

# PROJETS ARDUINO - PEIP2

*Année scolaire 2021-2022*

## ***“THE ULTIMATE 80’s JUKEBOX”***



**Étudiants : MAIGNOT Audrey ; VICART Antoine**

Ecole Polytechnique Universitaire de Nice Sophia-Antipolis, Parcours des écoles d'ingénieurs  
Polytech 2ème année

1645 route des Lucioles, Parc de Sophia Antipolis, 06410 BIOT

# SOMMAIRE

Introduction.....	3
Chapitre I : L'algorithme.....	5
Chapitre II : Le développement du projet.....	6
II.1. Le menu.....	6
II.2. La lecture des musiques.....	7
II.2.1. Le module MP3 UART.....	7
II.2.2 Le réglage du volume par le potentiomètre.....	7
II.3. La connexion radiofréquence.....	7
II.4. La guirlande LED Adafruit NeoPixel.....	8
II.4.1. Idées initiales.....	8
II.4.2 Résultat final.....	8
II.5. Le “monnayeur”.....	8
II.6. La structure.....	9
Conclusion.....	10
Bibliographie.....	10

# Introduction

Dès le départ, nous savions que notre projet Arduino serait centré autour de la musique, ce domaine étant un centre d'intérêt commun. Contrairement à ce qu'on aurait pu s'attendre, nous ne nous sommes pas projetés dans le futur en créant un super robot qui chante mais nous sommes plutôt retournés dans le passé pour remettre au goût du jour le fameux jukebox de l'époque. Nous avons choisi le thème des années 80 pour son côté rétro, bien que la période du jukebox était déjà révolue dans ces années-là. Voici dans un premier temps ce que nous avons prévu à la genèse du projet, avant de vous dévoiler ce qu'il en est vraiment au final.

“Mettons-nous donc dans la peau d'un client qui utilise ce jukebox :

Le client arrive devant le jukebox et lit sur un écran LCD : « Insérez une pièce ». Il insère nécessairement une pièce de 50 centimes lui permettant d'écouter une musique. Une fois la pièce insérée, l'écran affiche « Genre ? » et le client en choisit un en s'aidant des 6 boutons latéraux. Une fois qu'il a fait son choix, il appuie sur le bouton central « OK » (il peut toujours cliquer sur le bouton « Retour » s'il s'est trompé et veut donc changer de genre). Ensuite, il peut sélectionner une des 6 musiques incluses dans le genre sélectionné à l'aide de nouveau des boutons latéraux. Il clique ensuite sur « OK » pour lancer la musique. Les musiques sont jouées par une enceinte reliée à un module MP3 UART. Une fois la musique terminée, il faut réinsérer une pièce de 50 centimes pour lire une autre musique. Tout cela fonctionne bien évidemment grâce à une carte Arduino, possiblement des batteries, résistances etc.

Les genres disponibles sont les suivants :

- Pop
- Rock
- Disco
- Arcade
- 80's Movie
- 80's Best Of

Quant à la fonctionnalité sans fil, la connexion radiofréquence, nous avons décidé d'utiliser un module Bluetooth relié à une guirlande de LED lumineuse afin de pouvoir contrôler cette dernière à distance avec son smartphone. Le client a le choix entre plusieurs « motifs » que nous, les créateurs, avons préprogrammé.”

Voilà donc ce que nous avons écrit avant de commencer à concevoir quoi que ce soit. Bien des choses ont évolué depuis.

Premièrement, le client n'insère plus nécessairement une pièce de 50 centimes. Nous avons conçu un “monnayeur” avec tout simplement un capteur de distance donc n'importe quelle pièce, à condition qu'elle reste au-dessus de ce capteur, fait totalement l'affaire.

Deuxièmement, les genres et les musiques ne se choisissent plus avec 6 boutons latéraux, un bouton “OK” et un bouton “Retour” mais avec un encodeur permettant d'économiser les 6 boutons de

sélection et le bouton "OK". Cependant, nous avons enlevé la possibilité de revenir en arrière, s'étant dit que l'erreur de sélection serait quand même assez peu fréquente.

Dernièrement, on a totalement modifié la fonctionnalité Bluetooth. Au lieu qu'elle contrôle la guirlande LED, elle permet de contrôler depuis son smartphone le volume, la guirlande étant maintenant indépendante du Bluetooth bien qu'on ait conservé les différents thèmes lumineux.

