

Rapport technique

Projet BLISS (Beatbox & Light)

Barthélemy DEMEER, Solène LE SINQ, Lou PORTELLI, Isidor TOUKAM

Mai 2023

PROJET BLISS (BEATBOX & LIGHT)	1
I. INTRODUCTION	3
1. CAHIER DES CHARGES	3
2. SCHEMA DE PRINCIPE	4
II. DESCRIPTION TECHNIQUE	5
1. MATERIEL UTILISE	5
2. FONCTIONNALITES	6
III. BILAN	8
1. ACHEVEMENT FINAL	8
2. RETOUR D'EQUIPE	9

I. Introduction

Le soirées SupOpticiennes sont reconnues pour leurs jeux de lumières exceptionnels. C'est pourquoi nous avons choisi de travailler sur le sujet beat box & light. Également car on souhaitait travailler sur un projet plutôt axé sur le code. L'association LaserWave utilise un ordinateur pour contrôler les différentes LEDs Eurolite®. Notre objectif est de supprimer cet intermédiaire en associant à chaque pad un set de couleurs, sans passer par l'ordinateur mais par la carte Nucléo (microprocesseur).

1. Cahier des charges

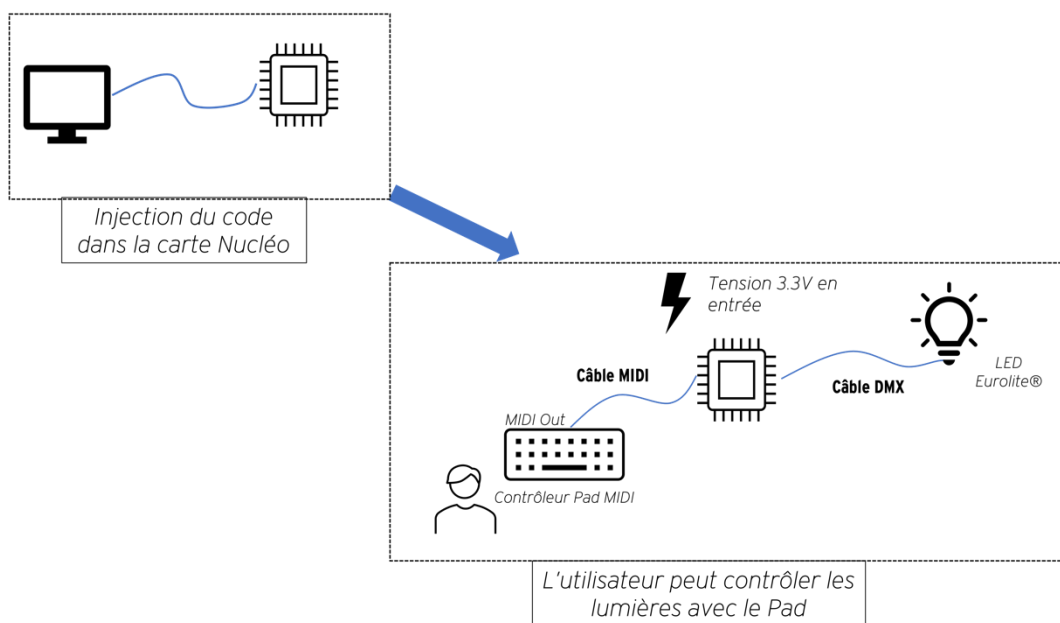
Notre projet répond ainsi au cahier des charges suivant :

- L'interface utilisateur doit être simple, autant lors de l'utilisation du dispositif que pour la modification du code et l'implantation de nouveaux sets de lumière.
- Mise en place d'une banque avec différents sets de lumière répondant au tableau suivant
- L'utilisateur doit simplement brancher les sorties DMX et MIDI du microcontrôleur Nucléo puis appuyer sur un pad pour modifier le set de couleur.

Numéro du pad	Nom du pad	Couleurs
1	Winter	Les 4 spots sont allumés au couleur de l'hiver (2 blancs et 2 bleus clairs)
2	Spring	Les 4 spots sont allumés au couleur du printemps (vert, rose et jaune)
3	Summer	Les 4 spots sont allumés au couleur de l'été (orange, vert et bleu)
4	Fall	Les 4 spots sont allumés au couleur de l'automne (orange, marron et jaune)
5	Carrousel 1	Boucle de 4 sets de couleurs prédéfinis
8	Carrousel 2	Idem que le 1 à la différence que en cliquant sur un autre pad, la boucle se stoppe instantanément
9	BLISS	Chaque spot s'allume d'une couleur choisie par un des membres du groupe
10	Random	Les 4 spots sont allumés avec des couleurs choisies aléatoirement
11	Purple	Les 4 spots s'allument respectivement en marron, rose, bleu clair et violet
12	Turquoise	Les 4 spots s'allument en bleu et vert
13	Blackout	Tous les spots sont éteints
14	Red	Tous les spots sont rouges
15	Green	Tous les spots sont verts
16	Blue	Tous les spots sont bleus

Tableau 1 : Banque des sets de lumière

2. Schéma de principe



II. Description technique

1. Matériel utilisé

- Partie DMX

On utilise des Eurolite LED Party, composé de trois blocs principaux qui réalise des mélanges RGB permettant d'obtenir toutes les couleurs. Il y a également trois bandes de LED blanches pour augmenter la luminosité. Nous avons quatre lampes, une maitresse, reliée à la carte Nucléo, et les trois autres esclaves de la première. Pour assurer la compatibilité avec notre code, nous prenons soin d'assigner à chaque lampe l'adresse correcte via les boutons utilisateur.

