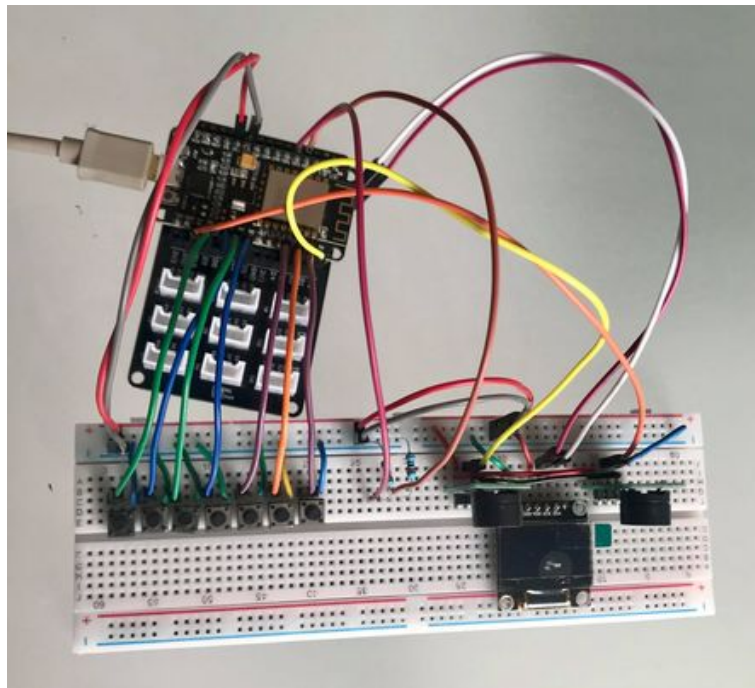


## Bureau d'Etude C++: A One Man Band

May 18, 2021

---



**Réalisé par :**

Javier      MORALES  
Rebekka    LIE

**Présenté à :**

DEAU    Raphaël

# Introduction

Dans ce bureau d'étude, nous avons créé un "one man band" ou un loop station permettant d'enregistrer des boucles musicales en direct en appuyant sur les boutons - comme si nous étions plusieurs à jouer dans un band en même temps. En effet, l'utilisateur peut enregistrer une mélodie avec un petit clavier fait à partir de petits boutons. Puis, il peut enregistrer une nouvelle mélodie par dessus, et ainsi de suite. Finalement, il sera possible de jouer toutes les mélodies enregistrées en même temps, ce qui peut même donner lieu à des accords.

## 1 Conception

### 1.1 Matériel utilisé

Afin de pouvoir faire marcher le OneManBand, nous nous sommes servis d'une carte Arduino ESP8266, et de plusieurs capteurs et actionneurs. D'abord, sept boutons servent comme clavier afin de pouvoir jouer la note voulue. Ensuite, les buzzeurs permettent de reproduire les différentes mélodies. Cependant, ce type de n'accepte que des signaux carrés, ce qui rend impossible la reproduction de plusieurs mélodies sur un même buzzeur. Il est donc nécessaire d'en avoir un pour chaque nouvelle mélodie. Accessoirement, un LED intégré dans la carte indique à l'utilisateur qu'une mélodie est en train de s'enregistrer. Finalement, un système de LED clignotant pourra être implémenté dans le futur, ainsi qu'un écran LCD.

### 1.2 Diagramme de classe

La figure 1 montre le diagramme de classe de notre projet. La classe OneManBandApp est la classe qui constitue la couche d'application. La classe Melody est constitué d'un tableau de notes, et elle permet d'enregistrer une mélodie, ainsi que de reproduire toutes les mélodies enregistrer avec une méthode statique. La classe Device est une classe virtuelle qui représente un élément de hardware quelconque. Les classes Button, Buzzer, et LED héritent donc de la classe Device. Elles permettent d'initialiser les éléments de hardware et de configurer le microcontrôleur afin de pouvoir les piloter correctement. La classe LED est implémenté mais pas utilisé car il manque des pins sur la carte. La classe Display n'est pas encore implémenté, et peut constituer une amélioration dans le futur.

