

Arduino : Introduction à la programmation physique

Prérequis

Pour aborder ce cours sur Arduino, il est essentiel d'avoir des notions de base en électricité (tension, courant, résistance) et une familiarité avec les concepts fondamentaux de la programmation (variables, conditions, boucles). Ces notions sont généralement abordées en classe de troisième et en début de seconde. Ce cours s'inscrit dans le programme de physique-chimie de seconde, plus précisément dans le thème de l'énergie et de la matière, en introduisant une approche pratique de la modélisation et de la commande de systèmes physiques.

Chapitre 1 : Découverte d'Arduino Uno

1.1 Qu'est-ce qu'Arduino ?

Arduino est une plateforme électronique open-source basée sur une carte microcontrôleur facile à programmer. Elle permet de créer des projets interactifs, des robots, des objets connectés et bien plus encore. L'intérêt principal d'Arduino réside dans sa simplicité d'utilisation et sa grande communauté d'utilisateurs, offrant une multitude de ressources et de tutoriels.

1.2 La carte Arduino Uno : Composants et Fonctions

La carte Arduino Uno est le modèle le plus populaire. Elle est composée de plusieurs éléments essentiels :

- **Microcontrôleur ATmega328P** : Le cerveau de la carte, responsable de l'exécution du programme.
- **Connecteurs d'alimentation** : Permettent d'alimenter la carte via un adaptateur secteur, un câble USB ou une batterie.
- **Ports d'entrée/sortie (I/O)** : Des broches permettant de connecter des composants électroniques (LED, résistances, capteurs, moteurs, etc.). Ces broches peuvent être configurées en entrée (pour lire des informations) ou en sortie (pour contrôler des composants).
- **Port USB** : Permet de connecter la carte à un ordinateur pour programmer et communiquer avec elle.
- **Régulateur de tension** : Assure une tension stable pour le microcontrôleur.

1.3 L'environnement de développement Arduino (IDE)

L'IDE Arduino est un logiciel gratuit qui permet d'écrire, de compiler et de téléverser des programmes sur la carte Arduino Uno. Il est basé sur le langage de programmation C++, simplifié pour faciliter l'apprentissage. L'IDE propose une interface conviviale avec un éditeur de code, une console de

surveillance et des bibliothèques de fonctions pré-écrites.

Chapitre 2 : Premiers pas avec la programmation Arduino

2.1 Structure d'un programme Arduino

Un programme Arduino (appelé "sketch") est composé de deux fonctions principales :

- **`setup()`** : Cette fonction est exécutée une seule fois au démarrage de la carte. Elle est généralement utilisée pour initialiser les broches d'entrée/sortie et configurer les paramètres de communication.
- **`loop()`** : Cette fonction est exécutée en boucle infinie après l'exécution de ``setup()``. Elle contient le code principal du programme, qui est répété continuellement.

2.2 Allumer une LED : Exemple pratique

Voici un exemple simple pour allumer une LED connectée à la broche 13 de la carte Arduino Uno :

```
```cpp
```

```
void setup() {
 pinMode(13, OUTPUT); // Configure la broche 13 en sortie
}
```

```
void loop() {
```

```
 digitalWrite(13, HIGH); // Allume la LED
 delay(1000); // Attend 1 seconde
 digitalWrite(13, LOW); // Éteint la LED
 delay(1000); // Attend 1 seconde
}
```
```

Dans cet exemple, ``pinMode(13, OUTPUT)`` configure la broche 13 comme une sortie. ``digitalWrite(13, HIGH)`` envoie une tension de 5V à la broche 13, ce qui allume la LED. ``digitalWrite(13, LOW)`` envoie une tension de 0V, ce qui éteint la LED. ``delay(1000)`` met le programme en pause pendant 1000 millisecondes (1 seconde).

2.3 Variables, conditions et boucles

- **Variables** : Permettent de stocker des données (nombres, caractères, etc.).
- **Conditions (if/else)** : Permettent d'exécuter un bloc de code uniquement si une condition est vraie.
- **Boucles (for/while)** : Permettent de répéter un bloc de code plusieurs fois.

Chapitre 3 : Interaction avec l'environnement

3.1 Lecture d'une entrée analogique

Arduino peut lire des valeurs analogiques provenant de capteurs (photocellule, potentiomètre, etc.). La fonction `analogRead(pin)` permet de lire la tension sur une broche analogique (A0 à A5) et de la convertir en une valeur numérique entre 0 et 1023.

3.2 Contrôle d'un moteur DC

Un moteur DC peut être contrôlé en modifiant la tension qui lui est appliquée. La fonction `analogWrite(pin, value)` permet de générer un signal PWM (Pulse Width Modulation) sur une broche numérique, ce qui permet de contrôler la vitesse du moteur. La valeur `value` doit être comprise entre 0 et 255.

3.3 Exercice : Contrôle de la luminosité d'une LED avec un potentiomètre

Connectez un potentiomètre à une broche analogique (A0) et une LED à une broche numérique (9). Écrivez un programme qui ajuste la luminosité de la LED en fonction de la position du potentiomètre.

```
```cpp
```

```
const int potentiometrePin = A0;
const int ledPin = 9;
```

```
void setup() {
```

```
 pinMode(ledPin, OUTPUT);
}
```

```
void loop() {
```

```
 int valeurPotentiometre = analogRead(potentiometrePin);
 int luminosite = map(valeurPotentiometre, 0, 1023, 0, 255);
 analogWrite(ledPin, luminosite);
}
```
```

Dans cet exemple, `analogRead(potentiometrePin)` lit la valeur du potentiomètre. `map(valeurPotentiometre, 0, 1023, 0, 255)` convertit la valeur du potentiomètre (entre 0 et 1023) en une valeur de luminosité (entre 0 et 255). `analogWrite(ledPin, luminosite)` ajuste la luminosité de la LED en fonction de la valeur de luminosité.

Résumé

- **Arduino** est une plateforme électronique open-source pour la création de projets interactifs.
- **Microcontrôleur ATmega328P** est le cerveau de la carte Arduino Uno.
- **IDE Arduino** est le logiciel utilisé pour programmer la carte.
- ``setup()`` est la fonction exécutée une seule fois au démarrage.
- ``loop()`` est la fonction exécutée en boucle infinie.
- ``pinMode()`` configure une broche en entrée ou en sortie.
- ``digitalWrite()`` envoie une tension HIGH (5V) ou LOW (0V) à une broche numérique.
- ``analogRead()`` lit la tension sur une broche analogique.
- ``analogWrite()`` génère un signal PWM sur une broche numérique.
- **Variables** stockent des données.
- **Conditions (if/else)** exécutent un code en fonction d'une condition.
- **Boucles (for/while)** répètent un code plusieurs fois.
- La fonction ``map()`` permet de remapper une valeur d'une plage à une autre : ``map(valeur,fromLow ,fromHigh ,toLow ,toHigh)``.

From:
<https://wikiprof.fr/> - **wikiprof.fr**

Permanent link:
https://wikiprof.fr/doku.php?id=cours:lycee:generale:seconde_generale_et_technologique:physique_chimie:arduino

Last update: **2025/06/25 19:10**

