

# Colocarina



## Projet Arduino Peip2 2017/2018

*Par les talentueux BERTIN Marylou et GUARDIOLA Jérémy*

(Projet totalement inspiré du youtuber Allen Pan - Sufficiently Advanced )

# Sommaire

## I. Introduction

- Objectifs
- Problématique
- Matériels

## II. Réalisation du projet

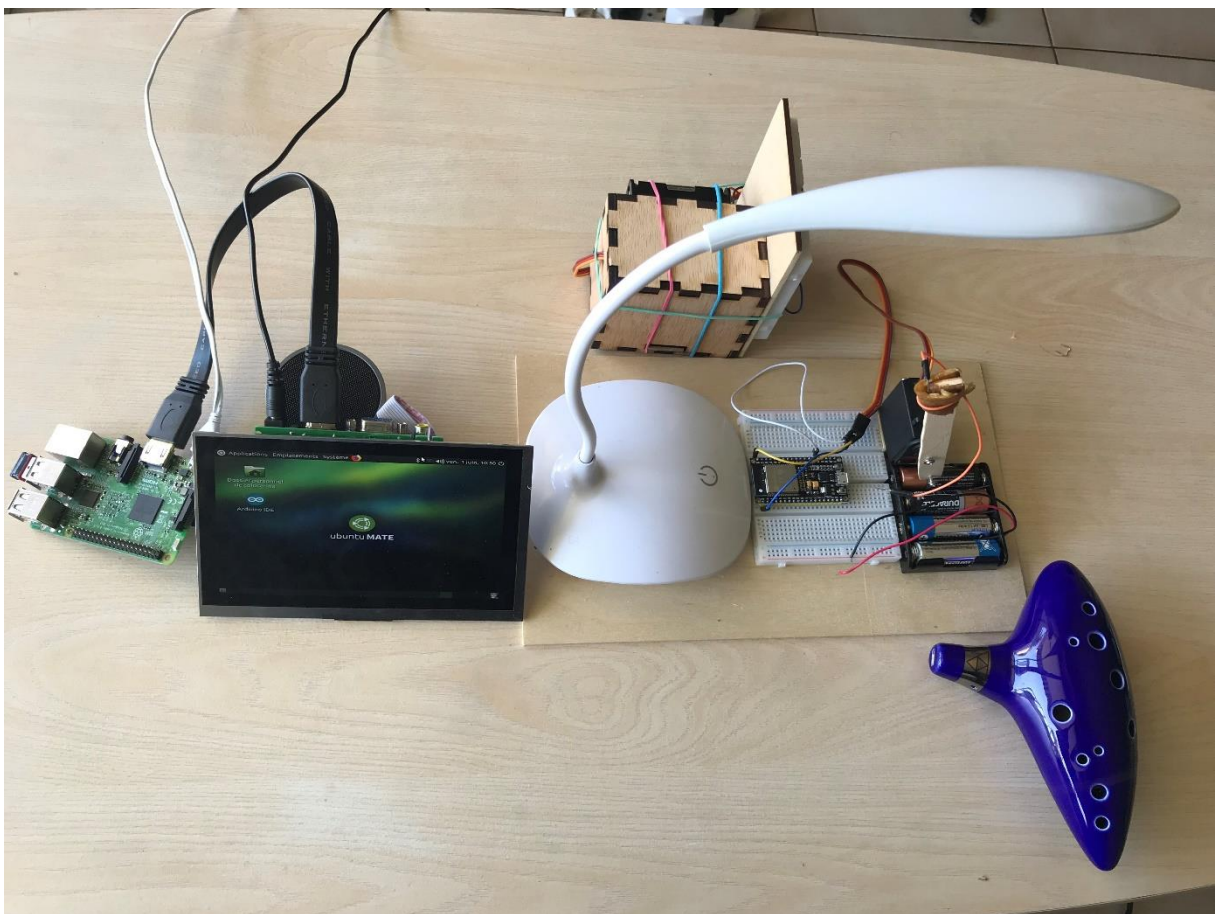
- Montage
- Programmes
- Problèmes rencontrés

## III. Conclusion

- Réussite
- Echs et déception
- Remerciements (pour les meilleurs)

# Introduction

Lors du choix du projet Arduino, nous nous sommes concertés pour choisir un sujet qui pourrait être ludique, intéressant et qui toucherait à notre passion mutuelle : la musique. Nous sommes arrivés après de longues recherches sur un projet complexe mais très amusant, celui d'une borne domotique réagissant à la musique joué par un Ocarina que nous avons baptisé, en l'honneur de notre colocation, le Colocarina. Notre but étant de créer un outil de domotique qui faciliterait la vie et réagissant à la musique.



