

MASTERMIND ARDUINO

RAPPORT DE SÉANCE : Semaine de 9 décembre

Objectif de la séance :

- Étude de la faisabilité
- Dimension projet
- Programmer bande de LEDS

1) ÉTUDE DE LA FAISABILITÉ

Notre projet est composé essentiellement de LED (en tout : $(4*12)+(4*12)+4 = 100$ LEDs)
Pour minimiser les branchements et l'utilisation des ports on utilise des **rubans de LEDs Neopixels**.

1^{ère} étape : comprendre comment fonctionne et comment brancher un ruban de LEDs : on commence par regarder sur Internet quelques vidéos et sites explicatifs

Ce que j'ai regardé :

<https://www.youtube.com/watch?v=EvRxQbhDxCo> (comment brancher ruban)

https://www.youtube.com/watch?v=sTwj7fs-Z_4 (comment couper/souder ruban)

<http://www.fablabredon.org/wordpress/2017/12/17/lumiere-sur-larduino-avec-de-la-couleur-et-des-led/> (comment coder ruban)

2^{ème} étape : étude du codage des différentes couleurs d'une LED RGB

La couleur d'une LED RGB est caractérisée selon 3 paramètres : teneur en rouge, teneur en vert, teneur en bleu.

Exemple :

Rouge = {255, 0, 0}

Vert = {0, 255, 0}

Jaune = {255, 125, 0}

Violet = {255, 0, 255}

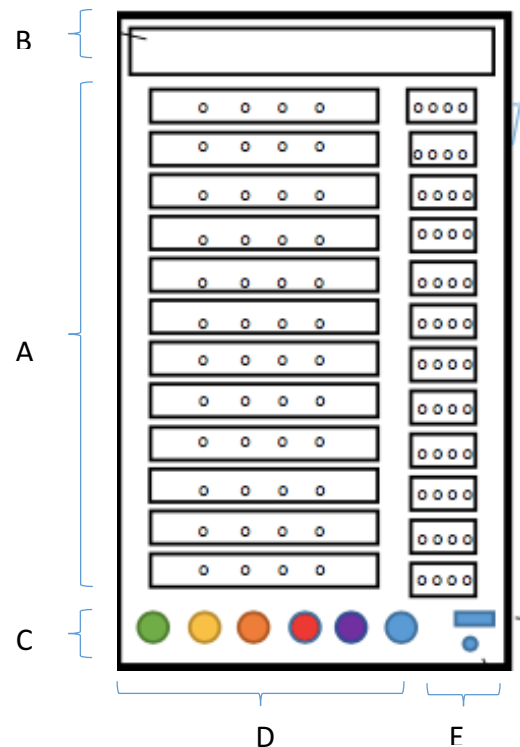
3^{ème} étape : Répartition des LEDS

Plusieurs possibilités :

- ~~découper 25 bouts de rubans de 4 LEDs~~
- ~~découper 8 bouts de rubans de 12 LEDs~~
- serpentins de LEDS ✓

2) DIMENSION PROJET

Notre projet sera dans l'idée sous la forme suivante :



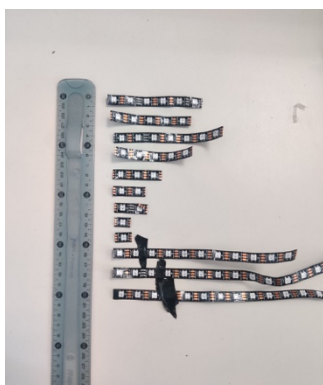
Soit constitué de plusieurs parties :

- Le corps ($A * D$)
- La trappe (B)
- Les boutons ($C * D$)
- Les LEDS de vérification (=corps secondaire) ($A * E$)

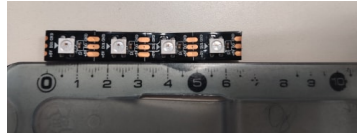
Idéalement notre Arduino sera **assez compact**, mais **suffisamment grand** pour ne pas avoir résultat trop chargé.

On commence par visualiser globalement les dimensions du projet :

Le **corps** est constitué de 12 parties de rubans de LEDs, cela représente **une vingtaine de centimètres** :



Plus précisément, chaque bout de rubans a une hauteur de 1cm et une longueur de 6,5 cm (+ on laisse entre chaque ligne 0,5 cm d'espace + un cadre de 1 cm d'espace tout autour du corps.)



