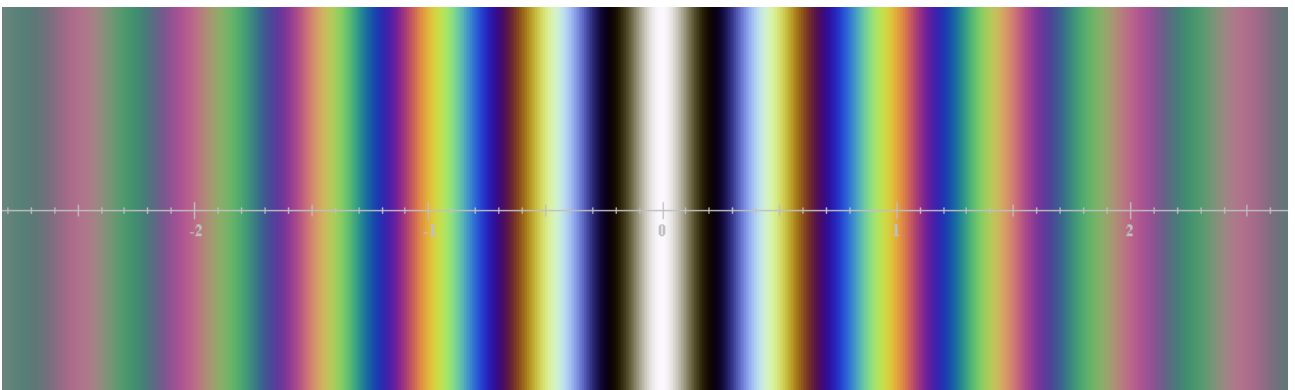


# Rapport Scientifique

## Projet Art & Sciences

### Interférences



# SOMMAIRE

I. Outils utilisés et Limitations

II. Compte-rendu d'expérience

III. Bibliographie

Annexes

## I. Outils utilisés et Limitations

Le Projet de Hugo Pétigny porte sur une réappropriation de l'expérience des fentes de Young. Une expérience qui a été réalisée par Thomas Young en 1801. Cette expérience permet de mettre en évidence la nature ondulatoire de la lumière. Elle illustre la dualité onde-particule : les interférences montrent que la matière présente un comportement ondulatoire, mais la façon dont elles sont détectées (impact sur un écran) montre leur comportement particulaire.

Tout d'abord, lors de nos premières discussions, l'idée première de Hugo Pétigny était la suivante. Projeter des raies de lumières créées par des fentes dites de Young. Un capteur de lumière récupérerait l'intensité lumineuse et la convertirait en un message sonore envoyé à des enceintes. Le son produit par ces enceintes serait récupéré par un capteur sonore qui ferait varier le moteur en charge du défilement des raies de lumières. Un problème se pose dans un tel système fermé. La vitesse de défilement des raies influence le capteur lumineux puis le capteur sonore. Enfin le capteur sonore modifie la vitesse de défilement des raies. La valeur de départ s'influence elle-même, c'est le problème de l'œuf et de la poule. D'où viens réellement l'information ? C'est pour cela que nous avons décidé au début du projet de ne plus avoir un système fermé en supprimant le capteur sonore.

Nous souhaitons que le projet puisse se rapprocher le plus possible de l'expérience de Young. La réalisation des fentes pour l'expérience demandait des outils de découpe qui ne sont pas en la possession de l'ESA ni de Polytech'Lille. C'est pour cela que le projet s'inspire de cette expérience sans pouvoir la reproduire de façon fidèle. Malgré cette limitation, après quelques tests, la vitesse de défilement des fentes peut également créer un début d'interférence par la présence de plus grandes zones irisées aux extrémités des raies lumineuses.