# **Problématique**

*Comment mêler musique et électronique dans un projet ludique ?*

## **Matériels utilisés :**

- Carte Raspberry pi3
- 2 x Carte Wifi ESP32
- Ocarina
- Micro statique
- Lampe
- 2 x Servo-Moteur
- Verrou
- Batterie
- Piles + Coupleurs
- Ecran LCD
- Clavier Bluetooth
- Plaques de branchements

# Réalisation du projet

Notre projet correspond à une borne fixe composé d'un micro et d'une carte raspberry pi3, qui reçoit une information fréquentielle produite par un ocarina, et selon la musique que l'on joue, la carte raspberry redirige vers un module grâce à une connexion et exécutera la tâche demandée. Nous avons donc créé les différents modules.

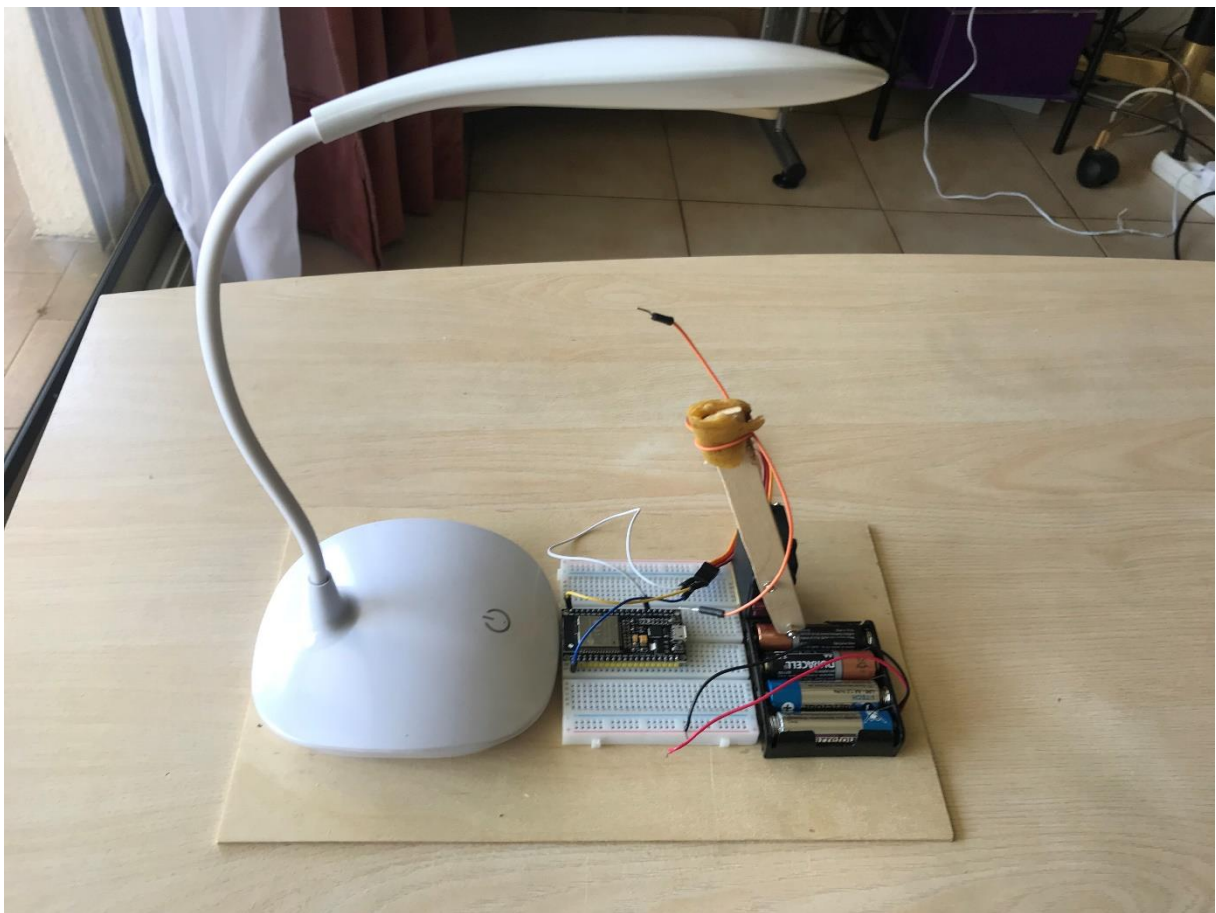
Dans un premier temps, M. Masson nous avait conseillé de créer un ocarina électronique au cas où notre analyse fréquentielle ne fonctionnerait pas, mais cette phase intermédiaire nous a fait perdre énormément de temps à cause de problèmes informatiques. Après concertation 1mois et demi après, nous nous sommes dit que c'était une perte de temps sachant que nous possédions déjà tous les programmes. Dans l'idée, il ne nous restait qu'à créer les modules et à agencer les programmes pour que la connexion wifi fonctionne.

