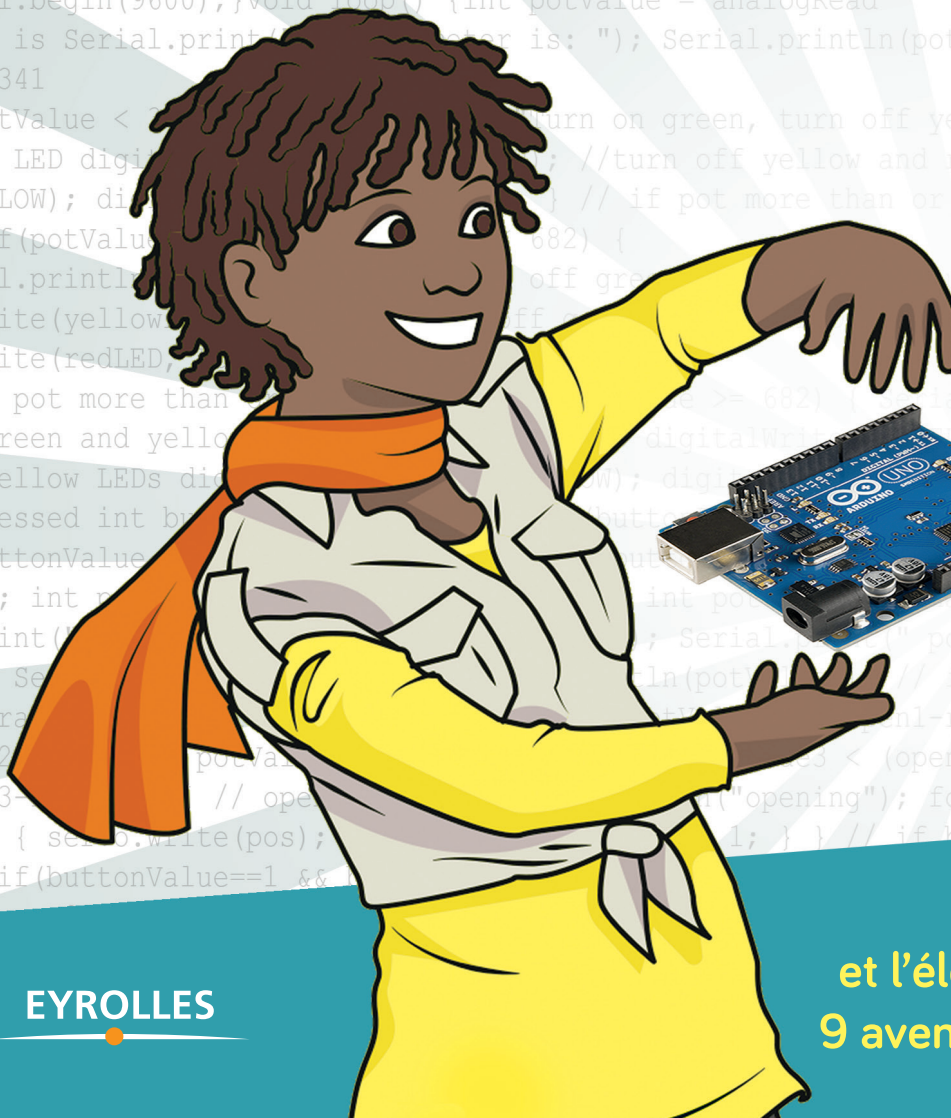


Becky Stewart

À l'aventure avec

# ARDUINO

Dès 10 ans



EYROLLES

Découvre Arduino  
et l'électronique grâce à  
9 aventures trépidantes !

# À l'aventure avec ARDUINO

## Pars à la découverte d'Arduino !

Amusant et facile à lire, ce livre te fera découvrir l'électronique et surtout le formidable potentiel d'Arduino, un petit microcontrôleur programmable qui permet de réaliser plein de projets ludiques. Avec des composants simples, du carton et de la colle, tu seras guidé dans la mise en œuvre de montages de difficulté croissante, allant d'un panneau d'affichage de LED à un jeu de labyrinthe à bille. Combinant avec intelligence théorie et pratique, cet ouvrage fera de toi un pro d'Arduino et des circuits électroniques. Dans notre société toujours plus numérique, cela te donnera un super avantage pour la suite de ton parcours scolaire.

Dans ce livre, tu fabriqueras :

- un carillon à vent électronique
- un mini coffre-fort à combinaison
- une boule de cristal qui s'illumine par magie
- un labyrinthe à bille qui mémorise ton score
- un affichage lumineux sur la manche d'un vêtement
- et bien d'autres projets !

## À propos de l'auteur

Ingénieur et professeur, Becky Stewart est la cofondatrice de Codasign, un organisme de formation qui propose des ateliers sur Arduino et les technologies numériques à l'attention du jeune public.



## À qui s'adresse cet ouvrage ?

Aux 10-15 ans, parents et enseignants.

Sur [www.editions-eyrolles.com/go/arduino](http://www.editions-eyrolles.com/go/arduino)

Télécharge toutes les ressources (code source, vidéos) du livre.

À l'aventure avec

**ARDUINO**

M. O'HANLON et D. WALE. – **Apprendre à coder en Python grâce à Minecraft.** *À paraître.*  
N°14292, 2016, 350 pages.

**Dans la collection « Pour les kids »**

D. WEINACHTER. – **Cahier d'activités Python pour les kids.** *Dès 10 ans.*  
N°14332, 2016, 64 pages.

J. R. BRIGGS. – **Python pour les kids.** *Dès 10 ans.*  
N°14088, 2015, 332 pages.

F. PAIN. – **Cahier d'activités Scratch pour les kids.** *Dès 8 ans.*  
N°14218, 2015, 64 pages.

THE LEAD PROJECT. – **Scratch pour les kids.** *Dès 8 ans.*  
N°14117, 2015, 160 pages.

**Dans la collection « Serial Makers »**

E. BARTMANN. – **Le grand livre d'Arduino (2<sup>e</sup> édition).**  
N°14117, 2015, 586 pages.

C. PLATT. – **L'électronique en pratique 2.** *36 expériences ludiques.*  
N°14179, 2015, 336 pages.

C. PLATT. – **L'électronique en pratique.** *36 expériences ludiques.*  
N°13507, 2013, 344 pages.

J. BOYER. – **Réparez vous-même vos appareils électroniques.**  
N°13936, 2014, 384 pages.

M. RICHARDSON et S. WALLACE. – **À la découverte du Raspberry Pi.**  
N°13747, 2013, 176 pages.

A. BANKS, MACUSER et IFIXIT. – **Réparez vous-même votre Apple.** *Une seconde vie pour votre iPhone, MacBook, iMac...*  
N°14251, 2015, 146 pages.

F. BOTTON. – **Les drones de loisir.**  
N°14219, 2015, 224 pages.

R. JOBARD. – **Les drones.** *La nouvelle révolution.*  
N°13976, 2014, 190 pages.

B. PETTIS, A. KAZIUNAS FRANCE et J. SHERGILL. – **Imprimer en 3D avec la MakerBot.**  
N°13748, 2013, 226 pages.

M. BERCHON. – **L'impression 3D (2<sup>e</sup> édition).**  
N°13946, 2014, 232 pages.

C. BOSQUÉ, O. NOOR et L. RICARD. – **FabLabs, etc.** *Les nouveaux lieux de fabrication numérique.*  
N°13938, 2015, 216 pages.



BECKY STEWART

# À l'aventure avec ARDUINO

EYROLLES

ÉDITIONS EYROLLES  
61, bd Saint-Germain  
75240 Paris Cedex 05  
www.editions-eyrolles.com

© 2015 John Wiley and Sons, Ltd.

*All Rights Reserved. Authorised translation from the English language edition published by John Wiley & Sons Limited. Responsibility for the accuracy of the translation rests solely with Groupe Eyrolles and is not the responsibility of John Wiley & Sons Limited. No part of this book may be reproduced in any form without the written permission of the original copyright holder John Wiley & Sons Limited.*

Traduction autorisée de l'ouvrage en langue anglaise intitulé *Adventures in Arduino* par Becky Stewart (ISBN 978-1-118-94847-7), publié par John Wiley & Sons.

Traduit et adapté de l'anglais par Jean Boyer.

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans l'autorisation de l'Éditeur ou du Centre Français d'exploitation du droit de copie, 20, rue des Grands Augustins, 75006 Paris.

© Groupe Eyrolles, 2015, ISBN : 978-2-212-14314-0

*Aux étudiants à qui l'on a toujours dit  
qu'ils devaient choisir entre les arts et la science.*

## **Remerciements**

Je remercie tout d'abord Alexandra Deschamps-Sonsino, sans qui je n'aurais jamais écrit ce livre. J'exprime également ma sincère gratitude à Alex pour son travail avec Tinker qui a donné le premier élan à la communauté Arduino de Londres.

Rien de tout cela n'aurait été possible sans le soutien des personnels d'éducation de Codasign. Ainsi, grâce à Melissa Coleman et Pollie Barden, j'ai beaucoup appris sur la manière d'enseigner Arduino, et j'apprends constamment, avec Adam Stark, à mieux former les enfants à la programmation informatique. Je n'oublierai pas non plus de remercier Emilie Giles, car Codasign aurait stagné sans vous : merci pour tout ce que vous avez fait. Je remercie aussi tout particulièrement Liat Wassershtrom, pour ses retours et son expertise.

Je voudrais également remercier les équipes éditoriales et artistiques qui m'ont aidée à mettre en forme cet ouvrage. Il a été grandement amélioré grâce à vos conseils.

Pour terminer, merci à ma famille qui m'a soutenue dans tout ce que j'ai entrepris. Merci à mes parents, qui ont mis à ma disposition un endroit calme pour écrire. Merci enfin à Ben, qui a gentiment accepté d'ajuster les dates de notre mariage et de nos vacances en fonction de mes travaux d'écriture.





# Table des matières

## Introduction ..... 1

Qu'est-ce qu'Arduino ? .....	1
Que vas-tu apprendre dans ce livre ? .....	2
De quoi auras-tu besoin ? .....	2
Les outils nécessaires .....	11
Les logiciels indispensables .....	15
Autre matériel utile .....	15
Que dois-tu connaître avant de lire ce livre ? .....	16
Comment ce livre est organisé ? .....	16
Présentation du texte .....	18
Le site web d'accompagnement .....	19
Pour obtenir de l'aide .....	20
Un mot du traducteur .....	20

## Aventure 1

### Installe et teste ton Arduino ..... 21

Ce dont tu as besoin .....	22
Téléchargement et installation du logiciel Arduino sur ton ordinateur .....	22
Installation du logiciel Arduino sur un Mac .....	24
Installation du logiciel Arduino sur un PC (Windows) .....	25
Installation du logiciel Arduino sur un ordinateur Linux .....	27
À la découverte de l'IDE Arduino .....	28
Vérification de l'installation avec le test du clignotement .....	29
Téléchargement du code de clignotement .....	31
Recherche des problèmes courants .....	34
Réalisation d'un circuit avec une LED .....	41
Ce dont tu as besoin .....	41
Compréhension des schémas des circuits .....	42
Utilisation d'une plaque de montage .....	44
Construction de ton premier circuit .....	45
Plus d'aventures en Arduino .....	48

## Aventure 2

### Utilise des capteurs ..... 51

Ce dont tu as besoin .....	51
Ajout de plusieurs LED .....	52

Envoi de messages à ton ordinateur . . . . .	60
Lecture de données en provenance d'un potentiomètre . . . . .	65
Comment prendre des décisions dans ton code ? . . . . .	69
Construction d'un panneau d'affichage de messages . . . . .	70
Ce dont tu as besoin . . . . .	71
Compréhension du circuit . . . . .	72
Réalisation du prototype sur ta plaque de montage . . . . .	72
Écriture du programme . . . . .	73
Construction de ton panneau d'affichage . . . . .	79
Plus d'aventures avec Arduino . . . . .	85

## Aventure 3

### Utilise des servomoteurs . . . . . 87

Ce dont tu as besoin . . . . .	87
Description des différents types de moteurs . . . . .	89
Contrôle d'un servomoteur avec Arduino . . . . .	90
Répétition d'actions . . . . .	93
Entrée numérique avec un bouton-poussoir . . . . .	95
Construction de ton coffre-fort à combinaison . . . . .	99
Ce dont tu as besoin . . . . .	100
Comprendre le circuit . . . . .	102
Le prototype sur plaque de montage . . . . .	102
Écriture du programme . . . . .	104
Réalisation du coffre-fort . . . . .	114
Soudage des fils . . . . .	116
Insertion de l'électronique . . . . .	118
Plus d'aventures avec Arduino . . . . .	119

## Aventure 4

### Utilise des registres à décalage . . . . . 121

Ce dont tu as besoin . . . . .	122
Organisation de ton code . . . . .	123
Utilisation des fonctions . . . . .	123
Utilisation des boucles for . . . . .	128
Comment obtenir plus de sorties avec un registre à décalage ? . . . . .	130
Comment fonctionne un registre à décalage ? . . . . .	132
Réalisation des connexions de ton registre à décalage . . . . .	134
Ajout de LED . . . . .	135
Écriture du programme . . . . .	137

Ajout d'autres registres à décalage . . . . .	143
Réalisation de l'affichage lumineux de ton nom . . . . .	145
Ce dont tu as besoin . . . . .	147
Comprendre le circuit . . . . .	148
Réalisation du prototype sur plaque de montage . . . . .	149
Écriture du programme . . . . .	149
Fabrication de tes lettres lumineuses . . . . .	155
Soudage des fils . . . . .	155
Insertion du circuit électronique . . . . .	157
Plus d'aventures avec les registres à décalage . . . . .	157

## Aventure 5

### Diffuse des sons . . . . . 159

Ce dont tu as besoin . . . . .	159
Création d'une liste . . . . .	160
Dévoile tes intentions . . . . .	162
Utilise une boucle et un tableau . . . . .	164
Mets cela en pratique . . . . .	164
Production de bruit . . . . .	168
Câblage de ton circuit . . . . .	170
Écriture du programme . . . . .	170
Réalisation d'un carillon à vent électronique . . . . .	176
Ce dont tu as besoin . . . . .	178
Comprendre le circuit . . . . .	179
Réalisation du prototype . . . . .	179
Écriture du programme . . . . .	181
Fabrication de ton carillon à vent . . . . .	182
Plus d'aventures sonores . . . . .	188

