

Arduino

Ariel MEGNIGBETO

27 novembre 2023



# Table des matières

<b>1</b>	<b>Introduction à Arduino</b>	<b>7</b>
1.1	Qu'est-ce qu'Arduino? . . . . .	7
1.2	Quelles sont les principales caractéristiques d'une carte Arduino? . . . . .	7
1.3	Installation de l'environnement de développement Arduino . . . . .	9
1.4	Votre première expérience avec Arduino : clignoter une LED . . . . .	10
<b>2</b>	<b>Comprendre les composants de base</b>	<b>13</b>
<b>3</b>	<b>Capteurs de base</b>	<b>15</b>
<b>4</b>	<b>Actionneurs</b>	<b>17</b>
<b>5</b>	<b>Affichage</b>	<b>19</b>
<b>6</b>	<b>Communication sans fil</b>	<b>21</b>
<b>7</b>	<b>Capteurs avancés</b>	<b>23</b>
<b>8</b>	<b>RFID et sécurité</b>	<b>25</b>
<b>9</b>	<b>Contrôle à distance</b>	<b>27</b>
<b>10</b>	<b>Conception de circuits</b>	<b>29</b>
<b>11</b>	<b>Alimentation et régulation</b>	<b>31</b>
<b>12</b>	<b>Programmation avancée</b>	<b>33</b>
<b>13</b>	<b>Projets avancés</b>	<b>35</b>
<b>14</b>	<b>Dépannage et résolution de problèmes</b>	<b>37</b>
<b>15</b>	<b>Conclusion et ressources supplémentaires</b>	<b>39</b>



# Préface

Bienvenue dans le monde passionnant de l'Arduino, une plateforme qui a révolutionné la manière dont nous abordons l'électronique et la création de projets interactifs. Si vous tenez ce livre entre vos mains, il est fort probable que vous ayez déjà une lueur d'intérêt pour Arduino, que ce soit en tant que débutant curieux ou en tant qu'expert chevronné. Quel que soit votre niveau, vous êtes au bon endroit.

L'Arduino est bien plus qu'une simple carte de développement électronique. C'est une passerelle vers un univers de possibilités infinies, où votre imagination est la seule limite. Que vous souhaitiez créer des robots autonomes, des systèmes de contrôle domotique, des dispositifs de mesure sophistiqués ou simplement vous amuser à apprendre l'électronique, l'Arduino est là pour vous accompagner.

Ce livre a été conçu pour vous guider tout au long de votre voyage avec Arduino, en commençant par les concepts les plus fondamentaux et en progressant jusqu'aux projets avancés. Vous découvrirez comment utiliser chaque composant électronique, qu'il s'agisse de LEDs, de capteurs, de moteurs ou de modules de communication sans fil. Nous vous expliquerons en détail comment les connecter à votre Arduino, comment les programmer pour accomplir des tâches spécifiques et surtout, comment les combiner pour créer des projets uniques et fascinants.

Chaque chapitre de ce livre vous rapprochera davantage de la maîtrise de l'Arduino, en vous fournissant des instructions claires, des exemples de code, des schémas de connexion et des projets pratiques. Vous apprendrez à résoudre des problèmes, à expérimenter, à créer, à échouer (et à réussir) tout cela fait partie intégrante du processus d'apprentissage.

L'objectif de ce livre n'est pas seulement de vous donner des connaissances techniques, mais aussi de vous inspirer à explorer, à innover et à réaliser vos propres idées. Les projets proposés ne sont que le point de départ, car nous croyons fermement que la créativité n'a pas de limites. Alors, plongez dans ce livre, prenez votre Arduino, et commencez votre aventure vers la découverte de ce monde électronique fascinant.

Nous vous souhaitons une expérience d'apprentissage enrichissante et amusante avec Arduino. Que ce livre devienne votre guide fidèle, votre source d'inspiration et votre partenaire dans la réalisation de projets électroniques incroyables. Prêt à démarrer ? Lets Arduino !



