Demo 8 #5

5. Modify the demo program so the input audio is from the microphone.

Python code : 08.py

```
import pyaudio
import wave
import struct
import math
from myfunctions import clip16

WIDTH = 2
CHANNELS = 1
RATE = 16000
DURATION = 10
N = DURATION * RATE

# Set parameters of delay system
Gdp = 1.0           # direct-path gain
Gff = 0.8           # feed-forward gain
delay_sec = 0.05    # 50 milliseconds
# delay_sec = 0.02
delay_samples = int( math.floor( RATE * delay_sec ) )

print('The delay of {0:.3f} seconds is {1:d} samples.'.format(delay_sec, delay_samples))

# Create a buffer to store past values. Initialize to zero.
BUFFER_LEN = delay_samples   # set the length of buffer
buffer = [ 0 for i in range(BUFFER_LEN) ]

# Open an output audio stream
p = pyaudio.PyAudio()
stream = p.open(format      = pyaudio.paInt16,
                channels    = 1,
                rate        = RATE,
                input       = True,
                output      = True )


# Get first frame (sample)
input_string = stream.read(1)

k = 0       # buffer index (circular index)
```

```
print("* Start *")
for n in range(0, N):
# while len(input_string) > 0:

   # Convert string to number
          input_value = struct.unpack('h', input_string)[0]
          # input_string = stream.read(1, exception_on_overflow = False)
          # input_tuple = struct.unpack('h', input_string)
          # input_value = input_tuple[0]
   # Compute output value
          output_value = Gdp * input_value + Gff * buffer[k]

   # Update buffer
          buffer[k] = input_value

   # Increment buffer index
          k = k + 1
          if k >= BUFFER_LEN:
      # The index has reached the end of the buffer. Circle the index back to the front.
                    k = 0

   # Convert output value to binary string
          output_string = struct.pack('h', int(clip16(output_value)))

   # Write output value to audio stream
          stream.write(output_string)

   # Get next frame (sample)
          input_string = stream.read(1)

print("* Finished *")

stream.stop_stream()
stream.close()
p.terminate()
```

Comments:

To make the input audio is from the microphone, we should make 'input = TRUE'. Also initialized WIDTH = 2, CHANNELS = 1, RATE = 16000, DURATION = 10,N = DURATION * RATE. I removed wave file and its properties, also got frames from stream instead file.

When I finished all of these operations and run the program. The program could not be stopped and I turned to TA for help. He told me the input_string could not be 0 so the program keep in the while loop. After that I add for loop to control the duration of 10 seconds. It worked well.