## Aventure 6

### Ajoute des bibliothèques . . . . . 191

Ce dont tu as besoin . . . . .	191
Les sorties analogiques . . . . .	192
Contrôle de la brillance d'une LED . . . . .	195
Mélange de lumières colorées . . . . .	198
Les capteurs capacitifs . . . . .	208
Ajout d'une bibliothèque . . . . .	208
Réalisation du câblage de ton circuit . . . . .	210
Écriture du programme . . . . .	211

Fabrication de ta boule de cristal . . . . .	215
Comprendre le circuit . . . . .	216
Réalisation du prototype sur plaque de montage . . . . .	217
Écriture du programme . . . . .	218
Construction de ta boule de cristal . . . . .	222
Plus d'aventures avec les bibliothèques . . . . .	225

## Aventure 7

### Utilise l'Arduino Leonardo . . . . .227

Ce dont tu as besoin . . . . .	227
Découverte de l'Arduino Leonardo . . . . .	228
Première connexion de ta carte Arduino Leonardo . . . . .	228
Un comportement identique à celui d'un clavier . . . . .	231
Détection de la lumière . . . . .	235
Réalisation de ton circuit . . . . .	236
Écriture du programme . . . . .	238
Réalisation d'une manette de jeu . . . . .	240
Ce dont tu as besoin . . . . .	241
Construction de ton circuit . . . . .	241
Écriture du programme . . . . .	243
Fabrication du boîtier de ta manette de jeu . . . . .	253
Assemble le tout . . . . .	254
Plus d'aventures avec l'Arduino Leonardo . . . . .	255

## Aventure 8

### Utilise l'Arduino LilyPad USB . . . . .257

Ce dont tu as besoin . . . . .	258
Découverte de l'Arduino LilyPad USB . . . . .	258
Un clignotant utilisant une carte Arduino LilyPad . . . . .	260
Prototypage des circuits souples . . . . .	262
Utiliser les tableaux astucieusement . . . . .	264
Passage de données entre fonctions . . . . .	270
Réalisation d'un vêtement à persistance de vision . . . . .	273
Ce dont tu as besoin . . . . .	274
Comprendre le circuit . . . . .	274
Écriture du programme . . . . .	277
Fabrication de ton vêtement à vision persistante . . . . .	280
Plus d'aventures avec l'Arduino LilyPad . . . . .	283

## Aventure 9

### La grande aventure : un labyrinthe à bille. . 285

Ce dont tu as besoin . . . . .	286
Première partie : le décompte des points . . . . .	287
Détection des vibrations avec des capteurs piézoélectriques. . . . .	287
Définir un seuil pour le marquage des points . . . . .	289
Ajouter des effets sonores . . . . .	292
Mémoriser le score . . . . .	293
Deuxième partie : création du labyrinthe . . . . .	295
Dessiner le labyrinthe . . . . .	296
Créer les règles du jeu . . . . .	297
Réaliser le prototype du circuit . . . . .	298
Troisième partie : écriture du programme . . . . .	300
Début de partie . . . . .	300
Fin de partie . . . . .	301
Début d'une nouvelle partie . . . . .	304
Mémoriser le score maximal . . . . .	306
Ajouter les sons . . . . .	307
Quatrième partie : fabrication du labyrinthe à bille . . . . .	315
Fabriquer ton labyrinthe . . . . .	315
Assembler les composants piézoélectriques . . . . .	317
Assembler les LED et le bouton-poussoir . . . . .	318
La touche finale . . . . .	319
Encore plus d'aventures : va plus loin avec Arduino . . . . .	320

## Annexe A

### Pour aller plus loin . . . . . 321

Plus de cartes, d'interfaces, de capteurs et d'actionneurs . . . . .	321
Les cartes interfaces (shields) . . . . .	322
Les capteurs et les actionneurs . . . . .	322
Tutoriels sur le Web . . . . .	322
Le site officiel d'Arduino . . . . .	322
Les sites de fabricants . . . . .	323
Les blogs . . . . .	323
Les vidéos . . . . .	323
Livres . . . . .	324
Sur Arduino . . . . .	324
Sur l'électronique de base . . . . .	324

Sur les circuits souples et les projets vestimentaires. . . . .	325
Autres sujets spécialisés . . . . .	325

## **Annexe B**

### **Où se procurer outils et composants . . . . .327**

Les kits de démarrage . . . . .	327
Les magasins spécialisés . . . . .	328
Les boutiques en ligne . . . . .	328
Les boutiques en ligne expédiant depuis l'Europe. . . . .	328
Les boutiques en ligne expédiant depuis les USA ou le Canada . . . . .	329
Deux autres sites incontournables . . . . .	329

### **Glossaire . . . . .331**

### **Index . . . . .337**

# Introduction

**ES-TU UN** aventurier ? As-tu envie de créer des projets qui t'amèneront à développer des nouvelles compétences et à maîtriser des outils inconnus ? Veux-tu découvrir comment utiliser la technologie pour concrétiser tes idées ? Es-tu curieux de savoir comment combiner programmation informatique et circuits électroniques avec des ciseaux et du papier, et même avec une aiguille et du fil ? Si la réponse est sans hésitation « oui », alors ce livre est fait pour toi !

## Qu'est-ce qu'Arduino ?

Arduino est un outil permettant de construire des dispositifs qui peuvent interagir avec l'environnement qui les entoure. Tu peux t'en servir pour y relier des capteurs détectant du son, de la lumière ou des vibrations, qu'il utilisera alors pour allumer une lumière, changer sa couleur, mettre en route un moteur, et bien d'autres choses. Arduino est un système magique, qui se situe au cœur de toutes ces actions. Il collecte des informations à partir de ses capteurs, évaluant ainsi le monde réel qui l'entoure. Il prend ensuite des décisions basées sur les données recueillies et provoque en retour des actions, sous forme de sons, de lumière, ou encore de mouvements.

Arduino se présente généralement sous la forme d'une carte électronique bleue, qui a à peu près la taille de ta main. Cette carte comporte des inscriptions en blanc qui permettent de repérer ses différents éléments. Tous les composants et les circuits de la carte sont visibles et accessibles. Il existe plusieurs types de cartes Arduino, et elles ne sont pas toutes bleues. Mais tu en apprendras davantage dans la section « De quoi auras-tu besoin ? » page 2, ainsi que dans les aventures 7 et 8.

Arduino est un microcontrôleur, autrement dit un ordinateur très simple. Il ne peut pas faire beaucoup de choses en même temps, mais ce qu'on lui dit de faire, il le fait très bien. Sans le savoir, tu utilises déjà beaucoup de microcontrôleurs dans la vie de tous les jours, car ils sont présents dans beaucoup d'appareils, comme les fours à micro-ondes et les machines à laver.

Il existe de nombreux types de microcontrôleurs mais ce qui est particulier avec Arduino, c'est qu'il est conçu pour être utilisé par les débutants. Donc, si tu es novice en programmation ou en électronique, tout va bien. Mais attention, ne sous-estime pas Arduino pour autant : il peut aussi s'adapter à de gros projets !

# Que vas-tu apprendre dans ce livre ?

Lorsque tu seras arrivé à la fin des aventures de ce livre, tu auras appris à installer l'environnement de programmation d'Arduino sur ton ordinateur, à l'utiliser pour écrire un programme et à le télécharger dans ton Arduino. Tu auras aussi découvert comment travailler avec trois types de cartes Arduino : la Uno, la Leonardo et la LilyPad USB.

Tu auras également appris les concepts de base de la programmation informatique, que tu pourras ensuite mettre en œuvre sur d'autres systèmes qu'Arduino. En effet, le langage d'Arduino est basé sur le langage C/C++. Cela signifie qu'en apprenant à programmer Arduino, tu sauras également comment s'effectue la programmation sur un ordinateur portable ou sur un Raspberry Pi.

En plus de la programmation, tu te seras familiarisé avec l'électronique et plus particulièrement avec ses différents circuits. Tu auras appris comment utiliser des capteurs pour détecter les signaux provenant du monde qui nous entoure, tels que la lumière ou les mouvements. Et tu sauras comment générer des actions en retour, par exemple produire un son ou allumer une lampe.

À la fin du livre, tu auras ainsi acquis une large connaissance de ce qui peut être réalisé avec Arduino. Tu seras alors prêt à concevoir et réaliser tes propres projets !

## De quoi auras-tu besoin ?

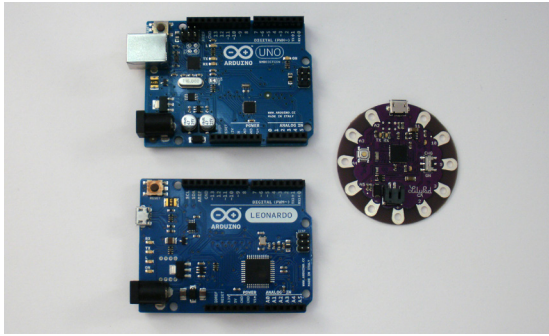
Cette section décrit tous les éléments dont tu auras besoin pour réaliser les projets de ce livre, sachant que de nombreux projets utilisent les mêmes composants.

Il est de plus en plus facile de se procurer des cartes Arduino (on en trouve même à la librairie Eyrolles !). Les boutiques spécialisées en électronique, comme Selectronic à Paris et à Lille, en ont toujours en stock. Ces magasins commercialisent également les autres composants dont tu auras besoin pour les projets de ce livre. Si tu ne peux pas te rendre dans une de ces boutiques, va sur Internet car de nombreux sites web proposent ces produits, ils sont répertoriés dans l'annexe B.

Le composant le plus important dont tu auras besoin est, bien entendu, une carte Arduino. Il existe différentes cartes Arduino, mais la Uno est la plus répandue et c'est celle que tu utiliseras le plus dans ce livre. Tu auras également besoin d'une carte Arduino Leonardo pour l'aventure 7 et d'une carte Arduino LilyPad USB pour l'aventure 8. Ces différentes cartes sont présentées sur la figure I-1.

Tu auras en plus besoin d'un câble USB pour relier la carte Arduino à ton ordinateur. Pour l'Arduino Uno, il s'agira d'un câble standard ; pour la Leonardo et la LilyPad USB, un câble micro USB sera nécessaire. Les deux sont visibles sur la figure I-2.



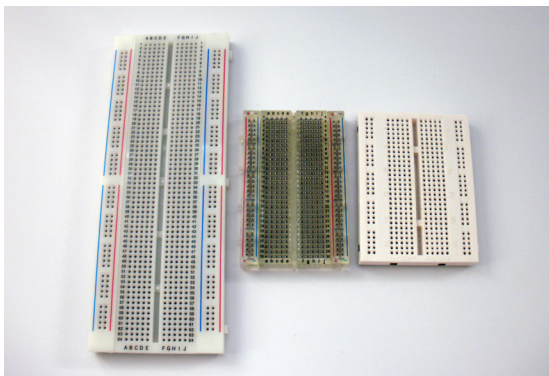


**FIGURE I-1** L'Arduino Uno (en haut à gauche), l'Arduino Leonardo (en bas à gauche) et l'Arduino LilyPad USB (à droite).



**FIGURE I-2** Un câble USB standard et un câble micro USB.

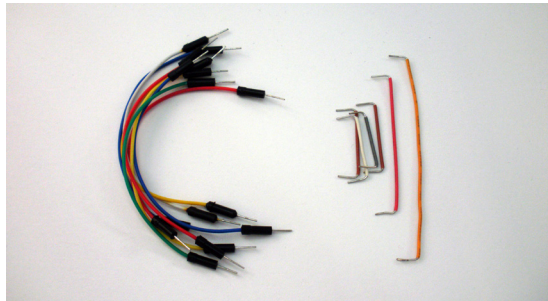
Tu vas aussi utiliser des plaques de montage pour relier facilement des composants électroniques sans avoir à les souder. Ces plaques existent dans différentes couleurs et dimensions. Les plus grandes sont utiles pour les projets les plus compliqués, qui nécessitent de nombreux composants ; les plus petites seront à utiliser pour les projets que tu souhaiteras réaliser dans un espace limité. Deux plaques de tailles différentes, réalisées avec des plastiques différents, sont montrées sur la figure I-3.



**FIGURE I-3** Des plaques de montage de diverses dimensions et couleurs.

L'aventure 3 est le seul projet qui utilise une plaque de montage pour sa réalisation. Les autres aventures n'en utilisent une que pour tester leur circuit. Il sera plus facile de travailler avec une plus grande plaque, mais si tu ne peux t'en procurer qu'une petite, cela conviendra très bien.

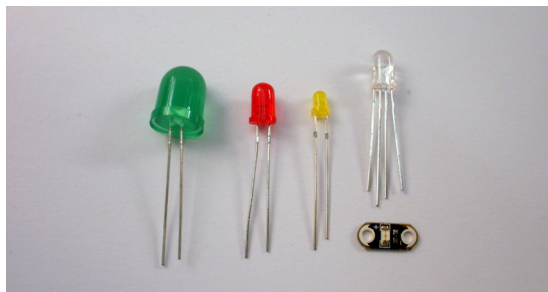
Les fils de pontage sont des fils que tu utiliseras pour réaliser les prototypes de tes circuits pour tester de nouveaux concepts. Certains sont rigides et de petite longueur, comme ceux montrés sur la figure I-4, à droite. D'autres sont plus souples avec des broches de connexion à chaque extrémité, comme ceux de gauche.