Nous avons choisi d'utiliser deux cartes Arduino pour réaliser le projet. Une carte Arduino sera en charge du contrôle moteur et l'autre de la partie lumineuse/sonore. Hugo et moi-même avons eu des cours plus ou moins poussés sur le codage avec Arduino ce qui permet durant ce projet une compréhension commune du codage de la partie technique. ( code Moteur et code Capteur Lumineux et Génération de son en [ANNEXE](#) )

Le codage du capteur lumineux fut notre première étape ( Voir II.Compte-rendu d'expérience ). La valeur donnée par le capteur varie entre 0 et 1023 ( 10bits ). Pour transformer cette information en son, il nous fallait un spectre sonore. Hugo Pétigny veut utiliser des fréquences entre 0 et 80Hz. Lorsque deux fréquences de ce spectre sont jouées

ensemble on obtient des ondes binaurales. Il a donc fallu retransformer cette valeur du capteur lumineux en une fréquence entre 0 et 80 Hz. Malheureusement, avec un module Arduino il n'est pas dans nos compétences ( voir impossible ) de générer un son en dessous de 30 Hz. C'est pour cela que dans le code ( voir [ANNEXE](#) ) la génération de signal ( Sound1 ) se fait entre 30 et 80 Hz.

Une fois la génération du son codée, nous devons transmettre cette information. Les premiers tests furent réalisés avec un Haut-Parleur avec lequel le montage était facile. Nous avons réfléchi à comment relier le module Arduino à une connectique audio grandement utilisée : les ports et câbles Jack. Après un essai avec une carte Adafruit-music-maker-shield ( voir Bibliographie ), nous avons commandé des ports jack femelles ainsi que des cartes amplificatrice pour relier les enceintes à la carte Arduino. Il aura fallu un travail de soudage pour câbler correctement les ports aux amplificateurs. Comme il l'indique par son nom, l'amplificateur permet la lecture d'un signal plus faible envoyé par la carte Arduino au enceinte pourvues d'une connectique Jack.

Un problème toujours non résolu est la possibilité de jouer un son différent sur chaque enceinte. Après de nombreux tests avec et sans amplificateurs, nous n'arrivions pas à obtenir un son différent sur chaque enceinte. Pour le moment, nous avons réussi à diffuser un son qui varie en fonction de la luminosité reçu par le capteur lumineux.

Enfin, le choix du moteur fut axé sur sa vitesse. Hugo Pétigny pris pour moteur un moteur pas à pas. Composé de 2048 pas, on modifie la vitesse du moteur en modifiant le délai entre chacun de ses pas. Le délai minimum étant de 2 milliseconde, la vitesse maximale du moteur sera de 4 secondes par tour. J'ai proposé un code commenté différent du code proposé par le professeur de Hugo mais il fonctionne tout aussi bien ( voir code Moteur en [ANNEXE](#) ).

## II. Compte-rendu d'expérience

### [Expérience capteur lumineux / Haut Parleur](#)

*Branchement :*

HP1 branché sur le port PWM 13 de l'Arduino et la masse PWM 14  
HP2 branché sur le port PWM 12 de l'Arduino et la masse PWM 14  
Capteur Lumineux branché sur le port Analogique 0 ( A0 ) de l'Arduino

Entre la branche de sortie du capteur et la masse on place une résistance élevée ( 1M ohms ) pour que la tension aille dans le ports A0 et non vers la masse ( GND ).

Voir Code Capteur Lumineux et Génération de son en [ANNEXE](#)

Le code est fait de tel sorte que les fréquences du son Sound1 vers le HP1 ou du son Sound2 vers le HP2 soit toujours comprises entre 30 et 80 Hz.

L'expérience aura montré la variation de la fréquences sonore lors des variations d'intensités lumineuses mais l'écart de fréquences entre les deux Haut Parleurs ne fut pas réalisable car les deux Haut Parleurs produisaient le même son.

