

Massimo Banzi  
et Michael Shiloh

# Démarrez avec Arduino

4<sup>e</sup> édition

Traduit de l'anglais  
par Gérard Samblancat

**DUNOD**

Cet ouvrage est la traduction en langue française,  
par Dunod Éditeur, de l'ouvrage *Getting Started with Arduino* 4th edition  
de Massimo Banzi et Michael Shiloh

Authorized French translation of material from the English edition of

**Make: Getting Started with Arduino, 4th Edition**

ISBN 9781680456936 © 2022 Massimo Banzi and Michael Shiloh.

Original edition published by Make Community in 2022

This translation is published and sold by permission of O'Reilly Media, Inc.,  
which owns or controls all rights to sell the same.

Illustrations intérieures: Elisa Canducci et Judy Aime' Castro

Maquette intérieure: Maud Warg

Mise en pages: PCA

## À PROPOS DES AUTEURS

**Massimo Banzi** est le co-fondateur du projet Arduino. C'est un pionnier, et enseignant en Interaction Design et de l'électronique open source. En plus de son travail sur Arduino, il enseigne les techniques Cyber à l'université USI de Lugano (suisse), et l'Interaction Design au SUPSI de Lugano, et au CIID de Copenhague.

**Michael Shiloh** est professeur au California College of Arts, où il enseigne l'électronique, la robotique et la mécatronique. Sa formation d'ingénieur l'a d'abord amené à travailler sur divers projets avant de se découvrir une passion pour l'enseignement. Michael préfère utiliser ses compétences pour mener à bien les projets les plus créatifs et innovants. Il participe et anime souvent des conférences dans le monde entier. Il a commencé à travailler sur Arduino en 2013, ce qui lui a permis d'enseigner sa technique de conception en électronique open-source à une nouvelle audience.

© Dunod, Paris, 2011, 2013, 2015, 2023 pour la traduction française

11 rue Paul Bert, 92240 Malakoff

[www.dunod.com](http://www.dunod.com)

ISBN 978-2-10-084534-7

# TABLE DES MATIÈRES

À PROPOS DES AUTEURS	II
AVANT-PROPOS DE LA 4 <sup>E</sup> ÉDITION	7
AVANT-PROPOS DE LA 2 <sup>E</sup> ÉDITION	9
INTRODUCTION	13
<b>/&gt;1 LA MÉTHODE ARDUINO</b>	<b>17</b>
La fabrication de prototypes	17
Le bricolage	18
On aime la récupération	18
Transformer les jouets pour enfants	19
L'entraide	20
<b>/&gt;2 LA PLATEFORME ARDUINO</b>	<b>21</b>
La carte Arduino	21
Le logiciel IDE	23
Installer la plate-forme arduino sur votre ordinateur	24
Installation de L'IDE: MacOS	24
Installation des drivers: MacOS	24
Identification du port série: MacOS	25
Installation de l'ide: Windows	25

Installation des pilotes: Windows	26
Identification du port: Windows	26
Installation de l'IDE : Linux	27
Configuration des pilotes: Linux	27
Identification du port: Linux	27
<b>/&gt;3 VRAIMENT COMMENCER AVEC ARDUINO</b>	<b>29</b>
Anatomie d'un appareil interactif	29
Capteurs et actionneurs	30
Faire clignoter une DEL	30
Passez-moi le parmesan	33
Arduino n'est pas fait pour les lâcheurs	34
Les vrais bricoleurs écrivent des commentaires	34
Le programme, pas à pas	35
Qu'allons-nous construire ?	37
Qu'est-ce que l'électricité ?	38
Utiliser un bouton-poussoir pour contrôler une DEL	40
Comment ça marche ?	42
Un circuit, un millier de comportements possibles	43
<b>/&gt;4 TECHNIQUES AVANCÉES D'ENTRÉES SORTIES</b>	<b>47</b>
Essayer d'autres capteurs on/off	47
Commutateurs faits maison (DIY)	49
Contrôler la lumière par PWM	49
Utiliser un capteur de lumière au lieu du bouton	55
Une entrée analogique	55
Essayer d'autres capteurs analogiques	58
Communication série	59
Piloter des charges plus puissantes (moteurs, lampes...)	60
Des capteurs plus complexes	61
L'alphabet Arduino	62
<b>/&gt;5 FABRIQUER UNE LAMPE ARDUINO</b>	<b>63</b>
Préparation	64
Le codage	65
Réaliser le montage	70
L'assemblage final	72