**FIGURE I-4** Des fils de pontage.

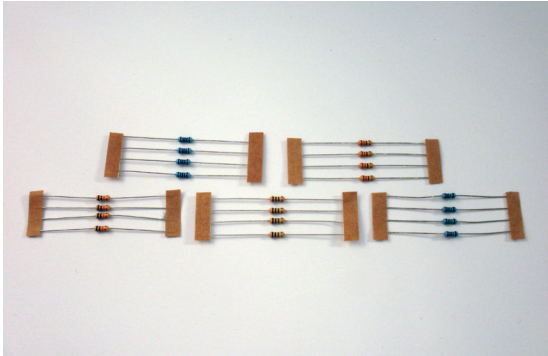
Les LED sont de petites lampes spéciales (LED signifie *Light Emitting Diode*, ou diode électroluminescente) qui sont disponibles dans une large sélection de tailles et de couleurs. Pour la plupart des projets de ce livre, tu pourras utiliser n'importe quelle taille et couleur, selon ta préférence. La taille la plus commune est de 5 mm de diamètre. Cependant, une diode de taille plus importante (10 mm, par exemple) pourra également être utilisée, pour plus de plaisir. La majorité des LED sont de couleur unique, mais, dans l'aventure 6, tu utiliseras une LED possédant quatre fils au lieu de deux : cette LED peut changer de couleur. Dans l'aventure 8, tu utiliseras un composant appelé LED LilyPad, spécialement conçu pour les circuits cousus sur du tissu.

Toutes les LED utilisées dans les projets de ce livre sont présentées sur la figure I-5.



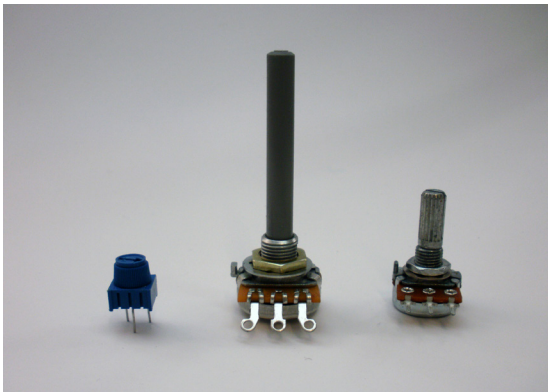
**FIGURE I-5** Différents types de LED dont une LED à changement de couleur (à droite) et une LED LilyPad (en dessous).

Les résistances sont des composants qui seront décrits dans les aventures. Elles sont disponibles avec différentes valeurs de résistance, mesurées en ohms ( $\Omega$ ). Tu n'auras pas besoin de beaucoup de résistances différentes pour les projets de ce livre. Les résistances étant de petits composants de faible prix, il serait bon d'en acheter des quantités plus importantes que celles requises. Tu auras besoin de résistances de 68 ou 100  $\Omega$ , 220  $\Omega$ , 10 k $\Omega$  (10 000  $\Omega$ ), 1 M $\Omega$  (1 000 000  $\Omega$ ) et 10 M $\Omega$  (10 000 000  $\Omega$ ). La figure I-6 montre les différentes résistances dont tu auras besoin.



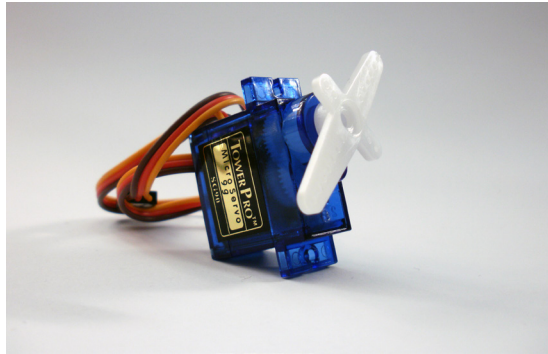
**FIGURE I-6** Les résistances nécessaires aux projets de ce livre : 100  $\Omega$  (en haut à gauche), 220  $\Omega$  (en haut à droite), 10 k $\Omega$  (en bas à gauche), 1 M $\Omega$  (en bas au milieu) et 10 M $\Omega$  (en bas à droite).

Les potentiomètres sont des composants électroniques que l'on utilise par exemple pour le réglage du volume sonore dans les appareils stéréophoniques. Ils se présentent sous différentes formes et dimensions. Certains peuvent s'insérer dans une plaque de montage, comme celui en bleu sur la figure I-7. D'autres nécessitent des fils soudés qui permettront de les relier à la plaque de montage, comme celui qui est au centre de la figure. Les modèles plus gros sont plus faciles à utiliser dans un projet, ce sont des potentiomètres pour montage en façade de boîtier.



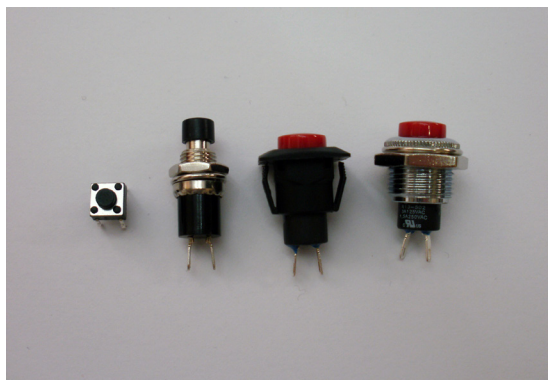
**FIGURE I-7** Trois types de potentiomètres.

Un servomoteur, comme celui de la figure I-8, est un type de moteur que tu utiliseras dans l'aventure 3.



**FIGURE I-8** Un servomoteur.

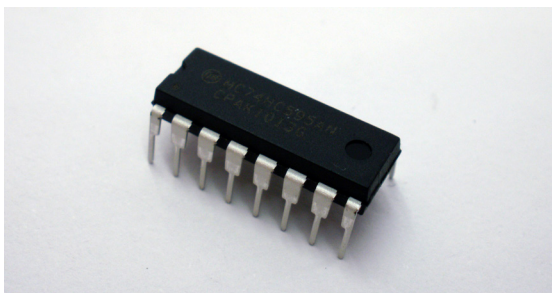
Les boutons-poussoirs sont d'autres composants se présentant sous différentes formes et dimensions. Tu n'as peut-être pas remarqué cela auparavant, mais il en existe de nombreuses sortes ! Tous les projets de ce livre utilisent des boutons-poussoirs qui ferment un contact (à l'opposé des boutons-poussoirs qui en ouvrent un). C'est donc des boutons de ce type qu'il conviendra d'acheter. Tant qu'il s'agit de poussoirs fermant un contact, tu pourras en utiliser n'importe quelle sorte. Les boutons-poussoirs tactiles sont de très petits interrupteurs qui peuvent s'insérer dans les plaques de montage ; ils sont pratiques pour tester les circuits. Pour les projets de ce livre, des boutons-poussoirs prévus pour être montés sur la façade d'un boîtier sont préférables. Les deux types sont présentés sur la figure I-9.



**FIGURE I-9** Un bouton-poussoir tactile (à gauche) et trois boutons-poussoirs différents prévus pour un montage en façade (à droite).

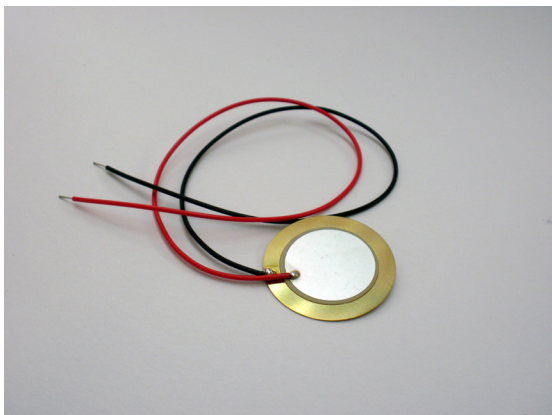
Dans l'aventure 4, tu apprendras à utiliser les registres à décalage. Ce sont de petits circuits intégrés noirs que tu peux utiliser pour contrôler un grand nombre de LED. Tu

auras besoin d'un circuit intégré de type registre à décalage 74HC595 (tu verras ce que cela signifie plus tard). C'est ce circuit intégré que tu devras acheter, il possède 16 broches, comme celui de la figure I-10.



**FIGURE I-10** Un registre à décalage.

Les transducteurs piézoélectriques sont des composants utilisés pour détecter des vibrations ou pour reproduire des sons, comme ceux d'un haut-parleur. Tu auras besoin d'un transducteur pour les aventures 5, 6 et 9. Ces composants se présentent parfois sous forme d'un boîtier en plastique noir, ce qui conviendra pour l'aventure 5, mais tu devras aussi disposer d'au moins cinq transducteurs sans boîtier (comme celui de la figure I-11), pour l'aventure 9.

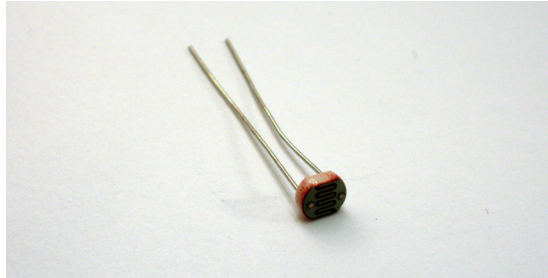


**FIGURE I-11** Un transducteur piézoélectrique.

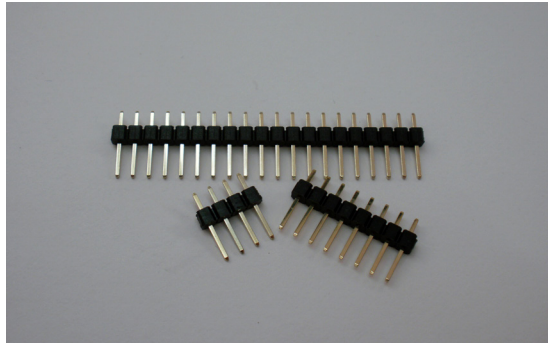
Une cellule photorésistive indique à Arduino la luminosité ambiante. Ce composant ressemble à celui de la figure I-12, d'autres peuvent être un peu plus grands.

Les barrettes de connexion de ce composant sont des petites broches métalliques de contact séparées par du plastique qui s'insèrent parfaitement dans les trous de connexion d'une carte Arduino Uno. Comme elles existent selon différents espacements, tu feras attention de bien choisir des barrettes à contacts mâles avec un

espacement de 2,54 mm, comme ceux de la figure I-13. Tu auras besoin d'une barrette de cinq contacts pour l'aventure 5. Tu pourras aussi t'en procurer des plus longues et les découper à l'aide d'une pince coupante.

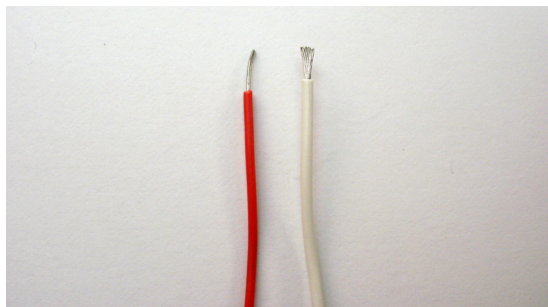


**FIGURE I-12** Une cellule photorésistive.



**FIGURE I-13** Des barrettes à contacts mâles.

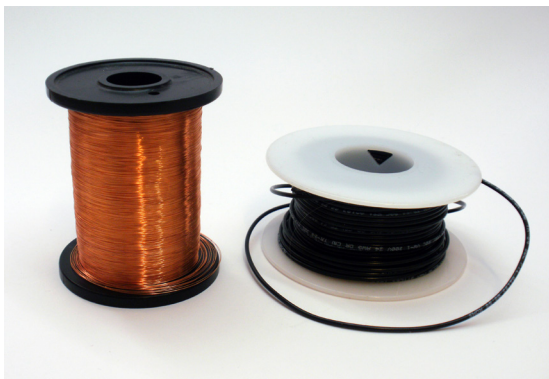
Lorsque tu penses aux circuits électriques, la première chose qui te vient à l'esprit est probablement un fil. Mais sais-tu qu'il en existe différentes sortes ? Un fil peut être constitué d'un seul brin métallique (fil rigide) ou de plusieurs brins torsadés (fil multibrins). La figure I-14 montre ces deux types de fils.



**FIGURE I-14** Du fil rigide (à gauche) et du fil multibrins (à droite).

Les fils rigides sont utiles avec des plaques de montage, mais ils sont peu souples. Les fils multibrins sont plus faciles à plier, mais tu devras étamer leurs extrémités à l'aide d'un fer à souder pour pouvoir les insérer dans une plaque de montage. Tu pourras décider, pour chaque projet, le type de fil que tu préfères : il n'y a pas de bon ou mauvais type à utiliser.

Les fils sont en général revêtus d'une enveloppe isolante ne conduisant pas l'électricité. Cette enveloppe peut être en plastique, comme pour le fil de droite sur la figure I-15, ou un vernis, comme le fil de gauche (qu'on appelle fil émaillé). Tu pourras choisir le fil qui te convient le mieux pour tes projets. Le fil émaillé conviendra bien pour le carillon de l'aventure 5, car il est très fin et permettra aux timbres du carillon de se balancer librement. Mais tu pourras aussi réaliser ce carillon en utilisant un autre type de fil très fin.



**FIGURE I-15** Du fil émaillé (à gauche) et du fil plastifié (à droite).

En électronique, la soudure est comparable à une colle conductrice. Elle se présente sous forme de bobines de fil à souder de différents diamètres, comme sur la figure I-16. Les projets de ce livre n'ont pas besoin de soudure spécifique, tu n'auras donc pas à te soucier du diamètre du fil que tu achèteras : n'importe lequel conviendra. Le seul point important sera de t'assurer d'acheter de la soudure pour l'électronique. N'achète surtout pas de la soudure pour la plomberie !



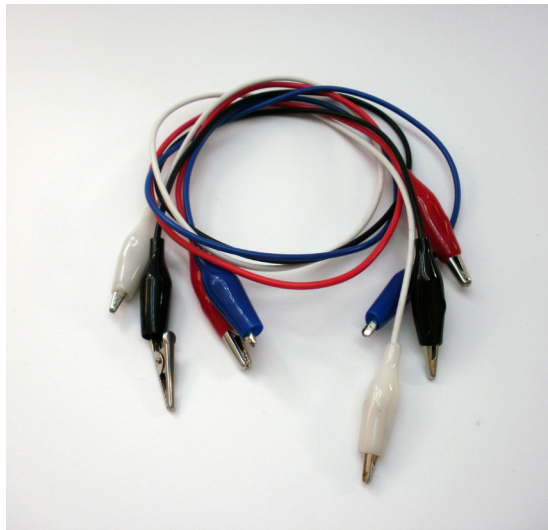
**FIGURE I-16** Des bobines de fil à souder.

