

<p style="text-align: center;">Urządzenia peryferyjne</p> <p style="text-align: center;">Laboratorium 5 – Karta dźwiękowa</p>	
<p>Michał Lewandowski, Dominik Kilijan</p> <p style="text-align: center;">Grupa F</p>	<p>Czwartek 17:30 – 20:30 TNP</p> <p style="text-align: center;">07.12.2023</p>

1 Wstęp teoretyczny

1.1 Zasady zapisywania dźwięku, technologie, parametry zapisu.

1.1.1 Zasady zapisywania dźwięku:

A. *Mikrofon:*

- Wybór odpowiedniego mikrofonu jest kluczowy. Różne rodzaje mikrofonów (dynamiczne, elektrostatyczne, pojemnościowe) mają różne charakterystyki i są lepsze do różnych zastosowań.

B. *Akustyka:*

- Warunki akustyczne pomieszczenia, w którym odbywa się nagrywanie, mają wpływ na jakość dźwięku. Izolacja od hałasów zewnętrznych i kontrola echa są istotne.

C. *Poziom sygnału:*

- Unikaj zbyt niskiego lub zbyt wysokiego poziomu sygnału. Nadmierny clipping (przesterowanie) lub zbyt mały sygnał mogą zniekształcić dźwięk.

D. *Przygotowanie sprzętu:*

- Upewnij się, że wszystkie urządzenia są prawidłowo podłączone i skonfigurowane. Sprawdź kable, mikrofony i inne urządzenia przed rozpoczęciem nagrywania.

E. *Format pliku:*

- Wybierz odpowiedni format pliku audio, takie jak WAV, AIFF, MP3 itp. Format zależy od zastosowania (np. profesjonalne studio, internetowe udostępnianie).

1.1.2 Technologie zapisu dźwięku:

A. *Analogowy vs. Cyfrowy:*

- Nagrywanie dźwięku może odbywać się na nośnikach analogowych (np. taśmy magnetofonowe) lub cyfrowych (np. pliki audio na komputerze). Współcześnie dominuje zapis cyfrowy.

B. Interfejs audio:

- Interfejsy audio przekształcają sygnał analogowy na cyfrowy i vice versa. Popularne interfejsy to USB, FireWire, Thunderbolt.

C. Konwertery Analogowo-Cyfrowe (ADC) i Cyfrowo-Analogowe (DAC):

- ADC zamienia sygnał analogowy na cyfrowy podczas nagrywania, a DAC dokonuje odwrotnej konwersji podczas odtwarzania.

1.1.3 Parametry zapisu dźwięku:

A. Częstotliwość próbkowania:

- Określa, ile próbek dźwięku jest pobieranych na sekundę. Standardowa wartość to 44,1 kHz, ale można spotkać także 48 kHz, 96 kHz itp.

B. Głębokość bitowa:

- Określa liczbę bitów przypisanych każdej próbce dźwięku. Wyższa głębokość bitowa oznacza lepszą jakość dźwięku, ale również większy rozmiar pliku.

C. Liczba kanałów:

- Określa, ile niezależnych ścieżek dźwiękowych jest nagrywanych. Standardowe wartości to mono (1 kanał) i stereo (2 kanały).

D. Kompresja dźwięku:

- Kompresja może zmniejszyć rozmiar pliku, ale może także wpłynąć na jakość dźwięku. Formaty kompresji, takie jak MP3, są powszechnie stosowane, ale dla profesjonalnych nagrań bezstratna kompresja jest preferowana.

Zasady te mogą się różnić w zależności od konkretnego zastosowania i rodzaju nagrania. W profesjonalnym studio nagraniowym stosuje się często bardziej zaawansowane technologie i parametry.

1.2 Środowiska programistyczne wspierające zapis, przetwarzanie i odtwarzanie dźwięku

DirectSound – część pakietu Microsoft DirectX umożliwiająca szybki dostęp do karty dźwiękowej i m.in. odtwarzanie i przechwytywanie dźwięków z urządzeń wejściowych. Obsługuje filtry i efekty dźwiękowe (jak np. echo, zniekształcenia, pogłos). Ta część DirectX działa tylko pod systemem Windows

ActiveX - zbiór gotowych kontrolki do obsługi między innymi audio. Został użyty w technologii WindowsForms w oparciu o język C#. ActiveX pozwala na przekazywanie danych pomiędzy różnymi aplikacjami działającymi pod kontrolą systemów operacyjnych Windows(np. Windows Media Player).

