



Vyšší odborná škola
a Střední průmyslová škola
elektrotechnická
Plzeň, Koterovská 85

ROČNÍKOVÁ PRÁCE S OBHAJOBOU

Téma: Synchronizace Světla a Hudby
Frekvenční Analýzou

Autor práce: Marek Vácal
Třída: 3.M
Vedoucí práce: Ing. Pavel JEDLIČKA
Dne: NAPSAT DATUM

Hodnocení:



**Vyšší odborná škola a
Střední průmyslová škola elektrotechnická Plzeň,
Koterovská 85**

ZADÁNÍ ROČNÍKOVÉ PRÁCE	
Školní rok	2023/ 2024
Studijní obor	78-42-M/01 Technické lyceum
Jméno a příjmení	Marek Vácal
Třída	3.M
Předmět	Kybernetika
Hodnoceno v předmětu	Kybernetika
Téma	Synchronizace Světla a Hudby Frekvenční Analýzou
Obsah práce	<ul style="list-style-type: none">• návrh a realizace hardwaru• software a řízení osvětlení• prozkoumání možností získávání zdrojových dat• frekvenční analýza• převod frekvence na barevné světlo
Zadávací učitel Příjmení, jméno	Ing. Pavel Jedlička
Podpis zadávajícího učitele	
Termín odevzdání	30. dubna 2024

Anotace

Jednalo by se světlo, u kterého by se dalo přepínat z normálního osvětlení na barevné LED, které by svítily podle hudby. Ovládání zařízení by bylo připojeno přes wifi, a to by dostávalo signály, z přístroje, který by zpracovával zvuk do formy, která by určovala barvu a intenzitu světla.

„Prohlašuji, že jsem tuto práci vypracoval samostatně a použil literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.“

V Plzni dne:

Podpis:

Obsah

1	Úvod	1
2	Plán fungování	1
3	Datová analýza	1
3.1	Sběr dat	1
3.2	Rychlá Fourierova transformace	4
3.3	propojení rfft s barvou	4
3.4	propojení rfft se silou	4
4	Překážky	4

1 Úvod

Jako vášnivý fanoušek fantasy a her se pravidelně setkávám s přáteli k hraní Dračího doupěte. Tato desková hra vás vtáhne do světa, kde jeden vypráví příběh a ostatní ho prožívají prostřednictvím svých postav. V naší skupině často využíváme hudební doprovod, který pomáhá vytvářet atmosféru scén a bojů, jež se odehrávají v příběhu.

Napadlo mě, že bychom mohli ještě více prohloubit atmosféru pomocí technologie. Chtěl bych vytvořit systém, který by dokázal převádět zvuk v reálném čase na barevný signál. Tento signál by se následně přenášel do LED pásky, která by se podle něj rozsvěcovala a pomáhala tak vizuálně ilustrovat události a emoce probíhající ve hře. Tím by se naše hraní stalo ještě více ponořujícím zážitkem.

2 Plán fungování

Program, který potřebuji, je napsán v programovacím jazyce Python, který jsem si vybral, protože s ním mám nejvíce zkušeností.

Zvuk bude nahráván skrze mikrofon a bude převáděn do potřebné formy. Poté na něm bude provedena Fourierova transformace. Podle toho jaké frekvence se ve zvukou budou nacházet se přiřadí dané barva. Dále se bude určovat intenzita, které bude podle toho jak silná je daná frekvence.

To vše se bude zpracovávat nejspíše na raspberry pi, které potom bude rozsvěcet LED pásek určitou barvou a intenzitou.

3 Datová analýza

3.1 Sběr dat

Využívaná data jsou sbírána skrze mikrofon, který je umístěn těsně u reproduktoru. Důvod, proč data nejsou brána přímo z prohlížeče je, protože prohlížeče mají ochranu proti kradení dat, aby nemohli být zneužity.

Pro práci s audio daty využívám python knihovnu pyaudio jak je možno vidět na obrázku 1. Na obrázku 2 je využití knihovny pro získání dat z mikrofonu. Můžeme zde vidět proměnou `p`, které dáváme datový typ `pyaudio.PyAudio`. Dva řádky níže je proměná `stream`, která nabírá data z `p`, která díky jejímu datovému typu je schopna brát data z mikrofonu (`input=True`). Zbytek udává další vlastnosti zvukových dat, podle proměných uvedených výše, a které budeme dále využívat.

Sbíraná data jsem si zobrazil pomocí knihovny `matplotlib`, jak je vidět na obrázku 3 a 4. Napřed je ale nutné převést data na čísla, abychom je mohly vykreslovat. To je vidět na obrázku 5, kde se v loopu čtou data z proměné `stream` a pak se dávají do proměné `dataInt` už jako čísla, která se poté vkládají jako velikost na ose `y`.

```

1 import numpy as np
2 import pyaudio as pa
3 import struct
4 import matplotlib.pyplot as plt
5 from scipy.fft import rfft

```

Obrázek 1: Knihovny

```

7 FORMAT = pa.paInt16
8 CHANNELS = 1
9 RATE = 44100 # in Hz
10 CHUNK = 1024
11
12 p = pa.PyAudio()
13
14 stream = p.open(
15     format = FORMAT,
16     channels = CHANNELS,
17     rate = RATE,
18     input=True,
19     output=True,
20     frames_per_buffer=CHUNK
21 )

