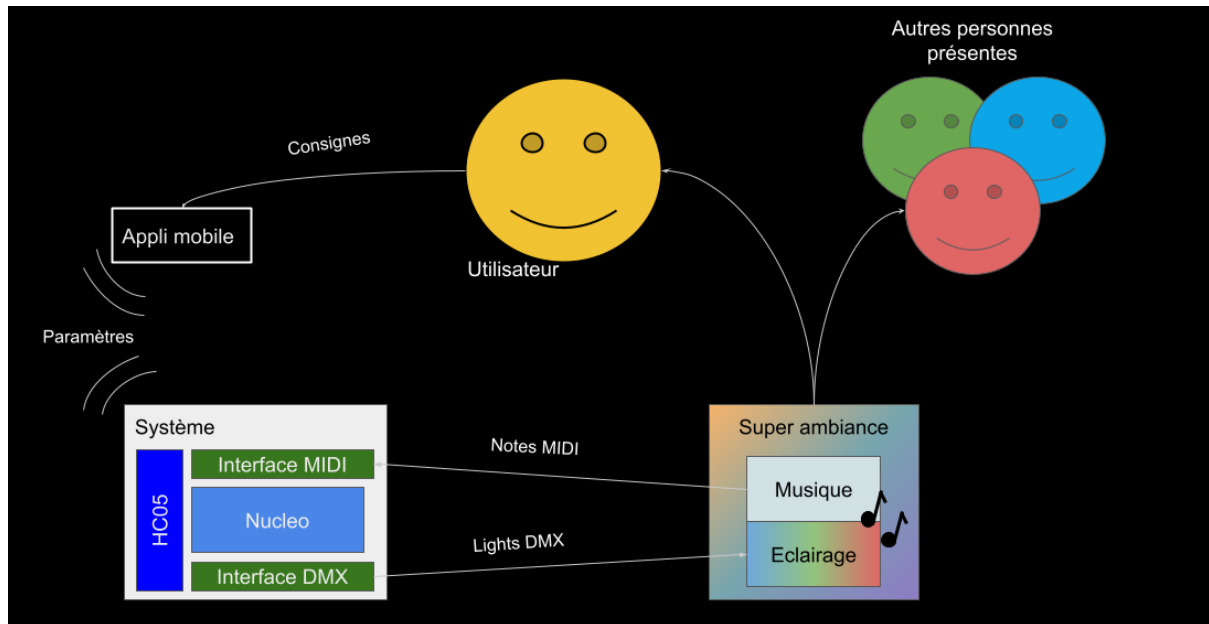


## Projet : Beatbox And Light

### Description du projet :

L'objet du projet est de réaliser un système capable d'adapter l'éclairage d'une soirée à la musique qui y est jouée.



Le système communique avec un système d'éclairage LED par le protocole DMX et avec le système sonore par le protocole MIDI, que ce soit *via* un fichier pré-enregistré dans ce format ou un musicien jouant en direct sur un synthétiseur.

Cette adaptation se base sur le principe d'une association note <> couleur (synesthésie) : les douze demi-tons de l'octave sont répartis sur douze couleurs. Quatre spots à LEDs indépendants peuvent être reliés à quatre pistes du fichier MIDI, ainsi un spot peut réagir au piano et un autre à la batterie.

Afin de fluidifier les transitions, la couleur envoyée ne dépend pas uniquement de la dernière note jouée mais également des deux précédentes.

L'intensité lumineuse augmente avec le volume de la note.

Un compteur permet de connaître le nombre de notes en train d'être jouées sur une piste afin d'éteindre les LEDs associées à cette piste lorsque la musique s'arrête. Selon la manière dont la partition MIDI est implémentée, ce compteur peut ne pas fonctionner.

Une application Android permet de modifier les paramètres suivants du système :

