

Christian Tavernier

Arduino

Maîtrisez sa programmation
et ses cartes d'interface (*shields*)

2^e édition

DUNOD

Maquette de couverture : APEX

Illustrations intérieures : Alain et Ursula Bouteville-Sanders

Illustrations couverture : © Oleg - stock.adobe.com

© Dunod, Paris, 2011, 2014,
2019 pour la nouvelle présentation
11 rue Paul Bert, 92240 Malakoff
www.dunod.com
ISBN 978-2-10-080732-1

Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5, 2° et 3° a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

TABLE DES MATIÈRES

Avant-propos	XI
1 • Qu'est-ce que l'Arduino ?	1
1.1 Qu'est-ce qu'un microcontrôleur ?	1
1.1.1 L'alimentation	2
1.1.2 L'horloge	2
1.1.3 Le circuit de reset	3
1.1.4 La programmation	3
1.2 Les différentes versions d'Arduino	4
1.3 L'Arduino Uno	5
1.3.1 Dimensions et organisation	5
1.3.2 Alimentation	6
1.3.3 Horloge	6
1.3.4 Reset	6
1.3.5 Les mémoires	7
1.3.6 Les entrées/sorties	8
1.3.7 Différences entre Arduino Uno Révision 1, 2 et 3	10
1.3.8 Schémas des Arduino Uno	10
1.4 L'Arduino Mega 2560	11
1.4.1 Les mémoires	12
1.4.2 Les entrées/sorties	12
1.4.3 Schéma de l'Arduino 2560	13
1.5 Langage machine et langage évolué	14
1.5.1 À l'origine : le langage machine	14
1.5.2 Les langages évolués	15
1.5.3 L'environnement de développement de l'Arduino	16
1.5.4 Le langage évolué de l'Arduino	18
1.6 L'environnement de développement	18
1.6.1 Installation de l'environnement de développement	19
1.6.2 Première configuration de l'environnement de développement	20

1.6.3	Utilisation de l'environnement de développement	21
1.6.4	Un premier programme juste pour « le fun »	23

2 • Environnement de développement et langage de programmation	25
2.1 Organisation générale d'un programme	26
2.1.1 Instructions et commentaires	26
2.1.2 Inclusions et définitions	27
2.1.3 Structure d'un programme	28
2.2 Données, variables et constantes	29
2.2.1 Les données numériques	30
2.2.2 Les données logiques et diverses	31
2.2.3 Les tableaux ou données indexées	32
2.2.4 Les données caractères et chaînes de caractères	32
2.2.5 Les conversions de types	35
2.2.6 Visibilité des variables	35
2.2.7 Gestion de la taille des variables	37
2.2.8 Constantes particulières	37
2.3 Fonctions arithmétiques et mathématiques	39
2.3.1 Opérateurs arithmétiques	39
2.3.2 Opérateurs arithmétiques composés	40
2.3.3 Fonctions mathématiques	41
2.3.4 Fonctions trigonométriques	42
2.4 Opérateurs de comparaison et opérateurs logiques	42
2.4.1 Opérateurs de comparaison	43
2.4.2 Opérateurs logiques booléens	43
2.4.3 Opérateurs logiques au niveau bit	44
2.5 Les structures de contrôle	46
2.5.1 Les prises de décision	46
2.5.2 Les boucles	50
2.5.3 Les sauts	51
2.6 Gestion du temps et des entrées/sorties	53
2.6.1 Gestion du temps	53
2.6.2 Entrées/sorties numériques	55
2.6.3 Entrées analogiques	57
2.6.4 Sorties analogiques	58
2.7 Entrées/sorties particulières	60
2.8 Fonctions diverses	62
2.8.1 Génération de nombres aléatoires	62