1.3 Formaty zapisu informacji dźwiękowej do zbioru (WAV, MP3, M4A, PCM itd.).

1.3.1 WAV (Waveform Audio File Format):

- **Charakterystyka:** Format bezstratny, zachowuje pełną jakość dźwięku.
- **Zastosowanie:** Często używany w profesjonalnych studio nagraniowych ze względu na wysoką jakość, ale pliki są większe niż w przypadku niektórych formatów kompresji.

1.3.2 MP3 (MPEG Audio Layer III):

- **Charakterystyka:** Format stratny, kompresuje pliki dźwiękowe usuwając pewne informacje, co powoduje utratę jakości, ale znacznie zmniejsza rozmiar pliku.
- **Zastosowanie:** Powszechnie stosowany w internecie do udostępniania muzyki, podcastów, plików dźwiękowych ze względu na niskie rozmiary plików.

1.3.3 M4A (MPEG-4 Audio):

- **Charakterystyka:** Format stosowany do kompresji dźwięku w standardzie MPEG-4, często używany z kodekiem AAC (Advanced Audio Coding).
- **Zastosowanie:** Popularny w sklepach internetowych z muzyką, w aplikacjach Apple, na urządzeniach iPod.

1.3.4 PCM (Pulse Code Modulation):

- **Charakterystyka:** Format bezstratny, przechowuje surowe dane audio w postaci cyfrowej.
- **Zastosowanie:** Często stosowany w kontekście nagrań bezstratnych, często używany w formie plików WAV.

1.3.5 FLAC (Free Lossless Audio Codec):

- **Charakterystyka:** Format bezstratny, kompresuje pliki bez utraty jakości w porównaniu do WAV.
- **Zastosowanie:** Popularny wśród audiofilów i entuzjastów muzyki, ponieważ oferuje wysoką jakość dźwięku przy mniejszym rozmiarze pliku niż WAV.

1.3.6 OGG (Ogg Vorbis):

- **Charakterystyka:** Format otwartego źródła, stosowany do kompresji dźwięku z minimalną utratą jakości.
- **Zastosowanie:** Popularny w grach wideo, na platformach streamingowych i w aplikacjach internetowych.

1.3.7 AAC (Advanced Audio Coding):

- **Charakterystyka:** Format stratny, stosowany w standardzie MPEG-4, oferuje lepszą jakość dźwięku niż MP3 przy podobnych rozmiarach plików.
- **Zastosowanie:** Powszechnie stosowany w plikach M4A, wideo HD, transmisjach strumieniowych.

2 Przebieg laboratoriów

Celem laboratoriów było napisanie programu, który będzie umożliwiał odtworzenie dźwięku w formatach wav oraz mp3, odtworzenie dźwięku z efektem echa. Wyświetlenie parametrów pliku audio w formatach wav oraz mp3 i nagranie własnego dźwięku za pomocą mikrofonu a następnie zapisanie go na komputerze.

W pierwszej kolejności pobraliśmy i sprawdziliśmy czy pliki audio są działające a następnie zabraliśmy się do napisania programu.

