

Sigal KABLA, Rojin ALOGLU et Laura MARTINEZ CASTRO

Contexte :

Dans le cadre des activités du FABLAB et de l'UE Processus d'Innovation, cette séance avait pour objectif de découvrir l'utilisation de la plateforme Arduino, un outil de créer des systèmes interactifs combinant électronique et programmation.

Contrairement aux séances précédentes centrées sur la fabrication additive, cette session introduisait la dimension électronique et programmable du prototypage, essentielle dans la conception d'objets interactifs et intelligents.

Qu'est-ce qu'Arduino ?

Arduino est une plateforme open-source composée d'une carte électronique programmable appelée microcontrôleur, d'un logiciel de programmation nommé Arduino IDE, ainsi que de différents composants électroniques tels que des LEDs, boutons, capteurs ou moteurs.

Cette plateforme permet de contrôler des objets physiques grâce à un programme informatique. Arduino agit comme le cerveau du système en recevant des informations, en les traitant puis en exécutant une action.

À quoi sert Arduino ?

Arduino est utilisé pour concevoir et tester rapidement des prototypes interactifs. Il permet notamment d'automatiser des actions, de contrôler des capteurs, de piloter des dispositifs électroniques et de développer des objets connectés.

Parmi les applications possibles, on peut citer l'allumage automatique d'une lumière, la mesure de la température ou de la luminosité, l'automatisation d'un système d'arrosage ou encore la création de dispositifs domotiques. Arduino constitue ainsi un outil central dans les démarches d'innovation et de prototypage rapide.

Principe de fonctionnement :

Le fonctionnement d'Arduino repose sur une logique simple en trois étapes.

1. La carte reçoit une information provenant d'un capteur ou d'une interaction utilisateur
2. Le microcontrôleur traite cette information grâce au programme écrit par l'utilisateur
3. Une action est exécutée, comme l'allumage d'une LED, l'activation d'un moteur ou l'émission d'un signal sonore.

Ce processus peut être résumé par la chaîne suivante : entrée, traitement, sortie.

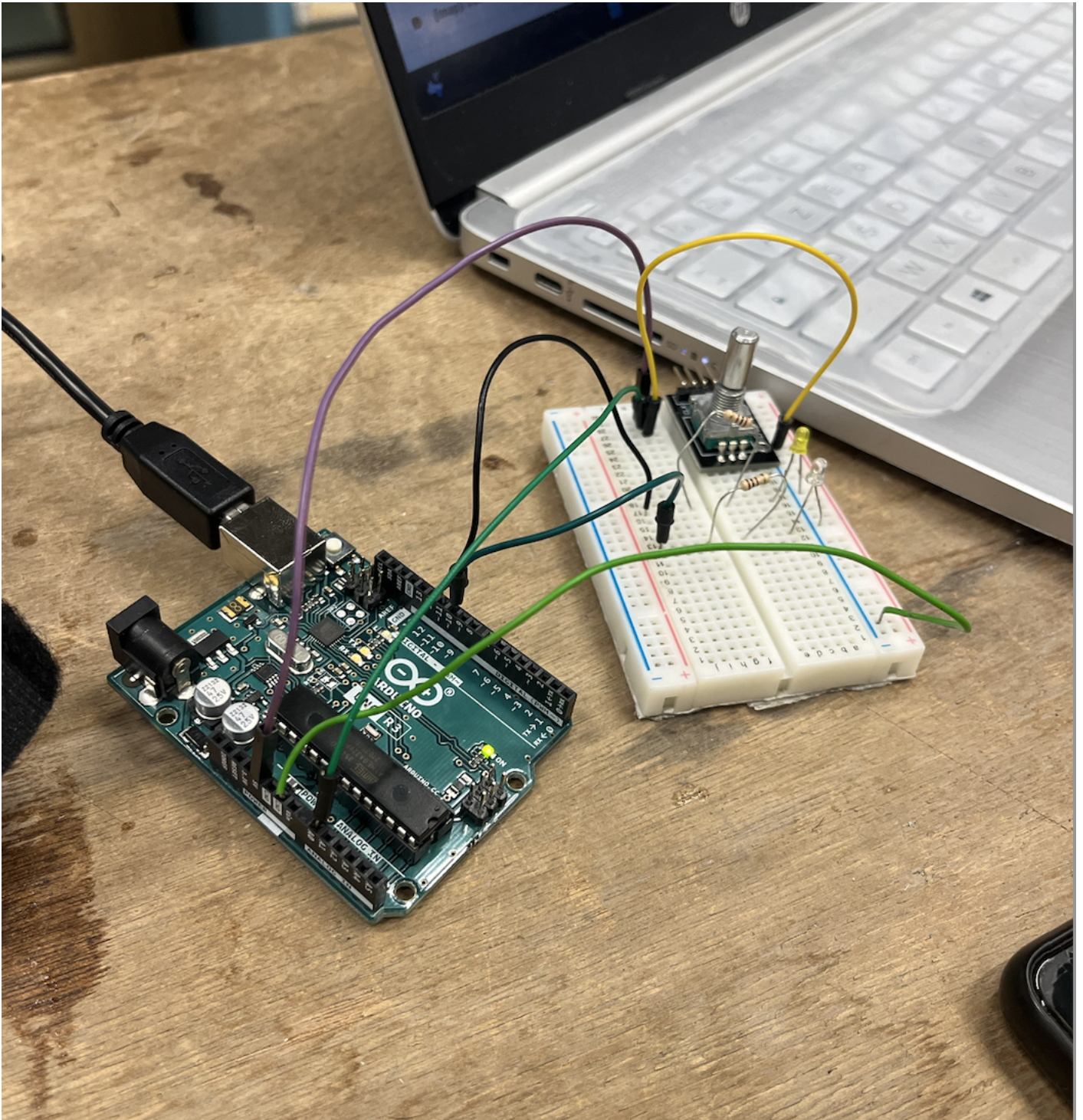
La programmation s'effectue à l'aide du logiciel Arduino IDE, dans lequel le code est écrit en un langage basé sur C/C++.