<b>/&gt;6</b>	<b>LE CLOUD ARDUINO</b>	<b>73</b>
	L'IDE du Cloud Arduino	73
	Le cloud IDE	73
	Le project hub	74
	Le Cloud IoT	75
	Caractéristiques du cloud IoT	76
	Arduino Cloud plans	77
<b>/&gt;7</b>	<b>UN ARROSAGE AUTOMATIQUE</b>	<b>79</b>
	Planifier	81
	Tester l'horloge (RTC)	83
	Tester les relais	88
	Le schéma de principe électronique	90
	Tester les capteurs de température et d'humidité	98
	Écrire le code	101
	Régler les heures de début et de fin	101
	Vérifier si c'est l'heure d'allumer ou d'éteindre	106
	Détecter la pluie	109
	Tout rassembler	110
	Assembler le montage	116
	Le Proto Shield	120
	Insérer votre projet sur la carte Proto Shield	121
	Souder le projet sur la carte Proto Shield	125
	Test de la carte Proto Shield assemblée	136
	Intégrer le projet dans un boîtier	137
	Test du système d'arrosage terminé	139
	Ce que vous pourriez encore faire vous-même	140
	Liste des composants de notre arrosage automatique	141
<b>/&gt;8</b>	<b>LA GAMME ARDUINO ARM</b>	<b>143</b>
	Quelles différences entre AVR et ARM ?	143
	Quelle différence avec 32 bits ?	144
	Quelles différences entre un microcontrôleur et un microprocesseur ?	144
	Qui est le meilleur : AVR ou ARM ?	145
	Les différentes cartes Arduino ARM	145
	Caractéristiques spécifiques	146
	Tension d'alimentation	146
	Courant d'utilisation	147

Convertisseur digital vers analogique	147
Le mode hôte USB	147
Les cartes Nano et MKR	148
<b>/&gt;9 DIALOGUER SUR INTERNET AVEC ARM: UN «CHECK» CONNECTÉ</b>	<b>149</b>
Le protocole MQTT (Message Queueing Telemetry Transfer Protocol)	150
Le circuit	150
Le broker MQTT de chez shiftr.io	154
Le code Arduino	154
La page Web	157
<b>/&gt;10 DÉPANNAGE</b>	<b>163</b>
La compréhension	163
Simplification et division	164
Exclusion et certitude	164
Tester la carte Arduino	164
Tester le montage sur plaque d'essai	165
Isoler les problèmes	166
Problèmes d'installation des drivers sous Windows	167
Problèmes avec l'IDE sous Windows	167
Identifier le bon port COM Arduino sur Windows	168
Autres techniques de débogage	169
Comment obtenir de l'aide en ligne	170
ANNEXES	173
A. La plaque d'essai	173
B. Lire les résistances et les condensateurs	174
C. Référence du langage Arduino	176
D. La famille Arduino	186
E. Les clones Arduino, dérivés compatibles et contrefaçons	188
F. Lire les schémas	189
INDEX	191



# AVANT-PROPOS DE LA 4<sup>E</sup> ÉDITION

Massimo et Michael ont été ravis de rédiger cette quatrième édition de leur ouvrage, pour tenir compte des dernières avancées en matière de conception électronique.

Cette édition a été augmentée de deux nouvelles parties: le chapitre 8 qui traite des puissantes cartes ARM Arduino, et le chapitre 9 qui décrit le système de développement Arduino en ligne, au travers d'un nouveau projet: le « Check Internet ».

En dehors de ces deux nouveaux chapitres, de nombreuses mises à jour ont été effectuées:

- La quatrième édition est écrite pour la version 2.0 de l'IDE Arduino.
- L'installation de l'IDE est plus facile, et les instructions spécifiques à Linux sont présentes.
- Les annexes donnent un aperçu de toutes les familles de cartes Arduino.
- Le chapitre sur la carte Leonardo a été remplacé par un aperçu du Cloud Arduino, d'IoT et du Hub Project.
- Pour ne plus gêner personne, certaines notations ont été modifiées:
  - La notation des signaux SPI est conforme à la norme du « Open Source Hardware » (<https://www.oshwa.org/a-resolution-to-redefine-spi-signal-names/>).
  - Les connexions se font sur barrettes sécables tulipes.

Certaines illustrations ont également changé, et de nouvelles ont été ajoutées. Les auteurs tiennent à remercier Elisa Canducci pour ses illustrations dans les premières et secondes éditions; ainsi que Judy Aime Castro pour ses dessins de la troisième édition.