### [Expérience Moteur](#)

*Branchement :*

Le moteur est branché sur une carte dédiée. Sur cette carte il y a 4 pin ( numéroté 1 2 3 4 ), ils sont branché respectivement au ports de la carte Arduino ( PWM 8 9 10 11 ). La carte dédiée est également dotée de pin d'alimentation + et - branché respectivement sur les port 5V et GND de la carte Arduino.

Voir Code Capteur Moteur en [ANNEXE](#)

La variable ' dir ' dans le code représente le sens de rotation. Comme c'est un booléen elle ne peut avoir que deux valeurs ' false ' ( sens horaire ) et ' true ' ( sens anti-horaire ).

Malheureusement, même en mettant le délai entre chaque pas du moteur à sa valeur la plus basse ( 2 ms ), la vitesse maximale du moteur reste lente à 4s par tour.

### III. Bibliographie

*Comment fonctionne l'Amplificateur PAM8403 :*

<http://store.logicneed.com/product/16084?fbclid=IwAR1iLPIIKRoayD-asPnXmOKw5am5oEW5OinuulYDpr5RIIS1E0Mihx6U1YE>

*Constitution d'un câble jack :*

<https://www.amazon.fr/P%C3%B4les-p%C3%B4les-souder-r%C3%A9paration-printemps/dp/B01GFOU09U?fbclid=IwAR0ykmkDt9ukLkpIDZ3V23ZY7U05wp3qTa6d3TsmfWSbKiKqhAytaVjEchc>

*Apprentissage du code Arduino ( documentation de l'ESA ) :*

[https://transat.stephaneabee.net/arduino/?fbclid=IwAR0hfGfXXhN7ZdUUvWVskAzs5z0cs0KMuYDonRD\\_1y4kEuL6n\\_5N2kc1t1Q](https://transat.stephaneabee.net/arduino/?fbclid=IwAR0hfGfXXhN7ZdUUvWVskAzs5z0cs0KMuYDonRD_1y4kEuL6n_5N2kc1t1Q)

*Génération de Fréquences Pures :*

[https://www.szynalski.com/tone-generator/?fbclid=IwAR2Jl2jxuPowTqzUZH7Voi6ed\\_KzPUQLcyJcywHS53-DTRBnFeSOVo1dx1I](https://www.szynalski.com/tone-generator/?fbclid=IwAR2Jl2jxuPowTqzUZH7Voi6ed_KzPUQLcyJcywHS53-DTRBnFeSOVo1dx1I)

*Comprendre la fonction Random() :*

<https://www.carnetdumaker.net/articles/generer-des-nombres-pseudo-aleatoires-avec-une-carte-arduino-genuino/?fbclid=IwAR1BMpkilt2c22Ar2hcs7pMWIeccWUoUyKPKl6zfKK9LkdMVHm8ilGcOE-4>

*Comprendre la fonction tone() :*

<https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/?fbclid=IwAR0Y5kOznVwaDfe-ZFZ0pJP7jW--PqMvQkdOvZjQ75YfOWhQX1PciN8jXfw>

*Câblage pour utilisation d'un Haut-Parleur :*

<https://www.carnetdumaker.net/articles/jouer-des-notes-de-musiques-avec-une-carte-arduino-genuino/?fbclid=IwAR0X9FnxW5VAnZOSiOeTKS8NcUmaa5kSQf7xoQLREH3oi5ucRXaLyGYgapY>

*Utilisation d'un module Bluetooth pour réduire le câblage si besoin :*

[https://eskimon.fr/tuto-arduino-907-utiliser-un-module-bluetooth-hc-05-avec-arduino?fbclid=IwAR283a77K179cu-A8sFO32ot\\_YsW-sjZWw\\_Ge9ENd4fIUu7G6yHPfqBZpAY](https://eskimon.fr/tuto-arduino-907-utiliser-un-module-bluetooth-hc-05-avec-arduino?fbclid=IwAR283a77K179cu-A8sFO32ot_YsW-sjZWw_Ge9ENd4fIUu7G6yHPfqBZpAY)