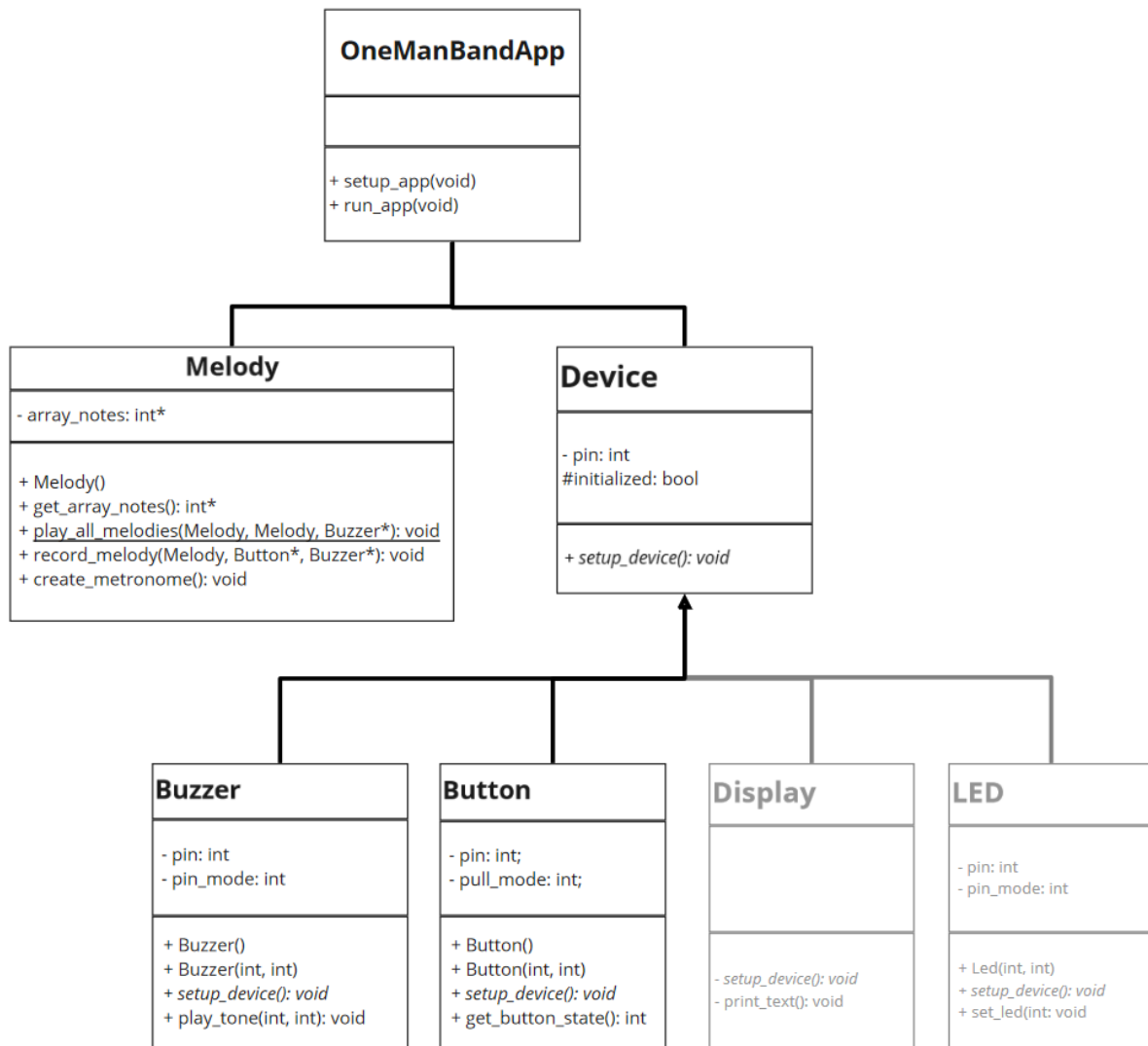


Figure 1: Diagramme de classe de OneManBand

### 1.3 Diagramme de séquence

La figure 2 montre le diagramme de séquence de l'application. Lorsque l'objet de OneManBand appelle la méthode `setup_device`, l'Arduino est configuré pour utiliser les boutons et les buzzers. Après, il appelle la méthode `run_app`, qui va enregistrer les mélodies. D'abord, la mélodie appelle la fonction `get_button_state()` pour savoir quel bouton a été appuyé. Ceci est fait tout les 10 ms. Puis, la méthode `play_tone()` permet de reproduire les notes enregistrés tout les 10 ms sur les buzzers. Finalement, lorsque toutes les mélodies ont été enregistrées, la méthode `play_all_melodies()` permet de reproduire toutes les mélodies.