```

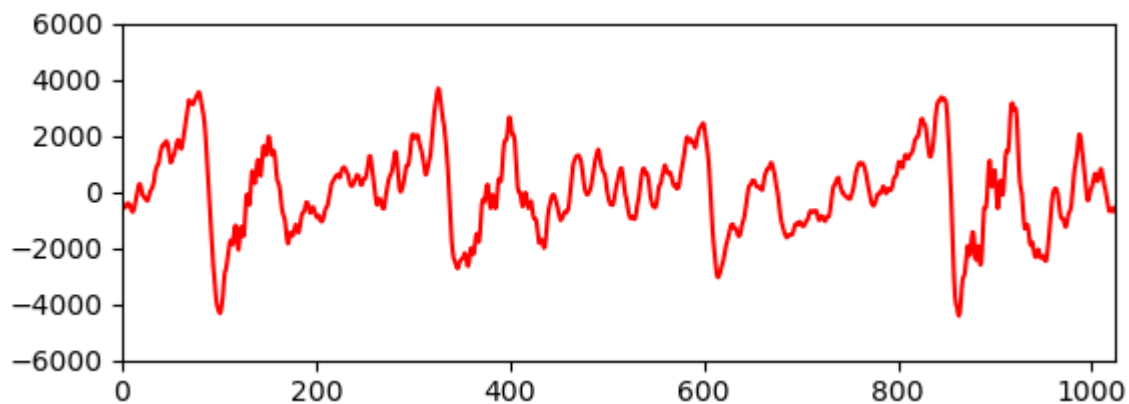
Obrázek 2: Sběr dat z mikrofonu

```

23 fig, [ax1, ax2] = plt.subplots(nrows=2, ncols=1)
24 x = np.arange(0, CHUNK, 1)
25 line, = ax1.plot(x, np.zeros(CHUNK), 'r')
26 ax1.set_ylim(-6000, 6000)
27 ax1.set_xlim(0, CHUNK)
28 fig.show()

```

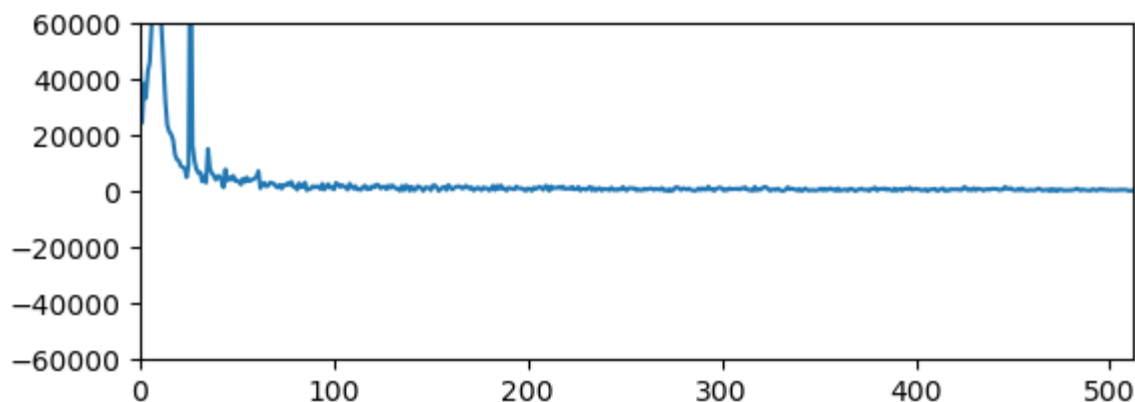
Obrázek 3: Kód na zobrazení zvuku



Obrázek 4: Zobrazení dat z mikrofону

```
37 while True:
38     data = stream.read(CHUNK)
39     dataInt = struct.unpack(str(CHUNK) + 'h', data)
40     line.set_ydata(dataInt)
41
42     fig.canvas.draw()
43     fig.canvas.flush_events()
```

Obrázek 5: Čtení a úprava dat



Obrázek 6: Zobrazení dat z Fourierovi transformace

```

42 fig.canvas.draw()
43 fig.canvas.flush_events()
44
45 real_time_rfft = rfft(dataInt)
46 ax2.clear()
47 ax2.plot(np.abs(real_time_rfft))
48 ax2.set_ylim(-60000, 60000)
49 ax2.set_xlim(0, (CHUNK/2))

```

Obrázek 7: Kód Fourierovi tranformace

3.2 Rychlá Fourierova transformace

Rychlá Fourierova transformace je algoritmus, který ze zadané zvukové stopáže zjistí přítomné zvukové frekvence, jak je možno vidět na obrázku 6. To, co ve svém kódu využívám je rfft z knihovny scipy, kde fft znamená Fast Fourier Transform a r znamená, že se počítá pouze v reálných číslech, tudíž je rychlejší než běžná fft. To je možno vidět na obrázku 7 s tím, že je také v loopu zmíněném v sekci sběru dat výše, protože je potřeba, aby odpovídala současným zvukovým datům.

3.3 propojení rfft s barvou

3.4 propojení rfft se silou

4 Překážky

Prvním problémem, se kterým jsem se setkal, byla obtížnost načtení knihoven do mého prostředí Pythonu. Po několika pokusech s terminálem jsem zjistil, že potřebné knihovny již mám nainstalované. Avšak i přesto, že byly knihovny nainstalované, stále se mi nedařilo je načíst. Nakonec jsem dostal radu, že je možné přidat potřebné funkce i přímo skrze nastavení. To fungovalo a já

mohl pokračovat v práci.

Když jsem se rozhodl použít Fourierovu transformaci, tak to byl docela skok do neznáma, protože se učí normálně až na vysoké škole, a tak jsem se s tím tak matlal se spoustou prokrastinace okolo, protože jsem byl dost stracen, ale nakonec to nějak funguje.