Après avoir réalisé tes projets, tu souhaiteras peut-être les faire fonctionner sans devoir les relier à ton ordinateur pour les alimenter en électricité. Tu pourras alors les alimenter avec une alimentation électrique ou une pile. Dans le premier cas, le plus simple sera d'utiliser un bloc d'alimentation branché sur le secteur électrique et possédant une sortie USB qui te permettra de relier le bloc secteur et ta carte Arduino avec un câble USB. Si tu souhaites utiliser une pile, le plus simple est d'en utiliser une de 9 V (volts) avec un adaptateur à fiche d'alimentation cylindrique ; sur la carte Arduino se trouve un connecteur noir, dans lequel tu peux brancher la fiche de l'adaptateur de la pile. Les deux possibilités sont montrées sur la figure I-17. Pour la carte LilyPad USB, tu pourras utiliser une batterie LiPo : tu le verras dans l'aventure 8.



**FIGURE I-17** Un adaptateur de pile 9 V avec fiche cylindrique (à gauche) et un bloc d'alimentation secteur à sortie USB (à droite).

Pour réaliser les circuits de l'aventure 8, tu auras besoin de fils munis de pinces crocodiles, comme ceux de la figure I-18. Tu utiliseras à la place des fils de pontage.



**FIGURE I-18** Des fils munis de pinces crocodiles.



Dans cette aventure, tu utiliseras également de la cordelette conductrice. Il en existe différentes sortes, mais elles sont toutes de couleur argent, comme celles de la figure I-19.



**FIGURE I-19** De la cordelette conductrice.

## Les outils nécessaires

Tout comme tu aurais besoin d'un marteau et d'une scie pour fabriquer un objet en bois, tu auras besoin d'outils spécifiques pour réaliser tes circuits électroniques.

Tu utiliseras une plaque de montage pour tester tes circuits, mais auras parfois besoin de t'étendre au-delà de la plaque. Ce sera le cas, par exemple, quand tu voudras ajouter des fils plus longs à un composant pour qu'il puisse se loger à l'intérieur du boîtier de ton projet. Il en sera de même quand tu souhaiteras fixer des composants entre eux, afin qu'ils ne puissent pas se détacher de ton montage.

Le premier outil indispensable est un fer à souder. La soudure est comparable à de la colle pour l'électronique, mais elle nécessite une haute température pour être appliquée (imagine une colle chaude pour l'électronique). Un fer à souder est un outil qui devient très chaud (plus chaud qu'un four) afin de faire fondre le fil à souder. N'utilise un fer à souder qu'en présence d'un adulte qui pourra t'aider.

Un fer à souder peut se brancher sur une prise électrique et se tenir d'une seule main, comme celui de la figure I-20. Il peut aussi s'insérer dans un boîtier connecté au secteur, avec réglage de température. Les deux types sont adaptés. Le plus important est d'en acheter un prévu pour la soudure de circuits électroniques et non pour la plomberie ou toute autre activité.

Les fils se présentent le plus souvent avec une gaine en plastique qui est un isolant ne conduisant pas l'électricité. Tu auras parfois à retirer cette gaine aux extrémités d'un fil afin de lui permettre de s'insérer dans une plaque de montage ou pour le souder à un

composant. Tu pourras essayer d'utiliser un couteau ou un cutter avec précaution pour retirer l'isolant plastique, mais cela risque d'être très frustrant. Il est bien préférable d'acheter l'outil adéquat pour cette opération : la pince à dénuder !



**FIGURE I-20** Un fer à souder.

Les pinces à dénuder se présentent selon différentes formes et tailles, comme tu peux le voir sur la figure I-21. Tu pourras choisir celle qui te convient le mieux.



**FIGURE I-21** Différents modèles de pinces à dénuder.

Les pinces coupantes, tu t'en doutes, coupent les fils. Assure-toi d'en trouver des petites, qui tiennent facilement dans ta main, car tu vas devoir travailler avec des composants et des fils de petites tailles. La figure I-22 montre le modèle de pinces coupantes que tu pourrais acheter.

Les pinces pliantes t'aideront à plier et donner forme aux fils, elles existent dans différentes formes et dimensions. Pour les projets de ce livre, tu auras besoin d'une petite pince pliante à usage général. Toutes les pinces de la figure I-23 conviendront.



**FIGURE I-22** Une pince coupante.



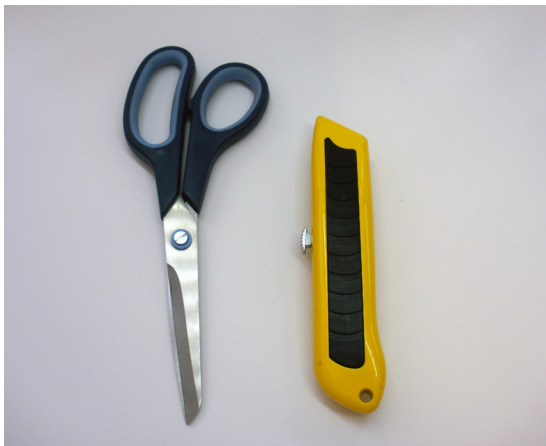
**FIGURE I-23** Deux modèles de pince pliante.

L'outil qui suit peut paraître un peu intimidant, mais il pourra devenir ton meilleur compagnon quand tu travailleras avec des circuits électroniques : c'est le multimètre ! Il mesure plusieurs paramètres (c'est pour cela qu'on le nomme ainsi) dont les résistances et les tensions, qui sont les plus utiles pour un débutant. On en trouve des très peu coûteux comme des très chers. Quand tu en choisiras un pour ton travail, tu n'auras pas à dépenser beaucoup, en particulier si c'est ton premier multimètre. En cherchant un multimètre en magasin, tu trouveras parfois la fonction de calibre automatique nommée *autorange*. Le calibre automatique signifie que tu n'as pas à connaître la valeur approximative de la mesure effectuée avant de la réaliser : le multimètre s'y adapte automatiquement. Par contre, tu devras absolument t'assurer qu'il permet d'effectuer des tests de continuité. Un test de continuité fait retentir un buzzer quand une connexion électrique est établie entre les deux fils de mesure du multimètre. La figure I-24 montre un multimètre peu coûteux, ne possédant pas le calibre automatique mais assurant les tests de continuité.

Les derniers outils qui te seront nécessaires ne sont pas spécifiques à l'électronique, mais ils seront essentiels pour construire les boîtiers de tes projets : des ciseaux et un couteau universel (figure I-25). Fais bien attention à ne pas te blesser en les utilisant !



**FIGURE I-24** Un multimètre.



**FIGURE I-25** Une paire de ciseaux et un couteau universel.

## Les logiciels indispensables

En parlant d'Arduino, il est facile de se limiter à la carte du circuit. C'est après tout la partie que tu vas physiquement placer dans tes projets. Cependant, Arduino a besoin d'un programme codifié pour pouvoir fonctionner. Tu devras d'abord écrire ce code à l'aide d'un ordinateur, puis le télécharger (dans le vocabulaire d'Arduino, on dit aussi « téléverser ») sur ta carte Arduino pour qu'il s'exécute.

La société qui fabrique les cartes Arduino conçoit également le logiciel qui t'aidera à écrire puis à téléverser ton code. Tu peux télécharger gratuitement ce logiciel depuis le site : <http://arduino.cc/en/Main/Software>

L'aventure 1 t'indiquera les étapes à suivre pour installer le logiciel sur ton ordinateur.

Les schémas des circuits ainsi que les plans de câblage des plaques de montage de ce livre ont été réalisés à l'aide d'un logiciel appelé Fritzing, qui est disponible gratuitement sur ce site : <http://fritzing.org/download>

Tu pourras également utiliser Fritzing pour concevoir tes propres projets.

## Autre matériel utile

Écrire un programme et construire un circuit ne représentent que la moitié d'un projet. Ce dernier ne prendra réellement forme que lorsqu'il sera placé dans une sorte de boîtier. Que ce soit un jeu ou une lampe interactive, tant qu'il n'est constitué que d'un circuit sur une plaque de montage, il n'a pas atteint tout son potentiel.

Pour réaliser les projets de ce livre, tu devras donc employer des techniques simples, en plus de tout ce que tu auras appris en électronique. Des ciseaux, du papier et de la colle constitueront ainsi le matériel de base pour de nombreuses aventures. De plus, tes projets seront encore plus beaux et plus personnels si tu leur ajoutes une petite touche de déco. En particulier, tu auras besoin des éléments suivants :

- une petite boîte en carton, ou une boîte à chaussures ;
- du papier cartonné, du carton et du papier ;
- de la peinture pour peindre des objets ;
- de la ficelle ou du fil ;
- de la colle blanche, un bâtonnet de colle ou un pistolet à colle ;
- un pinceau ;
- un ballon (pour l'aventure 6) ;
- une bille (pour l'aventure 9).

## Que dois-tu connaître avant de lire ce livre ?

Puisque tu as commencé la lecture de ce livre, je pense que tu dois t'intéresser à la technologie ! Il n'est pas nécessaire que tu aies déjà fait de la programmation sur ordinateur ou construit des circuits (puisque c'est ce que ce livre explique !). En revanche, je suppose que tu as déjà utilisé un ordinateur auparavant.

Tu auras besoin d'un ordinateur afin de travailler avec Arduino, mais son système d'exploitation importe peu – Mac OS X, Windows ou les différentes distributions de Linux (voir <http://playground.arduino.cc/Learning/Linux> pour connaître les distributions Linux qui peuvent être utilisées). Je suppose aussi que tu sais utiliser Internet et télécharger des fichiers, et que tu sais trouver et ouvrir les applications présentes sur un ordinateur.

Tu auras peut-être besoin du mot de passe de l'administrateur de ton ordinateur pour installer les logiciels. Si tu ne le connais pas, demande l'aide de quelqu'un qui le connaît quand tu les installeras.

## Comment ce livre est organisé ?

Ce livre va te guider dans la programmation de ta carte Arduino et dans la construction des circuits des neuf aventures. Chaque aventure commence par une introduction aux nouvelles connaissances dont tu auras besoin pour réaliser le projet proposé à la fin. Ces compétences peuvent correspondre par exemple à la découverte d'un nouveau composant, comme un moteur, ou encore à l'écriture intelligente d'un programme.

L'aventure la plus importante pour commencer est bien entendu l'aventure 1, car elle t'aidera à installer le logiciel nécessaire au téléchargement de ton code dans la carte Arduino. Si tu ne commences pas par cette étape, le reste des aventures n'aura que peu d'intérêt !

Les aventures 2 à 9 se réfèrent toutes aux aventures qui précèdent, il est donc recommandé de les suivre dans l'ordre. Si, bien entendu, tu préfères ne pas y consacrer l'attention nécessaire et les réaliser dans un autre ordre de ton choix, c'est tout à fait possible. Dans ce cas, tu pourras toujours chercher davantage d'aide sur un sujet particulier dans une aventure précédente s'il t'arrive de rencontrer quelque chose que tu ne connais pas.

En plus de t'aider à préparer ton ordinateur pour que tu puisses programmer ta carte Arduino, l'aventure 1 te guidera dans la réalisation de ton premier programme Arduino, qu'on appelle « sketch » ou « croquis » (traduction française du mot anglais *sketch*). À cette occasion, tu construiras ton premier circuit sur une plaque de montage pour contrôler une LED !

Dans l'aventure 2, tu apprendras à commander plus qu'une LED, à écrire un message depuis la carte Arduino vers ton ordinateur et à utiliser ton premier capteur, un

potentiomètre. Tu pourras mettre en pratique les connaissances acquises en réalisant un panneau d'affichage où un message indique que tu ne souhaites pas être dérangé.

Dans l'aventure 3, tu prendras connaissance d'un deuxième capteur : un bouton-poussoir. Tu le combineras avec le potentiomètre pour contrôler un moteur. Tu apprendras également à utiliser les boucles `for` dans un programme d'ordinateur pour répéter plusieurs fois la même suite d'actions. En combinant tout cela, tu réaliseras un coffre-fort à combinaison qui ne s'ouvrira que lorsque le bon code sera entré. Le « coffre-fort » n'est qu'une boîte en carton, il ne conviendra donc pas pour les bijoux de ta mère, mais il sera assez utile pour y cacher tes bonbons favoris.

L'aventure 4 te montrera comment scinder ton code en petits sous-ensembles, en utilisant ce qu'on appelle les fonctions. Tu utiliseras ensuite ces fonctions pour contrôler plusieurs LED à l'aide d'un circuit intégré spécial appelé registre à décalage. Dans le projet final de cette aventure, tu fabriqueras des lettres dans lesquelles seront intégrées des LED, dans le style des banderoles des anciens carnivals.

L'aventure 5 ajoute le son à la liste grandissante d'actions que tu peux contrôler avec Arduino. Tu y découvriras comment créer des listes dans le code pour jouer des petits morceaux de musique à l'aide d'un nouveau composant : le transducteur piézoélectrique. Tu réaliseras alors un carillon à vent amplifié, qui ajoutera une déformation électronique sonore à un ancien instrument.

L'aventure 6 introduit des contrôles plus subtils qui te permettront de modifier la luminosité d'une LED au lieu de simplement l'allumer ou l'éteindre. Tu étendras également les possibilités de ton Arduino en installant de nouvelles bibliothèques de code qui ne sont pas fournies avec le logiciel Arduino. Tu utiliseras ensuite tes nouvelles connaissances avec une LED tricolore pour créer une boule de cristal qui change de couleur de façon magique.

Dans l'aventure 7, tu découvriras une nouvelle carte Arduino : la Leonardo. Tu apprendras à maîtriser une de ses possibilités passionnantes : se comporter comme un clavier d'ordinateur. Tu utiliseras un nouveau capteur qui détecte la lumière et réaliseras une manette de jeu avec laquelle tu pourras jouer en faisant des vagues avec ta main.

