Projet Arduino

Rapport bibliographique

PeiP2

Année scolaire 2020-2021

Titre : Polightech

Étudiants : Alex INFANTINO
Louan PICON BANDEIRA DA SILVA

Encadrant : Pascal MASSON

Introduction :

Le projet Polightech est l'union de plusieurs mini-projets que nous souhaitons créer autour de l'utilisation de LED (diode électroluminescente) RGB (Rouge Vert Bleu). La simple utilisation de lumière permet de créer beaucoup d'ambiances variées et nous constatons que c'est un marché grandissant avec par exemple la gamme Hue de Phillips ou encore les produits de Xiaomi.

Au-delà du choix par l'utilisateur d'ambiances prédéfinies et la création possible d'ambiances, nous avons imaginé d'autres fonctionnalités comme la reproduction d'un système Ambilight (technologie phare de Phillips), la réalisation d'un procédé de clignotement et de changement de couleur des LED du ruban en fonction des notifications reçues sur notre téléphone et la commande vocale d'allumage des LED et de changement de couleur. Nous souhaitons rassembler toutes ces fonctionnalités autour d'une application smartphone pour plus d'ergonomie et d'interactivité avec l'utilisateur.

Tout d'abord, nous étudierons le différent matériel nécessaire à la réalisation du projet. Ensuite, nous verrons plusieurs projets existants qui correspondent aux fonctionnalités citées précédemment. Enfin, nous conclurons sur nos besoins matériels.

Étude du différent matériel nécessaire à la réalisation du projet :

Les cartes Arduino :

Il existe de nombreuses cartes Arduino, chacune ayant sa caractéristique propre, son cadencement horloge, son nombre d'entrées/sorties numérique, etc... Le choix de la carte est important car chaque projet a des ambitions différentes et donc des besoins différents.

La figure 1 montre un tableau comparatif des cartes Arduino.

Cartes Arduino / Caractéristiques	UNO R3 (classique & CMS)	UNO R3 Ethernet (classique & POE)	Leonardo	Mega 2560	Mega ADK	DUE	Esplora	Mini	Nano	Yun (classique & POE)	Zero PRO
Microcontrôleur	ATmega328P	ATmega328P	ATmega32u4	ATmega2560	ATmega2560	AT91SAM3X8E	ATmega32u4	ATmega328P	ATmega328P	ATmega32u4	ATSAMD21G18
Cadencement Horloge	16 MHz	16 MHz	16 MHz	16 MHz	16 MHz	84 MHz	16 MHz	16 MHz	16 MHz	16 MHz	48 Mhz
Tension d'entrée	7 - 12V	7 - 12V	7 - 12V	7 - 12V	7 - 12V	7 - 12V	7 - 12V	7 - 9V	7 - 9V	5V	5V
Tension de fonctionnement	5V	5V	5V	5V	5V	3,3V	5V	5V	5V	5V	3,3V
Entrée/Sortie Numérique	14/6	14/4	20/7	54/15	54/15	54/12	✗	14/6	14/6	20/7	14/12
ntrée-Sortie (PWM) Analogique	6/0	6/0	12/0	16/0	16/0	12/2 (DAC)	✗	8/0	8/0	12/0	6/1 (DAC)
Mémoire vive (Flash)	32 Ko	32 Ko	32 Ko	256 Ko	256 Ko	512 Ko	32 Ko	32 Ko	32 Ko	32 Ko	256 Ko
Mémoire vive (SRAM)	2 Ko	2 Ko	2,5 Ko	8 Ko	8 Ko	96 Ko	2,5 Ko	2 Ko	2 Ko	2,5 Ko	32 Ko
Mémoire morte (EEPROM)	1 Ko	1 Ko	1 Ko	4 Ko	4 Ko	✗	1 Ko	1 Ko	1 Ko	1 Ko	16 Ko
Interface USB	USB-B mâle	USB-B mâle	Micro-USB	USB-B mâle	USB-B mâle & USB-A pour Android	2 ports micro-USB (Native et programming)	Micro-USB	✗	Mini-USB	Micro-USB	2 ports micro-USB (Native et programming)
Port UART	1	1	1	4	4	4	✗	✗	1	1	2
Carte SD	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗
Ethernet	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗
Wi-Fi	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Dimensions	68x53mm	68x53mm	68x53mm	101x53mm	101x53mm	101x53mm	165x60mm	30x18mm	45x18mm	68x53mm	68x53mm

Figure 1 : *tableau comparatif des cartes Arduino*

Les modules Bluetooth :

Le module Bluetooth permet la communication de l'utilisateur, via un smartphone, avec la carte Arduino. L'application sera l'interface qui permettra à l'utilisateur de communiquer ses envies d'ambiances. Il obtiendra alors le résultat attendu grâce aux LED.

