

Projet Arduino – PeiP2

Année 2021-2022

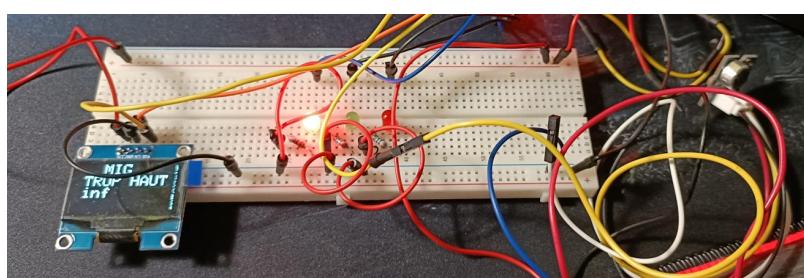
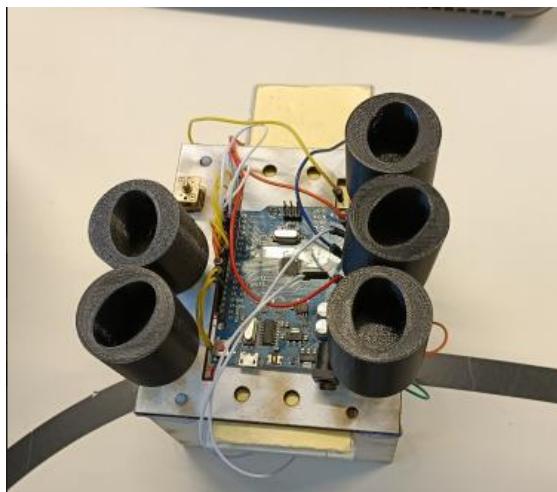
PROJET : Accordeur automatique de guitare.

Étudiants : Par MISSERI Marco et SERAFINO Jean-Adelin.

Encadrants : Pascal MASSON,

Christian PETER,

Amina BENOUAKTA.



Ecole Polytechnique Universitaire de Nice Sophia-Antipolis,
1645 route des Lucioles, Parc de Sophia Antipolis, 06410 BIOT

SOMMAIRE :

- **Introduction.**
- **Objectifs :**
 1. Cahier des charges.
 2. Ce qui a finalement été fait.
- **Vision globale du projet:**
 1. Présentation des différents modules.
- **Développement des parties du projet.**
- **L'algorithme.**
- **Les plannings:**
 1. Planning initial.
 2. Planning final.
- **Problèmes.**
- **Conclusion-perspectives.**
 1. Ce qui marche.
 2. Ce qui ne marche pas.
 3. Avec neuf séances de plus...
- **Bibliographie.**

I.Introduction:

Le projet Accordeur automatique de guitare est constitué de 2 parties:

-La première représente un accordeur traditionnel de guitare.

-La seconde est le module permettant d'accorder automatiquement la guitare.

Ces deux parties sont reliées entre elles par une connexion Bluetooth.

Nous allons donc dans ce rapport présenter les différentes parties de ce projet, tout en explicitant les tenants et les aboutissants.

II.Objectifs:

1. Cahier des charges:

RAPPEL:

Dans le cahier des charges réalisé fin novembre, nous avions prévu qu'à la fin des huit séances notre projet serait, en théorie, capable d'accorder automatiquement une guitare. Pour ce faire nous avions prévu:

1) Dans un premier temps, de réaliser un accordeur traditionnel de guitare composé d'un fréquencemètre, d'un programme permettant de traiter la fréquence et aussi d'une fonction affichage complété par des LED indiquant l'opération à effectuer: tendre ou détendre la corde.

2) Dans un second temps, de réaliser un module qui se fixerait directement sur la tête de la guitare pour l'accorder seule en fonction de la fréquence captée transmise par Bluetooth par l'accordeur de guitare.

2. Ce qui a finalement été fait:

Dans sa finalité, notre produit respecte globalement le cahier des charges :

-Il est capable dans un premier temps de capter une fréquence (Son).

-En fonction de la fréquence captée, d'indiquer si l'on doit tendre où détendre la corde.