Suite à cela, voici donc les plannings initiaux et finaux pour le développement du projet :

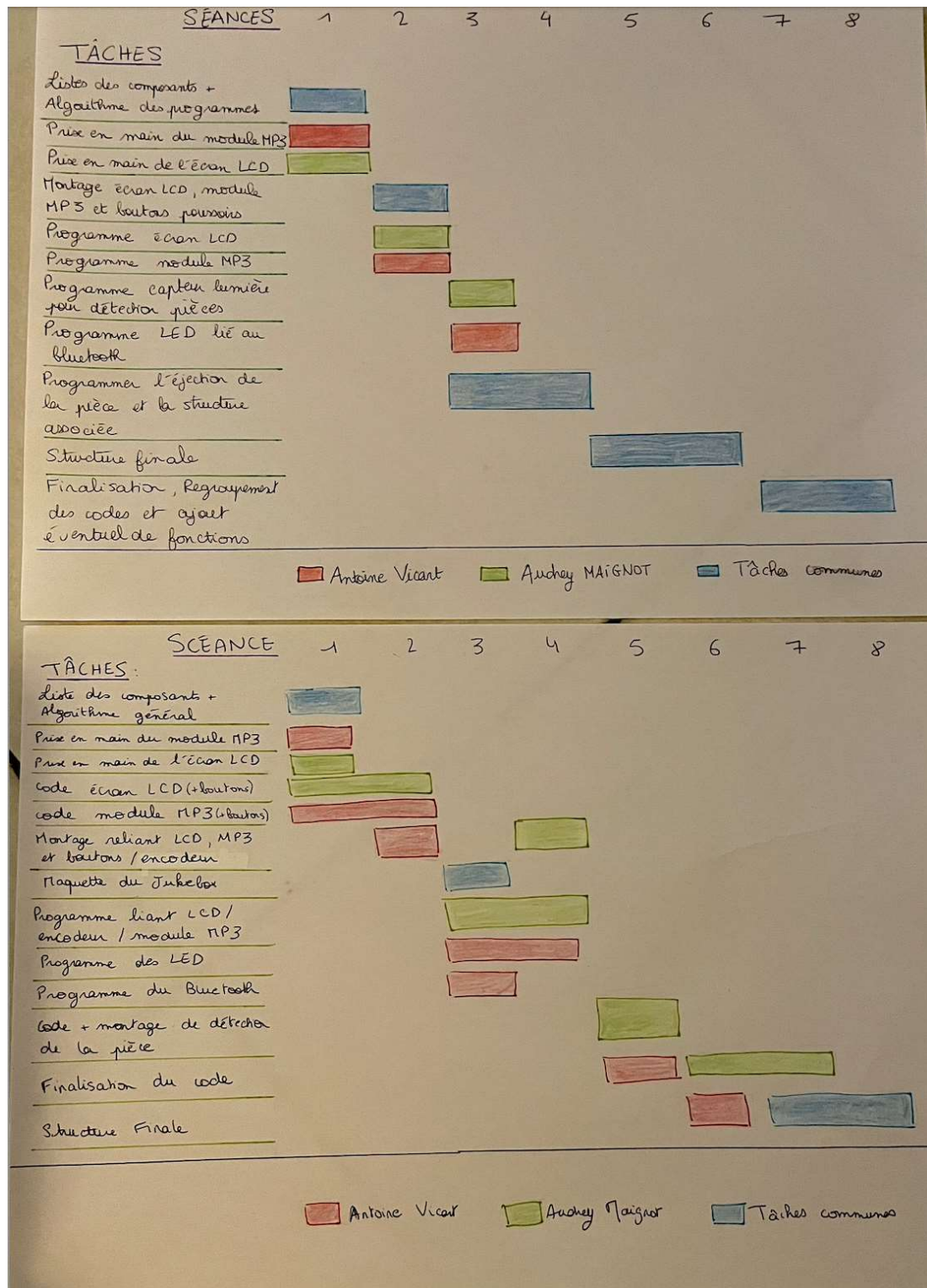


Figure 1. Plannings initiaux et finaux.

