ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

доц., к.т.н., доц.		О. О. Жаринов
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

ОСНОВЫ МОДЕЛИРОВАНИЯ И ОБРАБОТКИ АУДИОСИГНАЛОВ СРЕДСТВАМИ PYTHON

по курсу: МУЛЬТИМЕДИА ТЕХНОЛОГИИ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4116		Стонтов Е. О.
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2025

Цель работы

Получить представления о принципах формирования аудиосигналов и приобрести навыки работы со звуковыми файлами с использованием Python.

Вариант – 4

3 синусоидальных импульса, длительностью 0.5 сек каждый, с одинаковой частотой и с нулевыми паузами между ними, равными по длительности 0.5 сек, синхронно в обоих каналах.

Теоретические сведения

Мультимедиа — это технология, объединяющая различные формы представления информации, такие как текст, изображения, видео, анимация и звук, в единую интерактивную систему [1]. Аудиоконтент в мультимедиа играет ключевую роль, обеспечивая звуковое сопровождение, которое может включать музыку, речь, звуковые эффекты и другие аудиофрагменты. Также аудиоконтент может быть представлен отдельно, например - подкасты, аудиореклама, аудиокниги, музыка и др..

Основные задачи по обработке мультимедиа аудиоконтента:

- Воспроизведение - процесс передачи аудиосигнала от источника к слушателю, обеспечивающий восприятие звука;
- Редактирование - изменение и улучшение звуковых записей с помощью различных инструментов и программного обеспечения. Этот процесс может включать в себя обрезку, добавление эффектов, изменение громкости и многое другое.
- Визуализация - представление аудиосигнала в графической форме, что позволяет анализировать его структуру и характеристики.
- Мастеринг - финальный этап обработки аудиоматериала, включающий оптимизацию громкости, эквализацию и применение других техник для подготовки записи к распространению на различных платформах [2].
- И другие.

Ход выполнения работы

Была разработана программа, генерирующая звуковой сигнал,

состоящий из 3 импульсов, длительностью 0.5 сек. каждый с частотой сигнала 440 Гц (нота Ля), частотой дискретизации 44100 Гц и амплитудой сигнала – 0.5. Пауза между импульсами составляет 0.5 сек.. После генерации сигнала результат сохраняется в файл с расширением .wav. Код программы представлен в листинге 1.

Листинг 1 – Код программы для генерации и сохранения звукового сигнала

```
import numpy as np
from scipy.io.wavfile import write

fs = 44100            # Частота дискретизации
pulse_duration = 0.5  # Длительность одного импульса в секундах
pause_duration = 0.5  # Длительность паузы между импульсами
f = 440               # Частота сигнала (нота Ля)
amplitude = 0.5       # Амплитуда сигнала

# Создаем временную ось для одного импульса.
# np.linspace создаёт массив значений от 0 до pulse_duration с нужным
числом сэмплов.
t_pulse = np.linspace(0, pulse_duration, int(fs * pulse_duration),
endpoint=False)

# Вычисляем значения синусоиды: A*sin(2*pi*f*t)
pulse = amplitude * np.sin(2 * np.pi * f * t_pulse)

# Количество сэмплов, соответствующих длительности паузы.
num_samples_pause = int(fs * pause_duration)
# Массив из нулей для паузы.
pause = np.zeros(num_samples_pause)

# Объединяем массивы с сигналами и паузами
signal_mono = np.concatenate((pulse, pause, pulse, pause, pulse))
# Создаем стереосигнал
signal_stereo = np.vstack((signal_mono, signal_mono))

# Нормировка для привода к формату int16
signal_int16 = np.int16(signal_stereo / np.max(np.abs(signal_stereo))
* 32767)

# Записываем сигнал в WAV-файл
output_filename = 'signal.wav'
write(output_filename, fs, np.transpose(signal_int16))
```

Была разработана программа, которая открывает аудиофайл и строит графики:

- Визуализации сигнала для первых 0.05 сек. сигнала - рисунок 1;
- Визуализации сигнала для всего файла - рисунок 2;

- Амплитудного спектра - рисунок 3;
- Спектрограмму аудиосигнала.