*Câblage pour utilisation d'un capteur lumineux :*

[https://www.carnetdumaker.net/articles/mesurer-la-luminosite-ambiante-avec-une-photoresistance-et-une-carte-arduino-genuino/?fbclid=IwAR3couQ7kAm6\\_X7sRj8KPhSF-4Nf9xRBOroaX9oIP24z\\_fcEjpbBvdDQLs](https://www.carnetdumaker.net/articles/mesurer-la-luminosite-ambiante-avec-une-photoresistance-et-une-carte-arduino-genuino/?fbclid=IwAR3couQ7kAm6_X7sRj8KPhSF-4Nf9xRBOroaX9oIP24z_fcEjpbBvdDQLs)

*Recherches pour l'utilisation d'une carte Adafruit Music Maker Shield :*

[https://learn.adafruit.com/adafruit-music-maker-shield-vs1053-mp3-wav-wave-ogg-vorbis-player/play-music?fbclid=IwAR1Fs8B1BS26DL07hIAdZGSEHrv-OP-fGPYFIPEi0USLLbKCJOt-IV7\\_6eM](https://learn.adafruit.com/adafruit-music-maker-shield-vs1053-mp3-wav-wave-ogg-vorbis-player/play-music?fbclid=IwAR1Fs8B1BS26DL07hIAdZGSEHrv-OP-fGPYFIPEi0USLLbKCJOt-IV7_6eM)

*Cablage et Codage d'un Moteur Pas à Pas 2048 Steps :*

<https://retroetgeek.com/arduino/arduino-comment-utiliser-un-moteur-pas-a-pas-uln2003a-et-28byj-48/?fbclid=IwAR1BMpkilt2c22Ar2hcs7pMWleccWUoUyKPKl6zfKK9LkdMVHm8ilGcOE-4>

[https://create.arduino.cc/projecthub/debanshudas23/getting-started-with-stepper-motor-28byj-48-3de8c9?fbclid=IwAR2ZOiFKNUFy6le\\_uKwe2ACF6sck8Czpzus4\\_9bLC27jcNjjjspGGVUUYOI](https://create.arduino.cc/projecthub/debanshudas23/getting-started-with-stepper-motor-28byj-48-3de8c9?fbclid=IwAR2ZOiFKNUFy6le_uKwe2ACF6sck8Czpzus4_9bLC27jcNjjjspGGVUUYOI)