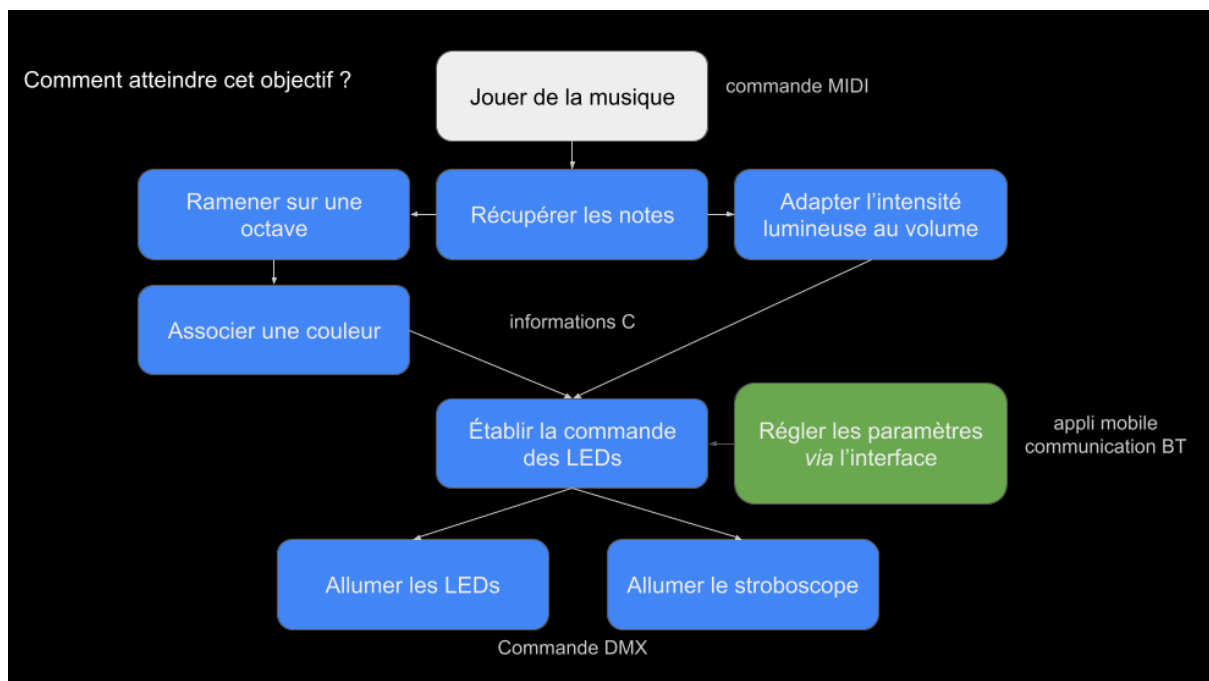
- Correspondance : pistes MIDI <> Adresses DMX
- Correspondance : notes <> Couleurs RGB
- Lancement et arrêt du mode "stroboscope" (non implémenté)

Cette application communique avec le système par la technologie *Bluetooth*.

### Description fonctionnelle :

Le système remplit les fonction suivantes :

- Extraire les notes du fichier MIDI
- Ramener les notes sur une octave
- Associer une couleur à chaque note de l'octave
- Adapter l'intensité de lumineuse au volume de la note
- Établir la commande des LEDs (couleur, intensité)
- Permettre à l'utilisateur de saisir des paramètres
- Communiquer ces paramètres de l'interface au système
- Mettre à jour les paramètres du système en fonction du réglage utilisateur
- Permettre à l'utilisateur de demander le démarrage ou l'arrêt du mode stroboscope d'un spot
- Communiquer cette demande de l'interface au système
- [Non implémenté] Démarrer ou arrêter le le mode stroboscope d'un spot à la demande de l'utilisateur



## **Structure du système :**

Le système est composé d'une carte Nucléo, à laquelle une interface DMX, une interface MIDI et un module Bluetooth HC05 ont été adjoints. Un téléphone (ou autre machine) fonctionnant sous Android et possédant son propre module Bluetooth est nécessaire pour utiliser l'application.

L'algorithme général reçoit à la fois les informations bluetooth (utilisées en paramètres du reste du programme) et les informations MIDI converties en DMX puis transmises aux LEDs. Le main contient deux parties, une partie Bluetooth déclenchée lorsqu'une information est reçue, et la boucle MIDI, exécutée lorsqu'une note est reçue depuis l'entrée MIDI.

Les fonctions essentielles du programme sont la fonction Bluetooth, la fonction de d'identification de canal MIDI, la fonction de correspondance chromatique et la fonction qui génère la trame DMX.

Nous pouvons présenter le code complet sous demande du lecteur.

## **Réalisation des fonction principales :**

### **Extraire les notes du fichier MIDI :**

Les informations reçues sont une trame MIDI, et consistent en trois octets :

Le premier indique s'il s'agit d'un NOTE ON ou d'un NOTE OFF, ainsi que le canal associé. Cette information est traitée et indique s'il faut allumer ou éteindre tel ou tel spot lumineux.

