

# Le cri des Arbres.

Rapport de synthèse d'une installation à venir.

Enoch HODONOU  
(POLYTECH)

Pierre  
DEMEULENAERE  
(ESA)

*« La conception d'une synchronisation son et lumière pour un Panorama. »*



# Sommaire.

Contexte de réalisation pour Mai

- Cahier des charges.
- Objectifs de la semaine d'atelier
- Etapes et résultats
- Problèmes
- Tâches à venir

Les appris

Annexe : documentation technique

# Contexte de réalisation.

Ce projet m'a conduit à interroger le bruit du monde urbain, en périphérie de mégalopole, dans des forêts, occupée et autonome. Le bruit vie et contamine. Voilà ! Le parasite .. L'informateur du flux, une forêt qui brûle, le bruissement de la vie.

Expositions Hypothétiques :

- 100% APV (Tourcoing)
- PRIST (Tourcoing)



*Claude Monet, Reflets verts, Entre 1914 et 1926 Deux «panneaux» à l'huile accolés sur toile marouflée sur le mur H. 200 ; L. 850 cm, musée d'Orsay, distrib. RMN Patrice Schmidt.*



*Sunn O))), Frost (C), 2019, Pyroclasts, Studio Electrical Audio.*

# Le cahier des charges

## Préambule

Dans le cadre du projet PRIST issu de l'entente entre l'Ecole d'Ingénieur POLYTECH-LILLE et l'Ecole d'Arts ESA, nous avons travaillé sur le projet *Le Cri Des Arbres*. Cette collaboration consiste en l'apport technique nécessaire afin de régir une interaction entre la lumière et le son dans un projet artistique.

## Attentes Techniques

Ce projet consiste à plonger l'utilisateur dans l'espace de la toile ou la lumière vibrent au rythme de la musique. Ce monde sera formé d'un petit demi-sas en U de 4m20 de diamètre avec une toile artistique tout au long du mur éclairé par des spots qui seront au-dessus de la tête de l'utilisateur placée à 2m40 avec en dessous de ceux-ci une enceinte infrabasse directionnelle pour avoir une douche de son infrabasse qui immergera l'utilisateur dans l'œuvre.

Pour les attentes techniques, nous les scindons en deux parties : une partie matérielle et une partie logicielle.

## Partie matérielle

Compte tenu du temps imparti pour le projet, nous faisons déjà quelques choix simples qui nous permettront de le développer le plus rapidement possible. En effet, nous choisissons d'utiliser une carte Arduino Uno pour le contrôle des différentes phases de la lumière. Le but de la gestion de la lumière est :

- Eclairer la toile qui sera à environ ~2m20
- La rendre vivante en la faisant vibrer au rythme des transitions musicales.