Différents modules Bluetooth existent mais regardons les plus utilisés :

- Module Bluetooth HC-06 : ce module est un module esclave ce qui signifie qu'il est piloté par un autre module Bluetooth (smartphone). Il se compose de trois entrées : VCC, GND et RX (R pour réception). Il y a aussi une sortie TX (T pour transmission).

- Module Bluetooth HC-05 : ce module est similaire au module HC-06. La seule différence est que le module HC-05 peut fonctionner en esclave ou en maître. Il peut donc contrôler d'autres modules Bluetooth. Ce module est représenté dans la figure 2.

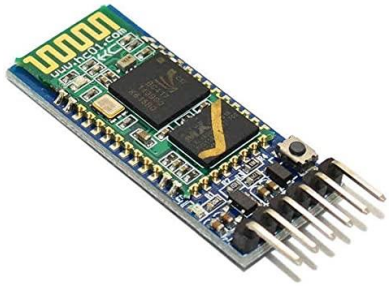

Figure 2 : *module Bluetooth HC-05*

L’application smartphone :

L’application smartphone est au centre de notre projet puisque l’utilisateur choisit l’ambiance qu’il désire grâce à l’application. Pour la créer facilement, de nombreux outils existent mais voyons le plus utilisé d’entre eux, App Inventor.

App Inventor est un environnement et un langage de programmation gratuit et libre d’accès. Elle a été développée par Google et elle est désormais entretenue par le MIT (Massachusetts Institute of Technology). Elle permet de réaliser des applications pour smartphones et tablettes.
Elle utilise une méthode de programmation graphique par blocs (voir figure 3) qui évite les erreurs de syntaxe et qui rend son accès plus facile.

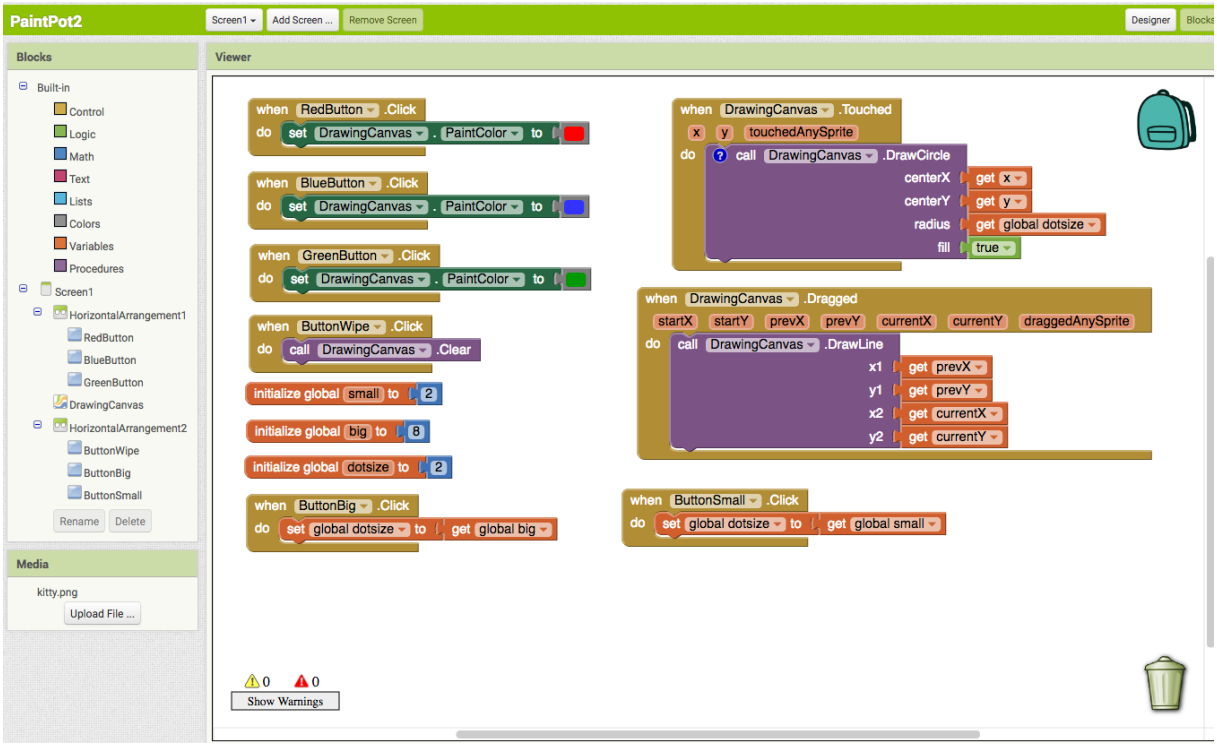

Figure 3 : *interface d’App Inventor*

Les microphones :

Un microphone est composé d’une plaque fixe et d’une plaque mobile. Les vibrations de l'air sont captées et vont déplacer la membrane mobile créant une variation du potentiel électrique qui va être retranscrit en un signal électrique.