-La connexion Bluetooth entre l'accordeur et le module d'accordage est établie.

-Les moteurs tournent en fonction de la fréquence captée.

En revanche, pour des raisons que nous expliciterons dans la partie problèmes de ce rapport, nous avons malheureusement dû supprimer la partie affichage (écran) que nous avions auparavant réalisé.

De plus, pour des raisons de manque de temps que nous verrons dans la partie problèmes, il nous a été compliqué de réellement tester la partie qui accorde automatiquement la guitare.

III. Vision globale du projet:

1. Présentation des différents modules:

a) L'accordeur:

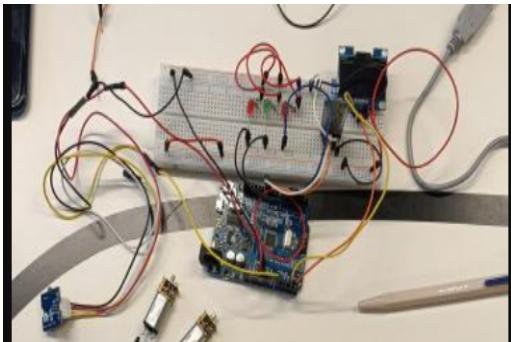
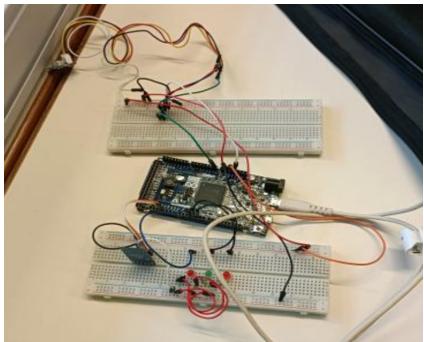


Photo 1: Montages de l'accordeur.

Pièce de l'accordeur.

- Sur les photos ci dessus on retrouve:
 - Le fréquencemètre (accompagné du module Grove-Loudness Sensor Noise).
 - Les LED participant à la fonction affichage du projet.
 - Le module Bluetooth HC05.
 - La pièce de l'accordeur.
 - L'écran que nous avons finalement enlevé.

b) Le module qui accorde automatiquement la guitare:

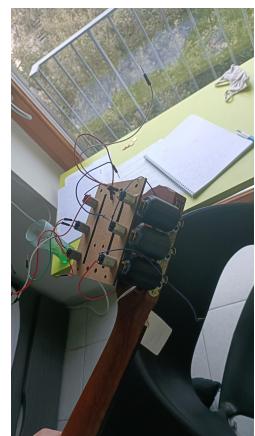
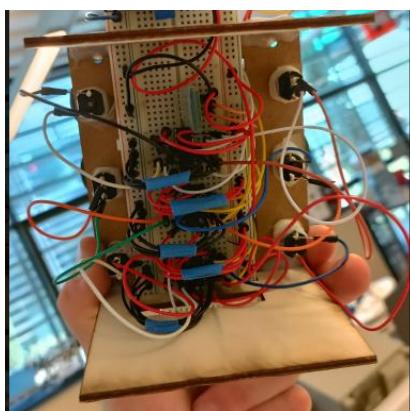
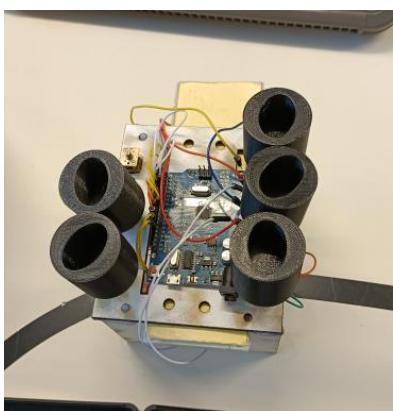


Photo 2: Pièces qui accordent automatiquement la guitare.

Pièce des supports moteurs

- Sur les photos ci-dessus on retrouve:
 - Les 6 moteurs.
 - Les puces L293D.
 - Le module Bluetooth HC06.
 - Les molettes en impression 3D.
 - La pièce du support moteur.