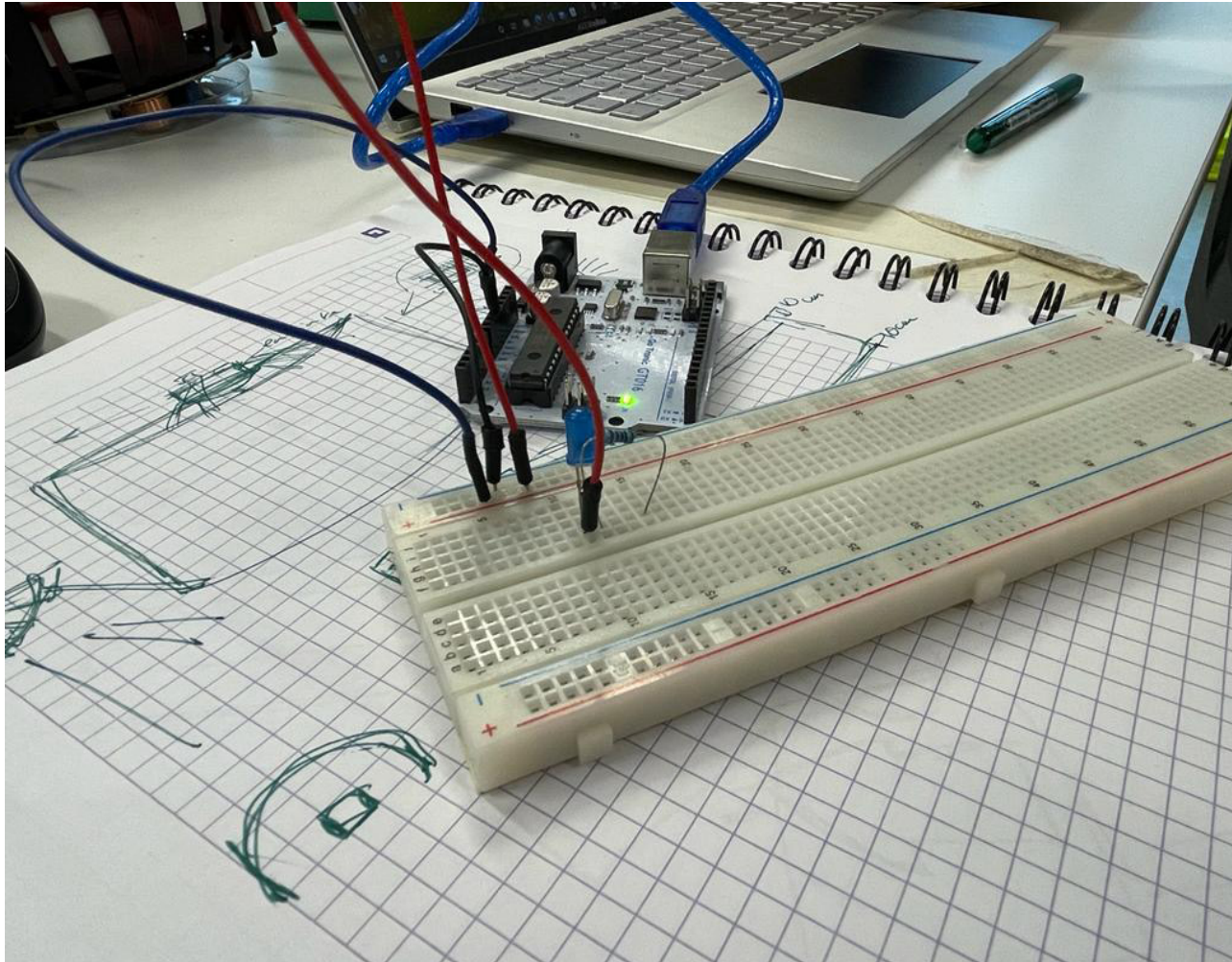
De plus, nous devons gérer l'interaction entre la lumière et le son. Pour ce faire nous décidons (encore une fois de plus sous le contrainte temps) d'utiliser un ordinateur qui nous permettra d'avoir un temps de développement réduit. L'interaction doit avoir les caractéristiques suivantes :

- Le son composé doit impacter l'intensité de la lumière qui éclaire la toile ; ce qui aura pour but de rendre le décor vivant.

