

# Rapport Technique

Thibaut Maroye  
Léa Viard  
Ellyne Liégeois  
Anthony Brasi  
Ferdinand Koci

Mai 2023

## 1 Introduction

### 1.1 Cahier des charges

Performances attendues :
Accès à 10 notes en même temps grâce à 10 lasers.
Accès à 6 banques de 10 sons

## 1.2 Schéma fonctionnel/Liste des fonctions

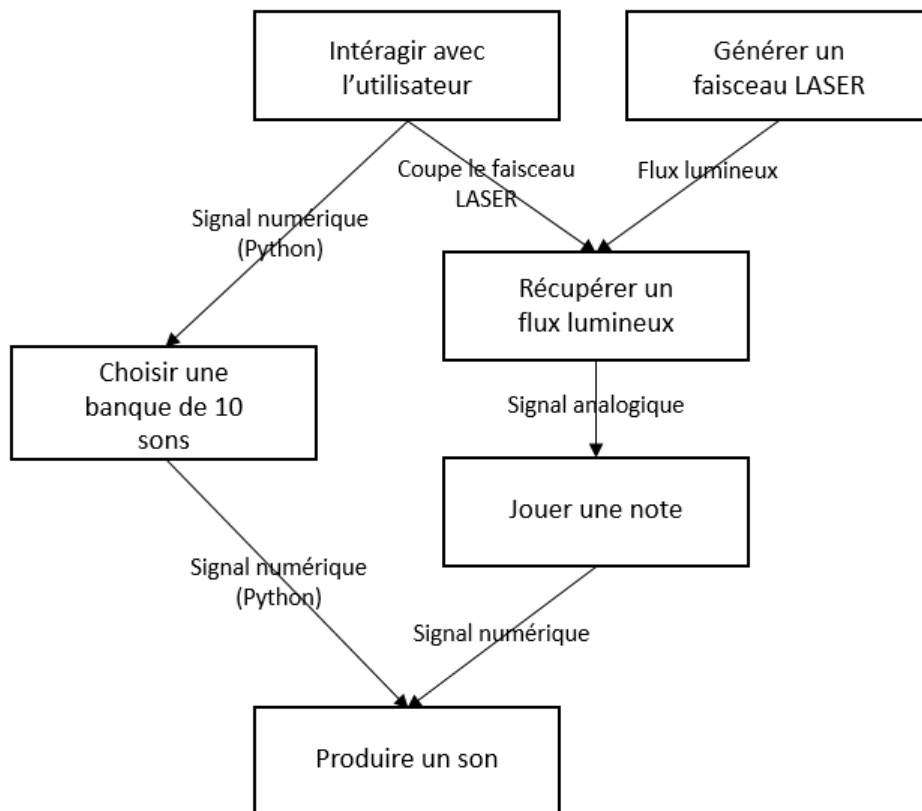


FIGURE 1 – Schéma fonctionnel de la harpe laser.

Fonctions supplémentaires (en bonus) :
L'utilisateur peut importer ses banques de sons personnelles
Réglages du volume sonore et du fade out de la harpe

## 2 Description du projet

### 2.1 Récupérer un flux lumineux

#### 2.1.1 Objectifs

Les deux boîtiers LASER produisent chacun 5 faisceaux parallèles, étendus spatialement dans une direction, ce qui nous donne une légère marge d'ajustement de leur positionnement. L'objectif de cette partie est de réaliser un montage électrique à l'aide de photodiodes et de récupérer le signal de coupure des faisceaux lors de l'utilisation de l'instrument avec une carte nucléo et un programme.

### 2.1.2 Réalisation

Nous avons commencé par programmer sur MBED en C++ un code afin de récupérer l'information "le faisceau lumineux est coupé ou non" .

```
#include "mbed.h"
InterruptIn photodiode1(D6); //Déclaration de la fonction
d'interruption
int main()

void coupe1(void);
int v1;

void coupe1(){
    v1=photodiode1.read(); lecture de la valeur numérique (0 ou 1) dans
le circuit de la photodiode 1, le seuillage est automatique
}

int main(){
    v1=photodiode1.read(); initialisation de la tension dans une des
branches
    photodiode1.rise(&coupe1); la fonction d'interruption se
déclenche à chaque augmentation ou chute de la tension dans la branche
(si le faisceau est coupé ou non)
    photodiode1.fall(&coupe1);
    while(1) {

printf("%d,%d,%d,%d,%d,%d,%d,%d,%d,%d\n",v1,v2,v3,v4,v5,v6,v7,v8,v9,v10
); //affichage en continu des valeurs de la tension pour chaque
branche, information récupérée par le programme python
        wait_us(1000);
    }
}
```