Pour la fonctionnalité de commande vocale d'allumage des LED et de changement de couleur, il faut un microphone de qualité qui retranscrit le message vocal. Si cela fonctionne bien, le microphone capte les ondes sonores de la voix et les transforme en signal électrique que nous utiliserons pour déclencher la commande allumant ou changeant la couleur sur le ruban de LED.

Ainsi différents modèles existent et ont des sensibilités différentes qui répondent à différents usages. Les principaux modèles de microphone sont représentés dans la figure 4.

Figure 4 : *différents modèles de microphone pour une carte Arduino*

- MAX4466 : ce microphone est le plus adapté pour des projets comme le changeur de voix, l'enregistrement et l'échantillonnage audio et les projets audio-réactifs.

- KY-038 : ce microphone est composé de quatre éléments. Le capteur qui effectue la mesure, l'amplificateur où le signal analogique est ensuite envoyé, un comparateur pour commuter la sortie numérique et une diode à sensibilité ajustable. Il est plus adapté à la surveillance de la température, la détection de proximité ou encore la surveillance d'alarmes.

- KY-037 : ce microphone est très proche du microphone KY-038 mais il a une plus grande sensibilité.

D'autre modèles sont envisageables pour des projets Arduino mais il existe un microphone de très bonne qualité que nous possédons tous, c'est celui de notre smartphone. Ce microphone est donc une option très intéressante pour des projets utilisant un smartphone.

Les rubans de LED :

Il existe deux types de rubans de LED : les analogiques et les numériques. Leurs différences résident dans les circuits de construction.

D'une part, les rubans RGB analogiques ont quatre circuits : un pour la puissance et un pour chacune des couleurs, comme le montre la figure 5.

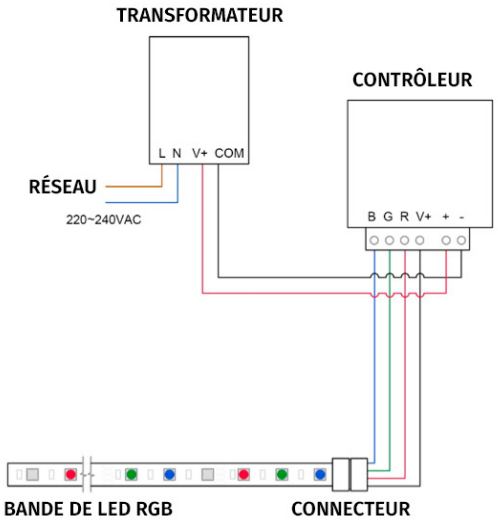

Figure 5 : *schéma d'un montage de ruban LED analogique*

D'autre part, les rubans LED numériques sont composées de trois à quatre circuits selon le type de puce utilisé. Il y a ainsi un circuit pour l'alimentation (5 VDC ou 12 VDC), un pour la masse et un pour les données. Il y a deux catégories de bandes LED numériques.

La première catégorie utilise le PWM (Pulse Width Modulation), qui synthétise des signaux continus à fréquence fixe sans retour d'information.

La seconde catégorie utilise le SPI (Signal Peripheral Interface), qui offre la possibilité à la bande de communiquer avec le dispositif sur lequel elle est connectée via le quatrième fil appelé Clock.

Le choix entre les deux catégories dépend finalement du dispositif que nous connectons aux bandes et de la compatibilité du futur projet, ainsi en Arduino nous pouvons travailler avec les deux.

En PWM, nous retrouvons principalement les WS281X avec 4 modèles différents :

- WS2811
- WS2812
- WS2812B
- WS2813

Le modèle WS2812B est une évolution des modèles WS2811 et WS2812. C'est une bande de 30, 60 ou même 144 LED en série et c'est celui que nous retrouvons le plus fréquemment dans les projets Arduino sur Internet bien qu'il ait une évolution : le WS2813.
La nouveauté du WS2813 est la présence d'un quatrième fil qui lui permet lorsqu'une LED tombe en panne et que la transmission d'information sur le bandeau s'arrête de jouer un rôle de pont et ainsi permettre le transfert d'informations à la LED qui suit celle endommagée pour ne pas stopper le fonctionnement du ruban.
Il existe d'autres modèles en PWM mais ce sont simplement des copies ou des variantes apportant une différence de luminosité ou de chaleur des couleurs.