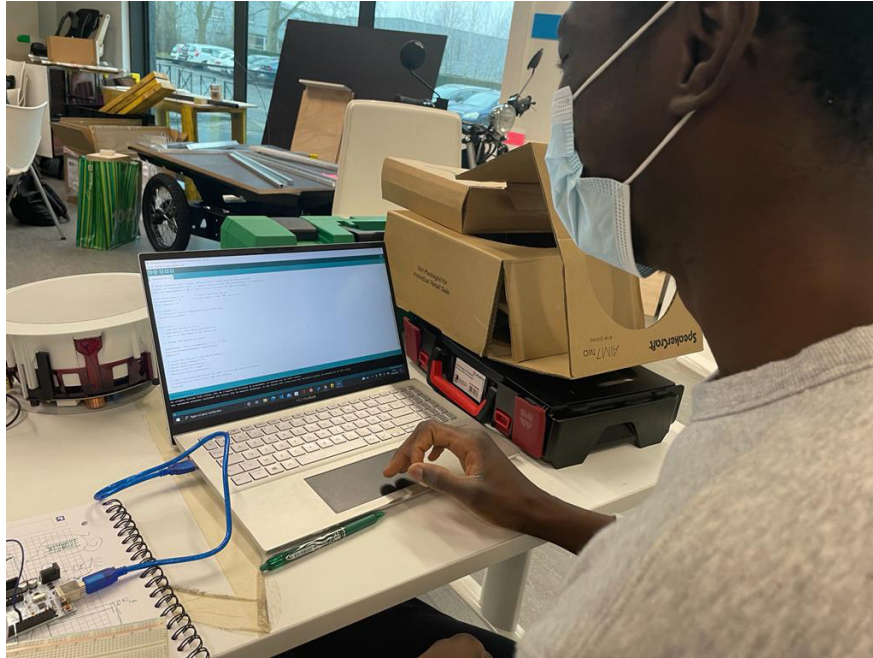
## Partie logicielle

Compte tenu de nos choix techniques, nous développons notre application avec Arduino IDE dans un premier temps pour la manipulation de la carte Arduino, puis nous utilisons ensuite Processing qui est un logiciel qui permet des interactions entre l'ordinateur et un arduino. L'amplificateur audio étant aussi relié à l'ordinateur, c'est par ce biais que nous comptons réaliser l'interaction voulue entre le son et la lumière au niveau logiciel.

# Etapes et résultats.



*Cablage et maquetage sur plaque de test avec une LED*



*Utilisation du programme Arduino avec fonction sinusoïdale pour concevoir une courbe d'intensité montante puis descendente.*



```
Interaction_Music_Lumi_re | Processing 4.0b3
Fichier Modifier Sketch Dépanner Outils Aide

Interaction_Music_Lumi_re
1 /**
2  * This sketch shows how to use the Waveform class to analyze a stream
3  * of sound. Change the number of samples to extract and draw a longer/shorter
4  * part of the waveform.
5  */
6
7 import processing.sound.*;
8
9 SoundFile sample;
10 Waveform waveform;
11
12 int samples = 100;
13
14 public void setup() {
15   size(640, 360);
16   background(255);
17
18   sample = new SoundFile(this, "Music.mp3");
19   sample.loop();
20
21   // Create the Waveform analyzer and connect the play
22   waveform = new Waveform(this, samples);
23   waveform.input(sample);
24 }
25
26 public void draw() {
27   // Set background color, noFill and stroke style
28   background(0);
29   stroke(255);
30   strokeWeight(2);
31   noFill();
32 }

Interaction_Music_Lumi_re
janv. 20, 2022 1:35:34 PM com.jsyn.devices.javasound.JavaSoundAudioDevice <init>
INFOS: JSyn: default output latency set to 80 msec for Windows 10
```

Modéliser une forme vague avec les données d'un son composé.



```
testVolume | Processing 4.0b3
Fichier Modifier Sketch Dépanner Outils Aide

testVolume
1 import processing.sound.*;
2 Amplitude amp;
3 SoundFile sample;
4
5 void setup() {
6   size(640, 360);
7   background(255);
8
9   sample = new SoundFile(this, "Music.mp3");
10  sample.loop();
11
12  // Create an Input stream which is routed into the Amplitude analyzer
13  amp = new Amplitude(this);
14  amp.input(sample);
15 }
16
17 void draw() {
18   background(255);
19   println(amp.analyze());
20
21   float ellipseSize = map(amp.analyze(), 0, 1, 10, 200);
22
23   circle(width/2, height/2, ellipseSize);
24 }
25
26
27
28
29
30
31
32
```

