

# MASTERMIND ARDUINO

## RAPPORT DE SÉANCE : Semaine de 13 janvier 2020

### Objectif de la séance :

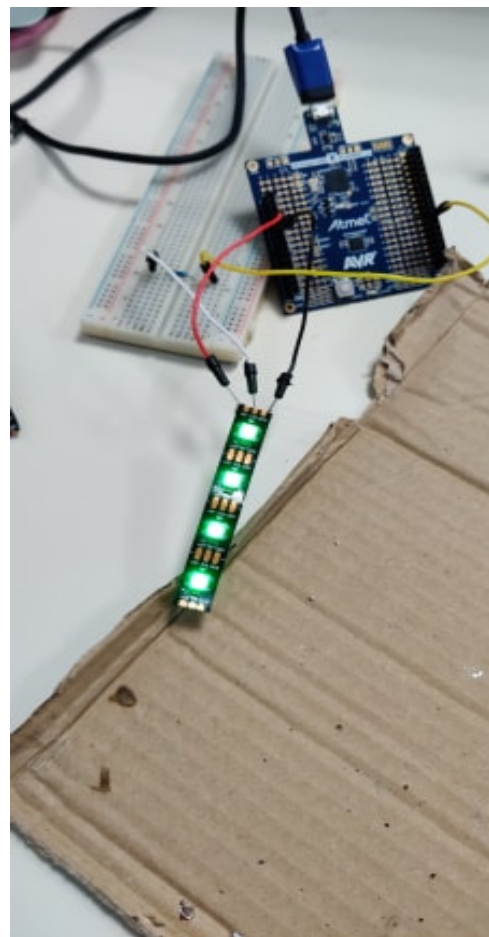
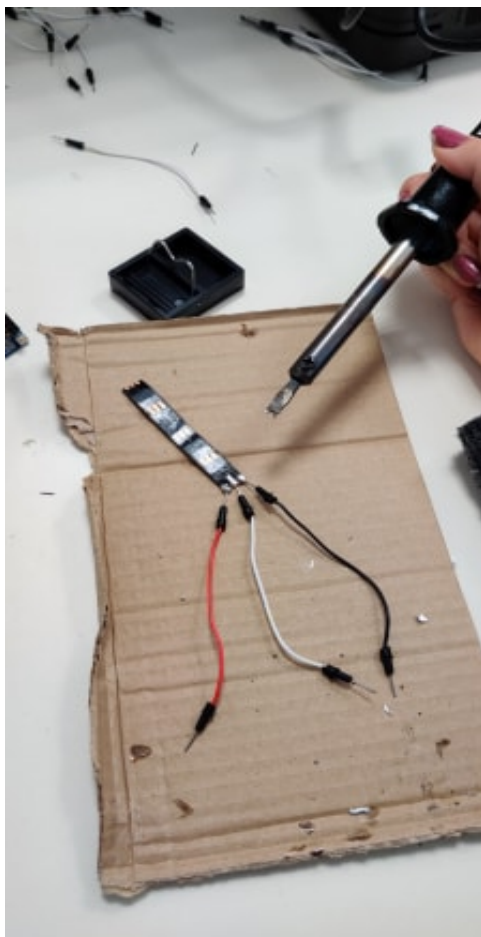
- Soudure LEDs
- ServoMoteur

### 1 ) SOUDURE LEDs

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**.

On réalise **deux serpentins de LEDs** : pour se faire on doit souder des portions de LEDs et tester les raccords à chaque soudure réalisée.

Le processus est assez long car **après chaque soudure on doit vérifier que tout fonctionne bien**. On utilise le programme pour tester les lumières RGB de chaque LEDs et on l'applique à chaque bout de bande de LEDs soudé.

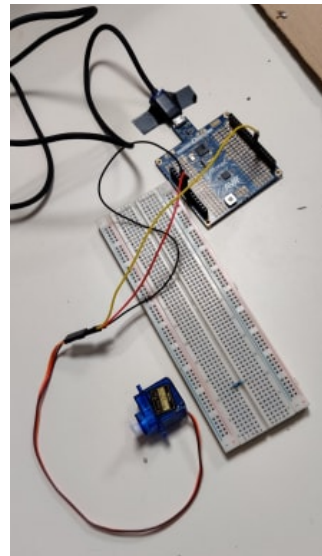
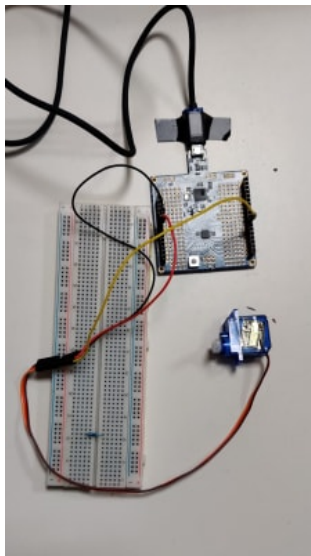


## 2) SERVOMOTEUR

Pour **ouvrir la trappe qui dévoile la solution finale**, on cherchait un moteur qui puisse se déclencher lorsque l'utilisateur trouvait la bonne réponse ou bien en cas d'échec.

Pour cela on utilise un ServoMoteur. Puisque c'était la première fois que j'utilisais un Servo moteur j'ai du chercher des tutoriels sur internet pour comprendre **comment le brancher et comment le programmer**.

Branchement du ServoMoteur :



puis on réalise un code test pour vérifier qu'il fonctionne bien :

```
Servo_test | Arduino 1.8.9

Servo_test
#include <Servo.h>

/* Créer un objet Servo pour contrôler le servomoteur */
Servo monServomoteur;

void setup() {
    // Attache le servomoteur à la broche D9
    monServomoteur.attach(9);
}

void loop() {
    // Fait bouger le bras de 0° à 180°
    for (int position = 0; position <= 180; position++) {
        monServomoteur.write(position);
        delay(15);
    }

    // Fait bouger le bras de 180° à 10°
    for (int position = 180; position >= 0; position--) {
        monServomoteur.write(position);
        delay(15);
    }
}

Téléversement...
Le croquis utilise 2108 octets (6%) de l'espace de stockage de programmes. Le max
Les variables globales utilisent 50 octets (2%) de mémoire dynamique, ce qui lais
```

Le code permet au servomoteur **d'effectuer des va-et-viens** : on vérifie ainsi qu'il fonctionne bien.

On applique maintenant l'utilisation du servo-moteur à notre projet. Le but est **d'effectuer un angle de 180° lorsque l'utilisateur trouve la bonne réponse ou lorsque la partie est finie.**