**Demo 5 #1**

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.

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

**Python Code: 05\_1.py**

import pyaudio

import wave

import struct

import math

def clip16( x ):

if x > 2\*\*15-1:

x = 2\*\*15-1

elif x < -2\*\*15:

x = -2\*\*15

else :

x = x

return (x)

gain = 1

wavfile = 'author.wav'

# wavfile = 'sin01\_stereo.wav'

print('Play the wave file %s.' % wavfile)

wf = wave.open( wavfile, 'rb' )

num\_channels = wf.getnchannels() # Number of channels

RATE = wf.getframerate() # Sampling rate (frames/second)

signal\_length = wf.getnframes() # Signal length

width = wf.getsampwidth() # Number of bytes per sample

print('The file has %d channel(s).' % num\_channels)

print('The frame rate is %d frames/second.' % RATE)

print('The file has %d frames.' % signal\_length)

print('There are %d bytes per sample.' % width)

p = pyaudio.PyAudio()

# Open audio stream

stream = p.open(

format = pyaudio.paInt16,

channels = num\_channels,

rate = RATE, #also change the frequency

input = False,

output = True )

if num\_channels == 1:

# Get first frame

input\_string = wf.readframes(1) #a sample from mono, 2 sample from stereo

while len(input\_string) > 0:

# Convert string to number

input\_tuple = struct.unpack('h', input\_string) # One-element tuple

input\_value = input\_tuple[0] # Number

# ?Compute output value

output\_value = int(clip16(gain \* input\_value)) # Integer in allowed range

#16 bits

# Convert output value to binary string

output\_string = struct.pack('h', output\_value)

# Write binary string to audio stream

stream.write(output\_string)

# Get next frame

input\_string = wf.readframes(1)

print('\* Finished')

elif num\_channels == 2:

# Read first frame

input\_string = wf.readframes(1)

while len(input\_string) > 0:

# Convert string to numbers

input\_tuple = struct.unpack('hh', input\_string) # produces a two-element tuple

# Compute output values

output\_value0 = int(clip16(gain \* input\_tuple[0]))

output\_value1 = int(clip16(gain \* input\_tuple[1]))

# Convert output value to binary string

output\_string = struct.pack('hh', output\_value0, output\_value1)

# Write output value to audio stream

stream.write(output\_string)

# Get next frame

input\_string = wf.readframes(1)

print('\* Finished')

stream.stop\_stream()

stream.close()

p.terminate()

**Comments:**

I wrote this code with if loop and determined the channels with getnchannels() so that the program could run either mono or stereo.

When the wave file was ‘author.wav’, “num\_channels = wf.getnchannels()” would get a value of 1, since it was mono. The program would go to “if num\_channels == 1” and run the loop with one output value.

When the wave file was 'sin01\_stereo.wav', “num\_channels = wf.getnchannels()” would get a value of 2, since it was stereo. The program would go to “elif num\_channels == 2” and run the loop with two output values.