Music to Dmx

Saas-en-boite

Contents

[Introduction 2](#_Toc19299064)

[Matériel utilisé 2](#_Toc19299065)

[Partie M. Hamilton-James : DMX 2](#_Toc19299066)

[Uart -> Dmx 2](#_Toc19299067)

[Types de projecteurs utilisés 3](#_Toc19299068)

[Modes utilisés 3](#_Toc19299069)

[Problèmes rencontrés 3](#_Toc19299070)

[Partie M. Nogueiras : Beat Detector 3](#_Toc19299071)

[Algorithme utilisé 3](#_Toc19299072)

[Problèmes rencontrés 4](#_Toc19299073)

[Partie M. Oliveira : Stream 4](#_Toc19299074)

[Conclusion 4](#_Toc19299075)

## Introduction

Pour ce projet, en entrée nous avons un Stream musical lu en Bluetooth, qui est ensuite analysé pour obtenir la BPM (beat per minute), finalement par rapport au BPM, nous contrôlons plusieurs types de projecteurs avec le protocole dmx.

## Matériel utilisé

* SparkFun Thing Plus - ESP32 WROOM
* SparkFun ESP32 Thing Plus DMX to LED Shield
* DMX Universe
* PMOD I2S Audio In (jack 3,5mm)
* Audio Out analogique (port jack 3,5mm)
* Digilent PmodMic - module microphone
* Appareil pour Stream musique *Bluetooth*

## Partie M. Hamilton-James : DMX

### Uart -> Dmx

A screenshot of a cell phone

Description automatically generated

Figure 1 : paquet dmx

Le protocole DMX est très similaire au protocole Uart, comme on peut le voir dans la figure 1, le start Frame et les 512 autres frames sont exactement identique a de l’Uart, à l’exception des 24 premiers bits de Start du paquet.

En sachant que nous avions a disposition que le protocole Uart, nous avons décidé de mettre un baud rate lent pour le premier message(90909bits/s) et le remettre a la vitesse du dmx (250kbits/s) pour commencer la communiquer.

### 

### Types de projecteurs utilisés

Nous avons utilisé 2 projecteurs différents, le mf078 qui est un projecteur RGB avec déplacement en x et en y, et le mpx405 multi-pack qui permet de connecter des lumières dessus.

### Modes utilisés

Nous avons instancié trois modes différents, le premier est le mode beat qui envoie une impulsion puissante sur le mf078 et les lampes du mpx405 et qui après 50 millisecondes diminue grandement l’intensité du mf078 et éteins les mpx405. Le deuxième mode est le mode off qui comme son nom l’indique, éteint toutes les lumières (très utile lorsque le mf078 est en mode strobe) et finalement le mode strobe qui donne au mf078 un effet stroboscopique rapide et allume les lumières pendant le temps d’un beat.

### Problèmes rencontrés

Nous avons eu quelques problèmes lors de l’envoie des données, mais très vite, nous avons réalisé qu’il fallait rajouter une boucle pour faire passer le temps et laisser au uart le temps de changer le baudrate.

Un autre petit souci a été le problème de documentations pour les différents pins.

Finalement le plus gros souci a été la maladie qui ne m’a pas aidé à coder.

## Partie M. Nogueiras : Beat Detector

### Algorithme utilisé

Afin de détecter les beats d'une musique, nous avons décidé d'utiliser la transformée de Fourier, afin d'en calculer l'énergie des basses fréquences et ainsi la comparer à une moyenne des énergies de la dernière seconde et comparer cette valeur multipliée par un certain coefficient (dans notre cas 190%).

Afin d'effectuer la transformée de Fourier, nous générons dans un premier temps, une matrice composée de "e^-(2πi/N)" à la puissance de sa coordonnée X multipliée par sa coordonnée Y dans celle-ci. Puis, en nous servant de cette matrice, nous multiplions notre échantillon normalisé à cette matrice et obtenons ainsi un vecteur contenant des nombres complexes représentant la valeur de chaque fréquence.

Pour calculer l'énergie qui nous intéresse de notre échantillon, nous sélectionnons la plage de fréquence que nous souhaitons analyser (pour notre projet nous avons choisi la plage de fréquences allant de 0 à environ 700 hertz) et en faisons une moyenne. Suite à cela, nous ajoutons cette nouvelle valeur à notre tableau contenant les 345 dernières valeurs (ce qui correspond pour un taux d'échantillonnage à 44'100 hertz à une seconde) et tirons l'énergie moyenne de la dernière seconde.

Normalement à ce stade, il aura fallu calculer la variance. Cependant, suite à plusieurs problèmes avec les valeurs de celle-ci et le bug que cela générait lors de l'exécution sur l'esp32, nous avons décidé de le retirer et d'établir un seuil uniquement grâce à la moyenne de la dernière seconde multipliée par un coefficient fixe. Ce coefficient est très sensible car c'est lui qui va nous permettre de détecter s'il y a eu un beat ou non et en fonction de la musique ou de la vitesse du beat celui varie plus ou moins fortement ce qui nous empêche d'avoir un coefficient générique pour n'importe quelle musique.

Une fois les calculs effectués et la comparaison des valeurs faite. Nous envoyons à la task dmx si elle doit allumer ou non ses lumières.

### Problèmes rencontrés

Suite à ces trois semaines, nous avons rencontré plusieurs difficultés, notamment au niveau des librairies. En effet, il nous a été très compliquer de trouver une librairie effectuant des FFT potable et n'avons malheureusement pas trouvé de fonction permettant d'effectuer une transformée inverse de Fourier qui soit opérationnelle. Nous avons donc décidé de créer notre propre librairie et nous sommes heurté à la difficulté de cette fameuse transformée inverse de Fourier. De ce fait, nous avions prévu d'utiliser un algorithme différent filtrant dans un premier temps les hautes fréquences et en retournant dans le domaine temporel un lissage puis un filtre en peigne permettant de générer des interférences constructives ou destructives afin de mieux faire ressortir les pics d'énergie et donc détecter les beats plus facilement. Comme parler dans l'algorithme utilisé finalement, nous avons eu des soucis avec la variance qui doivent probablement venir de la normalisation car les valeurs n'étaient pas comprises entre 0 et 1 et qui nous a empêché d'avoir un réglage plus "fin" au niveau du seuil.

## Partie M. Oliveira : Stream

## Conclusion

A la fin de notre projet, nous sommes très contents du résultat obtenu, il nous est possible de nous connecter en Bluetooth sur l'esp32 et d'envoyer de la musique qui sera traité par le microcontrôleur et générera un beat ou non et ainsi allumer ou non les lumières de notre univers dmx. Nous sommes également heureux d'avoir réussi à surmonter les différents problèmes rencontrés au cours du développement de notre projet même si nous n'avons pas eu le temps d'implémenter l'entrée et la sortie analogique.