Le second octet indique quelle note est jouée sous forme d'un entier.

Le dernier octet transmet l'information de vélocité (attaque) sous forme d'un entier.

### **Ramener les notes sur une octave :**

Cette fonction se fait simplement avec un modulo 12.

### **Associer une couleur à chaque note de l'octave :**

Après calcul du modulo, cette fonction utilise trois tableaux de 12 valeurs (R, G, B respectivement) et associe ainsi immédiatement chaque note à sa couleur. Le programme contient des tableaux par défaut, qui peuvent être modifiés *via* l'application mobile.

### **Adapter l'intensité de lumineuse au volume de la note :**

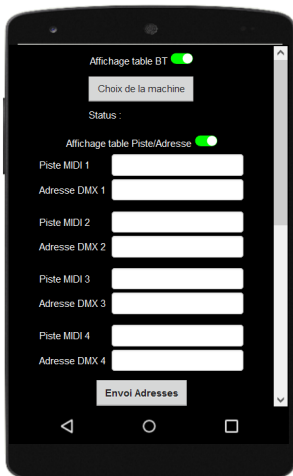
La vélocité fournissant une valeur entière, on l'utilise pour moduler l'intensité lumineuse reçue par chaque spot.

### **Établir la commande des LEDs (couleur, intensité) :**

Une fois les informations utiles récupérées, elles sont affectées aux cases utiles du tableau envoyé aux spots dans la trame DMX.

Chaque spot a besoin de plusieurs valeurs pour fonctionner comme souhaité, et cette sélection se fait au moyen de l'"adressage" : on indique une adresse à chaque spot, et

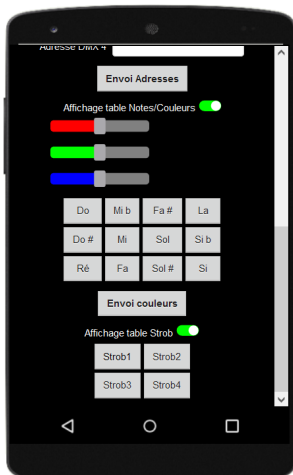
celui-ci “verra” les valeurs du tableau DMX à partir de cette adresse. Ainsi, un spot adressé à 12 et nécessitant 8 canaux pour fonctionner lira les cases 11 à 19 du tableau (les adresses commencent à 1, les indices du tableau à 0 ; ce décalage est évidemment pris en compte dans le programme).



### Permettre à l'utilisateur de saisir des paramètres :

L'application possède des champs d'écriture pour remplir les paramètres de pistes et d'adresses, et des “sliders” rouge, vert et bleu pour la définition des couleurs. Une fois que le paramètre à complètement été défini, les boutons “envoi adresses” et “envoi couleurs” permettent de lancer la transmission. Celle-ci commence par un unique caractère ('@' pour les couleurs ou '#' pour les adresses) suivi d'une série de nombres, écrits les uns après les autres selon un ordre préétabli qui donnent les valeurs à affecter.

Pour ne pas perdre une partie de cette série de valeurs, la fréquence d'envoi a été limitée à une valeur par 20 ms, et l'application attend une réponse du système avant de passer au message suivant afin de s'assurer que le message précédent a bien fini d'être traité (protocole “dialogue”).



### Mettre à jour les paramètres du système en fonction du réglage utilisateur :

Le système vérifie régulièrement si des consignes lui ont été envoyées et modifie les paramètres (variables c) en fonction.

### [Non implémenté] Démarrer ou arrêter le le mode stroboscope d'un spot à la demande de l'utilisateur :

Les spots disposent d'une fonctionnalité de stroboscope pouvant être intégralement commandée par la trame DMX. Ces stroboscopes devaient pouvoir être activés et désactivés selon le bon vouloir de l'utilisateur, par des commandes effectivement reçues par le

programme mais dont le traitement n'a pas pu être mis en place faute de temps. La communication se fait de la même manière que pour les paramètres.

### Validation du système :

Les fonctions suivantes ont été testées d'après les protocoles décrits.