## Montage

Nous nous sommes rendus au FabLab pour créer le module pour la lampe ainsi que pour le verrou.

Pour cette première, il fallait créer une plaque sur laquelle on collerait la lampe tactile et positionnerait le servo-moteur de tel sorte que lorsque l'on activerait la rotation, la barrette

collée dessus taperait sur le bouton tactile. Mais il fallait trouver une matière tactile pour pouvoir actionner la lampe (s'arracher un doigt n'étant pas la solution). Après essai de plusieurs matériaux dans notre appartement, l'objet le plus tactile fut la banane. Nous avons donc collé un morceau de peau de banane sur le bout de la barrette. Mais nous sommes conscients que la banane n'est pas une matière durable dans le temps (sachant que cela pourri en quelques heures). Ainsi, à chaque fois que le module est activé, le servo-moteur tourne et cette rotation allume la lampe. Nous avons également fixé la carte wifi et la source de courant sur cette plaque.



*Le module lampe*



Pour le verrou, il a fallu utiliser la découpe laser pour fabriquer une petite boîte dans laquelle on colle le verrou d'un côté et le servo-moteur de l'autre. La boîte est pile poil ajustée pour la taille de ces deux composants. On a également fait une découpe pour que le loquet puisse sortir de la boîte. Ainsi, lorsque l'on active le module, le servo-moteur se met en rotation et verrouille (ou déverrouille) la porte. On a également fixé la carte wifi ainsi que la source de courant sur la boîte. La boîte est ensuite fixée sur une porte et on visse la barre sur le mur à côté de la porte, qui permet de coincer le loquet du verrou, pour fermer la maison.



*Le module verrou*

# Programmes

Après avoir fabriqué les modules, il a fallu agencer les programmes pour que la musique de l'ocarina soit entendue, analysée, puis retranscrit au module correspondant. Nous avons donc créé des petites mélodies, analysé grâce à `audacity` la fréquence de chacune des notes jouées et enfin rajouté une marge d'erreur qui respecte les tons/demi-tons de chacune des notes. Ensuite nous avons téléchargés tous les packages nécessaires aux programmes que cela soit pour le programme python ou pour le programme Arduino. Puis, nous avons téléversés le programme Arduino dans les des cartes Wifi de chaque module. Ce programme permet, en fonction de la musique renvoyé par le broker de la carte raspberry (MQTT), d'activer la carte Wifi correspondante. Enfin, nous avons modifiés les fonctions des deux servo-moteurs pour qu'il fasse l'action demandée.

```

75 #*minA4 = A-BANDWIDTH
76 #*maxA4 = A+55
77
78
79 # Song note sequences
80 Suh = deque(['E', 'D', 'B', 'A', 'B', 'E'])
81 Time = deque(['A', 'E', 'A', 'D', 'A', 'C'])
82 Saria = deque(['D', 'E', 'B', 'A', 'B', 'A'])
83 Zelda = deque(['C', 'A', 'D', 'C', 'A', 'D'])
84
85 Heal = deque(['B', 'A', 'E', 'B', 'A', 'E'])
86 Test = deque(['C', 'A', 'C', 'A', 'C', 'E']) #Not a Zelda song
87 #heard note sequence deque
88 notes = deque(['C', 'C', 'C', 'E', 'C', 'C'], maxlen=6)
89
90 # Show the most intense frequency detected (useful for comparing)
91 frequencyOutput=True
92 freqNow = 420