IV. Développement des parties du projet:

1. Le fréquencemètre:

Le fréquencemètre est l'une des pièces moteurs du projet. Pour le réaliser, nous avons utilisé un micro Sensor pour arduino branché dans un premier temps sur une carte Arduino Uno. Comme cela ne fonctionnait pas, nous avons rajouté suite au conseil de Mr Masson un pont diviseur entre la carte et le Micro afin d'amplifier le signal. De plus, nous avons changé la carte: En effet, afin d'avoir une fréquence d'échantillonnage plus rapide et plus grande nous avons finalement utilisé une carte Arduino Due. Enfin, au niveau du code nous avons utilisé la bibliothèque "Arduino FFT.h". Ce code n'est pas à nous, nous l'avons trouvé sur un site internet mais nous y avons apporté quelques modifications:

Nous avons réduit la fréquence d'échantillonnage, ajouté une boucle "while" dans une boucle "for" avec un delay permettant de contrôler le temps entre chaque itération et enfin compléter le programme avec d'autres éléments de la bibliothèque "Arduino FFT.h".

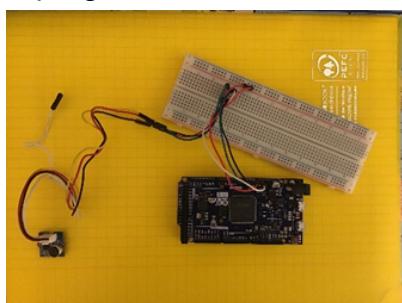


Photo 3: Montage du fréquencemètre.

2. La fonction affichage:

La fonction affichage a pour but de donner une indication à l'utilisateur sur l'opération à effectuer s'il ne possède pas le module qui accorde automatiquement la guitare. Au niveau du montage elle n'est composée que de trois LED et d'un écran. Le code est assez simple à comprendre: la fréquence captée est comparée aux fréquences de références de chaque cordes. Si la fréquence captée est trop grande (Photo 4) on affiche à l'écran "TROP HAUT" et l'une des LED rouge s'allume indiquant à l'utilisateur qu'il faut détendre la corde. Il en est de même si la fréquence est trop basse (cas analogue), enfin si la corde est juste on affiche à l'écran "Ok", la LED verte s'allume et on passe à la corde suivante. Comme expliqué précédemment nous avons malheureusement dû supprimer l'écran suite aux problèmes que nous avons eus. La fonction affichage de notre projet est donc dans sa finalité uniquement composée des LED.

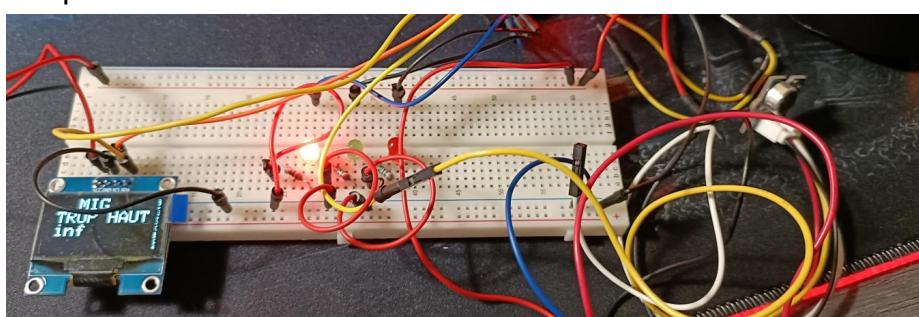


Photo 4: Fonction affichage du projet.

3. Le bluetooth:

Pour le bluetooth nous avons utilisé les modules HC05 et HC06 afin d'assurer la transmission de données entre l'accordeur et le module qui accorde:

Le premier module envoie au deuxième une variable “t” et une variable “n”. La variable “t” détermine le sens de rotation des moteurs, cette dernière peut prendre trois valeurs 6,7,8 et “n” détermine le moteur à faire tourner (“n” peut prendre 6 valeurs différentes variant de 0 à 5).