2.8.2	Manipulation de bits	63
2.8.3	Gestion des interruptions	65
2.9	Gestion du port série	67
3	Les cartes d'interface ou shields de base	71
3.1	Les shields prototypes	71
3.1.1	Le minimum vital	72
3.1.2	Les éléments optionnels	73
3.1.3	Bonnes et mauvaises cartes prototypes	73
3.1.4	Un shield qui n'en est pas un	76
3.2	Un shield idéal pour la formation	77
3.3	Un shield vide mais fort utile	79
3.4	Un shield pour y voir clair	81
4	Entrée/sorties numériques parallèles : poussoirs, claviers, LED, relais et afficheurs	85
4.1	Les entrées numériques parallèles	86
4.1.1	Poussoirs, commutateurs, capteurs de type tout ou rien	86
4.1.2	Comment vaincre les rebondissements ?	88
4.1.3	Claviers en matrices	90
4.1.4	Plusieurs touches avec une seule entrée	94
4.1.5	Entrées haute tension et opto-isolées	97
4.2	Les sorties numériques parallèles	98
4.2.1	Commandes de LED et de relais	99
4.2.2	Commande directe de charges alimentées en continu	102
4.2.3	Commande directe de charges reliées au secteur	103
4.2.4	Commande directe d'afficheurs à LED	104
4.2.5	Commande directe d'afficheurs à LED multiplexés	106
4.2.6	Commande d'afficheurs à LED avec des circuits spécialisés	109
4.2.7	Utilisation d'afficheurs à cristaux liquides (LCD)	109
5	Entrées/sorties analogiques : capteurs, potentiomètres, gradateurs et moteurs	117
5.1	Les entrées analogiques	117
5.1.1	Utilisation de capteurs analogiques résistifs	118
5.1.2	Où il est question d'hystérésis	120
5.1.3	Utilisation de capteurs délivrant une tension analogique	121
5.1.4	Utilisation de capteurs délivrant « n'importe quoi »	124

5.2	Les sorties analogiques PWM	126
5.2.1	Gradateur de tension ou de puissance PWM	126
5.2.2	Tension continue à partir d'un signal PWM	129
5.3	La commande des moteurs	130
5.3.1	Les servos de radiocommande	130
5.3.2	Les moteurs pas à pas	137
5.3.3	Les moteurs à courant continu	142
6 •	Les interfaces séries synchrones et asynchrones, I2C, SPI, bus « un fil »	147
6.1	Les liaisons séries asynchrones	147
6.1.1	Interface matérielle ou logicielle	147
6.1.2	L'interface RS 232	150
6.1.3	Des signaux de contrôle parfois inutiles	153
6.2	Les liaisons séries synchrones	154
6.2.1	Le registre à décalage	154
6.2.2	Commande de LED ou d'afficheurs avec un registre à décalage	156
6.3	Le bus I2C	158
6.3.1	Généralités	159
6.3.2	Protocole du bus I2C	159
6.3.3	La bibliothèque Wire de l'Arduino	162
6.3.4	Exemple d'utilisation du bus I2C	165
6.4	L'interface ou bus SPI	170
6.4.1	Principe d'une liaison de type SPI	170
6.4.2	La bibliothèque SPI de l'Arduino	172
6.4.3	Exemple d'utilisation d'une liaison SPI	173
6.5	Le bus « un fil » de Dallas (Maxim)	177
6.5.1	Principe du bus un fil	177
6.5.2	La bibliothèque OneWire de Jim Studt	179
6.5.3	Exemple d'utilisation du bus un fil	182
7 •	Pour aller plus loin	185
7.1	Les shields évolués	185
7.1.1	Programmation des shields évolués	186
7.1.2	Reliez votre Arduino à Internet	186
7.2	Les autres Arduino originaux	192
7.2.1	L'Arduino Mini	192
7.2.2	L'Arduino Nano	194
7.2.3	L'Arduino Fio	195
7.2.4	L'Arduino Pro	196