FIGURE 2 – Code mbed : récupération de l'information lumineuse

Les commandes impliquant v1 sont répétées dix fois pour les dix photodiodes. Le code produit en continu une suite de dix chiffres comportant des 0 ou des 1. Si la photodiode est éclairée, alors du courant passe dans la branche et la tension au borne de la résistance correspondante est non nulle car  $V_i = R_i I_{photodiode}$ . En revanche, si le faisceau du laser est coupé, alors le courant ne sera pas suffisant pour dépasser le seuillage induit par la carte nucléo et la tension sera remplacé par 0. Les chiffres changent dès que la tension change dans chaque branche peu importe si elle augmente ou diminue., ce qui permet d'avoir une information claire sur quel faisceau est coupé et pendant combien de temps.

Nous avons ensuite réalisé le montage suivant qui se compose de 2 fois 5 photodiodes en parallèle accompagnées de résistances et reliées à une alimentation, une masse et la carte nucléo. Il s'agit d'un montage de photodétection simple pouvant capter cinq signaux séparément.

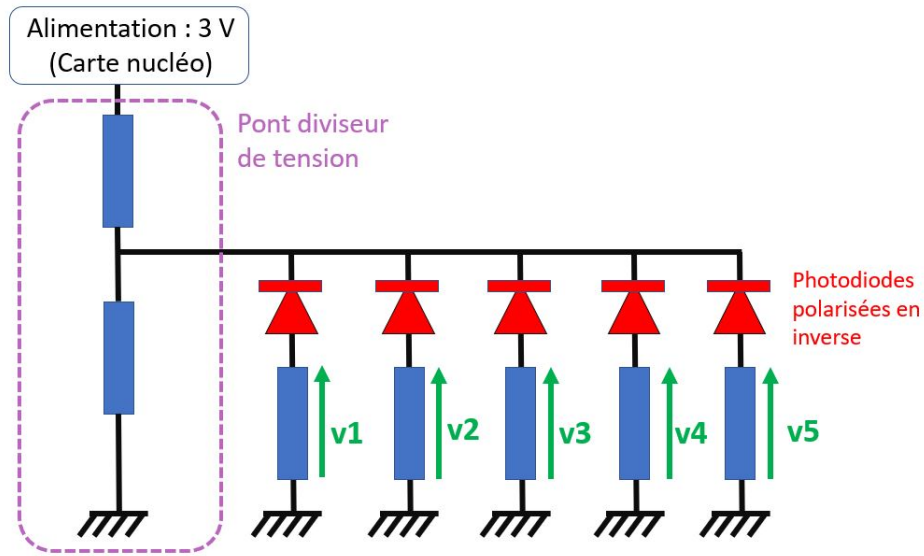


FIGURE 3 – Schéma du circuit à cinq photodiodes

Après des tests sur breadboard, nous avons monté les circuits et soudé les composants sur les plaquettes utilisées sur l'instrument final.

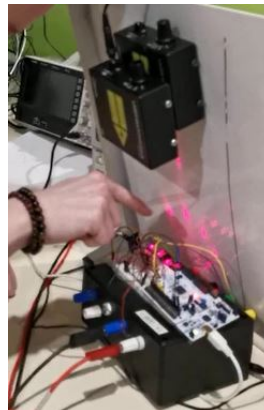


FIGURE 4 – Test sur breadboard avec cinq photodiodes



FIGURE 5 – Circuit soudé sur l'une des deux plaques

Pour avoir un vrai objet à proposer, nous avons en crée en parallèle une structure. Pour que le résultat visuel soit plus intéressant, et pour pouvoir activer deux photodiodes en plaçant un seul doigt à l'intersection de deux faisceaux, nous avons choisi de croiser ces derniers.