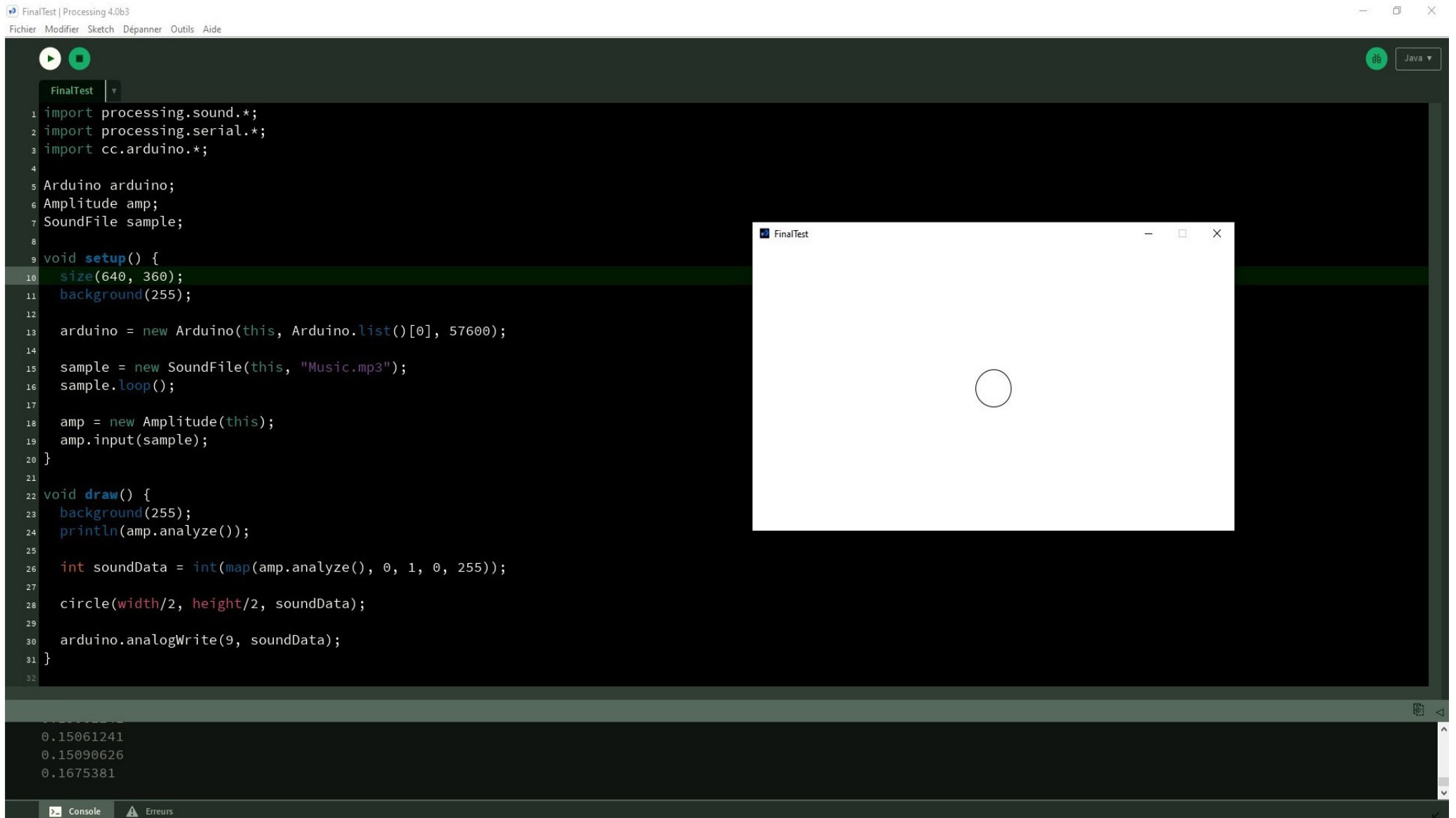
Console Erreurs

*Modélisation d'un cercle avec un son en utilisant la densité du son sur processing.*

FinalTest | Processing 4.0b3

Fichier Modifier Sketch Dépanner Outils Aide

```
FinalTest
1 import processing.sound.*;
2 import processing.serial.*;
3 import cc.arduino.*;
4
5 Arduino arduino;
6 Amplitude amp;
7 SoundFile sample;
8
9 void setup() {
10 size(640, 360);
11 background(255);
12
13 arduino = new Arduino(this, Arduino.list()[0], 57600);
14
15 sample = new SoundFile(this, "Music.mp3");
16 sample.loop();
17
18 amp = new Amplitude(this);
19 amp.input(sample);
20 }
21
22 void draw() {
23 background(255);
24 println(amp.analyze());
25
26 int soundData = int(map(amp.analyze(), 0, 1, 0, 255));
27
28 circle(width/2, height/2, soundData);
29
30 arduino.analogWrite(9, soundData);
31 }
32
```



