

Durant cette deuxième séance, nous avons commencé la partie Arduino et l'écriture de nos programmes : Eya a travaillé sur l'écran LCD pendant que je m'occupais des capteurs à leds infrarouges. Voici les différents points sur lesquels j'ai travaillé :

- **Branchement d'un capteur avec une led pour comprendre son fonctionnement**

```
1  const int capteur = 2;  
2  const int led=2;  
3  int val;  
4  
5  void setup() {  
6      Serial.begin(19200);  
7      pinMode(led,OUTPUT);  
8      pinMode(capteur, INPUT);  
9  }  
10 }  
11 void loop() {  
12     val=digitalRead(capteur);  
13     if (val==HIGH){  
14         digitalWrite(led,HIGH);  
15     }  
16     if (val==LOW){  
17         digitalWrite(led,LOW);  
18     }  
19 }
```

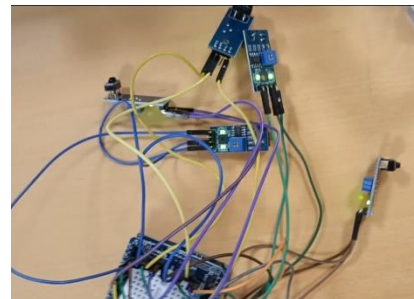
Pour commencer, j'ai fait un premier essai du fonctionnement du capteur à leds infrarouges pour être bien sûre de son fonctionnement en écrivant ce code permettant de détecter l'état du capteur et d'allumer ou éteindre la LED en conséquence.

Cette étape m'a permis de valider le fonctionnement d'un capteur en simulant une condition de détection avec une LED qui s'allume lorsque la pièce est détectée. Cela sert de base pour comprendre le principe de détection qui sera utilisé dans la suite du projet.

- **Branchements des 5 capteurs sur la carte et écriture du code**

J'ai connecté les 5 capteurs différents (10 centimes, 20 centimes, 1 euro, 50 centimes, 2 euros) à des sorties numériques puis j'ai écrit le code pour lire l'état de chaque capteur et afficher un message sur la console série en fonction de la pièce détectée, puis pour faire le calcul du total des pièces détectées.

Cela m'a permis d'étendre le système de détection à plusieurs types de pièces. Chaque capteur est associé à une valeur monétaire, et le total est mis à jour en fonction des pièces détectées.



- **Ecriture de la fonction pour compter les pièces lorsqu'elles passent devant les capteurs**

Développement d'une fonction `coinDetected()` pour compter le nombre de pièces détectées et intégration de cette fonction dans le code principal, appelée à chaque détection de pièce.

- **Assemblage des codes pour les capteurs et pour l'écran LCD**

Nous avons intégré nos deux codes précédemment développés en un seul programme qui contrôle le fonctionnement des capteurs ainsi que celui de l'écran LCD. L'assemblage des deux codes permet de détecter différentes pièces de monnaie, de les compter et d'afficher le total en euros sur un écran LCD :

- un message de bienvenue défile sur l'écran LCD au démarrage du système, offrant un visuel agréable.

```
lcd.backlight();
lcd.setCursor(3,0);
lcd.print("Welcome Back to");
lcd.setCursor(4,1);
lcd.print("the Coin S/C!");
delay(500);
for (int positionCounter = 0; positionCounter < 20; positionCounter++) {
  lcd.scrollDisplayLeft();
  delay(500);
}
```

- L'utilisation de la fonction `coinDetected()` permet de compter le nombre total de pièces détectées, affichant cette information à chaque détection.