En SPI il existe aussi différents modèles :

- WS2801
- LDP8806
- APA102
- SK9822/APA102C

Les deux premiers sont des modèles similaires et sont les plus utilisés notamment pour les projets de type Ambilight mais ils possèdent des désavantages. Ils sont limités à 30 LED par mètre et ils ont des problèmes de blanc.
Le APA102 et le SK9822/APA102C sont deux modèles presque identiques et considérés comme les plus performants grâce à des vitesses de transmissions et des fréquences de communication bien plus élevées que les autres modèles.

De plus, les rubans de LED peuvent avoir quatre niveaux de tension différents :

- 5 VDC : il est peu commun car une aussi petite tension ne sert qu'à des petits projets avec une connexion directe à un port USB le plus souvent.

- 12/24 VDC : ce sont les plus répandus de façon générale. Ils nous permettent de couper le ruban tous les 5 centimètres en 12 VDC ou tous les 10 centimètres en 24 VDC. Nous pouvons donc être plus précis

qu'avec des 220 VDC. Le principal problème est la longueur du ruban nécessaire. À partir d'une longueur d'environ 10 mètres nous pouvons avoir des chutes de tension. Ce n'est donc pas adapté à tous les projets.

- 220 VDC : il ressemble au 12/24 VDC mais il se connecte directement au secteur et n'a donc pas besoin d'un transformateur. Il est le plus adapté pour de grandes installations mais il est du coup moins pratique.

Autre paramètre important : le nombre de LED par mètre. C'est en fonction de ce rapport que nous allons avoir une différence de luminosité produite par le ruban. Ainsi, selon l'objectif d'éclairage, il faut avoir une plus grande ou moins grande densité de LED. Il y a donc plusieurs formats différents :

- 30 LED par mètre : la plus faible densité, elle a donc un coût de production moindre et peut convenir à de nombreux usages.

- 60 LED par mètre : le format moyen offrant un très bon équilibre entre coût et intensité lumineuse.

- 120 ou 144 LED par mètre : ces formats sont très utilisés quand il faut beaucoup d'intensité comme pour l'éclairage extérieur par exemple. Ils sont en revanche beaucoup plus cher.

Il existe aussi d'autres formats comme 100 LED par mètre ou encore 240 LED par mètre mais ils sont moins courants.
La figure 6 montre deux bandes de LED RGB à 30 et à 60 LED par mètre.

Figure 6 : *bandes LED RGB*

Étudions maintenant les puces utilisées par les différents rubans. Nous pouvons en recenser principalement quatre dans le marché :

- SMD 3528/2835 : ces puces sont utilisées pour les bandes les moins énergivores donc elles sont peu puissantes et à faible consommation. C'est donc l'option la plus économique.

- SMD 5050 : ce modèle est très répandu. Il offre un bien meilleur éclairage qui est évidemment accompagné d'une plus grande consommation.

- SMD 3014 : ces puces ont des performances lumineuses excellentes et une faible consommation.

- SMD 5630 : ce type de puce est utilisé pour atteindre le maximum de puissance. Son usage est le plus souvent à des fins professionnelles.

La figure 7 montre les différentes puces SMD étudiées.