FIGURE 6 – Faisceaux des boîtiers LASER croisés



FIGURE 7 – Premier dessin prévoyant l'allure générale de l'instrument

Pour ce faire, nous avons dessiné sur papier puis sur Solidworks les différentes pièces constituant la structure de l'instrument. Nous les avons produites à la découpeuse LASER puis assemblées, collées et peintes.



Nous avons essayé de penser la structure pour qu'elle soit la plus stable possible, tout en conservant un côté esthétique apporté par les pieds inclinés selon le même angle que les faisceaux, et en dissimulant les boîtiers LASER derrière la face avant de la partie haute.

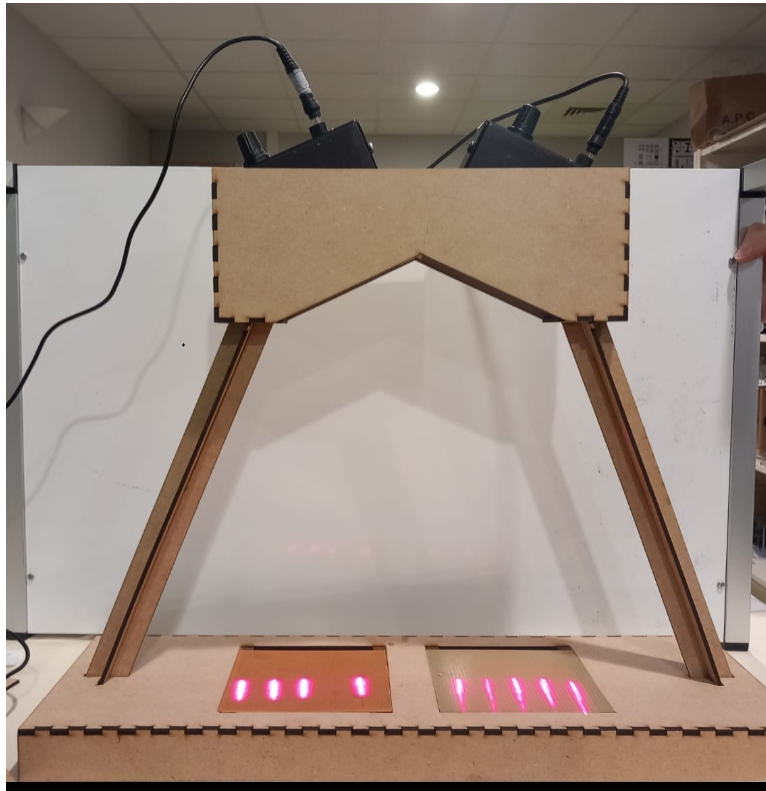


FIGURE 10 – Structure assemblée avant peinture

### 2.1.3 Résultat

La structure finale est la suivante :