4. Le module moteur:

Ensuite le module moteur dépend des informations envoyées par bluetooth, en effet:

Si “t” vaut 6 les moteurs tournent à gauche, si “t” vaut 7 les moteurs tournent à droite, lorsque “t” vaut 8 rien ne se passe (cette valeur est là à titre indicatif lorsque l'on conduit les tests). Il y a aussi deux variables : pin et pina dont les valeurs changeront en fonction du “n” reçu (si n = 0 alors pin = pin2 et pina = pin3).

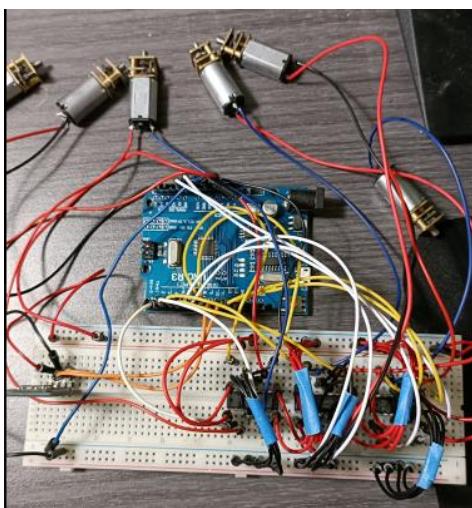


Photo 5: Montage des modules moteurs.

V.L'algorithme:

1)Algorithme de l'accordeur traditionnel:



2)Algorithme accordeur automatique.

int l = 1 or 2 (sense pas Bt)
 int n = 0 - 5 (,,,,,,)
 int pm2 = 2
 ;
 int pm13 = 13

Void Setup
 pmMode (pm2, OUTPUT)
 ;
 ;
 pm mode (pm13, OUTPUT)

Void Loop

```

if (n == 1) { pm2 = pm4;
    pm3 = pm5; }

if (n == 5) { pm2 = pm12;
    pm3 = pm13; }
    
```

```

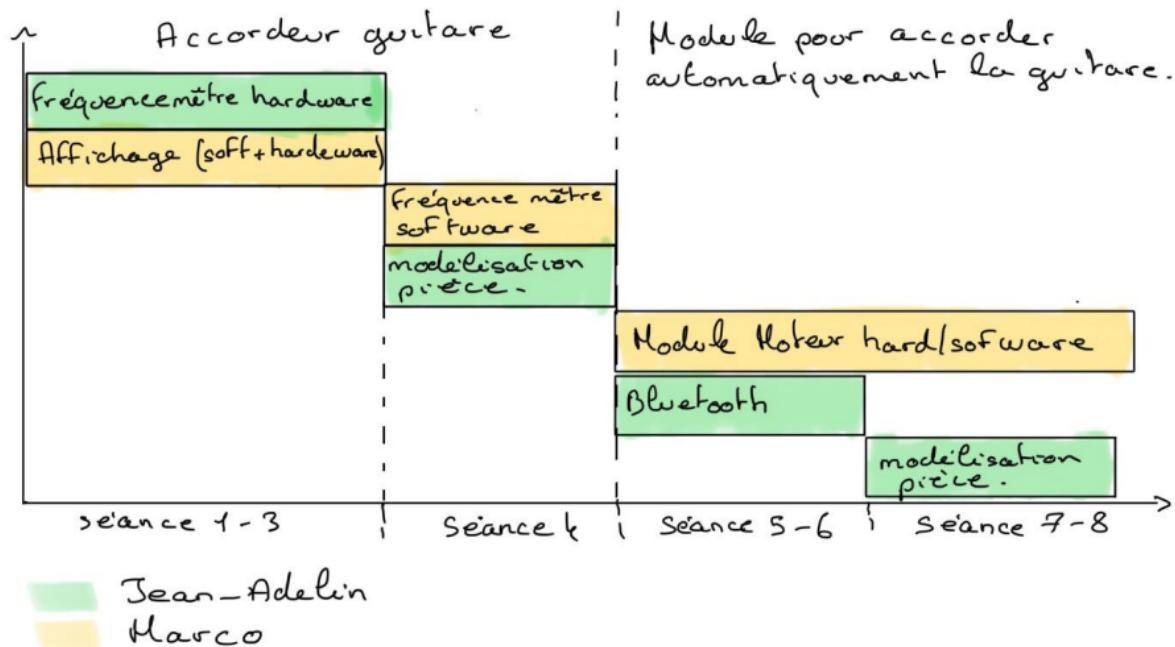
    if l == 1
        digitalWrite(pm2, High)
        digitalWrite(pm3, Low)
    delay(1,7)
    
```