  Код программы представлен в листинге 2.

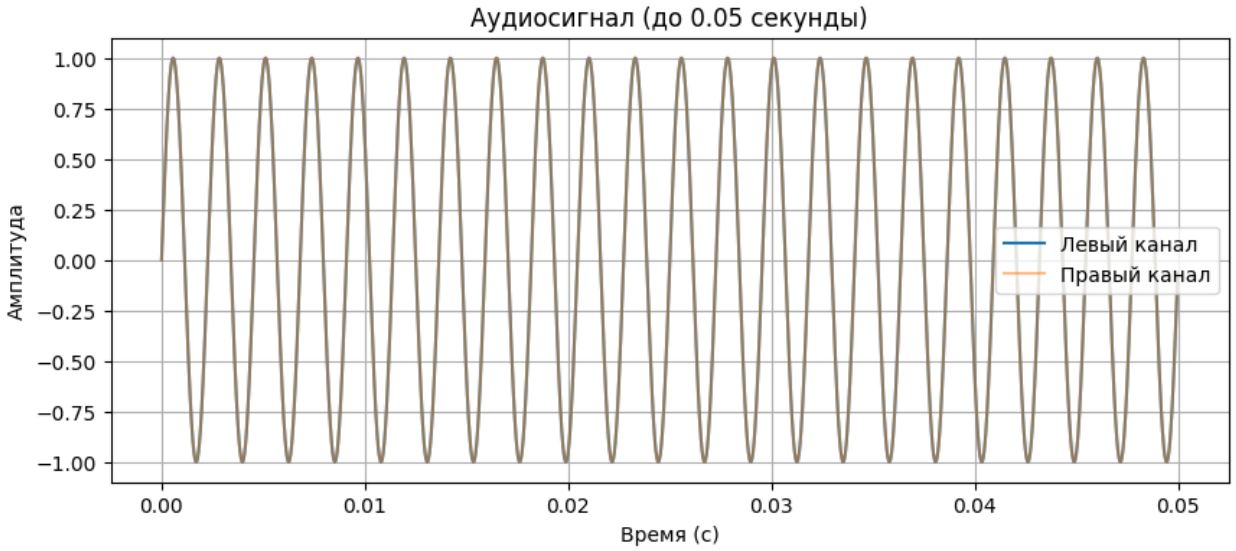

Рисунок 1 – Визуализация сигнала для первых 0.05 сек. сигнала

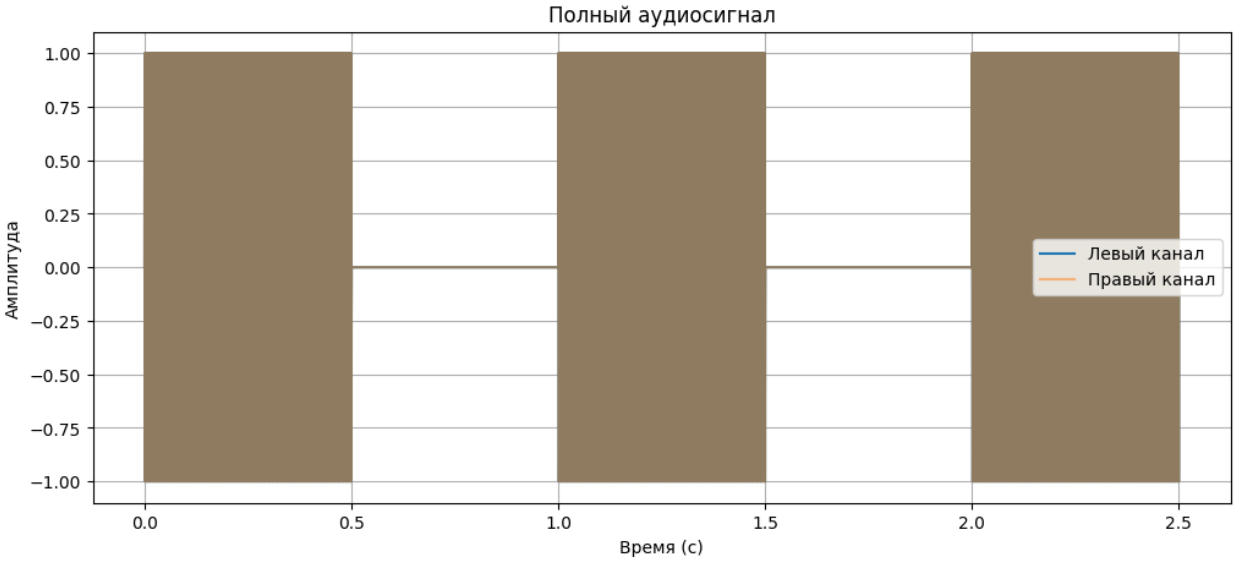

Рисунок 2 – Визуализация сигнала для всего сигнала

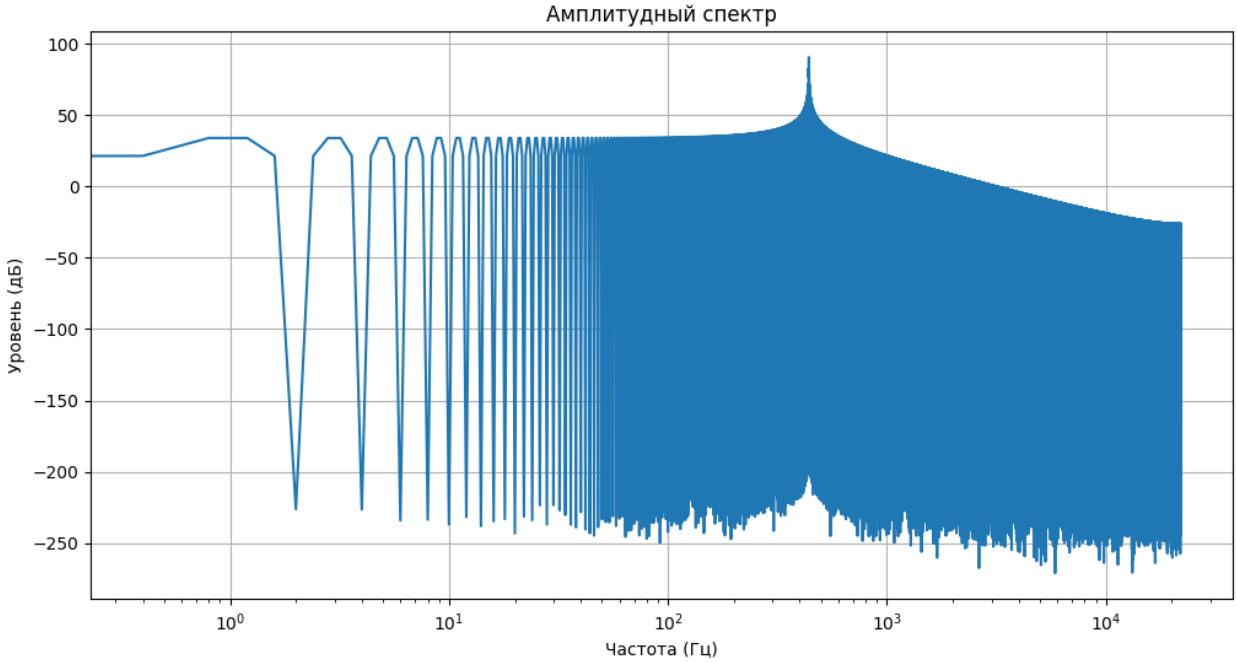

Рисунок 3 – Амплитудный спектр сигнала

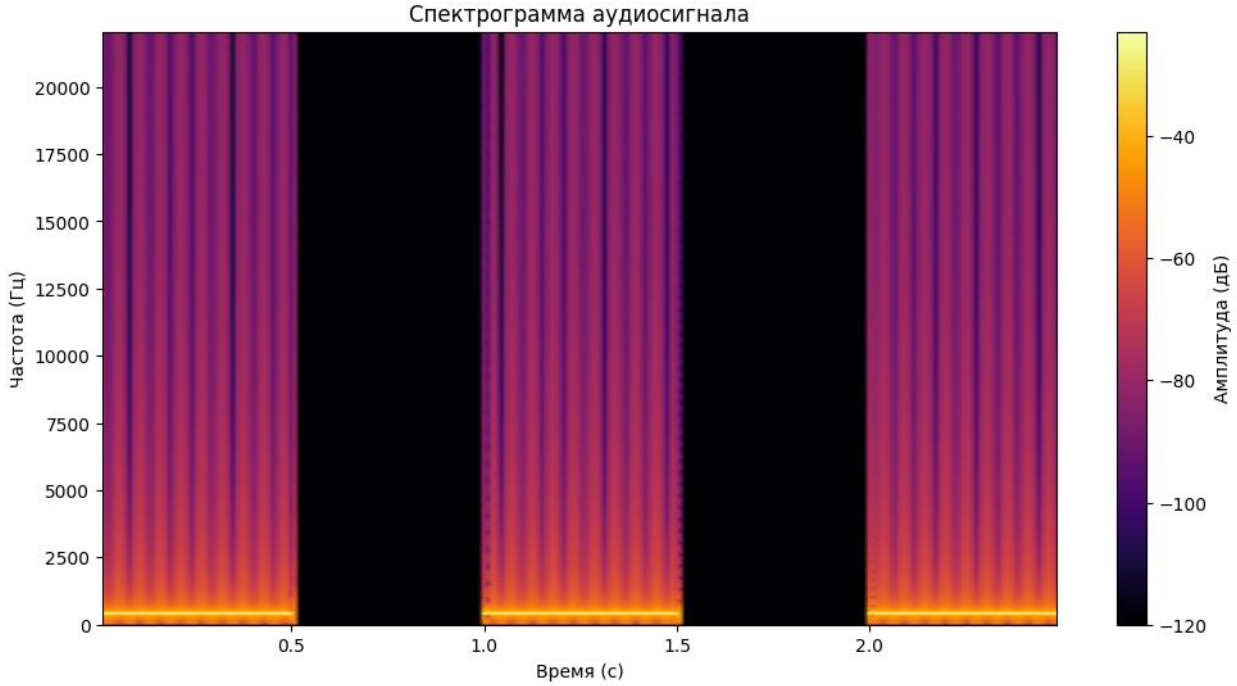

Рисунок 4 – Спектрограмма аудиосигнала

Листинг 2 – Код программы для построения графиков

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.io import wavfile
from scipy.signal import spectrogram, gaussian