Dans l'aventure 8, tu utiliseras encore une autre carte Arduino, la LilyPad USB. Tu réaliseras un circuit en utilisant une aiguille et de la cordelette conductrice à la place du fil électrique et du fil à souder. Cette aventure te permettra de maîtriser les tableaux, dans ton code, et de créer un vêtement qui affiche un message secret.

Enfin, la plus grande des aventures sera l'aventure 9 ! Tu auras la chance d'y mettre à profit l'ensemble des compétences acquises durant les précédentes aventures pour créer un jeu de labyrinthe à bille qui calculera automatiquement ton score, décomptera le temps restant et produira des effets sonores. Tu emploieras un transducteur piézoélectrique pour détecter les vibrations et reproduire des effets sonores.

Les annexes A et B te prépareront pour de nouvelles aventures au-delà de ce livre. L'annexe A t'orientera vers des livres et d'autres sources d'informations en ligne. L'annexe B t'indiquera où acheter les composants et les outils pour tes projets.

## Présentation du texte

Tout au long de ce livre, tu rencontreras des encadrés prévus pour t'aider.



Ces encadrés expliquent les concepts ou les termes avec lesquels tu n'es pas familier.



Ces encadrés te donnent des astuces pour t'aider dans l'écriture du code et le montage des circuits.



Ces encadrés contiennent d'importantes mises en garde afin que ton ordinateur et toi-même soyez protégés au cours de l'exécution des étapes d'un projet.



Dans ces encadrés, un rapide questionnaire te permettra d'évaluer ta compréhension ou d'approfondir un sujet particulier.



Dans ces encadrés, je t'explique des choses ou te donne des informations supplémentaires qui te seront utiles.



Ces encadrés contiennent des liens vers des vidéos (en anglais) qui te guideront dans les différentes étapes d'un projet.



Tu trouveras également deux autres types d'encadrés dans ce livre. L'encadré « Défi » te proposera de tirer profit de tes connaissances fraîchement acquises ou d'ajouter de nouvelles fonctions à tes projets. L'encadré « Exploration du code » te permettra de mieux comprendre les notions de programmation utilisées en codant Arduino.

En suivant les instructions du livre, tu devras entrer le code tel qu'il apparaît visuellement. Chaque « ; » est très important ! Par contre, les espaces entre les mots ne le sont pas : ils sont utilisés pour rendre le code plus lisible, mais ils ne sont pas importants pour Arduino. Par exemple, les deux lignes suivantes signifient la même chose :

```
if (i<4)
if ( i < 4 )
```

Ton sketch entier peut même être écrit sur une seule ligne de texte très longue, il fonctionnera quand même sur ton Arduino ! Ce sera cependant très difficile pour un autre programmeur de comprendre ce qui s'y passe. Il vaut mieux ajouter des espaces et des commentaires pour expliquer le contenu du programme : c'est la meilleure façon de coder.

Parfois, une ligne de code sera trop longue pour tenir sur une ligne du livre. Si tu vois le symbole ↵ à la fin d'une ligne de code, cela signifie que cette ligne et la suivante doivent être entrées sur une seule ligne de code dans le logiciel d'Arduino. Par exemple, les lignes suivantes doivent être entrées sur une seule ligne, pas sur deux :

```
Serial.println( "Hello, depuis la fonction setup() de ↵
ton Arduino Uno !" );
```

Pour que tu retiennes bien les notions de programmation expliquées dans ce livre, tu trouveras à la fin de chaque aventure un tableau qui liste les nouvelles commandes découvertes au cours du projet (fonctions, types de données, etc.).

Lorsque tu es arrivé à la fin d'une aventure, tu franchis une étape qui débloque un badge pour toi. Tu peux télécharger les badges de toutes les aventures en allant sur le site [www.editions-eyrolles.com/go/arduino](http://www.editions-eyrolles.com/go/arduino).

## Le site web d'accompagnement

Tout au long du livre, tu trouveras des références au site web d'accompagnement du livre : [www.editions-eyrolles.com/go/arduino](http://www.editions-eyrolles.com/go/arduino)

Sur ce site, tu pourras télécharger le code (en français et en anglais) des programmes des différentes aventures. Ces programmes seront classés par aventures, et par projets s'il y en a plusieurs dans une aventure. Tu pourras aussi visionner des vidéos (en anglais) qui t'aideront dans la réalisation matérielle des projets et dans leur programmation.

## **Pour obtenir de l'aide**

Dans l'annexe A de ce livre, tu trouveras de nombreux conseils pour te faire aider sur Arduino. Mais commence toujours par regarder le site d'Arduino ([www.arduino.cc](http://www.arduino.cc)), car même s'il est en anglais, il contient de nombreuses informations dans la section d'apprentissage *Learning*. De plus, tu pourras toujours poser tes propres questions sur le forum, qui existe lui en français :

<https://forum.arduino.cc/index.php?board=33.0>

## **Un mot du traducteur**

Les sites Internet et les ouvrages indiqués dans ce livre sont le plus souvent en langue anglaise. Toutefois, quand cela était possible, j'ai ajouté des sites ou des références en français qui traitent du même sujet. Tu pourras t'y rendre ou les consulter si tu ne peux pas lire l'anglais. Malheureusement, les sources d'informations les plus à jour et les plus complètes sont presque toujours dans la langue de Shakespeare... L'anglais technique étant généralement facile à lire, je t'invite à faire l'effort de parcourir ces documents. Je prends le pari, si tu as un niveau d'anglais scolaire du collège, que tu pourras rapidement comprendre les documents techniques dans cette langue. En plus, cela améliorera ton niveau en anglais.

Il est temps maintenant que tu partes à l'aventure !



## Aventure **1**

# Installe et teste ton Arduino

**TU VAS BIENTÔT** pouvoir créer des projets excitants qui feront le lien entre le monde physique et le monde digital ! Tu découvriras comment créer un programme qui déclenche un son, contrôle un moteur et fait clignoter la lumière. Arduino est l'outil parfait pour combiner des circuits et du code de programmation !

Tu suivras toujours les trois étapes suivantes, qui sont identiques pour chaque projet réalisé avec ton Arduino.

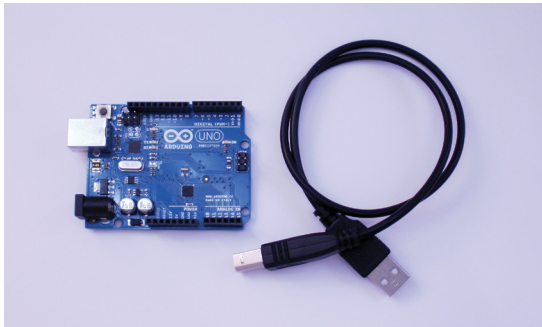
1. Écris sur ton ordinateur, à l'aide du logiciel Arduino, le code qui indique à ta carte Arduino Uno ce qu'elle devra faire.
2. Relie ta carte Arduino Uno à ton ordinateur pour y télécharger (téléverser) ton code.
3. Monte ton circuit puis relie-le à ta carte.

Commençons par le commencement. Avant de pouvoir faire quoi que ce soit, tu dois télécharger et installer le logiciel Arduino et préparer ton ordinateur pour programmer ta carte Arduino Uno. C'est ce que tu vas faire dans cette première aventure. Quand tu auras fait fonctionner le tout comme il se doit, tu pourras commencer ton premier projet Arduino : contrôler l'allumage et l'extinction d'une lampe.

## Ce dont tu as besoin

Pour commencer, tu as besoin des éléments suivants. La figure 1-1 montre les composants électroniques nécessaires :

- un ordinateur ;
- une carte Arduino Uno ;
- un câble USB ;
- une connexion Internet qui te permettra de télécharger le logiciel Arduino.




**FIGURE 1-1** Le microcontrôleur Arduino Uno et son câble USB.

## Téléchargement et installation du logiciel Arduino sur ton ordinateur

Pour pouvoir exécuter un programme Arduino, tu auras besoin, en plus de la carte Arduino Uno, d'un ordinateur et d'un logiciel spécial pour la faire fonctionner. Tu écriras d'abord sur un ordinateur le code qui va s'exécuter sur l'Arduino Uno, puis tu le téléchargeras dans ta carte. Cela te semble compliqué ? Ne t'inquiète pas, tu seras guidé pas à pas dans toutes les étapes. Et ce n'est pas aussi compliqué qu'il n'y paraît !

Tu vas utiliser un logiciel pour écrire le code de programmation, puis tu le téléchargeras. Ce logiciel s'appelle l'environnement intégré Arduino, ou *Integrated Development Environment (IDE)* en anglais. Tu le trouveras sur le site <http://arduino.cc/en/Main/Software> (voir la figure 1-2).

## Download the Arduino Software



**ARDUINO 1.0.6**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

**Windows** Installer  
**Windows** ZIP file for non admin install

**Mac OS X**

**Linux** 32 bits  
**Linux** 64 bits

[Release Notes](#)

**ARDUINO 1.5.8 BETA**

Arduino IDE that must be used for Arduino Yún and Arduino DUE. See the [release notes](#).

Windows Installer, ZIP file non admin  
 Mac OS X for Java 6, for Java 7  
 Linux 32 bit, 64 bit  
[Source](#)

ARDUINO 1.0.x / 1.5.x  
**PREVIOUS RELEASES**


Download the previous versions of the Arduino Software. Available for Windows, Linux, and Mac OS X.

ARDUINO IDE  
**INTEL GALILEO**  
**INTEL EDISON**

Download the Arduino IDE that supports the Intel Galileo and the Intel Edison boards. Available for Windows, Linux, and Mac OS X.

**FIGURE 1-2** Tu trouveras le logiciel IDE Arduino pour ton ordinateur sur le site web d'Arduino.

Un **environnement de développement intégré (IDE)** est un logiciel d'application utilisé pour écrire du code de programmation dans un langage particulier, on le nomme parfois « environnement de programmation ». Cette application permet de créer et modifier du code, mais aussi de le faire fonctionner (l'exécuter). De nombreux IDE offrent également des fonctions qui aident le programmeur à mettre au point (déboguer) leur programme. En d'autres termes, ils permettent de détecter les erreurs de programmation.



Tu vas maintenant télécharger et installer la dernière version du logiciel Arduino prévu pour le système d'exploitation de ton ordinateur en suivant les étapes ci-après. Quand ce livre a été écrit, la version courante de ce logiciel était la 1.6.5. Tu peux connaître la dernière version actuelle en te rendant sur le site <http://arduino.cc/en/Main/Software>. Tu y trouveras la dernière version en haut de la page. Après avoir installé le logiciel, tu pourras voir la version que tu utilises en lisant la barre du titre de la fenêtre de l'IDE Arduino.

Le logiciel peut fonctionner sur des ordinateurs Windows, Mac ou Linux. Toutefois, selon le système d'exploitation (*Operating Software* ou OS en anglais) utilisé, tu seras peut-être amené à installer un autre composant logiciel appelé **pilote**, en plus de l'IDE.

Cette aventure décrit ce que tu dois faire pour installer le logiciel. Tu peux aussi te rendre sur le site <http://arduino.cc/en/Guide/HomePage>, qui, en anglais, donne de précieuses indications pour l'installer.



Garde ta carte Arduino et le câble USB à portée de main lors de l'installation du logiciel, car tu pourrais en avoir besoin à certaines étapes, cela dépendra du système d'exploitation de ton ordinateur.



Un **pilote** (*driver*, en anglais) est un petit logiciel qui permet à ton ordinateur de communiquer avec un dispositif externe comme une imprimante ou un clavier.

## Installation du logiciel Arduino sur un Mac

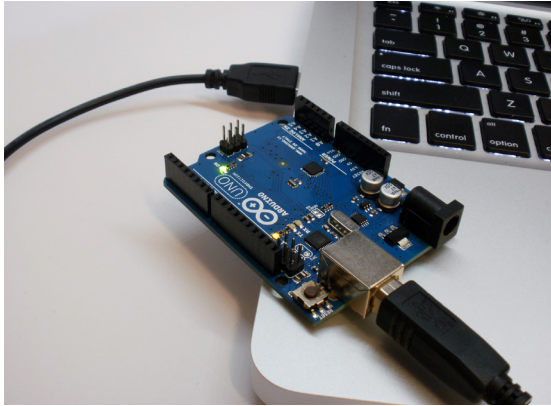
Il est très facile d'installer le logiciel sur un Mac. Il ne sera pas nécessaire d'installer un pilote en plus de l'environnement Arduino. Voici les étapes à suivre.

1. Ouvre la page <http://arduino.cc/en/Main/Software> dans ton navigateur Internet.
2. Clique sur le fichier zip **Mac OS X**.
3. Cherche le fichier **arduino-1.6.5-r3-macosx.zip** (ou la version la plus récente) et décompresse-le en double-cliquant dessus. Déplace alors le fichier **Arduino.app** dans ton dossier d'applications.
4. Après avoir installé le logiciel, relie le câble USB à ta carte Arduino puis l'autre extrémité à ton ordinateur. Un message indiquant qu'un nouveau périphérique réseau est détecté pourra alors apparaître à l'écran. Si cela se produit, annule ou ferme la fenêtre du message.



Si tu as un problème quelconque, rends-toi sur le site <http://arduino.cc/en/Guide/MacOSX> pour trouver de l'aide supplémentaire.

Pour voir une vidéo (en anglais) montrant l'installation de l'environnement Arduino sur un ordinateur fonctionnant sous Mac OS X, rends-toi sur le site d'accompagnement du livre [www.editions-eyrolles.com/go/arduino](http://www.editions-eyrolles.com/go/arduino).



**FIGURE 1-3** Relie le câble USB à la carte Arduino, puis connecte l'autre extrémité à ton ordinateur.

## Installation du logiciel Arduino sur un PC (Windows)

Tu dois suivre un certain nombre d'étapes pour installer le logiciel Arduino sous Windows. Tu devras en fait installer à la fois le logiciel et le pilote. Voici les étapes à suivre (il y aura peut-être quelques petites différences selon ta version de Windows, mais rien de bien méchant).