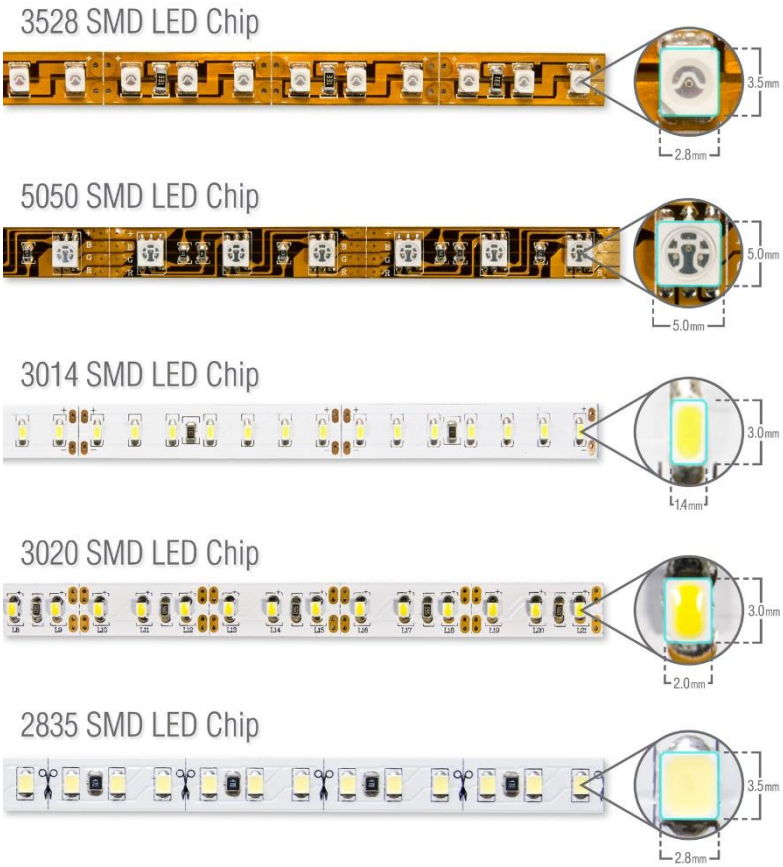

Figure 7 : *puces SMD*

Les logiciels Ambilight :

Une des parties de notre projet est de créer un système Ambilight. C'est une technologie qui permet de retranscrire les couleurs de l'écran en temps réel à des LED fixées à l'arrière de l'écran pour avoir un effet de prolongement, comme le montre la figure 8.

Figure 8 : *exemple d'utilisation du système Ambilight*

Il existe ainsi de nombreux logiciels qui permettent de découper les bordures de l'écran en plusieurs petites portions et d'en analyser la couleur pour envoyer l'information à la carte Arduino. Les plus utilisés sont Ambibox, Adrilight et Prismatik mais il en existe d'autres tel que Bambilight, Boblight, Screenbloom, etc…

Globalement, le fonctionnement des logiciels est le même. Après l'installation du logiciel, nous paramétrons le nombre de LED, la portion d'écran que chaque LED doit s'occuper, la disposition des LED (par exemple dans les coins ou non) ou encore le ratio de l'écran. La figure 9 est une capture d'écran de la fenêtre de paramétrage du logiciel Ambibox.

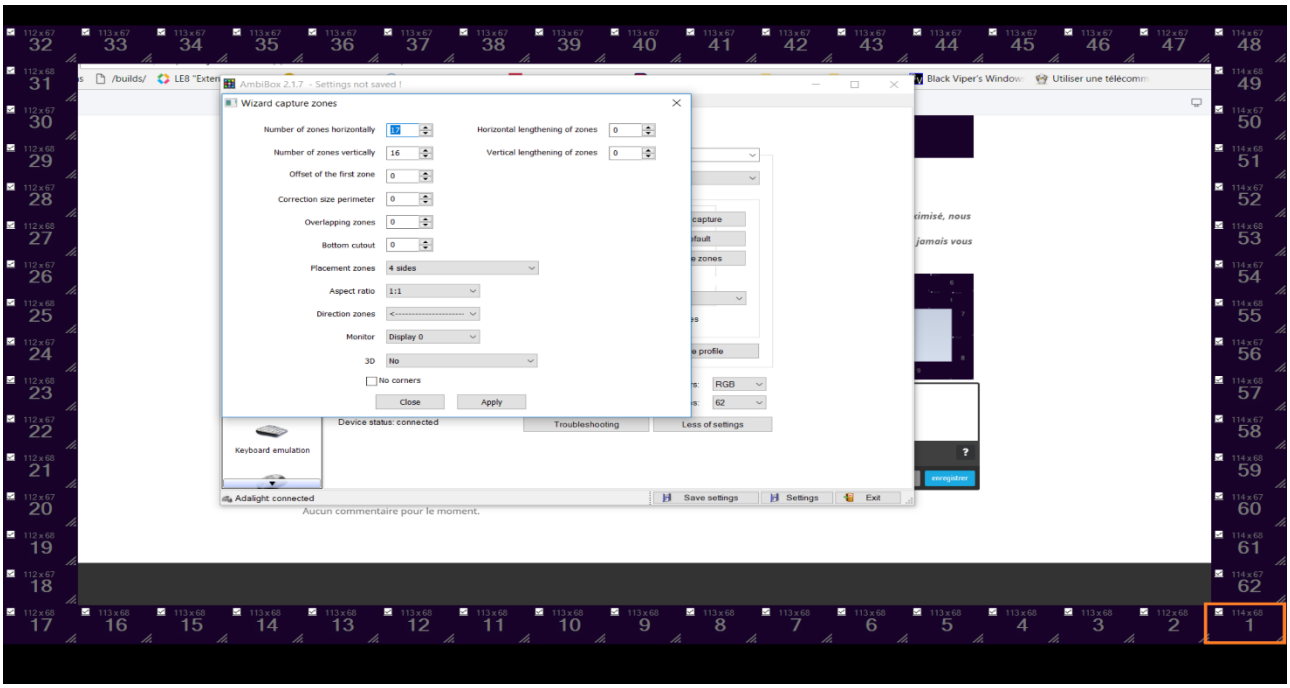

Figure 9 : *capture d'écran de la fenêtre de paramétrage du logiciel Ambibox*

Analyse de plusieurs projets existants :