import sys

# Загрузка аудиофайла
sample_rate, data = wavfile.read('signal.wav')
data = data / np.max(np.abs(data))
```

```
# Извлечение каналов
left_channel = data[:, 0]
right_channel = data[:, 1]

# Создание моно-сигнала для спектра и спектрограммы
mono_signal = right_channel

# 1. График аудиосигнала до 0.05 секунды
n_samples = int(0.05 * sample_rate)
time_short = np.arange(n_samples) / sample_rate

plt.figure(figsize=(10, 4))
plt.plot(time_short, left_channel[:n_samples], label='Левый канал')
plt.plot(time_short, right_channel[:n_samples], label='Правый канал',
alpha=0.5)
plt.xlabel('Время (с)')
plt.ylabel('Амплитуда')
plt.title('Аудиосигнал (до 0.05 секунды)')
plt.legend()
plt.grid()
plt.show()

# 2. График аудиосигнала для всей записи
duration = len(left_channel) / sample_rate  # Общая длительность
time_full = np.linspace(0, duration, len(left_channel),
endpoint=False)

plt.figure(figsize=(12, 5))
plt.plot(time_full, left_channel, label='Левый канал')
plt.plot(time_full, right_channel, label='Правый канал', alpha=0.5)
plt.xlabel('Время (с)')
plt.ylabel('Амплитуда')
plt.title('Полный аудиосигнал')
plt.legend()
plt.grid()
plt.show()