# Chapitre I : L'algorithme

Avant de commencer en détail le développement des différentes parties de notre projet, voici un organigramme expliquant tout le code permettant de faire fonctionner notre jukebox :

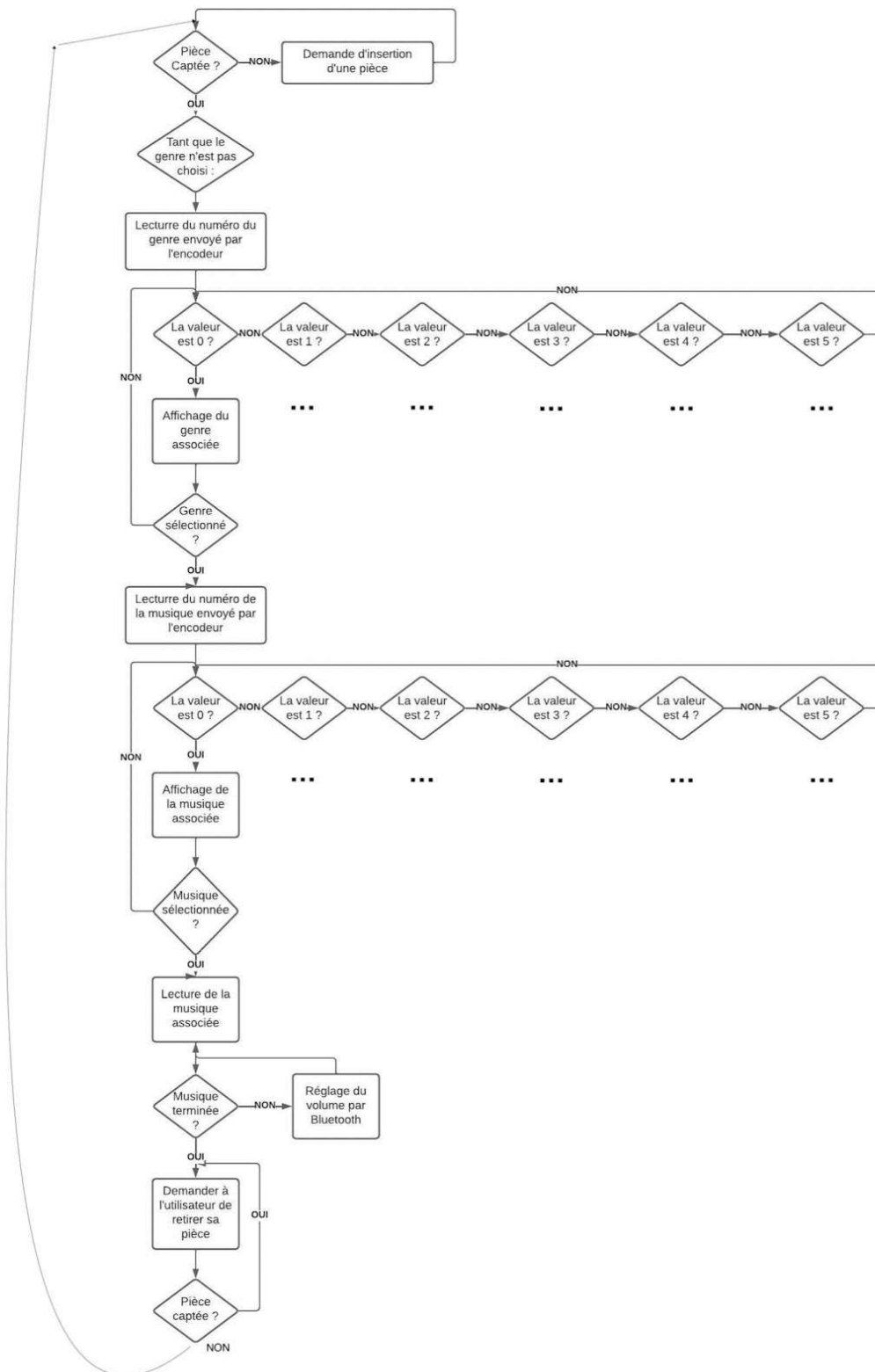


Figure I.1. *Algorithme.*

# Chapitre II : Le développement du projet

## II.1. Le menu

Nous avons commencé le projet en nous focalisant sur les parties les plus importantes du code, c'est-à-dire les liens à faire entre l'écran et l'encodeur pour accéder aux choix des musiques. À la base, nous avions pour projet d'utiliser 8 boutons poussoirs pour interagir avec le jukebox. 6 d'entre eux servaient à choisir un genre et les deux autres représentaient les boutons « OK » et « Retour ». Nous nous sommes donc lancés dans l'ébauche d'un code avec 6 boutons, le but étant de cliquer sur l'un deux, voir apparaître un genre, puis cliquer sur un autre pour voir apparaître le nom d'un autre genre. Seulement, nous avons rencontré des difficultés à établir le code puisque les boutons poussoirs n'ont pas de « mémoire ». Il fallait donc créer un code avec une mémoire pour chaque bouton. Nous n'avons pas réussi correctement ce code mais nous avons surtout eu une autre idée plus simple d'utilisation : utiliser un potentiomètre. Celui-ci regroupait alors les 6 boutons en un et minimisait le code. Nous avons donc testé des codes avec le potentiomètre, qui était raccordé au bouton poussoir « OK ». Ce bouton permettait de sélectionner le genre souhaité. Cependant, un de nos professeurs nous a suggéré d'utiliser un encodeur. Ceci permettait de combiner les deux et était, selon nous, bien plus simple d'utilisation. Nous avons donc fait un code avec un « compteur » qui s'incrémente à chaque fois que l'on tourne l'encodeur et qui prend des valeurs de 0 à 5 (pour les six genres) qui tournent en boucle. Ainsi, on crée un menu que l'on peut parcourir avant de faire son choix, en tournant la molette. Lorsque le choix est fait, il suffit d'appuyer sur l'encodeur pour sélectionner le genre souhaité. Le numéro du genre est alors enregistré. Le programme fonctionne de la même façon pour parcourir les musiques à l'intérieur de chaque genre.