1. Dans ton navigateur Internet, rends-toi sur la page de téléchargement Arduino <http://arduino.cc/en/Main/Software>. Commence par télécharger le fichier nommé `arduino-1.6.5-r2-windows.zip`.
2. Cherche le fichier zip téléchargé et décompresse-le dans le dossier **Program Files**. Il devrait contenir plusieurs dossiers et fichiers, dont un dossier nommé **drivers** et un fichier nommé `arduino.exe`. Si tu veux, tu peux alors cliquer avec le bouton droit de ta souris sur le fichier `arduino.exe` et placer un raccourci vers ce fichier sur le Bureau de l'ordinateur.
3. Relie le câble USB à ta carte Arduino puis l'autre extrémité à ton ordinateur, comme le montre la figure 1-3. Tu devrais voir des voyants s'allumer sur ta carte Arduino Uno. Ceci indique simplement qu'elle est alimentée électriquement. Ton ordinateur va maintenant rechercher un pilote à utiliser avec la carte Arduino. Il se peut qu'il en trouve un de lui-même. Cependant, selon la version de Windows utilisée, tu peux avoir à suivre différentes étapes pour terminer l'installation du

pilote. Tu auras peut-être besoin du mot de passe de l'administrateur de l'ordinateur et de l'aide de quelqu'un ayant une plus grande expérience des ordinateurs. Tu peux toujours te rendre sur le site <http://arduino.cc/en/Guide/Windows> pour trouver des instructions détaillées (en anglais). De nombreux sites français proposent également des tutoriels d'installation ou de démarrage tels que [http://www.mon-club-elec.fr/pmwiki\\_reference\\_arduino/pmwiki.php?n=Main.DebuterInstallationWindows](http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.DebuterInstallationWindows).

4. Clique sur le menu Démarrer et ouvre le Panneau de contrôle.
5. En étant dans le Panneau de contrôle, rends-toi dans Système et Sécurité. Clique sur Système et quand la fenêtre Système apparaît, ouvre le Gestionnaire de périphériques.
6. Dans le Gestionnaire de périphériques, regarde sous la rubrique Ports (COM & LPT) : tu devrais y voir un port nommé **Arduino Uno (COMxx)**. Si tu ne vois pas la section COM & LPT, regarde dans Autres périphériques et cherche **Périphérique inconnu**.
7. Clique avec le bouton droit sur le port **Arduino Uno (COMxx)** et choisis l'option Mise à jour du pilote.
8. Si tu utilises Windows XP ou si la fenêtre de recherche de pilotes apparaît, va sur le site <http://arduino.cc/en/Guide/UnoDriversWindowsXP> et suis les copies d'écran pour installer les pilotes situés dans le dossier que tu as téléchargé et décompressé.
9. Si la fenêtre de recherche de pilotes n'apparaît pas et si, à la place, une fenêtre avec l'option Rechercher automatiquement un pilote à jour et Rechercher un pilote sur mon ordinateur apparaît, clique sur Rechercher un pilote sur mon ordinateur.
10. Va dans le dossier que tu as téléchargé et décompressé, puis dans le dossier **drivers** qui s'y trouve, et choisis **arduino.inf**.

Wouah, c'était compliqué ! Heureusement, tu n'auras plus à le refaire : une seule fois suffit. Les étapes peuvent être modifiées lorsque de nouvelles versions de l'IDE Arduino apparaissent, ou s'il existe des mises à jour de ton système d'exploitation. Tu pourras toujours te rendre sur le forum français de ta carte Arduino en cas de problème d'installation, à l'adresse <https://forum.arduino.cc/index.php?board=33.0>. Tu pourras y lire les questions posées par d'autres utilisateurs et même poser tes propres questions, si tu ne trouves pas de sujets similaires aux problèmes que tu rencontres.



Si tu rencontres des problèmes, visite d'abord le site <http://arduino.cc/en/Guide/Windows> pour obtenir plus d'aide. Le site [http://mchobby.be/wiki/index.php?title=Guide\\_d%C3%A9marrage\\_Arduino](http://mchobby.be/wiki/index.php?title=Guide_d%C3%A9marrage_Arduino) en français pourra également t'apporter l'aide attendue.



Tu peux aussi visiter le site <https://learn.adafruit.com/lesson-0-getting-started/installing-arduino-windows> ou [www.dummies.com/how-to/content/how-to-install-arduino-for-windows.html](http://www.dummies.com/how-to/content/how-to-install-arduino-for-windows.html) ou encore <https://learn.sparkfun.com/tutorials/installing-arduino-ide/windows> pour davantage d'astuces, dont un guide pour l'installation sous Windows 8.

Si ton ordinateur fonctionne sous Windows XP, tu peux installer les pilotes en suivant les instructions sur les copies d'écrans du site <https://arduino.cc/en/Guide/UnoDriversWindowsXP>.

## Installation du logiciel Arduino sur un ordinateur Linux

Si ton ordinateur fonctionne sous Linux (par exemple si tu utilises un Raspberry Pi), tu devras d'abord consulter la documentation en ligne (en anglais) pour Linux et Arduino sur le site <http://playground.arduino.cc/Learning/Linux>. Nombreuses distributions de Linux existent, je ne les ai pas toutes citées ici. Si ton ordinateur fonctionne sous une de ces distributions, tu dois déjà savoir comment y installer un nouveau logiciel. Il te sera probablement facile d'installer l'environnement Arduino à partir d'un paquet (*package*, en anglais) en utilisant le gestionnaire de paquets. Voici comment réaliser l'installation sous une variante Debian de Linux, telle que Raspbian sur une carte Raspberry Pi. Sur la ligne de commande, entre la commande suivante :

```
sudo apt-get install arduino
```

Cette commande télécharge puis installe le logiciel. Tu peux aussi télécharger les versions 32 et 64 bits directement depuis la page de téléchargement <http://arduino.cc/en/Main/Software>. Tu n'auras pas à te soucier de la compilation du code source. Cependant, n'oublie pas de lire les instructions correspondant à la distribution que tu utilises à l'adresse <http://playground.arduino.cc/Learning/Linux>.

Si tu utilises un Raspberry Pi ou un autre système basé sur Linux, je suppose que tu sais comment utiliser la ligne de commandes pour installer des logiciels, quelle que soit la version de Linux que tu utilises. Si tu as besoin de plus d'informations ou de rafraîchir tes connaissances sur l'utilisation du Raspberry Pi, consulte l'ouvrage *À la découverte du Raspberry Pi* de Matt Richardson et Shawn Wallace (éditions Eyrolles, 2014).



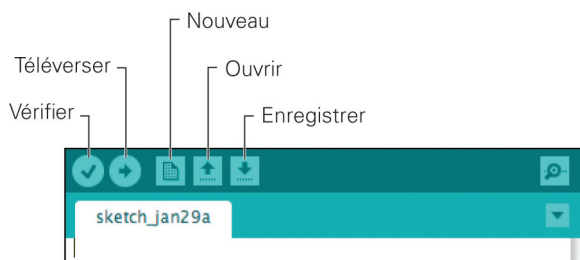
## À la découverte de l'IDE Arduino

Bien joué ! Maintenant que tu as installé le logiciel d'environnement Arduino, tu es prêt à utiliser l'IDE Arduino ! Il est temps de lancer le logiciel. Si ton ordinateur est un Mac, va dans le dossier Applications et double-clique sur Arduino. Si ton ordinateur fonctionne sous Windows, double-clique sur le raccourci que tu as créé sur le Bureau (si tu en as créé un comme indiqué précédemment dans cette aventure), ou va dans le dossier que tu as téléchargé précédemment et double-clique sur l'application Arduino. Si ton ordinateur fonctionne sous Linux, lance l'IDE Arduino depuis la ligne de commandes ou en double-cliquant sur l'icône du programme. L'icône du programme Arduino (voir la figure 1-4) est la même pour Mac, Windows ou Linux.

Lorsque l'IDE Arduino a terminé sa phase de lancement, une fenêtre similaire à celle de la figure 1-5 apparaît.



**FIGURE 1-4** L'icône du logiciel Arduino.



**FIGURE 1-5** La partie importante de l'IDE Arduino.

Par défaut, le logiciel Arduino possède des menus en anglais. Pour le faire fonctionner en français, effectue les opérations suivantes :

- après avoir lancé l'IDE Arduino, choisis le menu File puis Preferences avant de sélectionner Français (French) dans la case Editor language ;
- clique ensuite sur OK ;
- redémarre ensuite le logiciel Arduino qui devrait maintenant fonctionner en français.

Pour commencer, tu vas découvrir quelques-unes des fonctions de base de l'IDE Arduino. Il y a six boutons en haut de la fenêtre (voir la figure 1-5), le rôle de la plupart d'entre eux est facile à deviner. Les trois boutons du centre sont Nouveau, Ouvrir et Enregistrer. En cliquant sur Nouveau (surprise !), tu crées un nouveau fichier Arduino. Le bouton Ouvrir permet d'ouvrir un fichier Arduino existant. Enfin, Enregistrer permet de sauvegarder le fichier en cours. Tu remarqueras qu'il y a aussi un bouton isolé à l'extrémité droite de la fenêtre : tu t'en serviras au cours de l'aventure 2. Il ne te reste plus que les boutons de gauche à découvrir.

Le bouton qui ressemble à une case cochée est le bouton de vérification. Quand tu cliques dessus, le logiciel Arduino **compile** le code de ton programme : autrement dit, il prend le code que tu as écrit et le traduit en quelque chose de compréhensible par la carte Arduino. Une seconde action peut être déclenchée à l'aide de ce bouton. S'il y a des parties de code que le logiciel Arduino ne comprend pas, par exemple un point virgule manquant ou une erreur typographique durant la saisie des instructions, un message d'erreur est affiché en bas de la fenêtre de l'IDE Arduino. Celui-ci tente d'être explicite dans son message, mais ce dernier n'a parfois pas de sens ! Plus loin, dans cette aventure, tu verras les messages d'erreurs les plus fréquents et ce qu'ils signifient.

**Compiler** le code est un processus qui prend le code écrit par l'être humain et le transforme en instructions qui peuvent être comprises par une machine.



Ne te soucie pas encore trop de la façon dont cela fonctionne. Tout ce que tu dois retenir pour l'instant, c'est que le bouton Vérifier est utilisé pour traduire ton code en quelque chose que Arduino peut comprendre, et aussi pour savoir si le code contient des erreurs simples. Après avoir vu cela, clique sur le bouton Téléverser. Ce bouton compile et télécharge (téléverse) le code dans l'Arduino Uno. Le bouton Vérifier peut être utile quand tu écris un programme et que tu ne disposes pas de la carte Arduino Uno près de toi, car cela permet au moins de vérifier que ton code ne contient pas d'erreurs simples. En revanche, tu ne peux pas savoir si ton code fonctionne correctement avant de l'avoir téléversé dans ta carte Arduino.

**Téléverser** le code est un processus qui prend le code traduit sous une forme comprise par Arduino et le transmet à la carte Arduino qui va pouvoir le lancer.



## Vérification de l'installation avec le test du clignotement

D'habitude, tu peux écrire un programme pour un ordinateur comme un portable ou un Raspberry Pi, puis le lancer sur le même appareil que celui sur lequel tu l'as écrit. Avec Arduino, c'est légèrement différent, car une carte Arduino ne peut pas se programmer elle-même. En effet, tu dois d'abord écrire le code Arduino sur un ordinateur qui peut exécuter l'IDE Arduino. Le code qu'un être humain peut écrire est lu par l'IDE

puis traduit (ou compilé) en un code que la carte Arduino comprend. L'IDE télécharge ensuite le code compilé sur la carte Arduino.

Les programmeurs utilisent des petits programmes simples pour vérifier que leur ordinateur fonctionne tel qu'ils le souhaitent. Ces petits programmes sont appelés les programmes *Hello World* par les Anglo-Saxons, ou *Bonjour tout le monde* en français.

Un programme *Hello World* est un programme très simple. Si tu apprends un nouveau langage de programmation, tu écriras probablement un programme qui écrira simplement la phrase « Hello World » (ou « Bonjour tout le monde ») sur l'écran. Mais attention, Arduino n'a pas d'écran ! Après avoir téléchargé le code de ton ordinateur vers la carte Arduino, il n'y a plus de communication avec l'ordinateur ! Le câble USB ne sert plus qu'à alimenter le microcontrôleur (tu verras cependant, dans la prochaine aventure, comment envoyer des messages à l'ordinateur à l'aide du câble USB).

La carte Arduino peut faire clignoter une lampe. Elle en possède même une qui est prévue pour cela. Une petite **LED** (diode électroluminescente ou *light emitting diode* en anglais) orange ou rouge se trouve près du nombre 13 inscrit sur la carte (voir la figure 1-6). Cette LED est utile, car tu peux la programmer pour l'allumer puis l'éteindre afin de vérifier rapidement que ta carte Arduino fonctionne. Tu n'as besoin de rien en dehors de ta carte Arduino pour vérifier que tout fonctionne comme prévu.

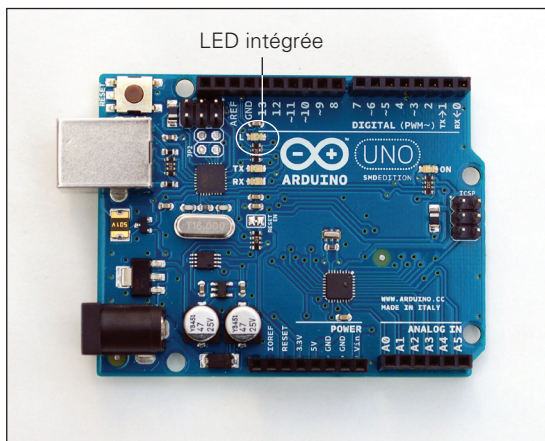
L'IDE Arduino comprend des exemples de code pour t'aider à débiter. L'un de ces exemples est le programme (également appelé **sketch**) de clignotement qui, pour Arduino, est utilisé à la place d'un programme *Bonjour tout le monde*. Tu vas maintenant l'utiliser pour vérifier que tout fonctionne comme tu veux.



Une **LED**, ou **diode électroluminescente**, est un composant électronique qui s'allume quand un courant électrique la traverse. Une diode ne laisse passer le courant électrique que dans un sens. Une LED laisse passer le courant seulement quand son fil le plus long est relié au pôle positif d'une source d'alimentation électrique et son fil le plus court au pôle négatif. Si les fils sont inversés, elle reste éteinte.