Analyse d'un projet Ambilight :

Nous analysons ici un projet afin d'installer la technologie Ambilight à l'arrière d'un écran (lien du projet : https://ambimod.jimdofree.com/2017/02/22/tuto-mettre-en-place-un-syst%C3%A8me-ambilight-sous-windows-avec-ambibox/). Tout d'abord, le matériel utilisé est une carte Arduino Nano (ce format ne semble pas obligatoire puisqu'aucune contrainte liée aux cartes Arduino Uno n'a été remarquée), une bande de LED de type WS2812B avec 62 LED munies de trois connecteurs d'angle pour améliorer l'ergonomie, les Sketchs AdalightAmbiboxFastLED ou AdalightAmbiboxFastLED2, au choix, ainsi que du logiciel AmbiBox pour Windows.

Ensuite, des mesures sont effectuées (avec des connecteurs d'angle il faudra bien penser à adapter la longueur des découpes de la bande). Il obtient finalement un rectangle de 19 par 12 LED (cela dépend de la taille de l'écran). Pour le câblage, nous avons deux options selon le type de bande de LED comme le montre la figure 10.

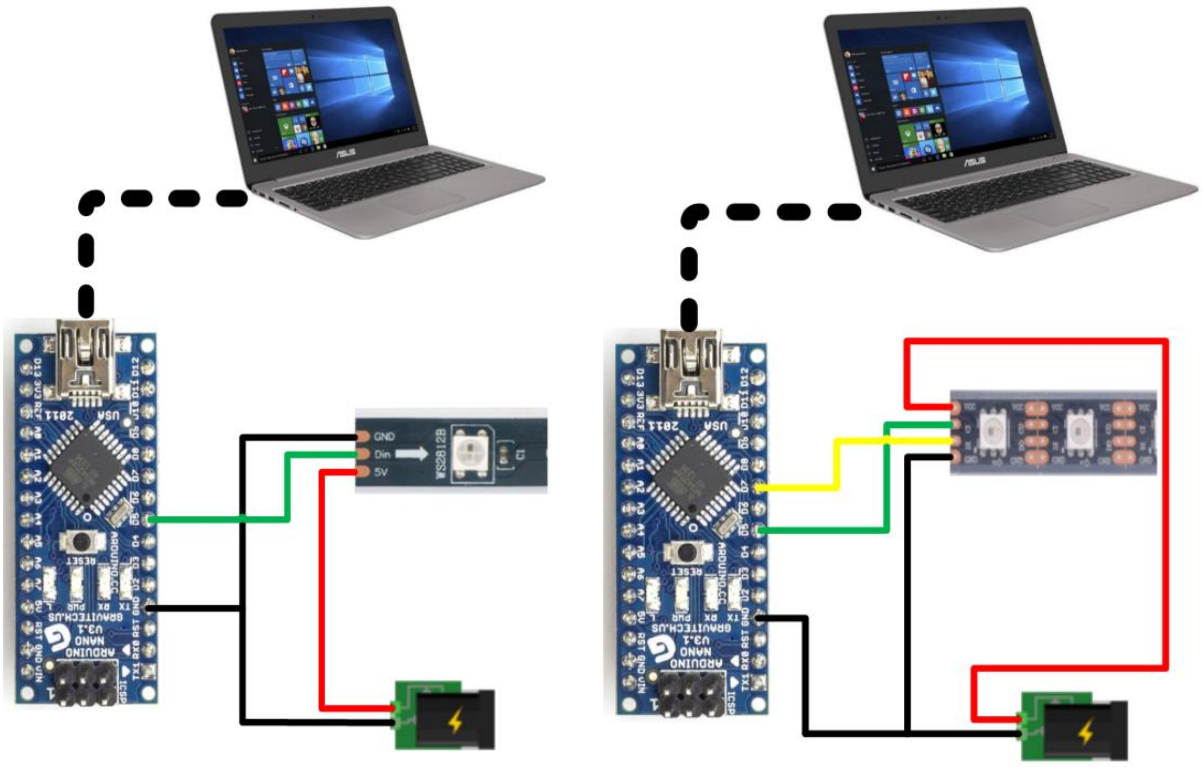

Figure 10 : *deux options de câblage*

La programmation se fait avec le Sketch AdalightAmbiboxFastLED, seul quelques modifications seront nécessaires en fonction du choix de bande de LED. Nous définissons ainsi le nombre de LED, la broche du fil DATA, l'utilisation ou non de Clock, le type de LED et l'ordre des LED. Le choix de la vitesse du port est important car il dépend des composants.