En parallèle, tout ce code est relié à l'écran LCD, afin que l'utilisateur puisse voir ce qu'il se passe. La prise en main de celui-ci ne fut pas compliquée. Les câblages étaient très simples car l'écran était relié à un port I2C, ce qui réduisait considérablement le nombre de fils. Pour revenir au code, nous avons donc relié l'affichage des genres sur l'écran à la valeur du compteur, reliée à la valeur du genre. Ainsi, le code vérifie la valeur et affiche le genre associé. Nous avons rencontré un problème quant à cet affichage puisque notre texte clignotait. Il était très difficile de voir ce qu'il s'affichait. Il a fallu tout simplement vérifier la valeur du compteur et de l'ancien compteur, pour que l'écran n'affiche le texte que lorsque l'on tourne la molette, c'est-à-dire quand la valeur du compteur change. Enfin, nous avons créé des fonctions « `musiqueRock()`, `musiquePop()`,... » qui sont appelées lors du choix de la musique et permettent d'afficher le titre des musiques en fonction du genre sélectionné, auquel nous avons attribué un numéro, comme dit précédemment. Ainsi, pour chaque genre, la fonction associée fait dérouler les musiques avec le même fonctionnement que le déroulement des genres.

## II.2. La lecture des musiques

Passons maintenant à l'élément essentiel d'un jukebox : les musiques.

### II.2.1. Le module MP3 UART

Le module MP3 UART est le composant crucial permettant la lecture des musiques, toute donnée transmise relative à cette dernière passant par ce module. Mais que permet-il de faire concrètement ?

Il permet tout d'abord d'y insérer une carte MicroSD, sur laquelle se trouvent les musiques, puis, grâce à une librairie qui lui est propre, d'envoyer des commandes de lecture aux enceintes. Ces commandes peuvent être de plusieurs types dont voici un exemple que l'on a très souvent utilisé : `“sendCommand(CMD_PLAY_WITHFOLDER, 0X0101);”`. Cette commande permet tout simplement de lire la 1ère musique située dans le 1er dossier, à condition que cette musique et ce dossier se nomment tous les deux “01” sur la carte MicroSD. Suivant donc ce qu'a demandé le client, telle ou telle musique va se lancer grâce à cette commande. Ces quelques lignes de code forment les fonctions `“lireMusiqueGenre()”` qui sont ensuite appelées dans la boucle principale. Dans chaque fonction `“lireMusiqueGenre()”`, on a stocké pour chaque musique un entier dans une variable correspondant à sa durée en millisecondes. Cette variable est ensuite utilisée dans le code principal, à l'aide d'un `“while”`, pour contrôler le volume via Bluetooth tant que la musique n'est pas terminée.