Michael Shiloh





# AVANT-PROPOS DE LA 2<sup>E</sup> ÉDITION

C'est un défi passionnant que l'on m'a confié il y a quelques années : enseigner à des concepteurs le strict minimum en électronique, pour qu'ils puissent rendre les maquettes de leurs futures créations plus interactives.

Après avoir commencé à enseigner d'une manière tout à fait scolaire, j'ai réalisé plus tard que cela ne fonctionnait pas aussi bien que je l'avais espéré. J'ai alors essayé de m'imaginer assis en cours, en train d'écouter nombre de théories ennuyeuses et sans applications pratiques immédiates.

En réalité, lorsque j'étais moi-même à l'école, je connaissais déjà assez bien l'électronique avec ma propre expérience pratique et empirique ; mais très peu par ses aspects théoriques.

J'ai alors essayé de reproduire la méthode qui m'avait véritablement permis d'apprendre l'électronique :

- j'étudiais chaque composant qu'il m'était possible de tenir en main et apprenais progressivement ce qu'il était ;
- j'essayais parfois de modifier des connexions dans les montages pour voir ce qui se passait. Cela donnait parfois un petit nuage de fumée... ;
- je construisais des kits vendus dans les magazines et les modifiais à loisir en leur ajoutant de vieux composants, pour obtenir de nouvelles choses.

Comme un enfant, j'étais toujours fasciné de découvrir le fonctionnement des choses.

Cette passion a grandi au point que je démontais tous les vieux appareils de la maison, et ceux que l'on m'amenait. Les objets les plus intéressants furent, entre autres, une machine à laver, un très vieil ordinateur venant d'une compagnie d'assurances accompagné d'une énorme imprimante, de nombreux modules et d'un lecteur de cartes magnétiques...

Après tout cela, je savais ce qu'étaient les composants électroniques, et à peu près à quoi ils servaient. Je disposais en plus de nombreux magazines d'électronique que mon père avait achetés au début des années 1970. Je passais des heures à les lire, et à regarder les schémas, sans y comprendre grand-chose.

Le fait de relire continuellement ces articles, ainsi que la connaissance que j'avais acquise avec tous mes démontages, ont commencé à me mettre sur la voie.

Mais je franchis encore un grand pas quand mon père m'offrit à Noël une boîte de construction électronique. Chaque composant y était présenté sous la forme d'un cube en matière plastique aimanté, et pouvait s'assembler aux autres composants, en faisant autant de connexions. Je ne savais pas alors que ce jeu était de conception allemande et avait été conçu par Dieter Rams dans les années 1960.

Avec ce nouvel outil, je pouvais très rapidement assembler les composants, et voir ce qui se passait. Je construisais des radios, des amplificateurs, des montages qui faisaient toutes sortes de sons, des détecteurs de pluie, des petits robots...

J'ai longtemps cherché le mot qui résume le mieux cette façon de travailler : sans véritable schéma, en partant d'une idée, et en arrivant au final à un résultat inattendu. C'est probablement ce que l'on pourrait appeler du « bricolage ».

La meilleure définition que je connaisse de ce bricolage vient d'une exposition à l'Exploratorium de San Francisco :

Le bricolage est ce qui arrive quand on essaie de fabriquer quelque chose que l'on ne connaît pas suffisamment, guidé par le caprice, l'imagination et la curiosité. Quand on bricole, il n'y a pas d'instructions, mais il n'y a pas non plus d'erreurs, pas de bonne ou mauvaise méthode de faire les choses. Il faut comprendre comment les choses fonctionnent et pouvoir les réassembler.

Bidules, machines, objets mal assortis mais fonctionnant parfaitement, voilà ce qu'est le « bricolage ». C'est un processus qui mélange le jeu et l'investigation. (cf. [www.exploratorium.edu/tinkering](http://www.exploratorium.edu/tinkering))

De mes débuts, je savais combien il fallait de connaissances pour être capable de concevoir ses propres montages à partir de composants de base.

Un autre déclic m'est venu à l'âge de 14 ans quand, en voyage à Londres avec mes parents, j'ai visité le Musée des sciences. Une section consacrée aux ordinateurs venait juste d'ouvrir. J'y ai découvert les bases du langage binaire, ainsi que la programmation.

J'ai alors réalisé que, pour de nombreuses applications, les ingénieurs n'avaient plus à utiliser les composants électroniques de base, mais pouvaient se contenter de rendre leurs montages intelligents en utilisant des microprocesseurs. Un logiciel pouvait remplacer des heures de conception électronique et rendre la mise au point bien plus rapide.