```

## Programmation des musiques

[illegible]

### Détection active des notes



# Problèmes rencontrés

On peut dire qu'il y en a eu bien trop ! Reprenons tout cela dans un ordre chronologique. A l'époque où nous avons essayé de faire un ocarina informatique, nous avons eu énormément de problèmes d'informatique, les idées étaient là mais notre niveau en informatique ne nous permettait pas de les coder. Ensuite, nous nous sommes rabattus sur la carte Raspberry mais encore une fois, même en suivant les tutoriels, rien ne marchait. Nous avons donc changé d'écran car il était dysfonctionnel et d'autres peip2 utilisant une carte raspberry nous ont aidés à programmer la carte raspberry. De là, nous nous sommes dit que ça irait bien vite, qu'il suffirait de télécharger arduino et quelques bibliothèques et ça serait bon, mais non. Les problèmes ont commencé par un crash de firefox, impossible à réparer. Nous avons donc dû rebooter la carte à neuf. Ensuite ça a été la librairie esp32 qui n'a pas voulu fonctionner sur cette carte raspberry. Nous étions focalisés sur l'utilisation de la carte raspberry sans réaliser que nous pouvions très bien programmer les cartes ESP32, sur un ordinateur. Et en effet, après de nombreuses erreurs de compilation, le programme a bien voulu fonctionner. Le dernier de notre souci a été que le programme ne voulait pas être envoyé sur les différentes cartes wifi sans être branché à un ordinateur. C'est par une dernière recherche internet que nous nous sommes rendu compte qu'il fallait les brancher sur le 3,3volts au lieu du 5volts. Et enfin le programme final a fonctionné, réagissant parfaitement à chacune des deux musiques.

# Conclusion

Notre projet a finalement été fonctionnel, pas aussi vite que prévu ce qui ne nous a pas laissé le temps de s'occuper de deux de nos modules prévus à la base : faire dire l'heure à un réveil parlant et l'appel de son téléphone. De plus, nous avons possiblement prévu d'activer des leds en fonction de la note joué, ce qui n'a pas pu être fait. Nous n'avons pas non plus eu le temps de paramétrer la fonction heal, qui permettait de tout remettre à 0 quand on quitte son domicile. De plus le son de confirmation n'a pas voulu fonctionner à cause du port jack de la carte raspberry visiblement dysfonctionnel (il n'a pas reconnu l'enceinte).

Mais cela nous a permis de travailler en groupe pour mener à bien un projet qui nous tenait à cœur, de réfléchir ensemble pour régler les problèmes, découvrir de nouvelles applications dans le domaine de l'électronique grâce à la carte raspberry ainsi que les cartes ESP32. Il nous a permis de développer notre organisation, notre autonomie ainsi que notre créativité.



# Remerciements

Nous tenons à remercier l'école Polytech Nice/Sohia d'avoir pu nous financer et nous permettre de développer un projet qui à notre sens, est la meilleure expérience de toute la PeiP. Nous remercions également le FabLab qui nous a permis de découvrir de nouvelles technologies et nous laissés accéder à leur atelier en nous aidant quand nous avons besoin. Et surtout nous tenons à remercier nos deux merveilleux intervenants qui nous ont permis de faire ce projet éducatif, qui n'aurait pas eu lieu sans eux.

# Annexes

Programmes disponibles sur notre Github.