# Chapitre 1

## Introduction à Arduino

L'électronique est une discipline fascinante qui nous entoure au quotidien. Des téléphones intelligents aux appareils ménagers, en passant par les gadgets électroniques, les circuits électroniques sont omniprésents dans notre vie. Cependant, pour beaucoup de gens, l'électronique peut sembler être un domaine mystérieux et complexe, réservé aux experts. C'est là qu'intervient Arduino.

### 1.1 Qu'est-ce qu'Arduino ?

Arduino est une plateforme open-source de développement électronique qui a été créée pour rendre l'électronique plus accessible à tous, quel que soit leur niveau de compétence. C'est un outil formidable pour les débutants qui souhaitent plonger dans le monde de l'électronique, mais c'est aussi un outil puissant pour les ingénieurs et les makers expérimentés qui veulent créer des prototypes rapidement.

Au cœur d'Arduino se trouve une carte de microcontrôleur qui peut être programmée pour effectuer une grande variété de tâches. Cette carte est souvent appelée une "carte Arduino". L'une des cartes Arduino les plus populaires est l'Arduino UNO R3, qui fait partie de votre kit.

### 1.2 Quelles sont les principales caractéristiques d'une carte Arduino ?

Les cartes Arduino ont plusieurs caractéristiques clés qui les distinguent :

1. Microcontrôleur : Chaque carte Arduino est équipée d'un microcontrôleur, généralement de la famille AVR d'Atmel (maintenant Microchip) ou d'autres fabricants comme ARM. Ce microcontrôleur est le cerveau de la carte Arduino et exécute les programmes que vous lui téléchargez.
2. Entrées/Sorties (E/S) numériques : Les cartes Arduino disposent de broches d'entrées/sorties numériques que vous pouvez utiliser pour lire des signaux numériques (0 ou 1) ou pour générer des signaux numériques. Ces

broches peuvent être configurées en tant qu'entrées ou sorties en fonction de vos besoins.

3. Entrées analogiques : En plus des E/S numériques, les cartes Arduino ont également des entrées analogiques qui vous permettent de lire des signaux analogiques, tels que des tensions variables. Cela les rend adaptées à la lecture de capteurs analogiques comme les capteurs de température, de lumière, ou de pression.
4. Connectivité : Selon le modèle, les cartes Arduino peuvent être équipées de différentes options de connectivité, notamment USB, Ethernet, Bluetooth, ou Wi-Fi, ce qui permet de les relier à d'autres dispositifs ou à l'Internet.
5. Programmabilité : Vous pouvez programmer une carte Arduino à l'aide de l'environnement de développement Arduino, qui utilise un langage de programmation basé sur C/C++. Arduino fournit également une bibliothèque de fonctions pré-écrites pour simplifier la programmation.
6. Open-source : Arduino est une plateforme open-source, ce qui signifie que les schémas, les plans de conception, et le logiciel sont disponibles gratuitement. Vous pouvez également créer et partager vos propres extensions matérielles (shields) ou logicielles.
7. Polyvalence : Les cartes Arduino sont polyvalentes et peuvent être utilisées pour une large gamme de projets, de l'automatisation domestique à la robotique, en passant par les systèmes de contrôle, les dispositifs interactifs, les œuvres d'art électroniques, etc.
8. Communauté active : Arduino dispose d'une grande communauté en ligne de passionnés, de développeurs et de makers. Cela signifie qu'il existe de nombreuses ressources, tutoriels, forums et projets disponibles pour vous aider à apprendre et à résoudre vos problèmes.

