



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

ROČNÍKOVÁ PRÁCE S OBHAJOBOU

Téma: Synchronizace Světla a Hudby Frek-
venční 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	Hardware	2
3.1	Raspberry pi zero WH	2
3.2	Mikrofon	3
3.3	LED pásek Neopixel WS2812B	3
4	Datová analýza	3
4.1	Thonny	3
4.2	Sběr dat	3
4.3	Rychlá Fourierova transformace	4
4.4	HSL	5
5	Překážky	5

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, je napsán v programovacím jazyce Python. Python je programovací jazyk založený na programovacím jazyce C. Vybral jsem si jej pro tento projekt, protože je vcelku intuitivní a mám s ním již předchozí zkušenosti.

Jako první součástka bude mikrofón připojený k Raspberry pi. Ten se bude ideálně nacházet u reproduktoru, ze kterého bude vycházet zvuk k frekvenční analýze. Důvod, proč data nejsou brána rovnou z prohlížeče je takový, že stránky bývají chráněny před braním dat zvenčí. Další důvod je univerzálnost. Přístroj by měl být použitelný bez ohledu, na to, co je zdroj zvuku (počítač, mobilní telefon, apod.). Jediné co potřebuje by měl být zdroj energie nejen pro sebe, ale i pro LED pásek a zdroj zvuku na frekvenční analýzu.

Zvuková data budou sbírána v reálném čase a současně na nich bude prováděna Fourierova transformace. Výsledná data z Fourierovy transformace se převedou na barvu v RGB podle předem určené stupnice v závislosti na nejvýznamější frekvencím která nám výjde z Fourierovy transformace. Získaná barva se následně pošle do LED pásky a k barvě se přidá i intenzita svitu, která se také získá z výsledků Fourierovy transformace.

Samotný LED pásek je jednometrový a nejlépe potřebuje vlastní napájení. Bylo by sice možné ho napájet i skrze Raspberry pi, ale proud, který by procházel by mohl Raspberry pi poškozovat, a snižovat jeho životnost, zároveň by se mohla snižovat efektivita přenosu energie a vznikalo by tak zbytečně přebíječné teplo.

Ve výsledku to má fungovat tak, že zdroj, bude zároveň napájet Raspberry Pi i samotný LED pásek. Kód pro datovou analýzu by se měl automaticky spustit po připojení ke zdroji, čímž by měl celý přístroj fungovat bez vnějšího zásahu.

3 Hardware

3.1 Raspberry pi zero WH

Raspberry pi zero WH je jeden z nízkonákladových minipočítačů od Britské společnosti Raspberry Pi Foundation viz obr. 1. Tento model jsem vybral z několika důvodů. Je poměrně malý a levnější než jiné varianty. Má Wi-Fi a Bluetooth, což v tomto projektu nemá využití, ale jedná se o dobré funkce, které by se dala využít, kdybych chtěl na tomto projektu pracovat i do budoucna. Nachází se na něm model procesoru Broadcom BCM2835, který by měl zvládat počítat Fourierovu transformaci. Také se na něm nachází dva microUSB porty, jeden na napájení a druhý na datový vztup a jeden miniHDMI konektor. Jako ulehčič je zde microSD slot, ve kterém používám 16GB microSD kartu, ale je možno využít microSD kartu jakékoliv velikosti. Asi největší problémem je absence jacku, takže jsem při koupi mikrofону musel zdůraznit možnost pouze microUSB konektoru. Teoreticky by se dala využít redukce na microUSB, ale to je další část, která by se musela koupit a vyšlo by to drahé.



Obrázek 1: Raspberry pi zero WH

3.2 Mikrofon

U mikrofonu je vcelku jedno, jak je kvalitní, ale jde o to, jestli je kompatibilní s raspberry pi zero WH. Jak již bylo zmíněno, tak musí být skombinován s potřebnou redukcí na microUSB nebo se samotným mikroUSB. V mém případě jsem zvolil druhou možnost a používám Mikrofon Akaso microUSB viz 2. Jedná se o jediný mikrofon s microUSB co jsem našel, ale jakýkoliv mikrofon s microUSB by měl fungovat.



Obrázek 2: Mikrofon Akaso microUSB, pro V50X, Brave 4 Pro (SYZ0071-BK)

3.3 LED pásek Neopixel WS2812B

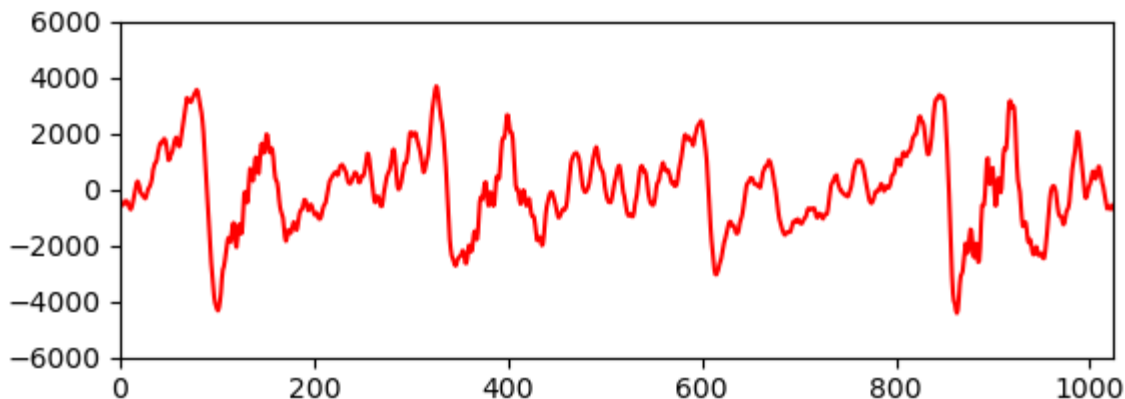
4 Datová analýza

4.1 Thonny

4.2 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 ??. Na obrázku 2 je využití knihovny pro získání dat z mikrofonu. Můžeme zde vidět proměnou p,