À mon retour, j'ai tout de suite commencé à économiser pour pouvoir m'acheter un ordinateur et apprendre à le programmer.

Après ça, mon premier et plus intéressant projet a été d'utiliser mon tout nouvel ordinateur ZX81 pour contrôler un robot à souder. Cela n'a peut-être pas l'air très excitant, mais c'était alors un grand défi, parce que j'en avais besoin, et que je venais juste d'apprendre à programmer. Mais il était déjà clair que l'écriture du programme prendrait moins de temps que la mise au point d'un circuit électronique complexe.

Une vingtaine d'années plus tard, je crois que ce sont toutes ces expériences qui m'ont permis d'enseigner à des gens dépourvus de connaissances mathématiques, et de leur transmettre le même enthousiasme et la capacité à « bricoler » que j'avais étant jeune et que j'ai toujours gardée.

Massimo Banzi

## **/ REMERCIEMENTS DE MASSIMO BANZI**

Je dédie cet ouvrage à Ombretta.

## **/ REMERCIEMENTS DE MICHAEL SHILOH**

Je dédie cet ouvrage à mon frère et à mes parents.

Je remercie tout d'abord Massimo pour m'avoir invité à travailler sur cette troisième édition, et pour m'avoir permis de rejoindre l'équipe Arduino. Cela a été un vrai privilège et une grande joie de travailler à ce projet.

Merci à Brian Jepson pour ses conseils, ses encouragements et son aide. À Frank Teng de m'avoir guidé. À Kim Cofer et Nicole Shelby qui ont fait un superbe travail d'édition.

À ma fille Yasmine pour ses encouragements incessants et sa haute opinion de moi. Je n'aurais rien fait sans son soutien.

Un dernier merci, mais pas le moindre à ma partenaire Judy Aime Castro, qui a passé des heures interminables à transformer mes croquis en superbes illustrations. Pour ses avis sur le livre et sa patience. Je n'aurais pas non plus fait cela sans elle.

## **/ UTILISATION DES PROGRAMMES D'EXEMPLES**

Ce livre est destiné à vous aider à réaliser vos projets. Vous pouvez réutiliser les programmes fournis dans le livre dans vos propres applications et documentations. Vous n'avez donc pas besoin de nous contacter pour une quelconque autorisation, à moins de reprendre une partie importante de code. Par exemple: écrire un programme qui utilise plusieurs lignes venant de ce livre ne demande pas d'autorisations. Vendre ou distribuer un CD-ROM d'exemples de ce livre nécessite une autorisation.

Répondre à une question en citant ce livre et un programme d'exemple ne demande pas d'autorisation. Incorporer une grosse partie d'un programme issu de ce livre dans vos produits ou publications nécessite une autorisation.

Toute référence à cet ouvrage sera appréciée, mais n'est pas obligatoire. Une référence comprend le titre, le nom des auteurs, de l'éditeur et le numéro ISBN de cet ouvrage. Par exemple: «*Démarrez avec Arduino, quatrième version, par Massimo Banzi et Michael Shiloh (Make Community LLC). Copyright 2022 Massimo Banzi et Michael Shiloh, 978-1-6804-5693-6*».

Si vous avez besoin de plus d'informations, n'hésitez pas à nous écrire à [books@make.co](mailto:books@make.co).

**En complément de cet ouvrage, téléchargez le code source  
des exemples, à l'adresse :**

<https://www.dunod.com/eau/9782100845347>



# INTRODUCTION

Pour commencer, définissons ce qu'on entend par Arduino, ce terme correspond avant tout à un système de développement open source de carte électronique permettant de créer des objets autonomes et interactifs. Le système Arduino a d'abord été conçu pour des artistes et concepteurs qui avaient besoin de modules électroniques, sans pour autant être des spécialistes en électronique.

Les cartes et logiciels Arduino sont entièrement *open source*. C'est-à-dire que les utilisateurs Arduino forment une communauté qui accepte de partager généreusement ses connaissances. Cela peut être très utile pour les débutants, car il est souvent possible de trouver de l'aide près de chez soi, ou quasiment toujours en ligne, et sur tous les sujets. Les exemples de projets complets ne sont pas juste montrés en photo, mais ils présentent aussi des instructions de fabrication détaillées.

La plate-forme de développement (IDE) peut être téléchargée gratuitement sur le site [www.arduino.cc](http://www.arduino.cc).

