

1) Objectifs du cours

À la fin de la séance, tu dois pouvoir :

- expliquer **à quoi sert** une carte Arduino ;
- identifier les éléments essentiels : **microcontrôleur, broches (pins), alimentation, USB** ;
- comprendre ce qu'est un **programme** (un “sketch”) Arduino ;
- comprendre la structure `setup()` / `loop()` ;
- réaliser un montage simple avec **3 LED** + résistances ;
- écrire et modifier un programme qui fait **clignoter 3 LED**.

2) Arduino, c'est quoi ?

2.1 La carte Arduino : un “petit ordinateur” spécialisé

Une carte Arduino est une **carte électronique programmable**.

Elle sert à **contrôler des objets** : allumer des LED, lire un bouton, mesurer avec un capteur, faire tourner un moteur, etc.

Ce n'est pas un PC : elle ne lance pas d'applications. Elle exécute **un seul programme**, en boucle, pour piloter des entrées/sorties.

2.2 Le microcontrôleur : le “cerveau”

Sur la carte, il y a un composant principal : le **microcontrôleur** (souvent ATmega328P sur Arduino Uno).

C'est lui qui :

- lit les **entrées** (bouton, capteur),
- commande les **sorties** (LED, moteur),
- exécute le programme stocké dans sa mémoire.

2.3 Les broches (pins) : les “portes”

Les broches servent à connecter le monde extérieur.

- **Broches digitales (D2, D3, ... D13) :**
Elles peuvent être **ON/OFF** (0 ou 1).
 - **HIGH** \approx 5V (allumé)
 - **LOW** \approx 0V (éteint)
- **GND** : la **masse** (le “0V”). Indispensable pour fermer le circuit.
- **5V** : alimentation 5 volts (utile pour certains montages).

2.4 Le câble USB : 2 rôles

Quand tu branches l'Arduino en USB :

1. tu **alimentes** la carte,
 2. tu **envoies** le programme dans le microcontrôleur.
-

3) Sécurité et bon sens en atelier

- Une LED **doit avoir une résistance** (souvent $220\ \Omega$ ou $330\ \Omega$), sinon elle peut griller.
 - Ne jamais relier directement **5V et GND** (court-circuit).
 - Toujours vérifier le câblage avant de brancher l'USB.
-

4) Matériel (par binôme)

- 1 carte Arduino Uno (ou compatible)
 - 1 câble USB
 - 1 breadboard (plaqué d'essai)
 - 3 LED (rouge/verte/jaune par exemple)
 - 3 résistances $220\ \Omega$ (ou $330\ \Omega$)
 - fils de connexion
-

5) Montage : faire clignoter 3 LED

5.1 Choix des broches

On va utiliser :

- LED1 sur **D8**
- LED2 sur **D9**
- LED3 sur **D10**

5.2 Sens d'une LED (important)

Une LED à 2 pattes :

- **patte longue = anode (+)**
- **patte courte = cathode (-)** (souvent côté méplat)

5.3 Câblage (logique simple)

Pour chaque LED :

- la **patte longue** va vers une **broche Arduino** (D8, D9, D10) **via une résistance**
- la **patte courte** va vers **GND**

Schéma logique (texte) pour une LED :

D8 → résistance → patte longue LED → patte courte LED → GND

Tu répètes pareil pour D9 et D10.

6) L'IDE Arduino : ce que les élèves doivent comprendre

6.1 Un programme Arduino s'appelle un “sketch”

On écrit du code (langage C/C++ simplifié par des fonctions Arduino).

6.2 Les 2 fonctions obligatoires

- `setup()` : exécutée **une seule fois** au démarrage.
- `loop()` : exécutée **en boucle**, sans arrêt.