### II.2.2. Le réglage du volume par le potentiomètre

Petite fonctionnalité en plus bien que quasiment indispensable : la possibilité de régler le volume des musiques grâce à un potentiomètre relié aux enceintes et au module MP3 UART. Le principe de fonctionnement est très simple : suivant l'orientation de la molette, le dispositif renvoie une valeur comprise entre 0 et 255 (PWM) et, plus la valeur est grande, plus le volume est élevé. Pour modifier à proprement parler le volume, on utilise une fonction de la même librairie que pour la lecture des musiques, la commande `“sendCommand(CMD_SET_VOLUME, V)”` permettant donc de régler le volume à une certaine intensité V.

## II.3. La connexion radiofréquence

Cette partie fait le lien avec la précédente : le réglage du volume. En effet, la connexion radiofréquence obligatoire que nous avons choisi pour notre projet consiste en l'utilisation du Bluetooth afin de pouvoir régler le volume des musiques depuis son smartphone. Nous avons utilisé l'application “Bluetooth Electronics”, étudiée préalablement en cours, en communication avec le module Bluetooth que nous avons aussi utilisé en cours. Le branchement fut donc une banalité, les quelques petites difficultés furent sur l'écriture du code. Il se trouve que pour la partie réglage du volume, le code est identique, la subtilité se trouvant sur la manière de transmettre l'information depuis le téléphone. Il se trouve que, sur le smartphone, l'interface proposée au client est une molette à bouger de gauche à droite. Quand justement elle bouge, le téléphone envoie une donnée sous forme d'une lettre au module Bluetooth qui comprend donc qu'il doit exécuter la partie du code sur la modulation du volume.

Cependant, il nous est fort regrettable de vous apprendre que tout cela ne marche pas. En réalité, tout fonctionnait jusqu'à la 8ème séance de projet puis ensuite plus rien. Ce qui est particulièrement dérangeant, c'est qu'il nous était totalement impossible de régler ce problème vu que, comme tout fonctionnait la semaine d'avant, il ne peut pas s'agir d'un problème lié au code et, comme le téléphone se connecte toujours au module Bluetooth, il ne s'agit pas non plus d'un problème lié aux branchements...

## II.4. La guirlande LED Adafruit NeoPixel

Voici maintenant l'élément esthétique que l'on a voulu rajouter à notre jukebox pour bien entrer dans le thème des années disco.

### II.4.1. Idées initiales

Cette première sous-partie est dans un sens liée à la grande partie précédente puisque, comme rien n'est jamais aussi facile qu'on ne le pense, la fonctionnalité Bluetooth n'était pas destinée au volume en premier abord, mais au contrôle de la guirlande LED de la marque Adafruit modèle NeoPixel. Cette guirlande permet très aisément de programmer chacune des LEDs la composant. L'objectif initial était donc que, suivant le genre dans lequel le client choisit une musique, un certain thème lumineux se lance sur la guirlande. La fonctionnalité Bluetooth permettait quant à elle de changer par n'importe quel autre thème le thème en cours depuis son smartphone. Cette dernière idée a vite été changée par la fonctionnalité Bluetooth que vous connaissez, mais la première est restée longtemps dans nos esprits. Seulement un seul problème, de taille certes, mais uniquement un nous a fait abandonner : la guirlande bloquait le bon déroulement du code, ce dernier étant trop volumineux pour une seule carte Arduino.

### II.4.2. Résultat final

Finalement, nous avons séparé le code de la guirlande du code principal en le mettant sur une seconde carte Arduino, entièrement indépendante donc de la première. Afin de parer à ce petit échec de ne pas avoir pu réaliser ce que l'on voulait initialement, nous avons bien amélioré les différents thèmes lumineux, chose relativement simple grâce à la possibilité de programmer LED par LED ce que l'on veut. On a enfin mis tous les thèmes à la suite qui tournent en boucle à partir du moment où le jukebox est sous alimentation.

Cependant, problème ici-aussi, pas aussi grave toutefois que pour le Bluetooth : dû encore à un bourrage de la mémoire de la 2<sup>de</sup> carte Arduino, la guirlande s'arrête nette au bout d'un certain temps avec seule possibilité pour la relancer de débrancher son alimentation personnelle et la rebrancher.