Spot	Lampe 1	Lampe 2	Lampe 3	Lampe 4
Adresse	1	9	17	25

Tableau 2 : Adresse des spots

Les adresses sont donc espacées de 8 emplacements. Cela est dû au codage des spots, qui doivent contenir 8 informations pour réaliser les différents sets de lumières.

Functions in DMX mode

Channel	Value	Function
1 Master dimmer	000 – 255	Total brightness 0-100 %
2	000 – 255	Red 0-100%
3	000 – 255	Green 0-100%
4	000 – 255	Blue 0-100%
5	000 – 255	White 0-100%
6 Strobe effect	000 – 000	No function
	001 – 255	Strobe effect with increasing speed (Channel 1, 2, 3, 4, 5)
Speed channel 7	000 – 255	Speed with increasing speed channel 7
7	000 – 050	No function
	051 – 100	Color change, "Switching"
	101 – 200	Color change, "Fading"
	201 – 255	Sound control

Tableau 3 : Codage des spots – Source : Mode d'emploi Eurolite

Pour notre projet, nous avons choisi de ne pas utiliser la fonctionnalité *Strob* pour alterner différentes couleurs. Nous avons mis en place d'autres méthodes, développée plus tard dans le rapport.

Concernant le codage, nous envoyons via une liste, les informations à chaque spot pour créer les couleurs souhaités. Dans l'idée de simplifier les écritures, nous avons mis en place une fonction *allumer* qui prend en argument la liste d'un spot et une liste correspondant à une couleur.

```
void allumer(int lampe[], int couleur[]){
    dmx_data[lampe[0]]=couleur[0];
    dmx_data[lampe[1]]=couleur[1];
    dmx_data[lampe[2]]=couleur[2];
    dmx_data[lampe[3]]=couleur[3];
    dmx_data[lampe[4]]=couleur[4];
    dmx_data[lampe[5]]=couleur[5];
    dmx_data[lampe[6]]=couleur[6];
}
```

Code de la fonction *allumer*

- o Partie MIDI :

On utilise une table midi MPD 32 (Akai Professional), qui est donc l'interface de contrôle des lumières. On récupère l'adresse de chaque pad pour l'assigner ensuite à un set de lumière. L'adresse des pads est donnée en hexadécimal grâce au logiciel MIDI Monitor. Nous avons mis en place une « banque » qui recense l'adresse des pads en fonction de leurs noms sur la table, toujours dans l'optique de faciliter l'utilisation.

```
/*          *** DECLARATION DES PADS ***          */
// Valeurs des pads donnés grâce au logiciel MIDI Monitor
int pad1=0x24;
int pad2=0x25;
int pad3=0x26;
int pad4=0x27;
int pad5=0x28;
int pad6=0x29;
int pad7=0x2A;
int pad8=0x2B;
int pad9=0x2C;
int pad10=0x2D;
int pad11=0x2E;
int pad12=0x2F;
int pad13=0x30;
int pad14=0x31;
int pad15=0x32;
int pad16=0x33;
```

Code de la « banque » des pads

- o Partie Nucléo :

La carte Nucléo permet de faire le lien entre le bloc DMX et le bloc MIDI. La carte en elle-même est donc agrémenter de 2 étages : un étage

DMX et un étage MIDI. La carte a besoin d'être alimentée pour délivrer les instructions. Puisque le but du projet est de s'affranchir de l'ordinateur, on utilise un adaptateur UBS-Secteur pour l'alimenter.

2. Fonctionnalités

Un des objectifs du projet BLISS est de réaliser une alternance de couleur sur un des sets lumineux. Pour cela, nous avons conçu deux méthodes différentes :

- Méthode 1 : Utilisation du *wait_us*

Notre première intuition était d'utiliser une boucle avec une instruction *wait_us* entre chaque set de lumière. Ainsi, à intervalles réguliers, les lumières changent. Cette solution est simple, cependant elle comporte un inconvénient. En utilisant cette méthode, lorsque l'utilisateur veut revenir à un set statique, la boucle se termine avant de pouvoir passer au set lumineux suivant. Le changement n'est donc pas assez rapide et cela présente un problème majeur dans le cas de l'utilisation avec de la musique.