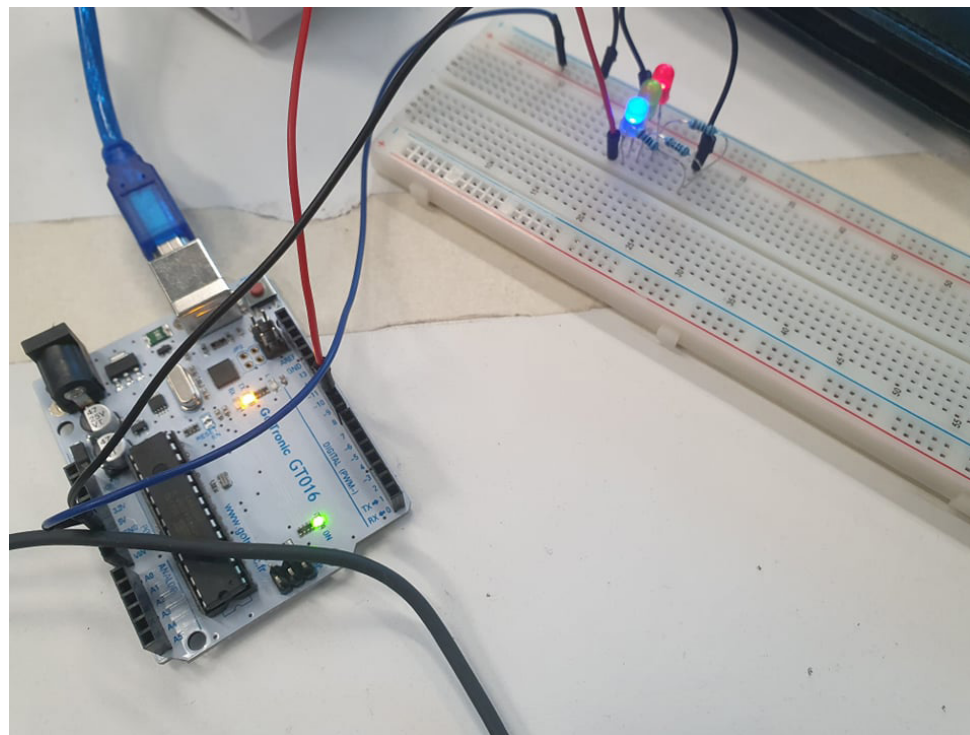
FinalTest

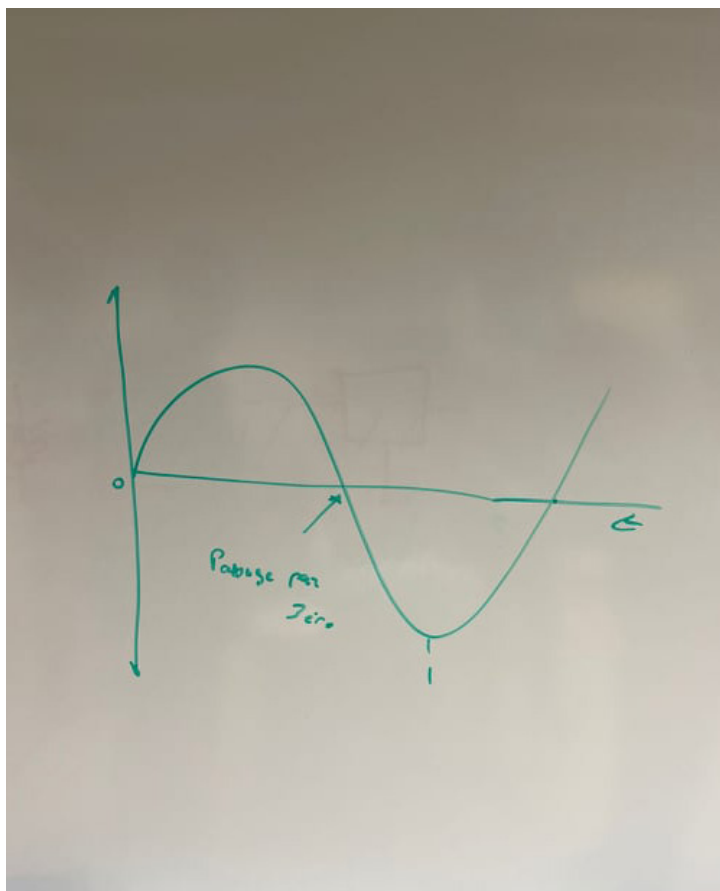
0.15061241  
0.15090626  
0.1675381

Console Erreurs

*Transmettre les données de densité du son composé à la Arduino.*

*Cablage et maquetage sur plaque de test avec trois LED*





*Rencontre avec Thery Flamand pour comprendre comment fonctionne le «Module variateur AC, 1 canal, logique 3.3V/5V 8A AC 50/60hz 220V/110V»*

# Problèmes .

Etant donné que l'établissement n'autorise pas les étudiants à manipuler des circuits électriques compatibles avec un courant 230V en raison d'assurance et de sécurité.