## II.5. Le monnayeur

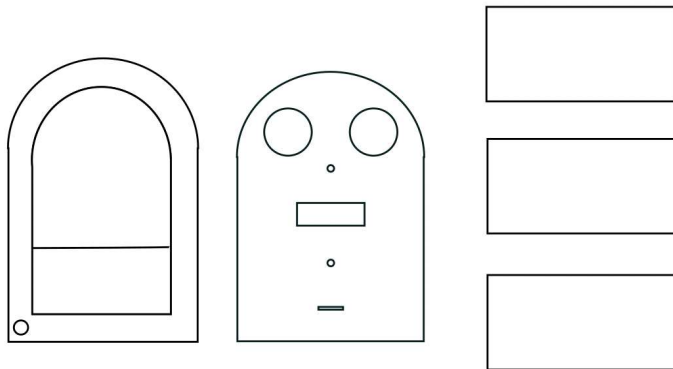
Après avoir terminé ces parties de code conséquentes, qui sont au cœur du jukebox, nous nous sommes penchés sur le captage de la pièce, qui permet à l'utilisateur d'accéder à tout le code décrit précédemment. Ici, la partie compliquée était de lier le captage de la pièce, donc le code, avec une structure adaptée. Nous avons donc pris en main un capteur de distance et nous nous sommes lancés dans une petite boîte en carton pour faire des tests, avant d'avoir la structure finale. Nous avons découpé une encoche pour insérer la pièce et nous avons percé le dessous de la boîte pour laisser passer le capteur. D'ailleurs, la structure finale de cette boîte fonctionne de la même façon. Nous avons très vite pris en main cette partie, que nous avons ensuite reliée au code principale expliqué précédemment. Nous avons rajouté quelques lignes de codes, reliées à l'écran LCD, demandant à l'utilisateur d'insérer une pièce, tant que celle-ci n'est pas captée. Pour cela, nous avons réglé une distance à capter. Par exemple, tant que la distance captée est supérieure à 60, l'écran LCD affiche un message disant d'insérer une pièce. Ensuite, nous avons dû nous occuper de la récupération de la pièce. Dans l'idéal, nous aurions voulu faire un système d'éjection de la pièce à la fin de la lecture de la musique. Cependant, cette partie aurait été trop conséquente par rapport au temps qu'il nous restait. Nous avons donc fait autrement. À la fin de la musique, un message demandant à l'utilisateur de retirer sa pièce s'affiche sur l'écran. Tant que celui-ci ne l'a pas fait, l'utilisateur ne peut accéder à rien et ne peut donc pas choisir d'autres musiques. Pour lier ce code à une véritable structure, nous avons découpé au laser une petite boîte en bois. Nous n'avons pas fixé la partie arrière de la boîte, de façon à



pouvoir l'ouvrir facilement pour récupérer la pièce. Cette boîte a été fixée à la face avant du jukebox, au niveau de l'encoche d'insertion de la pièce.

## II.6. La structure

Venons-en donc à la structure finale. Au total, le jukebox compte 10 pièces présentées ci-dessous (sans compter le "monnayeur"), que nous avons modélisées à l'aide du logiciel Inkscape. Ne connaissant pas ce logiciel, il a fallu le prendre en main assez rapidement mais ce problème a été surmonté sans grandes difficultés.



La première pièce à gauche représente la face arrière. Le contour représente la partie fixe, qui est trouée en bas à gauche afin de faire passer les fils d'alimentation. Les deux pièces du centre sont des portes qui permettent d'accéder à l'intérieur du jukebox. Celle du haut permet d'accéder à tout le circuit électronique et celle du bas à la boîte de récupération de la pièce, qui a été introduite précédemment.

La pièce du centre est la face avant du jukebox. Nous avons fait plusieurs trous afin de laisser passer les enceintes, le bouton réglage du volume, l'écran LCD, l'encodeur et la fente pour la pièce. Ces deux premières pièces ont été consolidées avec trois arceaux intérieurs de la même forme que celui de la face arrière. Deux d'entre eux ont été collés à la face avant. Le premier pour consolider et le deuxième pour venir coller la guirlande de LED sur le dessus, c'est pourquoi ses dimensions sont plus petites de quelques millimètres. Le dernier arceau a été collé à la face arrière. Pour une question pratique et pour encore plus de consolidation, nous avons ajouté des étagères internes au jukebox (les pièces de droite). En effet, celles-ci permettent au deux faces de ne pas s'affaisser mais elles sont surtout là pour tenir tous les composants électroniques. Celle du bas est en fait le socle inférieur, où les faces avant et arrière sont vissées, et où est posée la carte arduino gérant les LEDs. Celle du milieu supporte la carte Arduino gérant tout le code et la plaquette sur laquelle sont reliés les 5V et les GND de tous les composants et celle du haut supporte le module MP3 et le module Bluetooth.