Le logiciel de développement IDE est basé sur le langage Processing (<http://www.processing.org>), qui a été lui-même créé pour aider les artistes à utiliser les ordinateurs pour leurs créations, sans avoir à devenir d'abord informaticiens.

Il fonctionne aussi bien sur Windows, Macintosh que Linux.

Les cartes Arduino sont d'un prix très abordable (environ 20 €) et assez tolérantes aux erreurs de débutants. Et même si le composant principal de la carte est endommagé, il pourra être changé pour quelques euros.

Le projet Arduino a été créé dans un environnement pédagogique où il est devenu très populaire. La communauté *open source*, qui partage ses projets et son savoir, partage aussi ses méthodes d'apprentissage.

Il existe aussi une *mailing list* (<http://bit.ly/1vKh0wb>) qui facilitera certainement les échanges entre tous les enseignants qui utilisent Arduino.

Comme les cartes et les logiciels Arduino sont libres de droits, vous pouvez télécharger toutes sortes de plans pour construire vos propres cartes, ou simplement comprendre comment cela marche.

## / PUBLIC CONCERNÉ

Ce livre a été écrit pour des débutants, des gens qui souhaitent apprendre à fabriquer des montages électroniques, ou faire de la programmation sans avoir forcément un gros bagage technique. La façon dont il explique les choses pourrait rendre fou plus d'un ingénieur. En lisant les premiers chapitres de la première édition, certains ont même parlé de ratés. Parlons-en car c'est un point délicat : être bon dans une technique, et être bon pour expliquer cette même technique, sont deux choses différentes.

Ce livre n'est pas à proprement parler un livre de cours d'électronique ou de programmation, mais en le lisant, vous y apprendrez effectivement des choses sur ces deux domaines.

Lorsque le système Arduino a commencé à être connu, j'ai réalisé comment les concepteurs et les autres bricoleurs de tout horizon pouvaient l'utiliser pour créer toutes sortes d'appareils aussi beaux qu'originaux. J'ai réalisé que chaque utilisateur peut être tout à la fois un concepteur et un artiste à sa façon.

Massimo.

### NOTE

Arduino s'appuie également sur le travail de thèse accompli par Hernando Barragan sur la plateforme Wiring, alors qu'il étudiait sous les responsabilités de Casey Reas et de moi-même (Massimo) au IDII Ivrea (Interaction Design Institute).

## / QU'EST-CE QUE L'INTERACTION DESIGN ?

Arduino a été créé au départ pour enseigner l'Interaction Design, qui est une méthode de conception basée sur la fabrication de prototypes. Il existe de nombreuses définitions de cette méthode mais voici celle que nous préférons : « L'Interaction design c'est la conception des interactions ».

Aujourd'hui, cette méthode est utilisée pour améliorer les relations homme-machine. C'est un bon moyen de réaliser de belles – et parfois controversées – expériences entre les humains et la technologie. L'Interaction Design est un processus de création itératif, où le résultat devient progressivement de plus en plus fidèle aux attentes.

Le champ d'action spécifique de cette méthode qui est utilisée dans Arduino est ce que l'on appelle le *Physical Computing*.

## / QU'EST-CE QUE LE PHYSICAL COMPUTING ?

C'est une démarche créative qui permet aux concepteurs et artistes d'utiliser l'électronique pour créer des appareils innovants. Elle vise en particulier à créer des appareils pouvant communiquer avec nous, grâce à des capteurs et à des actionneurs, dont le comportement est régi par un logiciel exécuté par un microcontrôleur (une sorte de mini-ordinateur complet sur une simple puce).

Dans le passé, pour faire de l'électronique, il fallait toujours avoir à faire à des ingénieurs, seuls capables de concevoir des schémas composant par composant. Cette difficulté a rebuté nombre d'esprits créatifs. La plupart des outils étaient alors conçus pour des ingénieurs ou des personnes au niveau de connaissances comparable.

Mais ces dernières années les microcontrôleurs sont devenus bien meilleur marché et faciles à utiliser. Les ordinateurs sont aussi bien plus puissants et permettent d'utiliser facilement des outils de développement.

Le progrès que nous avons fait avec Arduino est d'avoir rendu ces outils accessibles, au point qu'un débutant puisse créer une application après seulement deux ou trois jours de pratique.

Avec Arduino, tout créateur peut apprendre très rapidement les bases de l'électronique et du fonctionnement de capteurs en tout genre.