Ensuite, il installe le logiciel Ambibox. Après quelques petites manipulations, il installe PlayClaw, un logiciel qui permet de récupérer ce qui est affiché à l'écran. À la fin de l'installation, il configure le dispositif Adalight pour aboutir à l'effet Ambilight. Cependant les LED sont peut-être allumées mais leur disposition dans le logiciel n'est pas du tout celle mis en place derrière l'écran, il faut donc afficher la disposition des LED comme le montre la figure 11.

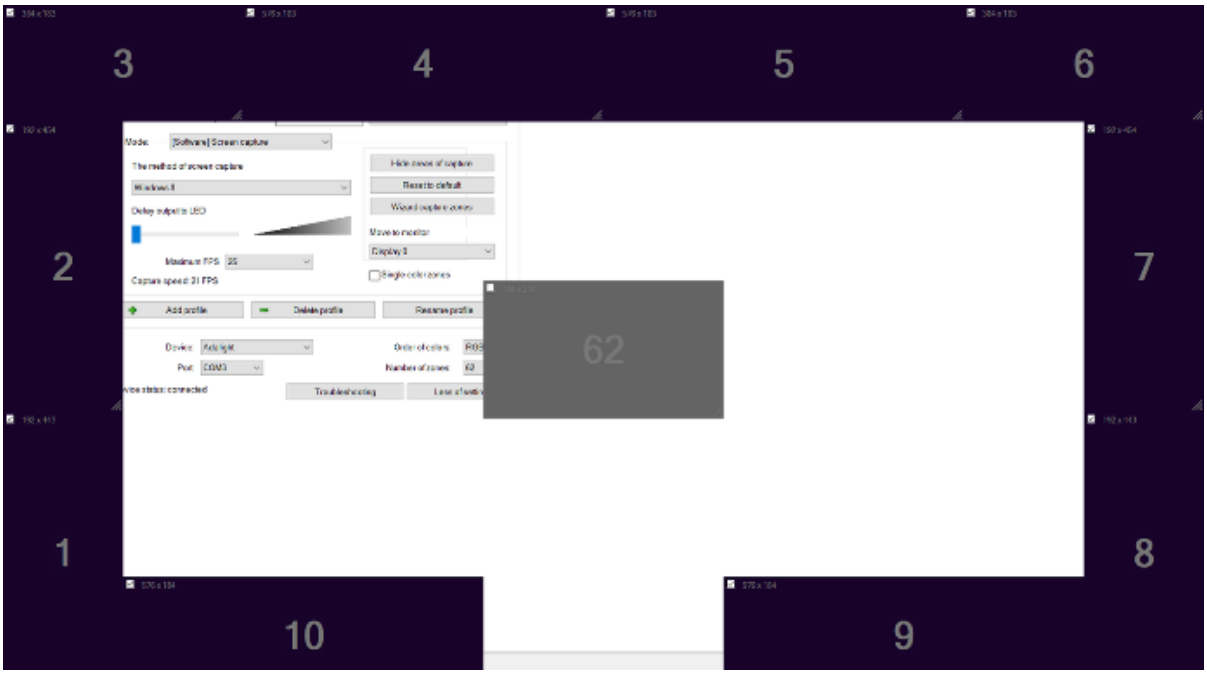

Figure 11 : *capture d'écran du paramétrage de la disposition des LED*

S'enchaîne donc un paramétrage pour que chaque case numérotée corresponde à une LED derrière l'écran. Un bon alignement est nécessaire pour avoir un bon effet sous peine de voir les lumières derrière l'écran ne pas correspondre à leur portion d'écran.
Finalement avec des modifications telles que la définition du nombre de cases dans la fenêtre, l'option « No corners » car il n'y a pas de LED dans les coins, la définition du ratio sur 16:9, etc…, on arrive à un résultat optimal.
Il suffit maintenant d'alimenter les LED et de lancer le logiciel pour que le système Ambilight fonctionne.

Analyse d'un projet d'ajustement des LED en fonction des notifications reçus sur le téléphone :

Cette autre composante de notre projet a pour but de créer une méthode modifiant les couleurs de notre ruban de LED en fonction des notifications reçues sur notre smartphone. Par exemple si nous recevons une notification Facebook, nous paramétrons Arduino pour que les LED s'allument en bleu, en rouge pour YouTube, etc....