- Méthode 2 : Utilisation du module *ticker*

Pour répondre à ce problème il nous a semblé judicieux de nous pencher sur le module *ticker* compris dans l'environnement Keil Studio. Cette fonctionnalité va nous permettre de déclencher une séquence sous l'action d'un pad, lançant donc un compteur. Cette séquence est conditionnée par une condition *if* et non par une boucle *for* ou *while*. Ainsi, la condition peut être rapidement interrompu et donc la séquence de défilement des sets s'arrête immédiatement lorsque la condition n'est plus satisfaite. Cette dernière méthode est plus complète quoique plus complexe aussi. Elle alourdi un peu le code car il est nécessaire de placer une non-satisfaction de la condition à l'entrée de chaque pad ainsi de ne pas rentrer par erreur dans la boucle.

```
void int_tik(){ // Utilisation du Ticker pour l'alternance des sets
  if(seq1==1){
    if(cpt==0){ // La séquence comprend 4 sets qui se répète jusqu'à ce que l'utilisateur revienne au Blackout
      allumer(lampe1,bleu);
      allumer(lampe2,bleu);
      allumer(lampe3,rouge);
      allumer(lampe4,bleu);
    }
    if(cpt==1){
      allumer(lampe1,blanc);
      allumer(lampe2,blanc);
      allumer(lampe3,blanc);
      allumer(lampe4,blanc);
    }
    if(cpt==2){
      allumer(lampe1,rouge);
      allumer(lampe2,rouge);
      allumer(lampe3,rouge);
      allumer(lampe4,rouge);
    }
    if(cpt==3){
      allumer(lampe1,bleu);
      allumer(lampe2,blanc);
      allumer(lampe3,blanc);
      allumer(lampe4,rouge);
    }
    if(seq1==1){
      cpt=cpt+1;
      if(cpt==4){
        cpt=0;
      }
    }
  }
}
```

Code pour la séquence de set à l'aide de *ticker*

Notre objectif était de pouvoir changer la période de changement de sets à l'aide de la variation des potentiomètres présents sur la table de pads. Nous n'avons pas eu cependant le temps de pouvoir nous pencher sur la récupération d'une telle donnée et donc notre dernier objectif n'a pu être abouti.

III. Bilan

1. Achèvement final

Au cours de la réalisation de ce projet, nous avons toutes et tous pu développer et acquérir de nouvelles compétences, et ce dans différents domaines :

- **Séparation de tâches à distance et travail en groupe**

En effet, nous pouvons avancer uniquement de façon hebdomadaire, lors des séances prévues en cours. Ainsi, il nous a fallu veiller à nous organiser en amont, notamment à la fin de chaque séance, de sorte à savoir sur quelle tâche chacune et chacun pourra travailler durant la semaine. De plus, il nous fallait une plateforme commune, de sorte à centraliser tous nos documents et notre avancée. Nous avons donc opté pour l'application Notion, remplissant parfaitement ce rôle.
- **Obtention de couleurs désirées par méthode RGB, en faisant varier les intensités de rouge, vert ou bleu**

Nous avons appris à former une immense palette de couleurs à partir de quelques couleurs basiques, à l'image de peintres composant différentes nuances de couleurs. Pour cela, nous avons le choix d'implémenter différentes intensités de rouge, vert, bleu et blanc dans chaque spot de lumière. Cette intensité était caractérisée par un entier entre 0 et 255 compris.
- **Utilisation du langage C++**

Nous avons eu une nouvelle opportunité d'utiliser le langage C++, quelque peu différent du langage C sur quelques aspects. Ainsi, nous avons pu bénéficier d'une toute première et nouvelle approche avec ce langage puissant et ses règles différentes.
- **Réalisation d'une bibliothèque**

Dans l'objectif de rendre notre code facile à comprendre et à utiliser, nous avons eu à réaliser deux bibliothèques : une dédiée aux différentes lampes à utiliser, et une dédiée aux différentes couleurs que nous allons utiliser. De ce fait, l'utilisation est simplifiée, et les noms utilisés sont instinctifs, de sorte à ce qu'une personne extérieure au projet comprenne clairement comment utiliser ces bibliothèques.
- **Établissement un cahier de charges**

Sur un aspect plus professionnel, nous avons appris à réaliser un cahier des charges précis et rigoureux, de sorte à exprimer clairement les capacités de notre système.
- **Synthèse des fonctions, d'explications de ces fonctions, et des méthodes utilisées pour y arriver**

Enfin, nous avons appris à synthétiser de manière claire et concise des compétences apprises, des connaissances mobilisées, et des processus permettant d'aboutir à un résultat.

2. Retour d'équipe

Lors de la première séance, nous avons établi un planning pour planifier nos séances. Ce dernier a été quelque peu perturbé par l'apparition de difficultés imprévues. En effet, nous avons pris beaucoup de retard à cause du matériel que nous ne maîtrisons pas et qui plus est, était partagé avec d'autre groupe. C'est pourquoi d'une séance à une autre, notre code ne correspondait plus aux paramètres appliqués aux sport et/ou aux pads. Dans ces cas-là, la motivation flanchait un peu au sein de l'équipe car voyant le retard s'accumuler, nous étions conscients que des choix devaient être fait au sein de nos objectifs.

En réponse à cela, nous avons mis en place un protocole de vérification du matériel à faire avant le début de chaque séance afin d'optimiser notre temps de travail avec un matériel fonctionnel et coopérant.