Enfin, la structure est recouverte de planches de plastique blanc thermoformable, qui diffusent la lumière des LEDs pour rendre un meilleur effet. Nous avons rencontré un problème quant à la taille des planches, qui étaient trop petites. Nous en avons donc collé trois et, pour ne pas voir les jointures, nous avons collé des petites lamelles de bois par dessus. Nous avons aussi customisé le jukebox en imprimant des étiquettes d'indication comme "CHOIX" ou encore "VOL." afin d'indiquer à l'utilisateur le rôle de chaque élément. Nous avons aussi peint en blanc toute la structure pour qu'elle soit assortie au plastique blanc et pour ne plus voir les traces de bois brûlé par la découpeuse laser.

# Conclusion

Pour finir, nous allons vous résumer ce qui fonctionne et ce qui ne fonctionne pas, maintenant que nous avons acquis de l'expérience, ce que l'on ferait autrement, et ce que l'on aimerait ajouter à notre jukebox si nous avions 8 séances supplémentaires.

Pour commencer, le menu, la lecture de la musique, le réglage du volume par le potentiomètre et le "monnayeur" fonctionnent, et c'est bien un minimum pour un jukebox. Mais voilà, tout ce qu'on a voulu ajouter comme fonctionnalités bonus, bien que la connexion radiofréquence était obligatoire, fonctionne ou fonctionne à moitié. Nous avons une guirlande LED qui s'arrête, se fige au bout d'un certain temps, cela étant dû à un trop-plein de mémoire dans sa carte Arduino et une fonctionnalité Bluetooth qui, sans qu'on en ait trouvé la raison comme expliqué dans la partie qui lui est consacrée, ne fonctionne pas. Donc, qu'est ce que l'on aimerait ajouter à notre jukebox si nous avions 8 séances supplémentaires ? Premièrement, régler les deux problèmes ci-dessus : essayer de brancher la guirlande à une carte Arduino MEGA, cette dernière possédant plus de mémoire, et déceler le problème lié au Bluetooth. Ensuite, il est assez évident que rajouter un bouton "Retour" ferait partie des fonctionnalités à ajouter les plus utiles, comme un vrai monnayeur avec éjection de la pièce et muni du balance permettant d'accepter qu'une certaine somme pour lire une musique. Nous pourrions très bien aussi rajouter des musiques, voire même des genres, ou encore permettre de choisir d'écouter la musique que l'on veut en se connectant en Bluetooth et en lisant une musique depuis son téléphone... Il pourrait y avoir encore plein de choses à ajouter, même pour un objet aussi simple qu'un jukebox.

Mais, concrètement, avant de penser à ce qu'on pourrait rajouter, qu'est ce que l'on modifierait ? Nous pourrions opter pour un écran plus grand et de meilleure qualité, de même pour les enceintes, faire ce que l'on voulait au départ concernant la guirlande LED, c'est-à-dire qu'un thème lumineux différent se lance suivant le genre dans lequel on se met. Vraiment, il y aurait encore plein de choses à faire.

Voici donc ce qui conclut ce rapport de projet, merci à vous d'avoir lu jusqu'au bout.

## BIBLIOGRAPHIE

- [\[GUIDE\] Piloter un afficheur LCD 16x2 I2C - Arduino France](#)
- [Gestion d'un bouton poussoir avec Arduino • AranaCorp](#)
- [Rotary Encoder Using Arduino Hardware Interrupts](#)
- [Improved Arduino Rotary Encoder Reading : 4 Steps \(with Pictures\) - Instructables](#)
- [Playing music in Arduino using SD Card Module — Steemit](#)
- [Catalex\\_MP3\\_board.pdf](#) (fourni par Mr MASSON)
- [Bluetooth Electronics \(keuwl.com\)](#)
- [Arduino Library Use | Adafruit NeoPixel Überguide | Adafruit Learning System](#)
- [Adafruit NeoPixel Arduino Code Examples with Simulation \(wokwi.com\)](#)
- [Tweaking4All.com - Arduino - LEDStrip effects for NeoPixel and FastLED](#)