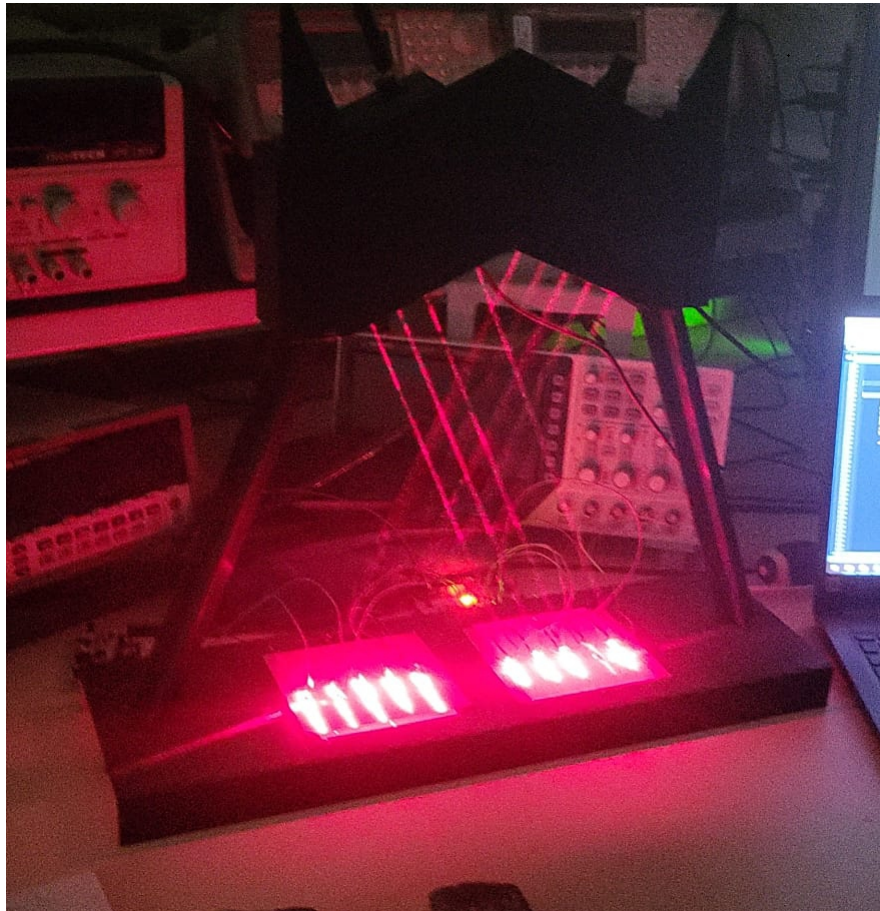


FIGURE 11 – Höllen Harfe assemblée

Elle permet d'intégrer les deux boîtiers LASER de manière suffisamment précise pour que les faisceaux arrivent sur les photos diodes. Ces dernières sont soudées sur des plaquettes qui sont encastrées sur la structure. Nous avons pris garde à laisser une marge de translation des plaquettes pour ajuster leur position en cas de déplacement des boîtiers LASER.

A l'issue de tous les tests, nous avons réussi à faire fonctionner le code pour seulement 4 photodiodes, probablement à cause d'un problème ayant survécu pendant le soudage des photodiodes, mais la cause précise reste inconnue.

## 2.2 Produire un son

### 2.2.1 Objectifs

Le but de cette partie est que l'information reçue par les photodiodes soit acheminée jusqu'à un ordinateur qui produira un son dépendant de quel laser a été coupé par le musicien.

### 2.2.2 Réalisation

L'information reçue par la carte nucléo est envoyée aux ports de communication de l'ordinateur par la fonction `print`. On choisit de récupérer cette information sous Python en utilisant la librairie `Pyserial`, via le code ci après. Il suffit ensuite de répéter cette lecture à chaque nouvelle arrivée d'information venant de la carte à l'aide d'une boucle `while` et de la condition `"if serialInst.in_waiting :"` et de connecter les fronts descendants et montant des bits correspondants à chacune des photodiodes à une fonction qui jouera un son (figure 13).



```

ports = serial.tools.list_ports.comports()
serialInst = serial.Serial()
serialInst.baudrate = 9600
serialInst.port = "COM11"
serialInst.open()
packet = serialInst.readline()
datap=packet.decode('utf').rstrip('\n').split()

```

FIGURE 12 – Code d’acquisition Pyserial

```

if data[0]==0 and olddata[0]==1:
    self.do_pressed()

if data[0]==1 and olddata[0]==0:
    self.do_released()

```

FIGURE 13 – Fronts vers fonctions

Le principe s’étend simplement à toutes les notes : les adresses des sons que l’on souhaite jouer sont stockées dans une liste et on joue chaque son sur des channels différents du mixer pygame. Pour pouvoir réaliser tout cela en parallèle de la mise à jour de l’interface graphique dont on parle ci-après, il est nécessaire d’utiliser la librairie "Threading".

## 2.3 Intéragir avec l’utilisateur

### 2.3.1 Objectifs

L’utilisateur doit pouvoir choisir la banque de son associée aux faisceaux lasers sur une interface numérique, et effectuer des réglages de volume et fade out.

### 2.3.2 Réalisation, Fonctionnement et Résultat

Nous avons réaliser cette interface sur python avec la librairie PyQt5 (Figure 14)

Les touches de piano permettent de tester les sons que l’on choisit dans "Sound Bank", parmi les 6 banques de sons disponibles sur le logiciel. Il y a aussi une option "Custom" permettant à l’utilisateur d’importer ses propres banques, qui doivent être des dossiers contenant des fichiers .mp3 correctement nommés (do, re, mi... do2, re2, mi2).