Les programmes Arduino sont souvent appelés **sketches**, ou encore **croquis**, comme les petits dessins que font les artistes.



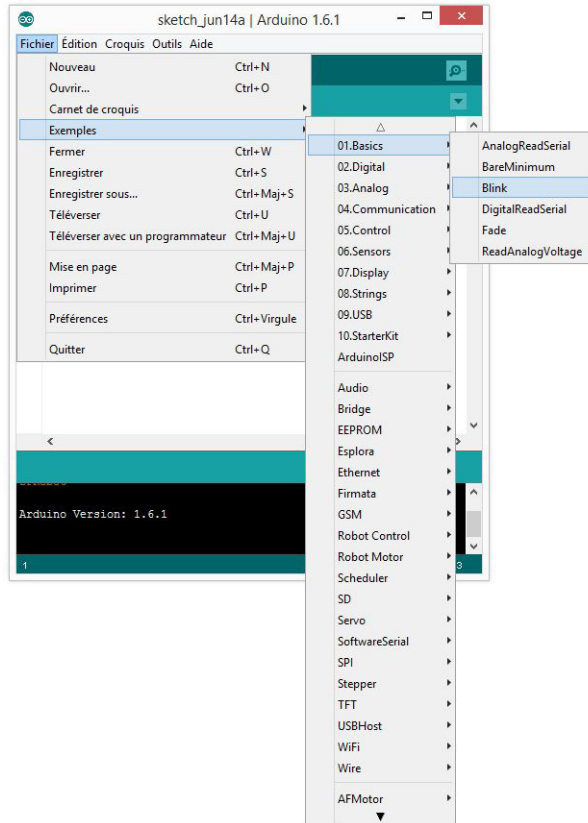
**FIGURE 1-6** La diode LED intégrée sur la carte Arduino près du nombre 13.

## Téléchargement du code de clignotement

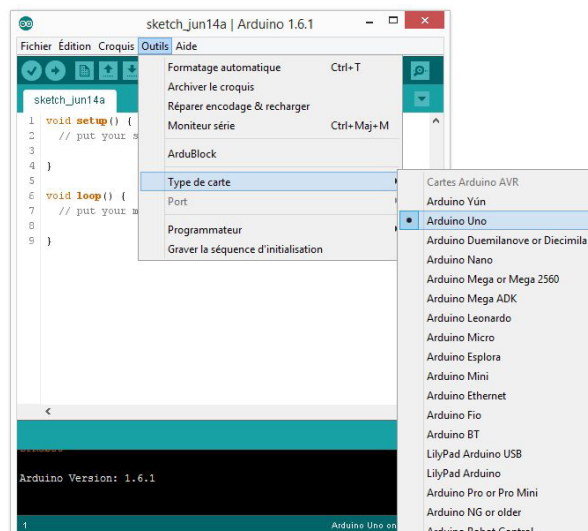
Pour vérifier que ton Arduino Uno peut recevoir des messages et un nouveau programme de ton ordinateur, tu dois compiler et télécharger le sketch de clignotement (*Blink sketch*). Va dans le menu Fichier > Exemples > Basics > Blink (voir la figure 1-7) afin d'ouvrir une nouvelle fenêtre contenant le sketch `Blink`.

Avant de télécharger ton code, tu dois vérifier deux réglages : le type de carte et le port de communication. Tu dois t'assurer que ces paramètres sont corrects chaque fois que tu démarres l'IDE Arduino. Après les avoir vérifiés, tu ne devras plus les changer jusqu'à ce que tu quittes et redémarres l'IDE à nouveau. Pour trouver le type de carte, choisis Outils > Type de carte (voir la figure 1-8). Tu devrais y trouver une liste de toutes les cartes Arduino. Assure-toi de choisir celle que tu utilises. (Ce sera normalement une Arduino Uno, mais si tu n'es pas sûr, le modèle est écrit sur la carte).

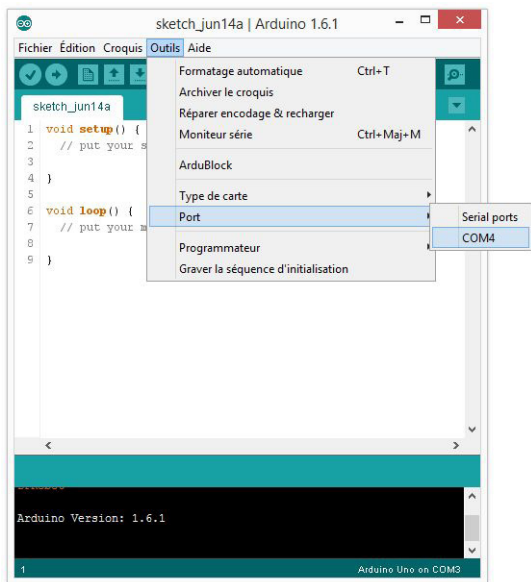
Tu trouveras le port dans le menu Outils > Port (voir la figure 1-9). Choisis le port que tu utilises de la même façon que tu as choisi le type de carte. Assure-toi que ta carte Arduino est reliée à ton ordinateur avec le câble USB, sinon le port n'apparaîtra pas dans la liste.



**FIGURE 1-7** Ouverture du sketch Blink qui se trouve dans les exemples inclus dans l'IDE Arduino.



**FIGURE 1-8** Sélection du type de carte utilisée.



**FIGURE 1-9** Sélection du port auquel est reliée ta carte Arduino.

Il peut être difficile de déterminer quel port tu dois choisir quand plusieurs ports sont indiqués dans la liste du menu. Tu peux consulter cette liste avant et après que la carte ne soit reliée. Si un port apparaît après que la carte est connectée, alors c'est celui-ci que tu devras choisir.



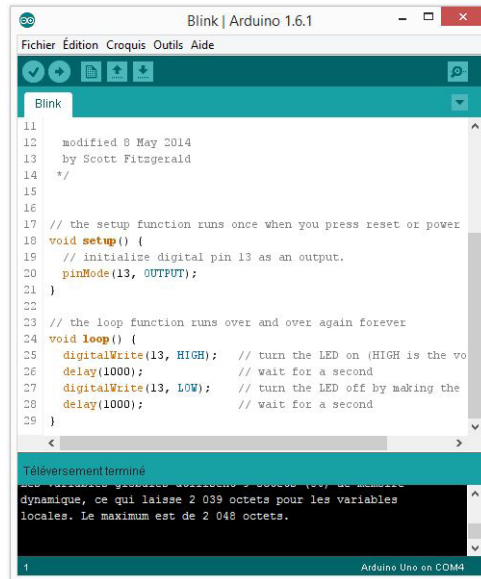
Si tu utilises un Mac, cherche le port qui commence par `/dev/tty.usbmodem`. Sur un ordinateur Windows, le port est simplement indiqué par `COM`. Sous Linux, le port commence par `/dev/ttyACM`. Il n'y a aucun risque si tu ne choisis pas le bon port. Si tu n'es pas sûr, choisis un port puis clique sur le bouton Téléverser. Si tu obtiens un message d'erreur, choisis le port suivant dans la liste et continue ainsi jusqu'à ce que tu trouves celui auquel ta carte Arduino est reliée.

Rappelle-toi que tu dois relier à l'ordinateur le câble USB qui est connecté à ta carte Arduino, sinon le port ne sera pas dans la liste des ports disponibles.



Après avoir choisi la carte et le port, tu es prêt à téléverser ton code : le croquis Blink que tu as ouvert. Pour cela, clique simplement sur le bouton Téléverser. Si tu as oublié de choisir le port, un message apparaîtra te demandant d'en choisir un.

Si tu as tout correctement préparé, tu verras un message dans la fenêtre du bas de l'IDE Arduino t'informant de la compilation du croquis puis de son téléchargement. Si le sketch est téléchargé sans problème, tu verras apparaître un message de fin (en caractères blancs).



```
Arduino IDE - Blink | Arduino 1.6.1
Fichier Édition Croquis Outils Aide
Blink
11
12   modified 8 May 2014
13   by Scott Fitzgerald
14   */
15
16
17 // the setup function runs once when you press reset or power
18 void setup() {
19   // initialize digital pin 13 as an output.
20   pinMode(13, OUTPUT);
21 }
22
23 // the loop function runs over and over again forever
24 void loop() {
25   digitalWrite(13, HIGH); // turn the LED on (HIGH is the vo
26   delay(1000); // wait for a second
27   digitalWrite(13, LOW); // turn the LED off by making the
28   delay(1000); // wait for a second
29 }
Téléversement terminé
Le compilateur gère les variables locales (V) de manière
dynamique, ce qui laisse 2 039 octets pour les variables
locales. Le maximum est de 2 048 octets.
1 Arduino Uno on COM4
```

**FIGURE 1-10** Le message de l'IDE Arduino après le téléchargement correct d'un sketch.

Regarde la LED à côté du nombre 13 sur ta carte. S'allume-t-elle pendant une seconde puis s'éteint-elle pendant une seconde ? Si oui, alors bravo, car tu as réussi ! Bien joué ! Si non, ou si un message en caractères orange est apparu dans la fenêtre du bas de l'IDE, la section suivante va t'aider à déterminer ce qui ne s'est pas bien passé.

## Recherche des problèmes courants

Lorsque quelque chose ne se passe pas correctement quand tu télécharges le code sur la carte Arduino, un message provenant d'un module appelé **avrduide** est affiché dans la fenêtre du bas de l'IDE (comme celui de la figure 1-11).

La carte Arduino est réalisée autour d'un circuit intégré microcontrôleur (calculateur) fabriqué par la société Atmel. Ce circuit intégré est un microcontrôleur de type AVR et le logiciel que l'IDE Arduino utilise pour communiquer avec la carte Arduino est appelé **avrduide**. Ainsi, quand tu obtiens des messages provenant d'**avrduide**, cela signifie que quelque chose s'est mal passé lors de la communication entre l'ordinateur et la carte. Généralement, c'est parce que l'ordinateur essaie d'envoyer un nouveau sketch à la carte



Arduino Uno par l'intermédiaire d'**avrdude**, mais l'ordinateur ne la trouve pas. Ces problèmes peuvent être dus au choix d'un mauvais port. Cependant, si tu as bien essayé tous les ports, il peut s'agir d'un autre problème.



**FIGURE 1-11** Une erreur courante quand l'ordinateur ne parvient pas à communiquer avec Arduino.

Le plus simple à faire, quand tu obtiens une erreur d'**avrdude**, est de déconnecter ta carte Arduino de son câble USB (cela supprime son alimentation électrique et la met à l'arrêt) puis de la reconnecter. Si tu as toujours des problèmes, essaie de déconnecter la carte Arduino, puis arrête l'IDE Arduino ainsi que tout autre programme. Redémarre ensuite l'IDE puis reconnecte l'Arduino Uno pour voir si tu réussis à télécharger un nouveau sketch.

Si tu as fait tout cela et que tu ne parviens toujours pas à télécharger des sketches sur ta carte Arduino, suis à nouveau toutes les étapes de l'installation de l'IDE et des pilotes.

Le site web d'Arduino (<http://arduino.cc>) est une aide précieuse renfermant de nombreux tutoriels. Il contient également un forum en français où tu pourras poser des questions. Tu y trouveras probablement des questions déjà posées par d'autres personnes ayant eu le même problème que toi.



## EXPLORATION DU CODE

Par chance, tu as maintenant téléversé et fait fonctionner ton code sur ta carte Arduino Uno.

Mais que fait réellement le code ? Tu sais que l'Arduino Uno allume puis éteint la LED située près du nombre 13, mais comment sait-elle qu'elle doit le faire ?



Une manière efficace de commencer l'étude d'un programme consiste à le lire avant de l'écrire. Après tout, à l'école, tu n'as pas appris à écrire avant de savoir lire ! Tu peux utiliser le sketch Blink comme introduction au code. Ne te soucie pas d'en connaître tous les détails dès maintenant, car il y a beaucoup de choses à apprendre. Tu découvriras tout un tas de nouveaux termes, mais tu n'auras pas besoin de retenir ce qu'ils signifient pour le moment. Tu consacreras plus de temps à les comprendre dans les aventures suivantes.

Si tu regardes à nouveau le sketch Blink, tu verras que la première partie est en gris. L'IDE Arduino aide à comprendre ce qui se passe dans un programme en changeant les couleurs du code selon ce qu'il fait. Le texte qui est en gris s'appelle un **commentaire**. Les commentaires sont des notes à l'attention des programmeurs pour les aider à comprendre ce qui se passe dans le code. Le long commentaire au début du sketch explique ce qu'il fait.



Les **commentaires** sont des notes à l'intérieur d'un programme qui expliquent ce qu'une ligne ou une partie de code fait. Chaque ligne de commentaire commence par `//`. Si tu veux écrire un commentaire de plusieurs lignes, il commencera par `/*` et se terminera par `*/`. Ces caractères spéciaux indiquent à l'ordinateur exécutant le programme d'ignorer la ou les ligne(s) de commentaires.

La version originale en anglais du sketch Blink :

```
/*  
  Blink  
  Turns a LED on for one second, then off for  
  one second, repeatedly.  
  
  Most Arduinos have an on-board LED you can  
  control. On the Uno and the Leonardo, it is  
  attached to digital pin 13.  
  If you're unsure what pin the on-board LED is  
  connected to on your Arduino model, check the  
  documentation at http://arduino.cc.  
  
  This example code is in the public domain.  
  
  modified 8 May 2014  
  by Scott Fitzgerald  
*/
```

La version francisée :

```
/*  
  Blink  
  Allume une LED pendant une seconde, puis l'éteint  
  pendant une seconde, de façon répétitive.
```

La plupart des cartes Arduino possèdent une LED intégrée que tu peux contrôler. Sur la Uno et la Leonardo, elle est reliée à la broche numérique 13.

Si tu n'es pas sûr de la broche à laquelle la LED est reliée, vérifie la documentation de ta carte sur le site <http://arduino.cc>.