Que le module variateur n'est pas disponible sur Lille et qu'il ne peut être livrée expressément nous nous retrouvons bloqués et ne pouvant expérimenter entièrement le processus de synchronisation son composé et lumières 230v.

# Liste des tâches pour Mai.

- Recevoir les composants manquants :

arduino,  
module variateur,  
stairvilles,  
ampoules,  
module bluetooth,  
prise à 3 fils femelles,  
boite étanche,  
multi prise.  
Peindre

- Concevoir l'application téléphonique  
- Connecté TX/RX au module bluetooth  
Lié le module variateur aux spots

# Annexe :

## Documentation Technique

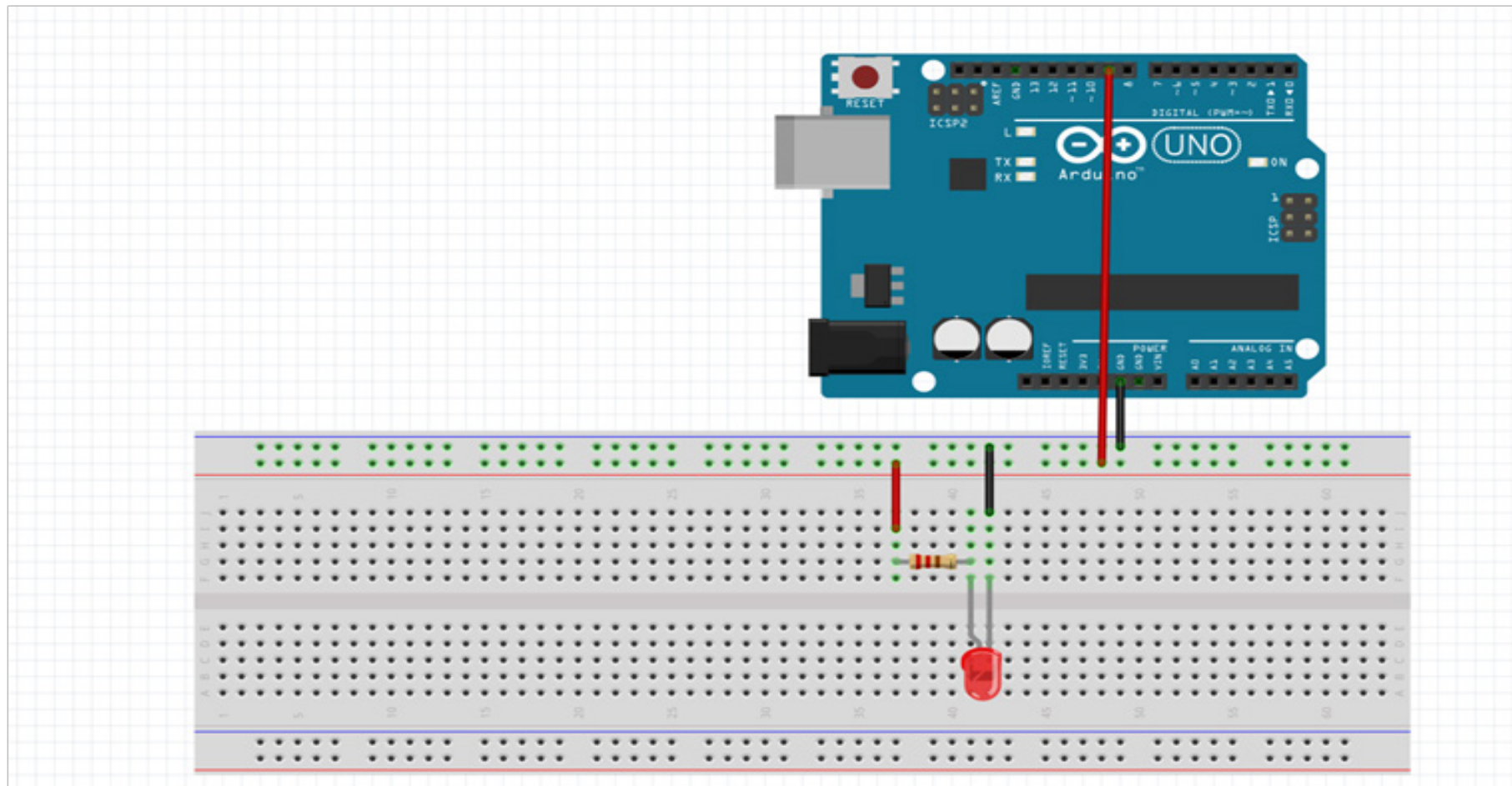
Dans cette documentation, nous expliquerons notre procédé au cours de ce projet et aussi toutes nos expériences réalisées. Nous présenterons également les aspects du programme pour chaque expérience réalisée. A la fin de ce document nous inséreront toutes les fiches techniques récoltées nécessaires au bon déroulement de ce projet.