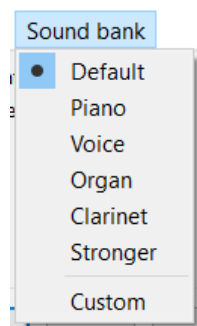


FIGURE 15 – Option Sound bank.

Le bouton connect sert à se connecter à la carte nucléo. Un texte s’affiche en haut à gauche de la fenêtre lorsque le programme est parvenu à se connecter à la carte. Enfin, nous avons ajouté les potentiomètres "volume" et "fade out" pour régler ces paramètres.

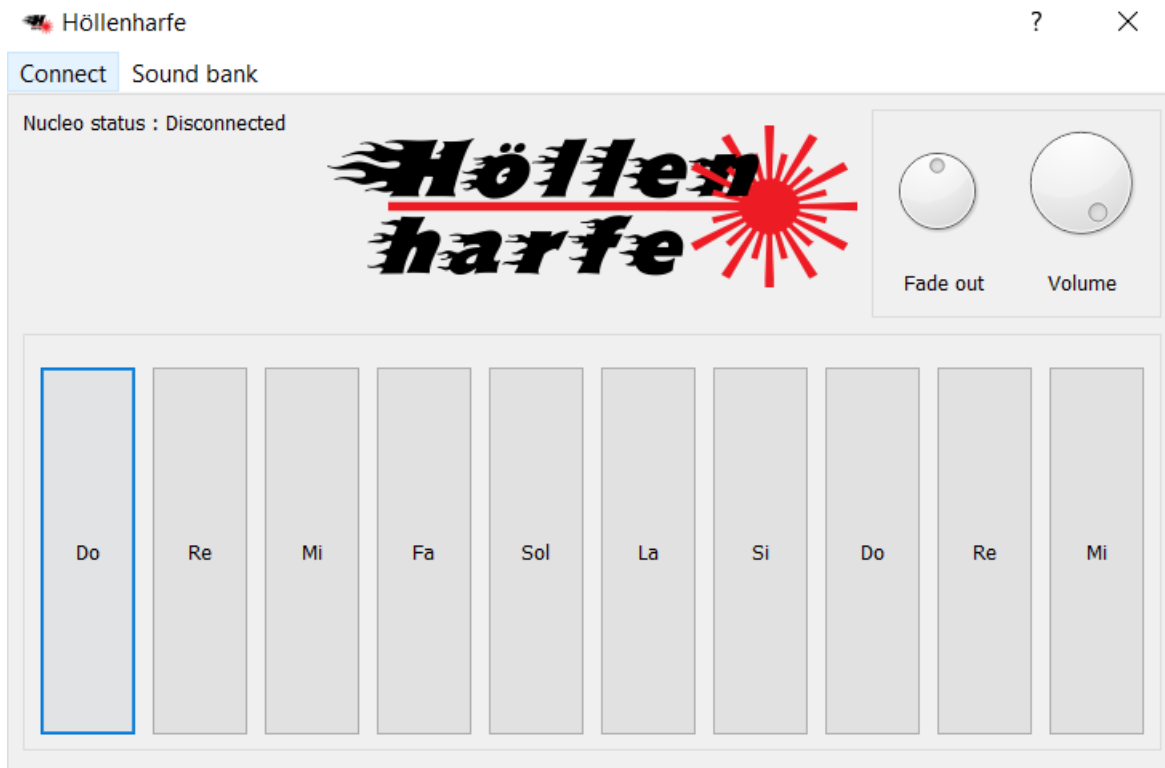


FIGURE 14 – Interface utilisateur

### 3 Bilan

#### 3.1 Avancement final

Au final, toutes les performances du cahier des charges sont validées, à l'exception de "Accès à 10 notes en même temps grâce à 10 lasers", car certaines 6 des photodiodes n'envoient pas d'informations comme on a pu le voir dans la partie récupérer un flux lumineux.

#### 3.2 Retour d'expérience de l'équipe

La répartition du travail a été faite ainsi. Les fonctions "produire un son" et "interfacer avec l'utilisateur" ont été assurées par Thibaut et Ferdinand, tandis que Léa, Ellyne et Anthony ont travaillé sur la fonction "récupérer un flux lumineux".

L'équipe a pu développer des compétences d'organisation du travail ainsi que des compétences techniques propres aux fonctions développées par chaque membre.