7) Programme : Blink avec 3 LED (version simple)

Copie-colle ce code dans l'IDE Arduino :

```
// On donne un nom aux broches pour mieux comprendre
int led1 = 8;
int led2 = 9;
int led3 = 10;

void setup() {
    // On dit à Arduino : ces broches seront des SORTIES
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
    pinMode(led3, OUTPUT);
}

void loop() {
    // LED 1
    digitalWrite(led1, HIGH);    // allume
    delay(500);                // attend 0,5 seconde
    digitalWrite(led1, LOW);     // éteint
    delay(500);

    // LED 2
    digitalWrite(led2, HIGH);
    delay(500);
    digitalWrite(led2, LOW);
    delay(500);

    // LED 3
    digitalWrite(led3, HIGH);
    delay(500);
    digitalWrite(led3, LOW);
    delay(500);
}
```

8) Explication détaillée, ligne par ligne

8.1 Déclarer des variables (mémoriser des numéros de broches)

```
int led1 = 8;  
int led2 = 9;  
int led3 = 10;
```

- `int` signifie “nombre entier”.
- `led1` est une variable qui contient **8**.
- Ça rend le programme plus lisible : on écrit `led1` au lieu de 8.

8.2 `setup()` : préparer la carte

```
void setup() {  
    pinMode(led1, OUTPUT);  
    pinMode(led2, OUTPUT);  
    pinMode(led3, OUTPUT);  
}
```

- `void setup()` : “fonction de préparation”.
- `pinMode(broche, OUTPUT)` : indique que la broche va **envoyer** un signal (sortie).
- Sans `pinMode`, Arduino ne sait pas si la broche doit lire ou commander.

8.3 `loop()` : le scénario qui se répète sans fin

Arduino exécute `loop()` encore et encore, tant qu'il est alimenté.

Allumer une LED

```
digitalWrite(led1, HIGH);
```

- `digitalWrite` = “écriture digitale”
- `HIGH` = ON \approx 5V \rightarrow la LED s'allume (si elle est bien branchée)

Attendre

```
delay(500);
```

- `delay(500)` stoppe le programme pendant **500 millisecondes**
- 500 ms = 0,5 seconde

Éteindre la LED

```
digitalWrite(led1, LOW);
```

- `LOW` = OFF \approx 0V \rightarrow LED éteinte

Puis on recommence le même principe pour LED2 puis LED3.

9) Comprendre le “cycle” complet

Le déroulement est :

1. LED1 s'allume 0,5 s
 2. LED1 s'éteint 0,5 s
 3. LED2 s'allume 0,5 s
 4. LED2 s'éteint 0,5 s
 5. LED3 s'allume 0,5 s
 6. LED3 s'éteint 0,5 s
- Puis **retour au début** (loop recommence).
-

10) Mise en route dans l'IDE (procédure élèves)

1. Brancher la carte en USB
2. Ouvrir l'IDE Arduino
3. Choisir la carte : **Outils** → **Type de carte** → **Arduino Uno**
4. Choisir le port : **Outils** → **Port** → **(COM... ou /dev/tty...)**
5. Cliquer sur **Vérifier (✓)** : compile le programme
6. Cliquer sur **Téléverser (→)** : envoie le programme dans la carte
7. Observer les LED

11) Activités rapides (différenciation)

Niveau 1 (facile)

- Changer la vitesse : remplacer 500 par 200 puis 1000.
- Expliquer ce que fait HIGH / LOW.

Niveau 2 (intermédiaire)

Faire clignoter **les 3 LED en même temps** :

- allumer les 3
- attendre
- éteindre les 3
- attendre

Niveau 3 (plus avancé)

Créer un “chenillard” plus fluide :

- LED1 ON, LED2 OFF, LED3 OFF
- LED1 OFF, LED2 ON, LED3 OFF
- LED1 OFF, LED2 OFF, LED3 ON

12) Erreurs fréquentes et diagnostic

- **Une LED ne s'allume pas** : LED à l'envers (anode/cathode), fil mal placé, GND absent.
- **Tout ne marche pas** : mauvaise broche (D8 vs D7), ou `pinMode` manquant.
- **Téléversement impossible** : mauvais port, câble USB “charge seulement”, mauvaise carte sélectionnée.