7.2.5	L'Arduino Pro Mini	198
7.2.6	L'Arduino LilyPad	199
7.3	Les « Arduino like »	200
7.3.1	Le Eleven, anciennement TwentyTen de Freetronics	201
7.3.2	Le Freeduino	202
7.3.3	Le Boarduino	203
7.4	Les « gros » Arduino	204
7.4.1	L'Arduino Leonardo	204
7.4.2	Les Arduino Due, Yun et Galileo	206
Webographie		207
Index		209

L'Arduino, ou plutôt les Arduino devrions-nous écrire car ils ont fait des petits, connaissent aujourd'hui un succès planétaire, pourtant, il ne s'agit au départ que d'une simple carte électronique de petit format supportant un microcontrôleur, entouré du minimum de composants nécessaires à son fonctionnement de base. Rien de très original jusque-là nous direz-vous, et surtout rien qui puisse expliquer ce succès.

En fait, cette carte supporte également, sur sa périphérie, une rangée de connecteurs dans lesquels peuvent venir s'enficher directement des cartes d'interface, appelées les « shields ». Ces cartes d'interface sont aujourd'hui très nombreuses et couvrent quasiment tous les besoins des applications courantes.

Mais ce n'est pas tout. L'Arduino est également équipé d'un connecteur USB permettant de le raccorder à un micro-ordinateur, PC ou Mac peu importe, utilisé pour écrire le programme destiné à le piloter, mais également pour programmer le microcontrôleur qui équipe l'Arduino sans aucune intervention matérielle de votre part.

Cerise sur le gâteau, le projet Arduino dans son ensemble, c'est-à-dire les cartes mais aussi les logiciels associés, sont « open source » c'est-à-dire sont publics, libres de droits et donc utilisables par tous sans devoir verser de quelconques royalties ou droits d'auteur.

En d'autres termes, l'Arduino permet à tout un chacun de réaliser à moindre coût des applications à base de microcontrôleur quasiment sans nécessiter de connaissances en électronique, sans avoir besoin de sortir le fer à souder et sans devoir investir dans des outillages spécialisés tels qu'outils de développement ou programmeurs par exemple.

C'est en grande partie à cette conjonction de facteurs que l'Arduino doit son succès, et nous allons voir, dans la suite de cet ouvrage, qu'il n'est pas usurpé.

■ Avertissement relatif aux schémas

À partir du chapitre 4, et jusqu'à la fin de cet ouvrage, vous allez pouvoir découvrir un certain nombre de schémas d'applications ou d'interfaces mettant en œuvre l'Arduino. Compte tenu de la relative universalité des entrées/sorties de ce dernier, nous avons adopté, les conventions suivantes.

Si, sur un schéma, ne figure aucune indication de nom ou de numéro de ligne d'entrée/sortie, cela signifie que n'importe quelle entrée/sortie numérique de

l'Arduino peut être utilisée, sans restriction. Sur l'Arduino Uno, cela signifie donc que l'on peut utiliser les entrées/sorties 2 à 13 incluses et sur l'Arduino Mega 2560 que l'on peut utiliser les entrées/sorties 2 à 53 incluses.

Si un numéro de ligne est indiqué, cela signifie :

- soit que le choix de la ligne est imposé parce qu'elle supporte une interface spécifique (Rx, Tx, SDA, SCL, /SS, MISO, MOSI, SCK, PWM) mais cela est alors clairement indiqué dans le texte associé ;
- soit que c'est cette ligne qui a été utilisée dans le programme associé au schéma.

Lorsque l'on doit utiliser une entrée analogique, celle-ci est toujours explicitement désignée sur les schémas afin que vous ne puissiez pas la confondre avec une entrée/sortie numérique, sous la forme AX où X est le numéro de l'entrée choisie. X peut varier de 0 à 5 pour l'Arduino Uno et de 0 à 15 pour l'Arduino Mega 2560.

Les entrées analogiques sont toutes échangeables entre elles sans restriction. Ainsi, un schéma utilisant par exemple A0, fonctionnera tout aussi bien si vous utilisez A1. Il faudra juste éventuellement penser à modifier le programme associé en conséquence.