3 Kod programu

```
import wave # utworzenie i wyświetlenie Nagłówka pliku wav
import pyaudio # Rejestracja dźwięku z mikrofonu
import sounddevice as sd # Odtwarzanie pliku wav
import numpy as np # Odtwarzanie pliku wav
from scipy.signal import convolve # utworzenie echa
import librosa # Nagłówek pliku mp3
import pygame # odtwarzanie pliku wav w formacie directsound

pygame.options['audio'] = ('directsound', 'silent', 'openal', 'pulse',
                           'xaudio2')

import pygame.media # DirectSound

def play_wav(file_path):
    # Odczytaj plik WAV
    with wave.open(file_path, 'rb') as wav_file:
        # Pobierz dane audio
        frames = wav_file.readframes(wav_file.getnframes())
        audio_data = np.frombuffer(frames, dtype=np.int16)

        # Odtwórz dźwięk
        sd.play(audio_data, wav_file.getframerate())
        sd.wait()

def add_echo(audio_data, delay, decay=0.5):
```

```

    # Generuj odpowiedź impulsową dla efektu echa
    impulse_response = np.zeros(delay)
    impulse_response[0] = 1.0
    impulse_response[-1] = decay

    # Konwolucja danych audio z odpowiedzią impulsową
    echoed_data = convolve(audio_data, impulse_response, mode='full')

    # Znormalizuj dane echa, aby uniknąć przeciążeń
    echoed_data = np.int16(echoed_data / np.max(np.abs(echoed_data)) * 32767)

    return echoed_data

def play_wav_with_echo(file_path, delay=44100, decay=0.5):
    # Odczytaj plik WAV
    with wave.open(file_path, 'rb') as wav_file:
        # Pobierz dane audio
        frames = wav_file.readframes(wav_file.getnframes())
        audio_data = np.frombuffer(frames, dtype=np.int16)

        # Dodaj efekt echo do danych audio
        echoed_data = add_echo(audio_data, delay, decay)

        # Odtwórz dźwięk z efektem echo
        sd.play(echoed_data, wav_file.getframerate())
        sd.wait()

def display_mp3_header(file_path):
    # Wczytaj plik MP3
    audio, sr = librosa.load(file_path)

    # Pobierz parametry
    channels = audio.shape[0] if len(audio.shape) > 1 else 1
    sample_width = audio.dtype.itemsize
    frame_rate = sr
    frame_width = audio.shape[1] * sample_width if len(audio.shape) > 1 else
sample_width
    length_in_seconds = librosa.get_duration(y=audio, sr=sr)

    # Wyświetl parametry
    print(f"Kanały: {channels}")
    print(f"Szerokość próbki: {sample_width}")
    print(f"Częstotliwość ramki: {frame_rate} Hz")
    print(f"Szerokość ramki: {frame_width} bajtów")
    print(f"Długość: {length_in_seconds} sekundy")

```

```

def play_directsound(file_path):
    audio_file = pygame.media.load(file_path) # Podmień ścieżkę do pliku
    audio

    # Utwórz odtwarzacz mediów
    player = pygame.media.Player()
    player.queue(audio_file)

    # Odtwórz dźwięk
    player.play()

    # Utrzymuj program w działaniu, aby umożliwić odtwarzanie dźwięku
    pygame.app.run()

def display_wav_header(file_path):
    with wave.open(file_path, 'rb') as wav_file:
        print("Nagłówek pliku WAV:")
        print("Format: {}".format(wav_file.getparams()))
        print("Kanały: {}".format(wav_file.getnchannels()))
        print("Szerokość próbki: {} bajtów".format(wav_file.getsampwidth()))
        print("Częstotliwość ramki: {} Hz".format(wav_file.getframerate()))
        print("Liczba klatek: {}".format(wav_file.getnframes()))
        print("Typ kompresji: {}".format(wav_file.getcompname()))

def register_sound_microphone(record_time_seconds, file_name, selected_format,
selected_num_of_channels, selected_rate,
                                selected_frames_per_buffer):
    paudio = pyaudio.PyAudio()

    stream = paudio.open(

        format=selected_format,
        channels=selected_num_of_channels,
        rate=selected_rate,
        input=True,
        frames_per_buffer=selected_frames_per_buffer

    )

    print("Rozpoczęcie nagrywania")

    frames = []
    for i in range(0, int(selected_rate / selected_frames_per_buffer *
record_time_seconds)):
        data = stream.read(selected_frames_per_buffer)
        frames.append(data)