Obrázek 3: Zobrazení dat z mikrofonu

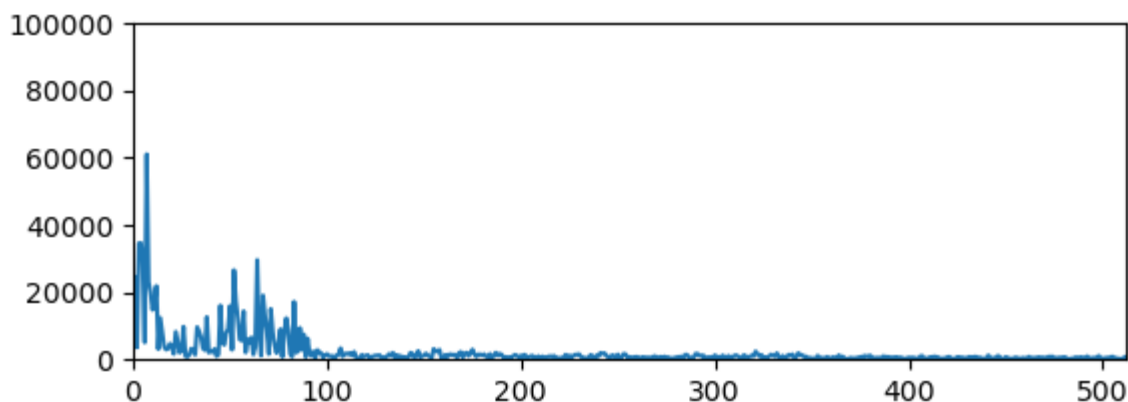
které dáváme datový typ `pyaudio.PyAudio`. Dva řádky níže je proměnná `stream`, která nabírá data z `p`, která díky jejímu datovému typu je schopna brát data z mikrofonu (`input=True`). Zbytek udává další vlastnosti zvukových dat, podle proměnných uvedených výše, a které budeme dále využívat.

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

```
import numpy as np
import pyaudio as pa
import struct
import matplotlib.pyplot as plt
from scipy.fft import rfft
from colorsys import hls_to_rgb
```

4.3 Rychlá Fourierova transformace

Rychlá Fourierova transformace je algoritmus, který ze zadané zvukové stopáže zjistí přítomné zvukové frekvence, jak je možno vidět na obrázku 4. 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 ?? s tím, že je také v loopu zmíněném v sekci sběru dat výše, protože je potřeba, aby odpovídala současným zvukovým datům.



Obrázek 4: Zobrazení dat z Fourierovi transformace

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N}$$

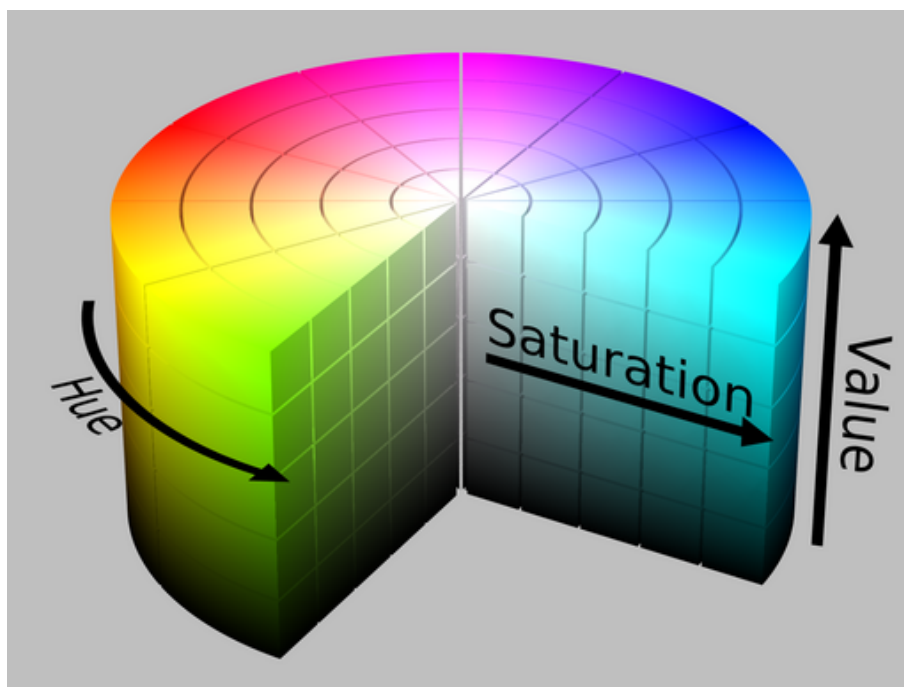
$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{i2\pi kn/N}$$

4.4 HSL

5 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.



Obrázek 5: HSL