L'utilisation d'Arduino est également diversifiée et peut couvrir de nombreux domaines, notamment :

1. Apprentissage de l'électronique et de la programmation : Arduino est souvent utilisé dans les écoles et les ateliers pour enseigner les concepts de base de l'électronique et de la programmation. Il offre une plateforme pratique pour apprendre à coder et à créer des circuits électroniques.
2. Projets artistiques interactifs : Les artistes utilisent Arduino pour créer des œuvres d'art interactives qui répondent aux mouvements, à la lumière, au son, etc. Cela ouvre de nouvelles possibilités d'expression artistique.
3. Domotique : Arduino peut être utilisé pour automatiser et contrôler les appareils domestiques, tels que l'éclairage, le chauffage, la climatisation, les systèmes de sécurité, etc.
4. Robotique : Arduino est couramment utilisé pour construire des robots, des véhicules autonomes et d'autres systèmes autonomes. Il permet de contrôler les moteurs, de lire des capteurs et de prendre des décisions en temps réel.
5. IoT (Internet des objets) : Arduino peut être connecté à Internet pour créer des projets IoT. Vous pouvez surveiller et contrôler des dispositifs à



distance, collecter des données environnementales, créer des applications de suivi, etc.

6. Contrôle de machines et d'équipements industriels : Dans des applications industrielles, Arduino peut être utilisé pour contrôler des machines, collecter des données de capteurs, ou même automatiser des processus de production simples.
7. Expérimentation scientifique : Les chercheurs et les scientifiques utilisent Arduino pour créer des dispositifs de mesure et d'expérimentation dans divers domaines, tels que la biologie, la physique, la chimie, etc.
8. Projets personnels et ludiques : Enfin, Arduino est largement utilisé par les amateurs et les bricoleurs pour créer des projets personnels, des gadgets amusants, des jouets électroniques et bien plus encore.

## 1.3 Installation de l'environnement de développement Arduino

Pour commencer à utiliser Arduino, vous devrez installer le logiciel Arduino IDE (Integrated Development Environment) sur votre ordinateur. Ce logiciel est disponible gratuitement sur le site Web d'Arduino et est compatible avec Windows, Mac et Linux.

Rendez-vous sur le site officiel d'Arduino à l'adresse. Cliquez [ici](#)

Téléchargez la version de l'Arduino IDE compatible avec votre système d'exploitation (Windows, Mac ou Linux). Suivez les instructions pour le téléchargement.

- Installation de l'Arduino IDE :
  - Sur Windows : Exécutez le fichier d'installation téléchargé et suivez les instructions à l'écran pour installer l'IDE.
  - Sur Mac : Faites glisser l'application Arduino IDE vers le dossier Applications.
  - Sur Linux : Suivez les instructions spécifiques à votre distribution pour l'installation de logiciels.
- Lancement de l'Arduino IDE : Une fois l'IDE installé, lancez-le depuis le menu de votre système d'exploitation.
- Configuration de la carte Arduino :
  - Allez dans le menu "Outils" de l'IDE.
  - Sélectionnez "Type de carte" et choisissez "Arduino/Genuino Uno" ou le modèle de carte que vous utilisez.
  - Sélectionnez également le port série correspondant à votre carte Arduino (vous pouvez le trouver dans le gestionnaire de périphériques).
- Écriture et téléversement de votre premier programme : Vous êtes maintenant prêt à écrire votre premier programme Arduino dans l'IDE. Vous pouvez trouver des exemples de programmes sous "Fichier" > "Exemples". Une fois votre programme écrit, cliquez sur l'icône "Téléverser" (flèche vers la droite) dans l'IDE pour charger le programme sur votre carte Arduino.
- Vérification du fonctionnement : Votre programme devrait être téléchargé avec succès sur votre carte Arduino, et vous devriez voir les résultats sur la carte (par exemple, des LED clignotantes si c'est un programme

d'exemple). Vous êtes maintenant prêt à explorer et à développer davantage vos propres projets Arduino.

## 1.4 Votre première expérience avec Arduino : clignoter une LED

La meilleure façon d'apprendre Arduino est de commencer par un projet simple. Dans ce chapitre, nous allons vous guider à travers la création de votre premier projet Arduino : faire clignoter une LED. Vous apprendrez à écrire un code simple, à le téléverser sur votre carte Arduino et à voir les résultats en direct.

