|  |  |
| --- | --- |
| **Ex. No. 2** | **Synchronized Animation and Audio** |
| Date of Exercise | 19-07-24 |

**Aim:**

To record and visualize audio waves.

**Algorithm**

1. Start
2. Import the required modules
3. Define the number of blocks, channels, format, and other parameters.
4. Open a stream and record audio using pyaudio.
5. Use matplotlib to plot the recorded audio.
6. Show the plot.
7. Stop

**Program**

1. **To display and visualize audio waves**

**Code:**

import numpy as np

import pyaudio

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

p = pyaudio.PyAudio()

info = p.get\_host\_api\_info\_by\_index(0)

numdevices = info.get('deviceCount')

for i in range(0, numdevices):

if p.get\_device\_info\_by\_host\_api\_device\_index(0, i).get('maxInputChannels') > 0:

print("Input Device id ", i, " - ", p.get\_device\_info\_by\_host\_api\_device\_index(0, i).get('name'))

SAMPLESIZE = 4096

SAMPLERATE = 44100

stream = p.open(format=pyaudio.paInt16,

channels=1,

rate=SAMPLERATE,

input=True,

frames\_per\_buffer=SAMPLESIZE)

plt.style.use('bmh')

fig, ax = plt.subplots()

x = np.arange(0, SAMPLESIZE)

line, = ax.plot(x, np.random.rand(SAMPLESIZE), lw=1)

ax.set\_xlim(0, SAMPLESIZE - 1)

ax.set\_ylim(-2\*\*15, 2\*\*15 - 1)

def init():

line.set\_ydata(np.ma.array(x, mask=True))

return line,

def animate(i):

try:

data = np.frombuffer(stream.read(SAMPLESIZE), dtype=np.int16)

line.set\_ydata(data)

except IOError as e:

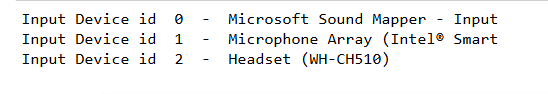
print("IOError:", e)

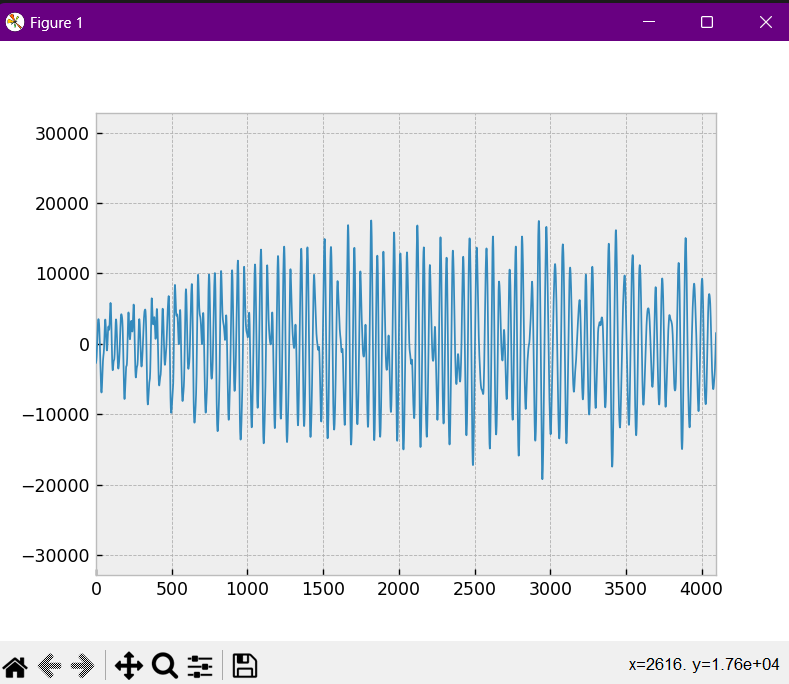
return line,

ani = FuncAnimation(fig, animate, init\_func=init, blit=True, interval=50, cache\_frame\_data=False)

plt.show()

**Output**

****



**Result**

The program to record and visualize audio waves has been successfully built and verified.