Un programme Arduino comporte toujours deux parties principales :

1. La fonction `setup()` s'exécute une seule fois au démarrage et sert à configurer les composants électroniques, par exemple définir une broche comme entrée ou sortie
2. La fonction `loop()` s'exécute ensuite en continu et permet de répéter les actions programmées.

Matériels :

- Carte Arduino Uno
- planche à pain
- Deux LED
- Deux résistances
- Fils Dupont
- Câble USB
- Ordinateur avec le logiciel Arduino IDE
- Potentiomètre



Code :

```
const int LED1 = 8;  
const int LED2 = 9;  
const int POT = A0;  
  
void setup() {  
  pinMode(LED1, OUTPUT);
```

```
pinMode(LED2, OUTPUT);
}

void loop() {
  int potValue = analogRead(POT); // 0 à 1023

  // Seuil simple : moitié de la course du potar
  if (potValue < 512) {
    // Allume LED1 seulement
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, LOW);
  } else {
    // Allume LED2 seulement
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, HIGH);
  }

  // Petit délai pour stabiliser la lecture
  delay(10);
}
```

Notre circuit n'a pas fonctionné, nous avons tenté de trouver des explications :

Pourquoi notre montage n'a pas fonctionné ?

En apparence, le montage semblait correct : nous avons bien utilisé deux résistances pour les LEDs, le programme Arduino se téléversait sans erreur, et l'Arduino était correctement alimenté. Cependant, le système ne fonctionnait pas comme prévu : tourner le potentiomètre n'allumait pas les LED de la manière attendue.

La principale raison de cet échec vient d'une confusion sur le composant utilisé. Nous pensions utiliser un potentiomètre analogique, alors que le composant branché était en réalité un encodeur rotatif. Contrairement à un potentiomètre classique, un encodeur rotatif ne fournit pas une tension analogique variable, mais des impulsions numériques. Or, notre code Arduino utilisait la fonction `analogRead(A0)`, qui ne peut fonctionner correctement qu'avec un vrai potentiomètre analogique. Ainsi, même si le code était syntaxiquement correct et bien téléversé sur la carte, il ne pouvait pas produire l'effet attendu, car le matériel ne correspondait pas au logiciel.

Ce que nous aurions dû améliorer :

En prenant du recul, plusieurs améliorations pourraient être suggérées :

- Mieux identifier les composants avant le montage : comprendre la différence entre un potentiomètre analogique et un encodeur rotatif aurait permis d'éviter cette erreur.
- Vérifier la cohérence entre le code et le matériel : un code utilisant analogRead() impose obligatoirement un capteur analogique.
- Tester étape par étape : par exemple, afficher la valeur du capteur dans le moniteur série pour vérifier si elle varie réellement.
- Soigner le câblage : même avec deux résistances, il est essentiel de vérifier que chaque LED est correctement reliée à une sortie numérique et à la masse commune (GND).
- Prendre le temps de documenter le montage avant de le brancher : cela correspond à une bonne pratique dans une démarche d'innovation.

Erreurs à éviter à l'avenir :

Cette expérience met en évidence plusieurs erreurs classiques à éviter :

- Penser que tous les boutons rotatifs sont des potentiomètres
- Utiliser un composant sans connaître son mode de fonctionnement (analogique vs numérique)
- Se fier uniquement au fait que le code se téléverse sans erreur
- Ne pas tester les valeurs des capteurs avant de construire la logique complète
- Négliger la correspondance entre matériel, logiciel et objectif fonctionnel

Conclusion :

Quand bien même le montage n'a pas fonctionné, cette erreur a été formatrice. Elle montre que, dans un processus d'innovation, les échecs techniques font partie de l'apprentissage. En tant qu'étudiantes en Management de l'innovation, cette expérience rappelle l'importance de la compréhension des contraintes techniques, même avec des connaissances limitées en électronique, et de la communication entre conception, expérimentation et validation.