# ANNEXES

## Code Moteur :

```
int STEPPER_PIN_1 = 8; // Pin 1 brancher sur PWM 8
int STEPPER_PIN_2 = 9; // Pin 2 brancher sur PWM 9
int STEPPER_PIN_3 = 10; // Pin 3 brancher sur PWM 10
int STEPPER_PIN_4 = 11; // Pin 4 brancher sur PWM 11

int step_number = 0;
void setup() { // Cette fonction dit au programme sur quels Pin le moteur est
  Serial.begin(9600);
  pinMode(STEPPER_PIN_1, OUTPUT);
  pinMode(STEPPER_PIN_2, OUTPUT);
  pinMode(STEPPER_PIN_3, OUTPUT);
  pinMode(STEPPER_PIN_4, OUTPUT);
  pinMode(Pot_PIN, INPUT);
}

void loop() {

  OneStep(false); // Le FALSE écrit ici représente le sens de rotation ('false' =
  sens horaire ; 'true' = sens anti-horaire)

  delay(2); // Délai entre chaque step ( minimum 2ms -> 2048 step x 0.002
  secondes = 4s / tour )

}

void OneStep(bool dir){
  if(dir){ // Fonction rotation en sens horaire
  switch(step_number){
  case 0:
  digitalWrite(STEPPER_PIN_1, HIGH);
  digitalWrite(STEPPER_PIN_2, LOW);
  digitalWrite(STEPPER_PIN_3, LOW);
  digitalWrite(STEPPER_PIN_4, LOW);
  break;
  case 1:
  digitalWrite(STEPPER_PIN_1, LOW);
  digitalWrite(STEPPER_PIN_2, HIGH);
  digitalWrite(STEPPER_PIN_3, LOW);
  digitalWrite(STEPPER_PIN_4, LOW);
  break;
  case 2:
```



```

digitalWrite(STEPPER_PIN_1, LOW);
digitalWrite(STEPPER_PIN_2, LOW);
digitalWrite(STEPPER_PIN_3, HIGH);
digitalWrite(STEPPER_PIN_4, LOW);
break;
case 3:
digitalWrite(STEPPER_PIN_1, LOW);
digitalWrite(STEPPER_PIN_2, LOW);
digitalWrite(STEPPER_PIN_3, LOW);
digitalWrite(STEPPER_PIN_4, HIGH);
break;
}
}else{ // Fonction rotation en sens anti-horaire
  switch(step_number){
case 0:
digitalWrite(STEPPER_PIN_1, LOW);
digitalWrite(STEPPER_PIN_2, LOW);
digitalWrite(STEPPER_PIN_3, LOW);
digitalWrite(STEPPER_PIN_4, HIGH);
break;
case 1:
digitalWrite(STEPPER_PIN_1, LOW);
digitalWrite(STEPPER_PIN_2, LOW);
digitalWrite(STEPPER_PIN_3, HIGH);
digitalWrite(STEPPER_PIN_4, LOW);
break;
case 2:
digitalWrite(STEPPER_PIN_1, LOW);
digitalWrite(STEPPER_PIN_2, HIGH);
digitalWrite(STEPPER_PIN_3, LOW);
digitalWrite(STEPPER_PIN_4, LOW);
break;
case 3:
digitalWrite(STEPPER_PIN_1, HIGH);
digitalWrite(STEPPER_PIN_2, LOW);
digitalWrite(STEPPER_PIN_3, LOW);
digitalWrite(STEPPER_PIN_4, LOW);

}
}
step_number++;
if(step_number > 3){
  step_number = 0;
}
}

```

## Code Capteur Lumineux et Génération de son :

```
int EnceintePin1 = 13, EnceintePin2= 12; // Enceinte1 en PWM 13 et Enceinte2 en PWM12

void setup()
{
  Serial.begin(9600);
  pinMode(EnceintePin2,OUTPUT); // Port Audio 1
  pinMode(EnceintePin1,OUTPUT); // Port Audio 2
}

void loop() {

  float valeur = analogRead(A0); // Ici on lit la valeur donnée par le capteur lumineux
                                 //( entre 0 et 1023 )

  Serial.println(valeur);
  float ecart;
  float Sound1, Sound2;
  ecart = random(30); // Donne un nombre aléatoire entre 0 et 30
  Sound1 = 30+50*(valeur/1023); // Calcule pour transformer les données du capteur
                                // Lumineux en Son ( Enceinte 1 )

                                // Fréquences comprises entre 30Hz et 80Hz
                                // ( Limitation de la fonction tone() )

  Sound2 = Sound1 + ecart ; // Pour créer le Son de l'Enceinte 2 on Ajoute au Sound1 une
                             valeur aléatoire entre 0 et 30

  if( Sound2 > 80 ) // Ici les 3 prochaines lignes assurent que la fréquence ne
                   dépassera pas les 80 Hz
  {
    Sound2 = 80;
  }
  Serial.println(Sound1); // Affichage de la fréquence du Sound1
  Serial.println(Sound2); // Affichage de la fréquence du Sound2

  tone(EnceintePin1,Sound1); // Fonction pour envoyé le Sound1 sur l'Enceinte1
  //delay(100);
  //tone(EnceintePin2,Sound2); // Fonction pour envoyé le Sound2 sur l'Enceinte2

  /*tone(EnceintePin1,40);
  tone(EnceintePin1,60);*/
  delay(100);
}
```