Voici un aperçu de ce que vous allez apprendre dans ce premier projet :

- La structure de base d'un programme Arduino.
- Comment définir des broches (pins) en tant qu'entrées ou sorties.
- Comment utiliser la fonction "digitalWrite" pour contrôler une LED.
- Comment téléverser votre programme sur la carte Arduino.

### Étape 1 : Matériel requis

- Une carte Arduino (par exemple, Arduino UNO R3).
- Une LED (Light Emitting Diode).
- Une résistance d'environ 220 ohms (pour protéger la LED).
- Deux fils de raccordement (jumpers).
- Un ordinateur avec l'Arduino IDE installé.

### Étape 2 : Connexion du matériel

images a ajoute

### Étape 3 : Écriture du code Arduino

Listing 1.1 – Code Arduino avec mise en forme LaTeX

```
1 //Definition de la broche de la LED
2 int ledPin = 13;
3
4
5 // Fonction setup() - s'execute une fois au
6 // d'emarrage
7 void setup() {
8     // Declare la broche de la LED comme une
9     // sortie
10    pinMode(ledPin, OUTPUT);
11 }
12
13 //Fonction loop() - s'execute en boucle
14 void loop() {
15     // Allume la LED
16     digitalWrite(ledPin, HIGH);
17     delay(1000); // Attend 1 seconde (1000
18                 // millisecondes)
19
20     // Eteint la LED
21     digitalWrite(ledPin, LOW);
```

#### 1.4. VOTRE PREMIÈRE EXPÉRIENCE AVEC ARDUINO : CLIGNOTER UNE LED11

```
19     delay(1000); // Attend 1 seconde
20 }
```

##### Explication du code :

- Nous commençons par définir la broche à laquelle est connectée la LED (13) en utilisant la variable `ledPin`.
- Dans la fonction `setup()`, nous déclarons `ledPin` comme une sortie en utilisant `pinMode(ledPin, OUTPUT)`. Cela indique à Arduino que nous allons contrôler cette broche pour allumer et éteindre la LED.
- Dans la fonction `loop()`, nous utilisons `digitalWrite(ledPin, HIGH)` pour allumer la LED, puis `delay(1000)` pour attendre une seconde. Ensuite, nous utilisons `digitalWrite(ledPin, LOW)` pour éteindre la LED et attendre à nouveau une seconde. Cette séquence crée un clignotement de la LED.

##### Étape 4 : Téléversement du programme

- Connectez votre carte Arduino à votre ordinateur à l'aide du câble USB.
- Ouvrez l'Arduino IDE.
- Sélectionnez le type de carte et le port série dans le menu "Outils", comme expliqué précédemment.
- Cliquez sur l'icône "Téléverser" (flèche vers la droite) pour télécharger le programme sur votre carte Arduino.

Après avoir suivi ces étapes, la LED devrait commencer à clignoter à un rythme d'une seconde d'allumage, une seconde d'extinction.

Félicitations ! Vous avez réussi à créer et à téléverser votre premier programme Arduino pour faire clignoter une LED. Ce projet simple est une excellente introduction à la programmation et à la manipulation de matériel avec Arduino. Vous pouvez maintenant explorer davantage en modifiant le code ou en ajoutant des composants pour créer des projets plus complexes.



## Chapitre 2

# Comprendre les composants de base



## Chapitre 3

# Capteurs de base





## Chapitre 4

# Actionneurs



## Chapitre 5

# Affichage



## Chapitre 6

# Communication sans fil



## Chapitre 7

# Capteurs avancés





## Chapitre 8

# RFID et sécurité



## Chapitre 9

# Contrôle à distance



## Chapitre 10

# Conception de circuits



## Chapitre 11

# Alimentation et régulation





## Chapitre 12

# Programmation avancée



## Chapitre 13

# Projets avancés



## Chapitre 14

# Dépannage et résolution de problèmes



## Chapitre 15

# Conclusion et ressources supplémentaires