Première expérience : Oscillation de la lumière sur fréquence donné  
Pour réaliser cette expérience, nous utilisons :

- Une carte Arduino
- Une led
- Une résistance pour limiter le courant dans la led
- Une platine d'essai



Nous les assemblons suivant le schémas suivant (réalisé sur le logiciel Fritzing pour un meilleur rendu) :



*Nous utilisons le pin 9 qui est un pin PWM ; ce qui va nous permettre de contrôler la tension en sortie.*

Ensuite, nous chargeons notre code(1) .

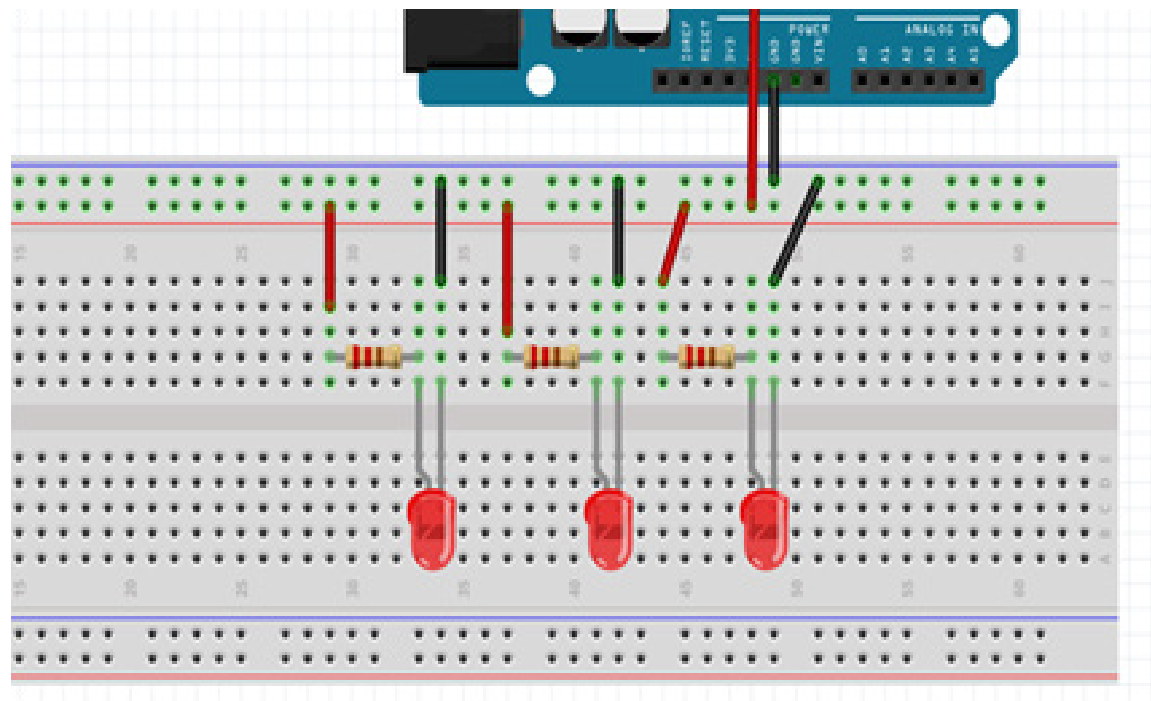
Experiencel.ino en annexe

<https://www.farnell.com/datasheets/1682209.pdf>

```
void loop() {  
  // change the analog out value:  
  analogWrite(analogOutPin, outputValue);  
  outputValue+=add;  
  if (outputValue == 255) add = -1;  
  if (outputValue == 0) add = +1;  
  
  // wait 20 milliseconds before the next loop for the analog-to-digital  
  delay(20);  
}
```

Ce code nous permet par variation progressive de faire clignoter la led à la fréquence 0,1 Hz avec des ondes en forme de dents de scies.