**Extraire les notes du fichier MIDI :** Affichage dans TeraTerm de la valeur de note reçue (entre 0 et 11).

**Ramener les notes sur une octave :** idem

**Associer une couleur à chaque note de l'octave :** Testé visuellement

**Adapter l'intensité de lumineuse au volume de la note** : Testé visuellement

**Établir la commande des LEDs (couleur, intensité)** : Affichage des valeurs dans TeraTerm.

**Permettre à l'utilisateur de saisir des paramètres** : cf fonction suivante

**Communiquer ces paramètres de l'interface au système** :

Un premier programme de type "echo" a été mis en place : le système recopie sur le terminal TeraTerm les messages qu'il reçoit de la liaison Bluetooth. Ceci nous permet de vérifier que le message théoriquement envoyé a bien été reçu.

**Mettre à jour les paramètres du système en fonction du réglage utilisateur** :

Paramètre "pistes <> adresses" : le système a été mis en fonctionnement pour un morceau MIDI à une unique piste et nous avons associé successivement cette piste à chacun des spots individuellement puis à plusieurs à la fois. Nous avons constaté que les spots s'activaient et se désactivaient conformément aux consignes.

Paramètre "notes <> couleurs" : En fonctionnement, nous avons pu passer d'un jeu de couleurs "à tendance rouge" à un autre "à tendance bleue" et constater le changement d'ambiance.

**Permettre à l'utilisateur de demander le démarrage ou l'arrêt du mode stroboscope d'un spot** : cf fonction suivante

**Communiquer cette demande de l'interface au système** :

Testé avec le programme "echo"

**[Non implémenté] Démarrer ou arrêter le le mode stroboscope d'un spot à la demande de l'utilisateur** :

Non implémenté, le stroboscope a fonctionné dans une version antérieure

## **Résultat final**

Le système est fonctionnel. Les spots s'allument et s'éteignent en rythme aux couleurs demandées.

Les paramètres peuvent être modifiés *via* l'application, même en cours de fonctionnement. Le protocole "dialogue" assure que les paramètres soient bien reçus dans la majorité des cas, bien qu'il reste de rares erreurs, en particulier lors de la transmission des couleurs, plus longue que celle des adresses.

## Planning

**[1/02] Séance 1 (CoVID Colin) :** Définition du projet, définition des tâches et des priorités, choix des outils de travail en équipe. Première prise en main du code Scriabine.

**[1/02] Séance 2 :** Appropriation du code et compréhension du protocole MIDI. Livrables intermédiaires.

**[1/03] Séance 3 :** Répartition des rôles (Stella, Nathanaëll + Colin sur le code principal, Noé sur l'appli). Les spots ne s'éteignent pas à la fin du morceau. Les transitions sont lissées par moyennage (coefficienté) sur les trois dernières notes

(nouvelle répartition des rôles : Colin sur l'appli, Noé sur l'ensemble du système)

**[8/03] Séance 4 :** Un compteur est mis en place pour éteindre les spots à la fin du morceau. La toute première version de l'appli est créée.

**[break]**

**Séance 5 (13/04) :** Gestion des pistes multiples. Retrait de l'image de fond de l'appli (poids, copyright).

**Séance 6 (20/04) :** Améliorations du code. Le Bluetooth ne marche pas, création du logiciel echo pour débogage.

**Séance 7 (17/05) :** Le Bluetooth fonctionne. L'usage d'une mauvaise carte nucleo fait perdre 3h sur la séance.

**Extra :** Nathanëll + Colin mettent en place le protocole "dialogue", Noé prends des photos pour la présentation.

**Séance 8 (23/05) :** Implémentation du programme global, derniers tests et présentation.

## Difficultés

Les difficultés rencontrées ont été techniques et conjoncturelles.

Parmis les premières, on retrouve :

- l'arrêt des notes à la fin du morceau
- la perte d'un temps considérable avec une mauvaise carte Nucleo
- la compréhension du protocole MIDI
- la compréhension de l'éditeur MIT App Inventor pour l'appli
- la compréhension du protocole Bluetooth
- la perte d'information lors de la transmission Bluetooth

Et parmi les secondes :

- Le CoVID de Colin qui a induit un retard dans l'appropriation du projet
- Divers problème de nature personnelle chez plusieurs membres de l'équipe
- Les examens pendant la période de rédaction des livrables

### **Analyse du travail en équipe**

Bien que ces rôles n'aient pas été fermement définis au début, le travail a rapidement été réparti entre deux équipes, la première, constituée de Stella et Nathanaël s'est penchée sur le cœur du système et l'implémentation du code C, tandis que Colin a développé l'interface mobile. Noé a pris du recul par rapport à ces équipes de sorte de pouvoir participer partout où c'était nécessaire et de garder une vue globale du projet et assurer la cohérence des actions des deux équipes.