# 3. График амплитудного спектра
fft_spectrum = np.fft.fft(mono_signal)
magnitude_spectrum = 20 * np.log10(np.abs(fft_spectrum))
freqs = np.fft.fftfreq(len(fft_spectrum), 1/sample_rate)

plt.figure(figsize=(12, 6))
plt.plot(freqs[:len(freqs)//2], magnitude_spectrum[:len(freqs)//2])
plt.xlabel('Частота (Гц)')
plt.ylabel('Уровень (дБ)')
plt.title('Амплитудный спектр')
plt.grid()
plt.semilogx()
plt.show()

# 4. Построение спектрограммы
g_std = 0.2 * sample_rate  # стандартное отклонение для окна Гаусса
(20% от частоты дискретизации)
window = gaussian(1024, std=g_std)  # Генерация гауссового окна
```

```
f, t, Sxx = spectrogram(mono_signal, fs=sample_rate, window=window,
nperseg=1024, noverlap=512, scaling='density', mode='magnitude')

plt.figure(figsize=(12, 6))
plt.pcolormesh(t, f, 20 * np.log10(Sxx + 1e-6), shading='gouraud',
cmap='inferno')
plt.colorbar(label='Амплитуда (дБ)')
plt.xlabel('Время (с)')
plt.ylabel('Частота (Гц)')
plt.title('Спектрограмма аудиосигнала')
plt.ylim(0, sample_rate // 2)
plt.show()
```

К сожалению, использование кода для построения спектрограммы из методических указаний не позволило построить наглядную спектрограмму, поэтому были внесены модификации, а именно:

- Была использована функция scipy.signal.spectrogram, которая является готовой оберткой для STFT, что упрощает код;
- Размер окна – 1024 сэмпла (в моем репозитории GitHub по ссылке - представлен результат использования окна размером 2*g_std, аналогично методическим указаниям, в файле spectr_incorrect.png);
- Параметр nperseg=1024 в функции spectrogram – длина сегмента;
- Параметр noverlap=512 в функции spectrogram – перекрытие сегментов (50% от длины сегмента) [3].

Как можно заметить из рисунков 1 и 2, 4, аудиосигнал соответствует условиям, приведенным в задании по варианту – аудиосигнал состоит из трех 3 синусоидальных импульсов.

На рисунке 3 можно наблюдать «пик» сигнала на частоте между 100 и 1000 герцами, что соответствует заданной частоте в 440 Гц.

На рисунке 4 можно заметить наибольшую мощность сигнала около 500 Гц, что соответствует заданию по варианту, также видны некоторые артефакты (узкие черные линии).

Также работа данной программы была проверена на звуковом эффекте «Задувание свечи» [4]. Результаты представлены на рисунках 5, 6, 7.

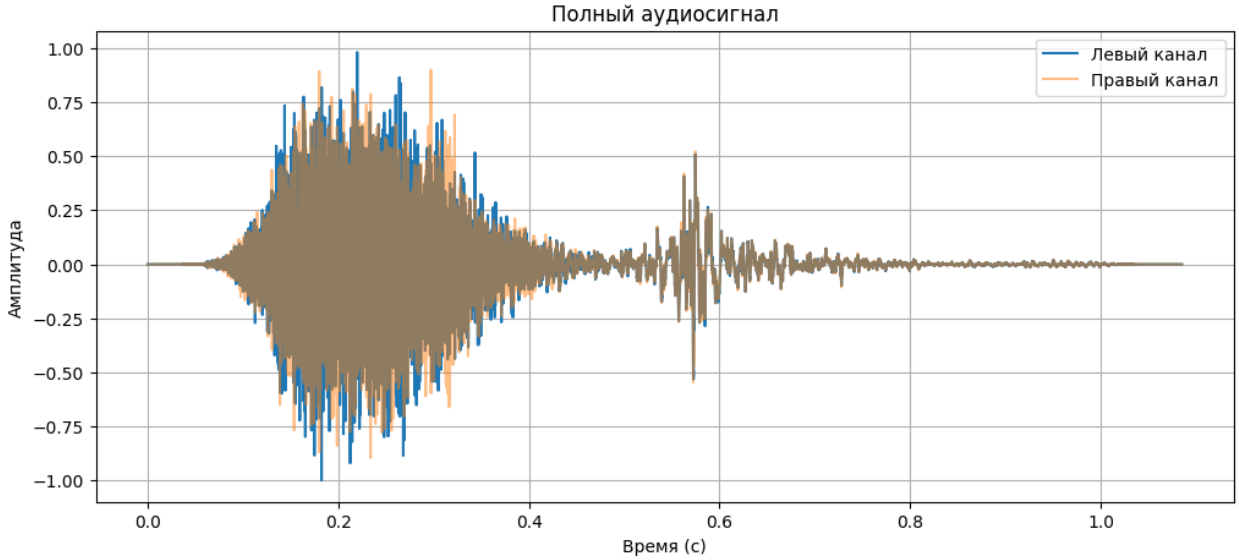

Рисунок 5 – Визуализация сигнала для звукового эффекта «Задувание свечи»

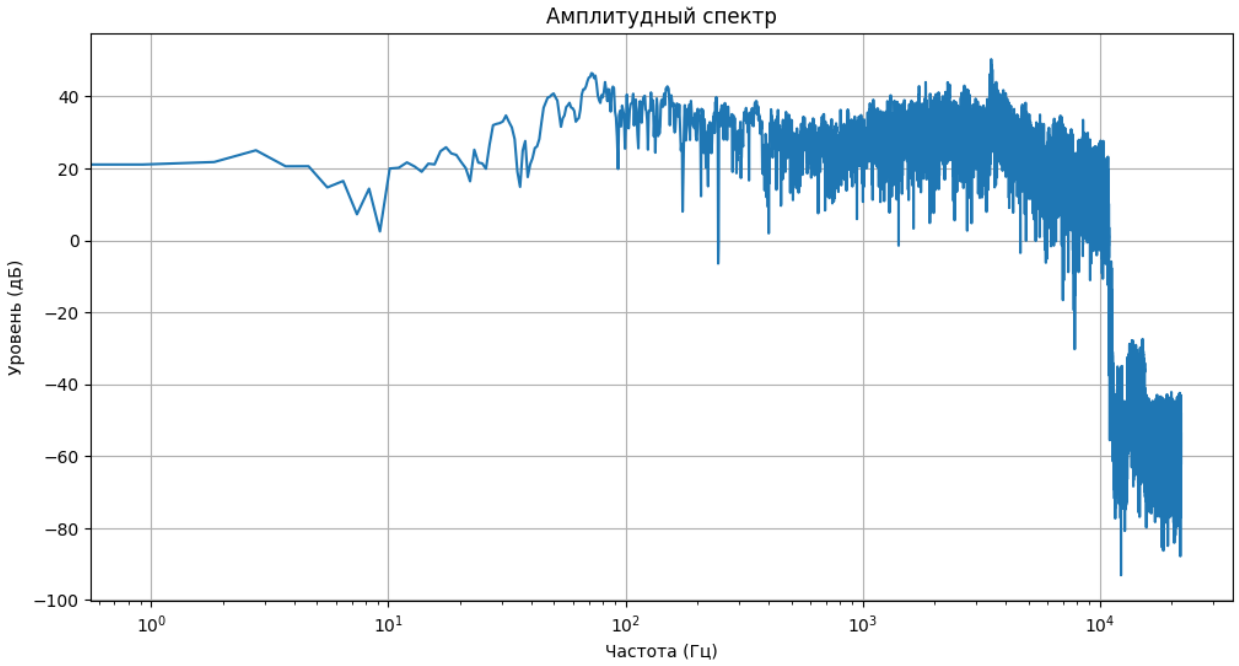