- La fonction `displayAmount()` gère l'affichage du total sur l'écran LCD, avec des ajustements du curseur en fonction de la valeur totale, en voici un extrait :

```
void displayAmount() {
  lcd.setCursor(0,0);
  lcd.display();

  if (Total>=0.00 && Total<10.0){
    delay(1300);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Total :");
    lcd.setCursor(9,1);
    lcd.print(Total);
    lcd.setCursor(12,1);
    lcd.print("e");
  }
}
```

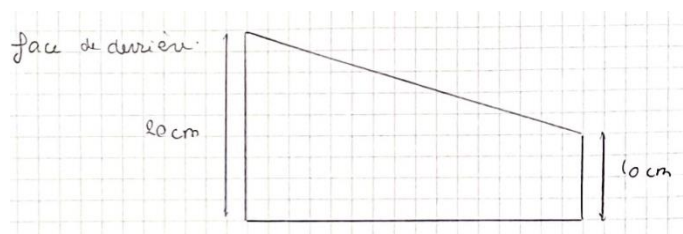
• Création d'une maquette aux dimensions

Une fois rentrée chez moi, j'ai entamé la reproduction de la machine en une maquette aux dimensions réelles.



Cela m'a permis de repérer les éventuels problèmes et d'ajuster les dimensions. Voici les différents points à modifier que j'ai notés :

- 1) Ajustement de l'angle en y de la pente pour la rampe : Il faut que nous augmentions l'angle d'inclinaison car avec celui prévu, les pièces ne glissent pas. L'écart entre les deux côtés de la pente est maintenant de 10 cm au lieu des 4 cm initialement prévus.



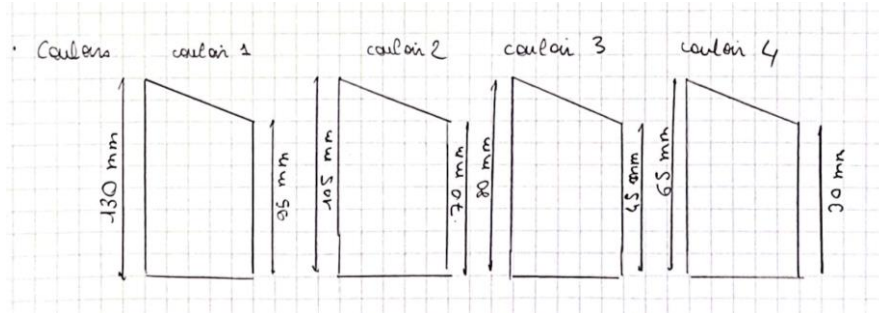
Mesures actuelles : 20 cm d'un côté et 10 cm de l'autre.

- 2) Ajustement de l'angle en x de la pente pour la rampe : l'angle en x est aussi trop faible : j'ai donc augmenté l'écart de la pente en x. L'écart entre les deux côtés de chaque couloir est maintenant de 3.5 cm au lieu de 2 cm.

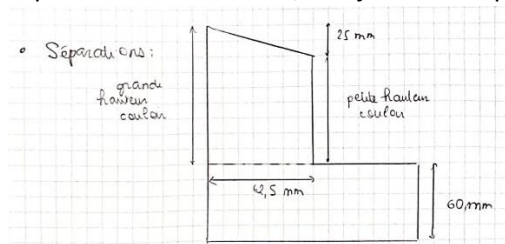
Mesures actuelles : 3.5 cm d'écart entre les deux côtés.

- 3) Uniformisation des dimensions des grands côtés des couloirs : J'ai ajusté les dimensions des grands côtés des couloirs pour maintenir une différence constante de 2.5 cm entre eux.

Dimensions actuelles : 13 cm, 10.5 cm, 8 cm, 6.5 cm.



- 4) Modification des séparations : Je pense qu'il vaut mieux regrouper les deux pièces de chaque séparation en une seule, en ajoutant simplement un rectangle de 6 sur 12 cm.



• **A faire prochainement :**

- Intégration des capteurs à la maquette pour vérifier le bon fonctionnement de tout le programme
- Validation des dimensions ajustées par la création d'une nouvelle maquette en carton plume.
- Modification en fonctions des pièces dans les fichiers svg