Ainsi nous analysons ici un projet très similaire au notre (lien du projet : https://create.arduino.cc/projecthub/geny-studio/notification-iot-using-neopixel-and-smartphone-945e81?ref=tag&ref_id=notifications&offset=5)

Concernant le matériel, nous avons besoin d'une carte Arduino Uno, d'un module Bluetooth HC-06 ainsi que d'un Adafruit NeoPixel Ring.
Ensuite, il suffit d'un smartphone et de l'application Notiduino qui a été développée par le créateur de ce projet.

Premièrement pour le montage il suffit de connecter le module Bluetooth et l'anneau (avec un câble vers le port d'alimentation et un autre vers la masse) à la carte (à noter qu'ici l'anneau est connecté au port d'alimentation mais pour nous il faudra connecter le ruban directement à une prise car les LED du ruban consomment bien plus d'énergie). La figure 12 montre le schéma du montage.

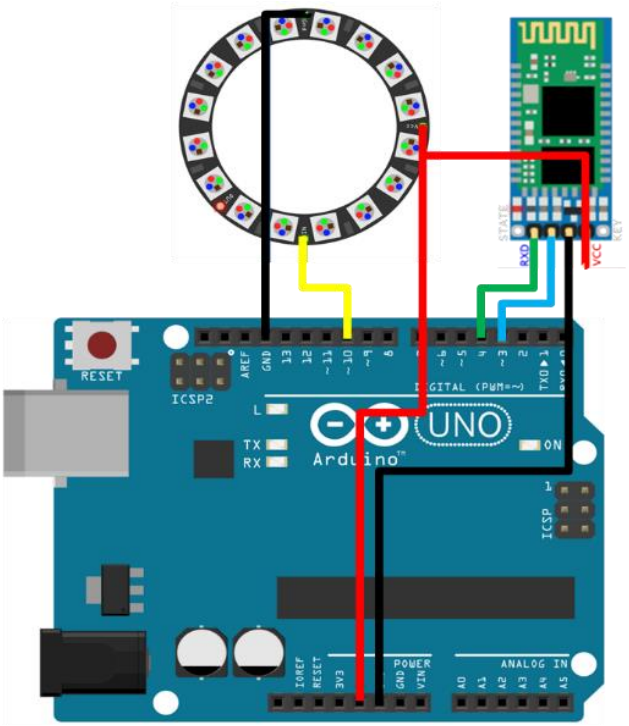

Figure 12 : *schéma du montage*

La création du code s'est faite avec les librairies SoftwareSerial et Adafruit qui semblent très pratiques dans ce genre de projet. Le code va récupérer un nombre transmit par le module Bluetooth qui, en fonction de sa valeur, va permettre à la carte de créer des nuances des couleurs rouge, vert et bleu, permettant ainsi d'offrir une grande palette de couleurs aux LED.

Après la mise en place du montage, l'écriture du code et le téléchargement et lancement de l'application Notiduino, nous sélectionnons les applications du téléphone dont nous voulons que les notifications soient affichées grâce aux LED et nous paramétrons la couleur associée souhaitée.

Finalement, le smartphone est connecté par Bluetooth à la carte Arduino et au moment de la réception d'une notification, l'anneau va s'adapter à la bonne couleur.

Dans notre projet nous souhaitons obtenir un clignotement de la couleur paramétrée, ce qui est légèrement différent par rapport à ce projet.

Conclusion :

Finalement, après étude des options qui s'offrait à nous, nous avons choisi le matériel pour le projet Polightech.

Concernant la carte Arduino nous choisissons la carte Arduino Uno car bien que basique elle répond à tous nos besoins. Il n'est donc pas nécessaire d'en prendre une plus avancé. L'équivalent de cette carte, la Xplained Mini, est très bien d'autant plus que nous sommes déjà en possession de celle-ci.

Pour le module Bluetooth nous optons pour le module HC-06 car nous n'avons pas besoin de contrôler d'autres modules Bluetooth avec celui-ci.

Pour créer l'application smartphone nous utiliserons App Inventor car c'est l'outil le plus utilisé et donc nous pourrons trouver de nombreux tutoriels pour obtenir un très bon résultat.

Pour le microphone nous restons sur le microphone de notre smartphone. Ce sera plus pratique et plus fonctionnel, d'autant plus que nous utilisons notre smartphone pour l'application.

Pour le ruban de LED nous optons pour le modèle WS2812B de NeoPixel à 60 LED par mètre car la technologie PWM nous semble la plus adapté et la plus simple d'utilisation. De plus, NeoPixel présente l'avantage d'avoir une librairie Arduino très pratique.

Pour le logiciel Ambilight nous nous orientons vers Ambibox car il semble être très complet et de nombreux projets similaires l'utilisent.