Dans cette expérience, nous réalisons l'interaction entre l'image et le son de notre projet. Avec une compilation, faite par l'artiste Pierre, nous récupérons des informations sonores à l'aide du logiciel Processing. L'on refait le même montage que précédemment mais cependant avec 3 leds en parallèle (pour simuler les 3 Spots qui éclaireront la toile).

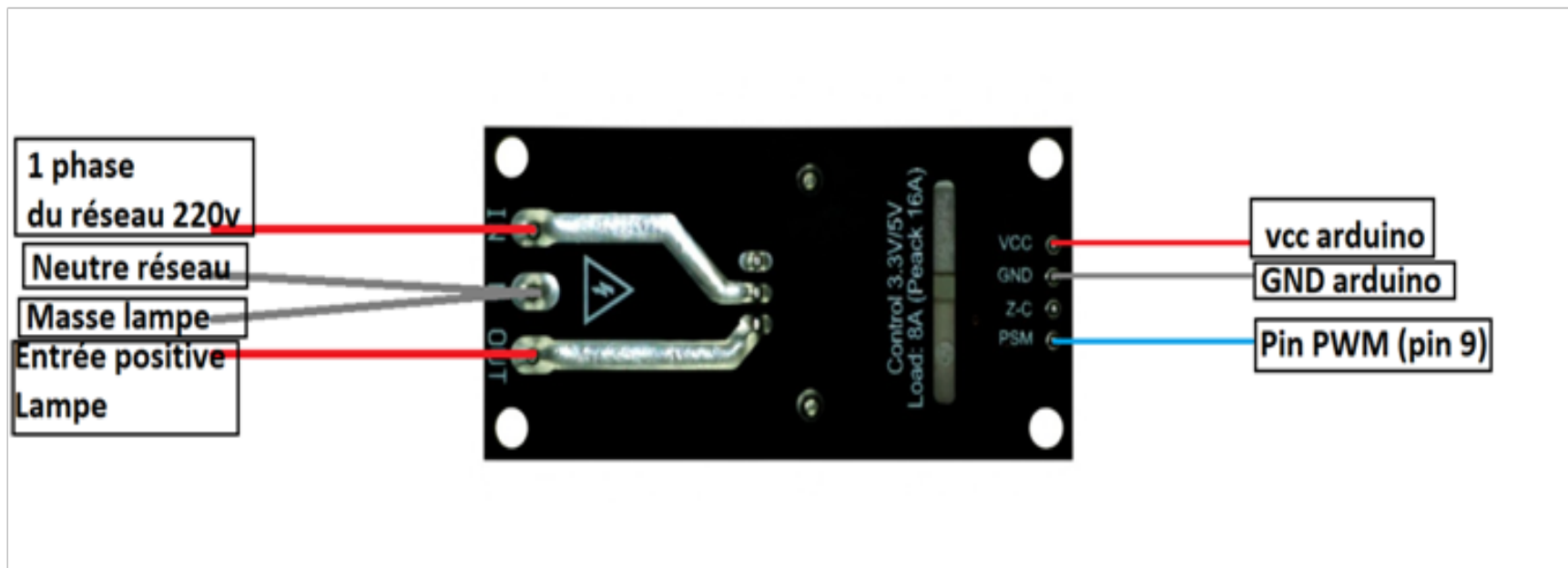


```
void draw() {  
  background(255);  
  println(amp.analyze());  
  
  int soundData = int(map(amp.analyze(), 0, 1, 0, 255));  
  
  circle(width/2, height/2, soundData);  
  
  arduino.analogWrite(9, soundData);  
}
```

Nous insérons alors le code(2) réalisé sous fritzing qui nous permet dans un premier temps de récupérer les informations sonores (méthode analyse() ) puis de le transformer( fonction map() ) en entier compris entre 0 et 255 (qui fait la valeur en décimal d'un byte). L'on passe ensuite cette valeur à la carte arduino qui va ainsi appliquer la tension adéquate à la pin 9 de l'arduino que nous utilisons.

(2) *FinalTest.pde*

Afin, de réaliser la connexion entre l'Arduino et la lumière, nous utiliserons un variateur que nous connecterons suivant les indications afin de réaliser le contrôle de la lumière.



<https://fichier.vs-elec.fr/v-elec/produits/1495/Glow.ino>