D'où :

Longueur du corps = $2*1 + 6,5 = 8,5 \text{ cm}$

Hauteur du corps = $2*1 + 12*1 + 11*0,5 = 25 \text{ cm}$

Longueur de la trappe = $8,5 \text{ cm}$

Hauteur de la trappe = 3 cm

Longueur du corps secondaire = $2*1 + 3,3 = 5,3 \text{ cm}$

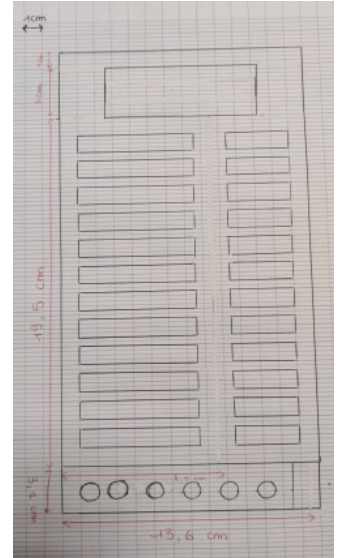
Hauteur du corps seconfaire = $19,5 \text{ cm}$

Longueur de l'emplacement boutons = $2*1 + 6*1,1 + 5*1 = 13,6 \text{ cm}$

Hauteur de l'emplacement boutons = $1*1,1 + 2*0,5 = 3,1 \text{ cm}$

Finalement, la caisse finale devra être de **dimension 13,6*32,1**

On prendra 15*35 pour être large et pouvoir brancher tous nos fils sans avoir de problèmes de places



3) PROGRAMMER BANDES DE LEDS

CHANGEMENT DE MISE EN PLACE :

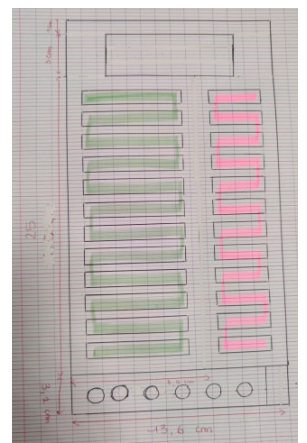
On pensait tout d'abord découper 25 bandes de 4 leds et les brancher sur notre carte Arduino pour les piloter idividuellement (par tranche de 4 LEDs).

Pour éviter d'avoir à alimenter 25 bandes différentes, on décide de faire un « **serpentin** » de **LEDs** afin d'avoir le moins d'alimentation possible à brancher.

On réalise deux serpentins distincts :

- En vert pour les LEDs principales du jeu
- En rose pour les LEDs secondaires du jeu

- Pour le serpentин vert, il sera constitué de 12 lignes de 4 LEDs
- Pour le serpentин rose, il sera constitué de 24 lignes de 2 LEDs



On a besoin de 6 couleurs différentes (codée en RGB) :

VERT : { 0 , 255 , 0 }

JAUNE : { 255 , 125 , 0 }

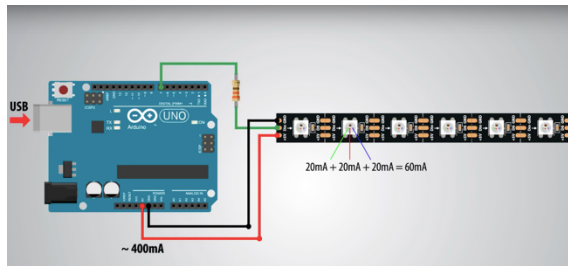
ORANGE : { 237 , 127 , 16 }

ROUGE : { 255 , 0 , 0 }

VIOLET : { 255 , 0 , 255 }

BLEU : { 0 , 127 , 255 }

Pour le branchement :



On installe la **bibliothèque FastLED** pour rédiger le code ARDUINO.

Prise en main de la bibliothèque et premiers codes pour tester l'allumage des LEDS :

```
Programmer_LEDS | Arduino 1.8.9

#include <FastLED.h>

const int LED_PIN=7;
const int NUM_LEDS=4;

CRGB leds[NUM_LEDS];

void setup() {
  FastLED.addLeds<WS2812, LED_PIN, GRB>(leds, NUM_LEDS);
}

void loop() {
  leds[0]= CRGB(255,0,0);
  FastLED.show();
  leds[1]= CRGB(0,255,0);
  FastLED.show();
  leds[2]= CRGB(150,0,255);
  FastLED.show();
  leds[3]= CRGB(255,200,0);
  FastLED.show();
}

Compilation terminée.
Le croquis utilise 3492 octets (11%) de l'espace de stockage de progr
Les variables globales utilisent 109 octets (5%) de mémoire dynamique

18 Arduino Nano sur /dev/cu.usbmodem1412
```

Premier problème qui se pose : seule la première LED s'allume (en rouge comme prévu), mais les suivantes restent éteintes.

On essaie les différentes couleurs, la première LED s'allume comme prévu de la couleur décidée lors du code, mais les LEDS suivantes restent éteintes → **à régler lors de la prochaine séance**