio. It is assumed the wave file is mono (single channel). The program `play_wav_stereo` plays a stereo (two channel) wave file.

In this demo, we use a `while` loop to write signal values to the audio output. Inside the loop, we use `unpack()` and `pack()` to convert between binary strings and lists of integers. In the `while` loop, we use the variable `gain` to amplify the signal and the function `clip16()` to keep the value within the range of a signed 16-bit integer to avoid potential overflow run-time errors. Correspondingly the format string `h` (`hh` for stereo) is used in the `unpack()` and `pack()` methods to set the encoding format.

Documentation for the `wave` module is at

https://docs.python.org/2/library/wave.html
https://docs.python.org/3/library/wave.html

2 Exercises

1. **Single program for mono and stereo.** Write a single Python program to play both mono and stereo wave files. The program should determine the number of channels by reading the wave file information. SUBMIT

   Verify that your program can play both mono and stereo wave files encoded with 16-bits per sample.

2. Modify your previous program so it can be used at the command line like

   ```
   >> python my_play_wave.py filename.wav
   ```

You will need to import the `sys` module.

For example, consider the Python demo program `demo_sys.py`

```
1 # demo_sys.py
2
3 import sys
4
5 for i in range(0, len(sys.argv)):
6     print('Argument %d is %s ' % ( i,  sys.argv[i] ) )
```

then at the terminal command line we get:

```
>> python demo_sys.py abc.wav 10 h20
Argument 0 is demo_sys.py
Argument 1 is abc.wav
Argument 2 is 10
Argument 3 is h20
```

3. Write a single Python program that can play wave files with either 16 or 24 bits per sample.

   (a) Note that `h` is used to set the encoding scheme to signed 16-bit integer in `unpack()` and `pack()` methods. What letters should be used for signed 24-bit integer?

   (b) What is the allowed range of values for a signed 24-bit integer? Write your own clipping function to avoid run-time overflow errors while playing wave files of signed 24-bit integer.

   (c) Modify the Python demo program so it can play a mono wave file formatted with either signed 16-bit integer or signed 24-bit integer. The PyAudio function `pyaudio.get_format_from_width` might be useful here.