Remarque:

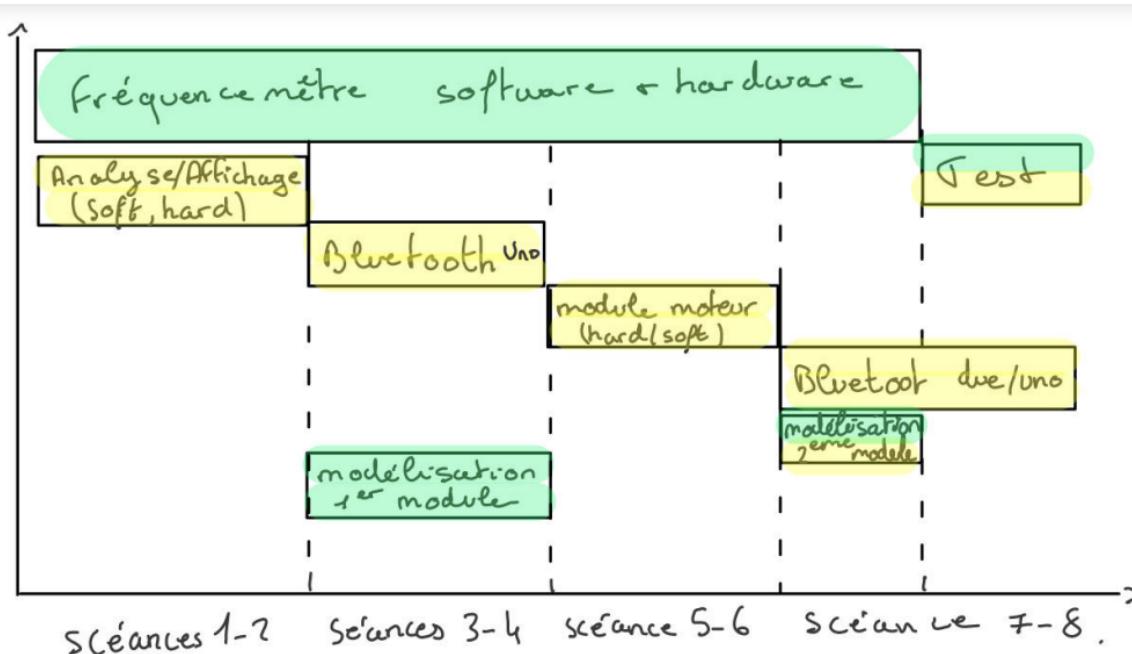
Les algorithmes sont disponibles sur le GitHub dans la section “algorithme”.

VI. Les plannings:

1. Planning initial:



2. Planning final:



Comparaison:

Comme il est possible de le voir ci-dessus, il nous a été compliqué de respecter le planning. En effet Jean-Adelin a passé jusqu'à sept semaines sur le fréquencemètre (Hardware +Software), Marco a lui été plus rapide sur l'affichage et a donc pris la plus grande partie du bluetooth en plus des modules Moteurs. Les modélisations de la fin ont été faites par Marco et le montage de la pièce qui accorde automatiquement à été fait conjointement.