```

```

print("Koniec nagrywania")

stream.stop_stream()
stream.close()
paudio.terminate()

wav_object = wave.open(file_name, 'wb')
wav_object.setnchannels(selected_num_of_channels)
wav_object.setsampwidth(paudio.get_sample_size(selected_format))
wav_object.setframerate(selected_rate)
wav_object.writeframes(b"".join(frames))
wav_object.close()

def display_menu():
    print("==== Menu Przetwarzania Dźwięku =====")
    print("1. Odtwórz plik WAV")
    print("2. Odtwórz DirectSound")
    print("3. Odtwórz dźwięk z echem")
    print("4. Wyświetl nagłówek MP3")
    print("5. Wyświetl nagłówek WAV")
    print("6. Nagrywaj z mikrofonu")
    print("7. Wyjście")
    print("=====")

    while True:
        choice = input("Wprowadź wybór (1-7): ")

        if choice == "1":
            file_in_name = input("Wprowadź nazwę pliku wejściowego audio: ")
            play_wav(file_in_name)
        elif choice == "2":
            file_in_name = input("Wprowadź nazwę pliku wejściowego audio: ")
            play_directsound(file_in_name)
        elif choice == "3":
            file_in_name = input("Wprowadź nazwę pliku wejściowego audio: ")
            play_wav_with_echo(file_in_name, delay=44100, decay=0.5)
        elif choice == "4":
            file_in_name = input("Wprowadź nazwę pliku wejściowego audio: ")
            # display_mp3_header(file_in_name)
        elif choice == "5":
            while True:
                choice = input("Wprowadź wybór (1-7): ")

                if choice == "1":
                    file_in_name = input("Wprowadź nazwę pliku wejściowego audio: ")
                    play_wav(file_in_name)
                elif choice == "2":
                    file_in_name = input("Wprowadź nazwę pliku wejściowego audio: ")

```

```

        play_directsound(file_in_name)
    elif choice == "3":
        file_in_name = input("Wprowadź nazwę pliku wejściowego audio: ")
        play_wav_with_echo(file_in_name, delay=44100, decay=0.5)
    elif choice == "4":
        file_in_name = input("Wprowadź nazwę pliku wejściowego audio: ")
        display_mp3_header(file_in_name)
    elif choice == "5":
        file_in_name = input("Wprowadź nazwę pliku wejściowego audio: ")
        display_wav_header(file_in_name)
    elif choice == "6":
        selected_record_time_s = float(input("Wprowadź czas nagrywania w
sekundach: "))
        selected_file_out_name = input("Wprowadź nazwę pliku wyjściowego:
")
        selected_file_format = pyaudio.paInt16
        selected_num_of_channels = int(input("Wprowadź liczbę kanałów: "))
        selected_rate = int(input("Wprowadź częstotliwość próbkowania: "))
        selected_frames_per_buffer = int(input("Wprowadź liczbę ramek na
bufor: "))
        register_sound_microphone(selected_record_time_s,
selected_file_out_name, selected_file_format,
                                selected_num_of_channels, selected_rate,
selected_frames_per_buffer)
    elif choice == "7":
        print("Zamykanie programu.")
        break
    else:
        print("Nieprawidłowy wybór. Proszę wprowadzić liczbę między 1 a
7.")

```

4 Wnioski i podsumowanie

Kod po laboratoriach został przetłumaczony na polski i zostały dodane komentarze.

Dzięki laboratoriom nauczyliśmy się korzystać z bibliotek – Wave, Sounddevice, pyAudio, Pyglet. Zdobyliśmy praktyczne doświadczenie w obsłudze danych audio – odczytywaniu, odtwarzaniu i nagrywaniu plików WAV, nauczyliśmy się pracować z różnymi parametrami audio, takimi jak częstotliwość próbkowania, liczba kanałów, liczba klatek na bufor. Zrozumienie tych pojęć było kluczowe do napisania programu a nam zapewniło praktyczne doświadczenie w dziedzinie cyfrowego przetwarzania dźwięku.

5 Spis literatury

1. <https://docs.python.org/3/library/wave.html>
2. <https://people.csail.mit.edu/hubert/pyaudio/>
3. <https://python-sounddevice.readthedocs.io/en/0.4.6/usage.html#playback>
4. https://www.w3schools.com/python/numpy/numpy_intro.asp
5. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.convolve.html>
6. <https://librosa.org/doc/latest/index.html>
7. https://pyglet.readthedocs.io/en/latest/programming_guide/media.html#guide-audio-driver-order