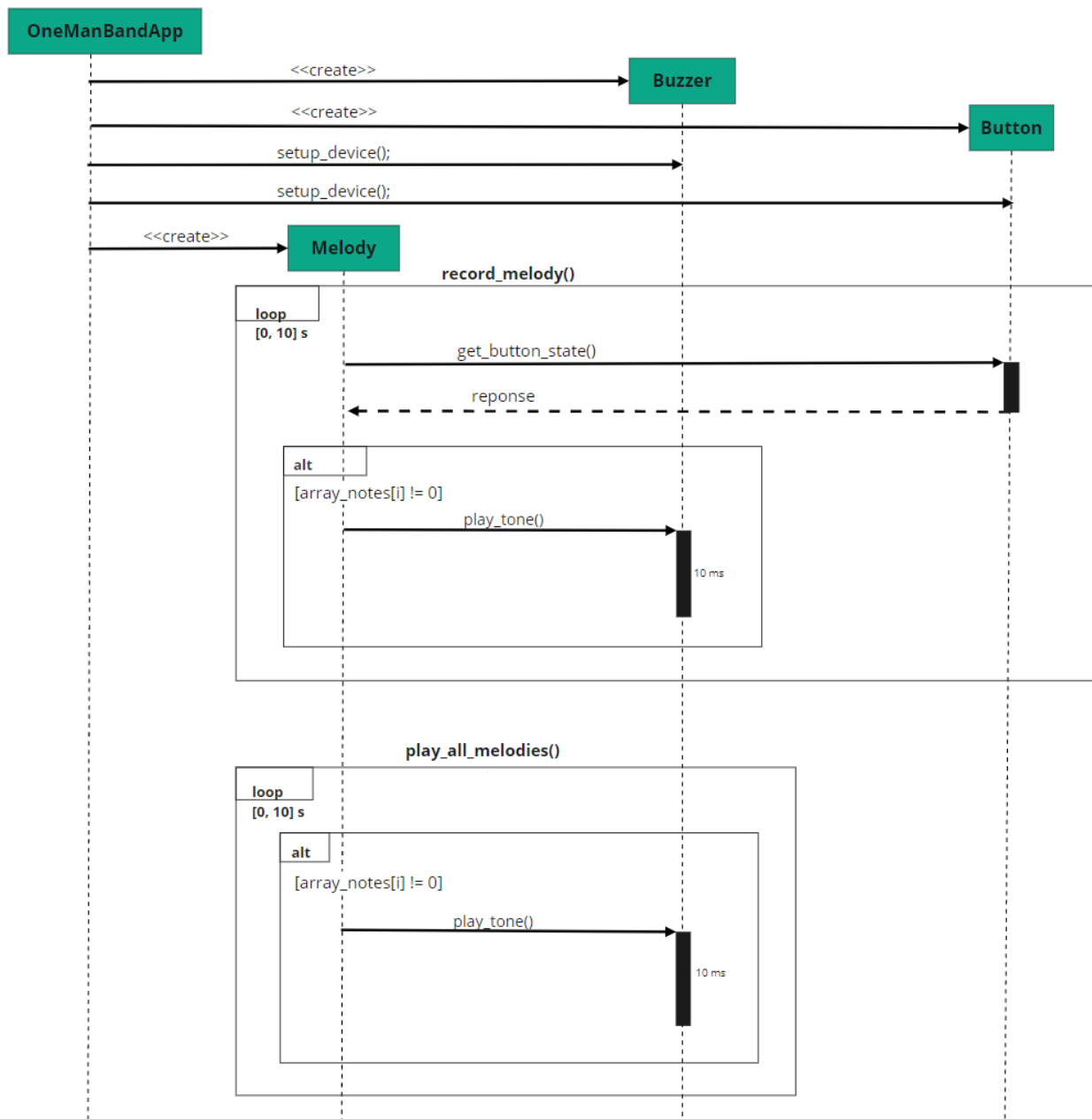


Figure 2: Diagramme de séquence de OneManBand

## 2 Conclusion

Ce BE nous a permis de mettre en pratique des compétences acquises au sein des TDs et cours sur un projet réel choisi par nous-mêmes. Nous avons bien aimé la liberté sur le choix du projet, ainsi de pouvoir mettre en place une maquette en réel.

Les soucis rencontrés au cours du BE étaient surtout par rapport à la configuration de l'IDE d'Arduino. De plus, on a eu quelques problèmes avec la librairie STD. On avait prévu au départ de structurer les notes d'une mélodie à l'aide d'une liste. Cependant, on a eu beaucoup de soucis lors du séquençage pour la stockage et la reproduction des mélodies. C'est pour cela qu'on a décidé simplement d'utiliser des arrays de longueur fixe, ce qui permet de parcourir plusieurs mélodies simultanément avec une période fixe. On a aussi rencontré une petite difficulté lors de

l'implémentation des LED. En effet, les boutons et les buzzeurs ont occupé presque tous les pins disponibles de la carte, ne laissant plus de place pour des LEDs. Ceci pourrait être résolu en configurant plusieurs boutons sur un même pin avec un multiplexeur. Le même problème peut être appliqué pour le display, qui n'a pas été implémenté.

Malgré quelques petites difficultés, nous avons été capables d'atteindre notre but initial. En effet, nous sommes très satisfaits du résultat final, même si l'étendue du projet a dû être légèrement réduite. Nous sommes restés très motivés tout au long du projet.