■ Compléments en ligne

Vous trouverez sur www.dunod.com/contenus-complementaire/9782100710409 et sur www.tavernier-c.com, à la page consacrée à cet ouvrage, un certain nombre de compléments organisés comme indiqué ci-après.

□ Répertoire fiches techniques

Ce répertoire contient les fiches techniques au format pdf de tous les circuits intégrés et composants particuliers utilisés sur les schémas présentés dans cet ouvrage. Vous pouvez vous y reporter pour connaître précisément leurs caractéristiques électriques et les limites à ne pas dépasser, ou bien encore pour découvrir leurs registres internes et la signification de leurs contenus, notamment pour ce qui est des circuits disposant d'une interface SPI, I2C ou bien encore bus un fil. Chaque fichier porte le nom du composant ou de la famille de composants qu'il concerne.

□ Répertoire schémas cartes

Ce répertoire contient les schémas de toutes les versions de cartes Arduino ainsi que les schémas des différents shields présentés ou utilisés dans l'ouvrage. Ces schémas sont directement issus de la documentation de leurs fabricants respectifs et sont disponibles sous forme de fichiers au format pdf. Les noms donnés aux fichiers sont sans ambiguïté quant aux schémas qu'ils contiennent.

□ Répertoire bibliothèques

Par précaution, vous trouverez dans ce répertoire les bibliothèques utilisées dans certains programmes de l'ouvrage ; bibliothèques qui sont proposées sur des sites autres que le site officiel consacré à l'Arduino et dont la pérennité ne peut donc être assurée.

□ Répertoire programmes

Comme son nom l'indique, ce répertoire contient les listings de tous les programmes présentés dans l'ouvrage. Ils sont tous au format ino, prêts à être compilés, et peuvent donc être copiés directement dans la fenêtre de l'éditeur de l'outil de développement de l'Arduino.

Ces programmes sont directement issus du PC de l'auteur ayant servi à leur développement et sont donc parfaitement conformes aux versions qui sont présentées dans cet ouvrage.

Les noms de fichiers correspondent aux repères donnés aux listings dans l'ouvrage (L61.pde pour le listing 6.1 par exemple).

■ Avertissement relatif aux versions d'Arduino

L'Arduino est un produit qui évolue très vite, trop même diront certains, et, même si cet ouvrage est régulièrement mis à jour, de nouvelles versions d'Arduino auront peut-être été commercialisées entre la dernière mise à jour de ce livre et son achat par vos soins.

Afin que vous puissiez néanmoins utiliser avec profit nos exemples de schémas et de programmes, nous avons donc fait le choix d'utiliser pour vous les présenter l'Arduino Uno à partir de la Révision 3. Toutes les versions ultérieures d'Arduino Uno sont en effet « upward compatible » de cette version, comme disent nos amis anglo-saxons, c'est-à-dire que, même si les versions ultérieures comportent des fonctions ou des possibilités supplémentaires, elles restent totalement compatibles des versions précédentes.

Sachez aussi que, vu le succès rencontré par l'Arduino, d'autres versions que l'Arduino « de base » ou Arduino Uno sont aujourd'hui commercialisées. On trouve ainsi sur le marché l'Arduino Leonardo, l'Arduino Due et l'Arduino Tre ou Galileo est même en cours de commercialisation au moment où ces lignes sont écrites.

Ces versions d'Arduino sont des évolutions du modèle de base, visant à lui offrir plus de puissance et plus de ressources internes ou d'entrées/sorties mais tout ce que vous aurez appris dans cet ouvrage avec l'Arduino Uno sera évidemment utilisable avec ces nouvelles versions.

L'outil de développement de l'Arduino est lui aussi pris de la même frénésie d'évolution et son numéro de version ne cesse d'augmenter au fil du temps mais, là aussi, les programmes développés avec les versions précédentes peuvent être exploités sans aucun problème avec les versions actuelles.