VII. Problèmes:

Durant ces huit séances nous avons dû faire face à un certain nombre de problèmes:

- Tout d'abord le fréquencemètre: en effet nous avons passé près de sept semaines dessus. Nous avons testé trois modules micros arduino différents, travaillé avec deux codes: un premier qui en fait ne marchait qu'avec le GBF, et après de nombreuses recherches, un autre utilisant une FFT. Comme expliqué précédemment ce problème s'est finalement résolu en ajoutant un pont diviseur entre le micro et la carte pour amplifier le signal, en modifiant la fréquence d'échantillonnage du code de la FFT et enfin en utilisant une carte Arduino DUE pour avoir une fréquence d'échantillonnage plus rapide.
- Ensuite, suite au changement de carte (DUE au lieu de UNO), cela a engendré deux nouveaux problèmes: la carte DUE n'est ni compatible avec la bibliothèque du bluetooth ni compatible avec la bibliothèque de l'écran. Pour le bluetooth nous avons réussi à trouver une solution: En effet, après un certain nombre de modifications de notre programme (bug dû au changement de carte) et en utilisant directement TX1 et RX1 pour connecter le BT à la place d'utiliser "software serial", nous avons réussi à assurer la transmission de données. En revanche pour l'écran, faute de temps nous avons dû l'enlever (bien que cela marchait très bien avec la UNO).
- Enfin, le temps (comme beaucoup de groupes j'imagine). En effet, nous aurions vraiment aimé avoir un peu plus de temps pour faire plus de tests, notamment des tests avec la guitare. Pour être tout à fait honnête c'est d'ailleurs un peu frustrant car il se trouve que nous commençons vraiment à avoir un projet abouti et donc les tests commençaient vraiment à devenir concluants.

VIII. Conclusion:

1. Ce qui marche:

- Le fréquencemètre.
- La connexion bluetooth.
- Les modules moteurs.
- Le projet dans sa globalité (peut-être pas chaque détail mais l'idée générale a été respectée).

2. Ce qui ne marche pas:

- L'écran complétant la fonction affichage (marche sur la Uno mais pas la Due).
- Il aurait fallu faire plus de tests pour avoir une meilleure précision.

3. Avec neuf séances de plus:

Avec neuf séances de plus nous aurions essayé de:

- trouver une solution pour inclure de nouveau l'écran avec la carte Due.

- travailler sur l'aspect esthétique du projet:
 - Refaire les montages à l'intérieur de la boîte de l'accordeur.
 - Fabriquer une pièce un peu plus "belle" pour l'accordeur automatique.
- Faire des tests avec la guitare.
- Une fois tout cela refait, nous aurions pu rajouter des fonctionnalités sur notre projet comme une fonctionnalité métronome où encore étendre notre projet à d'autres instruments à cordes comme le piano.

De plus, si nous devions reprendre notre projet depuis le début, nous ne perdrions pas autant de temps à tester les différents micros pour le fréquencemètre. Nous aurions donc directement créé notre propre micro en utilisant une Due ce qui nous aurait évité les nombreux bugs à la fin du projet avec le bluetooth et la fonction affichage.

En conclusion:

Nous gardons une très bonne expérience de ce projet même si tout n'a pas fonctionné parfaitement. Cela nous a permis de concevoir un objet sans forcément avoir de prérequis sur la manière dont il fallait le fabriquer. Cela nous a donc montré une première approche du travail d'ingénieur que nous souhaitons exercer dans trois ans.

Nous souhaitons aussi remercier tous les intervenants de TD ainsi que les personnes travaillant au FabLab: Merci donc à: Christian PETER, Amina BENOUAKTA, Xavier LEBRETON et Frédéric JUAN.

Et surtout merci infiniment à Mr Masson pour son aide, ses conseils et surtout sa patience !!

IX.Bibliographie:

[Pascal MASSON - Enseignement](#)

<https://fr.amen-technologies.com/arduino-based-guitar-tuner>

<https://www.mataucarre.fr/index.php/2017/12/07/mesurer-une-frequence-avec-un-arduino/>

[Grove - Sound Sensor - Seeed Wiki \(seeedstudio.com\)](#)

<https://www.youtube.com/watch?v=wbeV0J30LGQ&t=160s>

<https://www.instinctguitare.com/comment-jouer-des-harmoniques-naturelles-a-la-guitare/#:~:text=Par%20exemple%20sur%20la%20guitare%2C%20la%20fr%C3%A9quence%20de,220Hz%2C%20330Hz%2C%20440Hz%2C%20550Hz%20et%20ainsi%20de%20suite>

[Electronique - Réalisations \(sonelec-musique.com\)](#)