Рисунок 6 – Амплитудный спектр для звукового эффекта «Задувание свечи»

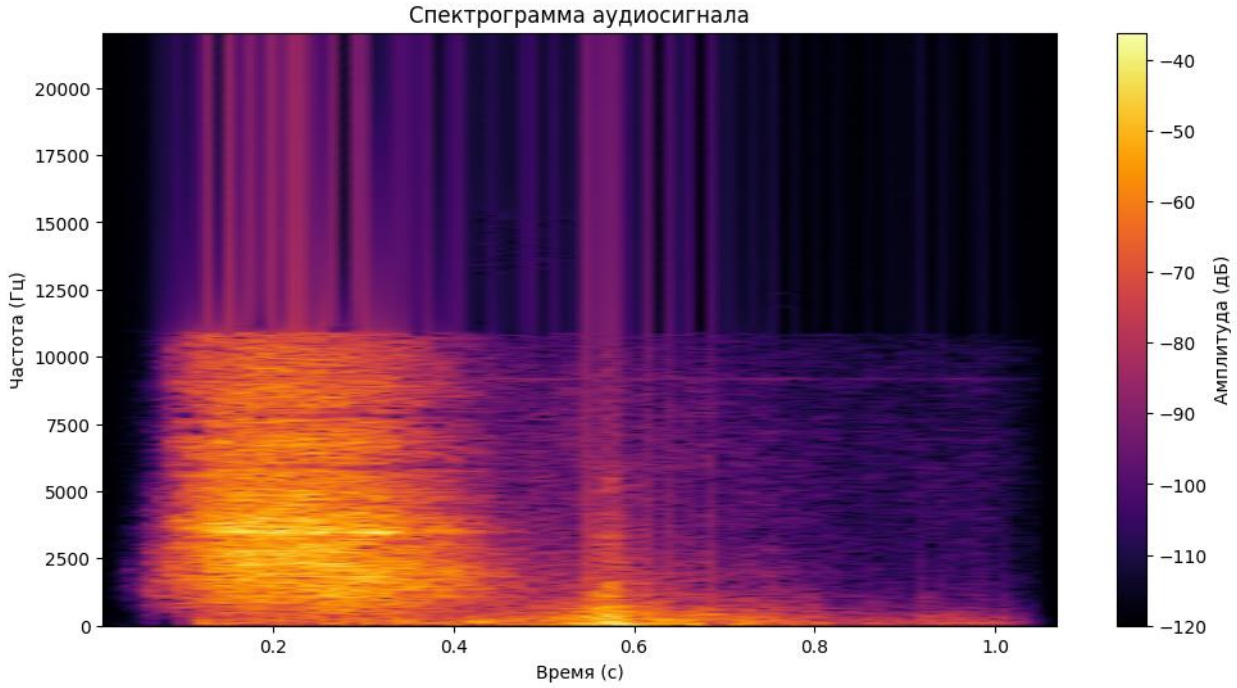

Рисунок 7 – Спектрограмма аудиосигнала

Вывод

В ходе лабораторной работы были получены навыки по работе с аудиофайлами на языке программирования Python, а именно – генерация аудиосигналов с определенными параметрами, визуализации аудиосигнала, построения амплитудного спектра сигнала и построение спектрограммы аудиосигнала.

Полученные графики соответствуют заданным параметрам сигнала (3 синусоидальных импульса, а также «пик» на амплитудном спектре на заданной частоте и наибольшая мощность сигнала на заданной частоте на спектрограмме).

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Мультимедиа // Википедия URL: https://ru.wikipedia.org/wiki/Мультимедиа (дата обращения: 10.02.2025).

2. Мастеринг: что это такое, чем отличается от сведения, в какой программе делать // SkillBox Media URL: https://skillbox.ru/media/cinemusic/mastering/ (дата обращения: 10.02.2025).

3. scipy.signal.spectrogram // SciPy URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.spectrogram.html (дата обращения: 17.02.2025).

4. Звуки свечи // ZVUKOGRAM URL: https://zvukogram.com/category/zvuki-svechi/ (дата обращения: 10.02.2025).