Cet exemple de programme est dans le domaine public.

```
Modifié le 8 mai 2014  
par Scott Fitzgerald  
*/
```

Les lignes qui ne commencent pas par `//` sont des lignes de code que l'ordinateur doit exécuter. Celles qui commencent par `//` sont ignorées par l'ordinateur, ce sont des notes expliquant ce que fait le code.

Chaque sketch Arduino doit posséder deux **fonctions** : `setup()` et `loop()`. Une fonction est un ensemble de lignes de code qui possède un nom. La portion suivante du code est la fonction `setup()`. Cette fonction ne s'exécute qu'une seule fois, elle est prévue pour les tâches qui ne doivent être réalisées qu'une fois au démarrage d'Arduino. Chaque fois que la carte démarre, elle cherche la partie du sketch contenant la fonction `setup()` et commence par exécuter le code qu'elle contient.

Une **fonction** est un ensemble de lignes de code qui possède un nom. Une fonction peut être réutilisée de façon répétitive. Elle peut utiliser des informations qui lui sont fournies en entrée et délivrer des informations en sortie lorsqu'elle a terminé, mais toutes les fonctions n'ont pas à le faire.



La version originale en anglais :

```
/* The setup function runs once when you press
   reset or power the board. */
void setup() {
// initialize digital pin 13 as an output
pinMode(13, OUTPUT);
}
```

La version francisée :

```
/* La fonction setup est exécutée quand tu
   appuies sur le bouton reset ou alimente la
   carte. */
void setup() {
// initialise la broche 13 en mode sortie
pinMode(13, OUTPUT);
}
```

Comme tu peux le voir, il y a une seule instruction à l'intérieur de la fonction `setup()` : c'est un appel à la fonction `pinMode()`. Chaque broche d'Arduino peut lire des informations ou en délivrer en sortie, mais elle ne peut pas le faire de façon simultanée. La fonction `pinMode()` indique si la broche est utilisée en entrée ou en sortie grâce à deux **arguments**. Le premier est le numéro de la broche que tu utilises. Le second est un mot-clé spécial (`OUTPUT`), qui indique à Arduino que tu souhaites utiliser la broche 13 en sortie et non pas en entrée.



Un **argument** est une information transmise à une fonction, que celle-ci utilise pour exécuter sa tâche. L'argument se place entre les parenthèses qui suivent le nom de la fonction. Dans le fragment de code suivant, par exemple, la fonction `delay(1000)` a comme argument 1000 : c'est le nombre de millisecondes pendant lesquelles on veut qu'Arduino attende avant d'exécuter la ligne de code suivante.

Le reste du code du sketch Blink est la fonction `loop()`. Après que l'Arduino Uno exécute le code contenu dans `setup()`, elle cherche la fonction appelée `loop()` et exécute alors l'ensemble du code qu'elle contient. Quand elle a terminé, elle exécute à nouveau le code de cette même fonction. Puis encore

à nouveau ! Et ainsi de suite ! La fonction `loop()` se répète indéfiniment (ou au moins jusqu'à ce que Arduino ne soit plus alimentée électriquement).

La version originale en anglais :

```
/* The loop routine runs over and over again
   forever */
void loop() {
  digitalWrite(13, HIGH); /* turn the LED on
  (HIGH is the voltage level) */
  delay(1000); // wait for a second
  digitalWrite(13, LOW); /* turn the LED off by
  making the voltage LOW */
  delay(1000); // wait for a second
}
```

La version francisée :

```
/* La fonction loop s'exécute de façon
   répétitive */
void loop() {
  digitalWrite(13, HIGH); /* allume la LED
  (HIGH est le niveau de tension HAUT) */
  delay(1000); // pause d'une seconde
  digitalWrite(13, LOW); /* éteint la LED en
  mettant le niveau de sortie BAS (LOW)*/
  delay(1000); // pause d'une seconde
}
```

La première ligne de code dans la fonction `loop()` allume la LED en utilisant la fonction `digitalWrite()`. Cette fonction reçoit deux arguments : le numéro de broche, et si oui ou non tu délivres un signal électrique sur la broche. L'argument du numéro de broche est identique à celui de la fonction `pinmode()`. Le second argument est un mot-clé, `HIGH` ou `LOW`. `HIGH` délivre une tension électrique en sortie alors que `LOW` n'en délivre pas.

La dernière partie du code, que tu rencontres pour la première fois, est la fonction `delay()`. Arduino exécute son code très rapidement, des millions de fois par seconde. C'est si rapide que tu ne pourrais pas voir la diode LED s'allumer et s'éteindre. Tu dois donc demander à ta carte de faire des pauses pour que tu puisses voir la LED clignoter. La fonction `delay()` lui commande d'attendre le nombre de millisecondes qui lui est fourni comme argument. Dans cet exemple, le délai est de 1 000 millisecondes, ce qui est égal à une seconde. Voici un résumé de ce que fait l'ensemble du croquis :

- la fonction `setup()` utilise la fonction `pinmode()` pour indiquer que la broche à laquelle est reliée la LED est utilisée en sortie ;

- dans la fonction `loop()`, la sortie de la broche 13 délivre une tension électrique à la LED ;
- Arduino fait ensuite une pause de 1 000 millisecondes ;
- puis la sortie de la broche de la LED ne délivre plus de tension ;
- et Arduino fait à nouveau une pause de 1 000 millisecondes ;
- ensuite, la fonction `loop()` s'exécute à nouveau.

Un dernier détail que tu as peut-être remarqué : c'est l'étrange ponctuation utilisée dans le code. La plupart des lignes se terminent par un point-virgule (;), comme un point à la fin d'une phrase. Un point indique qu'une phrase est terminée, ce qui permet une écriture nette et propre. Un point-virgule signifie la même chose pour un ordinateur : il permet à l'ordinateur de séparer les différentes lignes de code.

Tu as peut-être noté également que le code contient des espaces et des indentations (des retraits de textes). De même que les commentaires, ils permettent au code d'être plus facile à comprendre par le programmeur. L'ordinateur les ignore. L'indentation permet de voir facilement les lignes de code appartenant à une même fonction. Tu as sans doute remarqué que les lignes de code appartenant à la fonction `loop()` sont toutes situées entre les crochets { et } et sont indentées (en retrait). Les crochets { } indiquent à l'ordinateur que ces lignes appartiennent à la fonction `loop()`, alors que l'indentation rend cela visuellement plus clair.

Les espaces entre les parties du code sont également prévus pour le programmeur, l'ordinateur les ignore aussi. Par exemple, `delay(1000)` et `delay( 1000 )` sont identiques pour l'ordinateur, mais les espaces permettent une meilleure lisibilité du code pour certaines personnes. Tu pourras décider comment tu préfères écrire ton code dans les aventures suivantes !

## DÉFI



Essaie de modifier la durée pendant laquelle la LED s'allume puis s'éteint en changeant les arguments des fonctions `delay()`. Rappelle-toi que tu dois téléverser ton code à chaque fois que tu le modifies afin que le sketch avec les nouveaux changements soit bien dans Arduino.

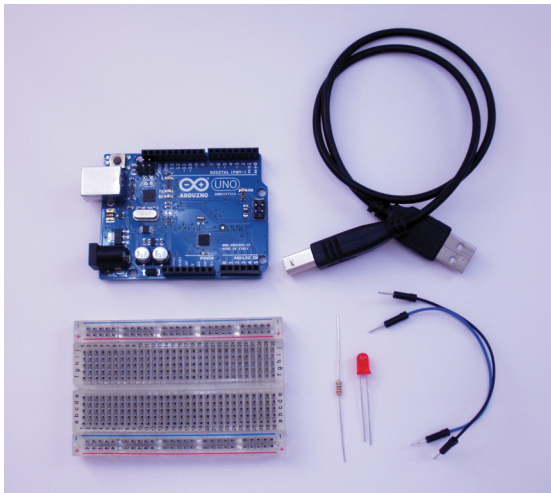
# Réalisation d'un circuit avec une LED

Maintenant que tu as une LED clignotante sur ta carte Arduino Uno, tu es prêt à aller plus loin en construisant ton premier circuit ! Tu vas d'abord te familiariser avec des outils dont tu auras besoin pour réaliser et tester de nouveaux circuits : les schémas et les plaques de montage. Ce sont les éléments essentiels pour réaliser tes propres projets que tu pourras ensuite partager, et aussi pour être capable de construire les projets créés par d'autres. Dans l'aventure suivante, tu franchiras une étape supplémentaire en construisant un boîtier pour ton circuit. Mais d'abord, familiarise-toi avec la façon dont le code et les circuits se combinent avec l'Arduino Uno.

## Ce dont tu as besoin

Tu dois disposer des éléments suivants pour construire ton circuit à LED. La figure 1-12 montre les composants électroniques nécessaires :

- un ordinateur ;
- une carte Arduino Uno ;
- un câble USB ;
- une plaque de montage ;
- une résistance de 220  $\Omega$  ;
- une LED ;
- 2 fils de pontage.

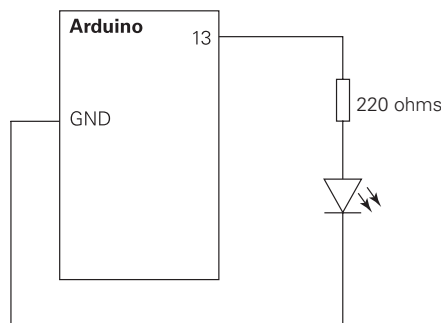


**FIGURE 1-12** Les composants électroniques dont tu as besoin pour réaliser ton circuit.

## Compréhension des schémas des circuits

L'électricité est un flux de charges électriques. Tu l'as vu dans la nature avec les orages et avec l'électricité statique qui survient quand tu marches sur un sol en moquette puis que tu touches une poignée de porte. Tu utilises aussi des circuits tous les jours pour contrôler la circulation de l'électricité. Par exemple, tu allumes et éteins l'éclairage d'une pièce à l'aide d'un interrupteur. Tu peux aussi allumer la télévision et changer de chaîne. Tout cela est réalisé à l'aide de circuits qui contrôlent l'électricité. Tu n'es pas encore prêt à réaliser des circuits comme ceux qui sont à l'intérieur d'un téléviseur, mais tu peux en construire un qui allume et éteint une lampe !

Même très simple, un circuit peut être réalisé de différentes manières. Par exemple, les LED existent en plusieurs tailles et couleurs. Tu peux alimenter une LED depuis l'Arduino Uno ou en utilisant une pile. Le schéma d'un circuit est un diagramme simplifié qui montre les informations importantes du montage en utilisant des symboles. La figure 1-13 montre le schéma du circuit que tu vas réaliser. Ce schéma contient trois symboles qui représentent l'Arduino Uno, la résistance (on verra cela plus tard) et la LED (le symbole triangulaire). Tu pourrais réaliser ce circuit avec une LED rouge de grande taille ou avec une petite diode de couleur verte, c'est ton choix. L'information importante est que tu dois relier cette LED à l'Arduino Uno et à une résistance.



**FIGURE 1-13** Le schéma du circuit à LED.

L'électricité peut être définie et mesurée de plusieurs façons. Parce qu'elle est invisible, il est difficile d'imaginer comment l'électricité agit, c'est pour cela que l'on utilise souvent une analogie avec l'eau. Le flux du courant électrique dans un fil est comparable à l'eau circulant dans un tuyau. L'eau s'écoulant au travers du tuyau est similaire au courant électrique (qui est mesuré en ampères et que l'on symbolise avec la lettre A) et la pression de l'eau est comme la tension électrique (qui est mesurée en volts et symbolisée par la lettre V). Le diamètre du tuyau combiné avec la quantité d'eau qui doit le traverser modifie la pression de l'eau. Un tuyau plus petit crée une pression d'eau supérieure à un gros tuyau pour la même quantité d'eau à faire passer. Le diamètre du tuyau représente la troisième propriété utilisée pour décrire l'électricité : la résistance électrique (mesurée en ohms et représentée par le symbole  $\Omega$ ).



Cela peut paraître compliqué et difficile à comprendre, mais n'essaie pas d'en saisir tous les détails dès maintenant. En travaillant avec l'électricité et en réalisant des circuits, tu verras comment tension, courant et résistance sont en relation. Revenons donc au schéma du circuit !

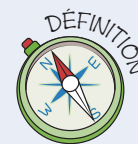
L'étude de l'électricité et des circuits est appelée la théorie des circuits. Si tu es intéressé par un apprentissage plus approfondi de cette théorie, il existe de nombreux tutoriels en ligne qui te permettront d'en savoir plus. J'aime beaucoup ceux de SparkFun (en anglais). Ils contiennent des animations qui aident à illustrer les concepts ! Commence par <https://learn.sparkfun.com/tutorials/what-is-electricity> et <https://learn.sparkfun.com/tutorials/voltage-current-resistance-and-ohms-law>.



Dans le schéma de la figure 1-13, le circuit est relié à la broche 13 et à la broche GND (masse) d'Arduino. La broche 13 est celle depuis laquelle l'électricité qui va allumer la LED circule. GND signifie la Terre, ou la Masse, ou le potentiel 0 V. Si tu utilises une pile pour alimenter un circuit, celle-ci possède deux connexions, l'une positive et l'autre négative. L'électricité circule du pôle positif vers le pôle négatif. Les broches 13 et GND de la Uno ont exactement le même rôle que le positif et le négatif d'une pile. L'électricité circule de la broche 13 vers la masse (GND) quand elles sont reliées à un circuit.

Le circuit contient deux composants en plus de la carte Arduino Uno : une résistance et une LED. Toutes les broches de la carte fournissent une tension de 5 V ce qui est trop pour une LED. Souviens-toi de l'analogie avec l'eau et imagine que la LED est une paille pour boire. Elle n'est pas aussi résistante qu'un tuyau d'eau, et si l'on force le passage de trop d'eau au travers de la paille, cela pourrait la faire éclater. Une **résistance** aide à contrôler le courant qui circule au travers de la LED. Dans ce type de circuit, la résistance limite la quantité de courant, c'est pourquoi on l'appelle « résistance de limitation de courant ».

Une **résistance** est un composant électrique qui résiste au passage du courant dans un circuit. Par exemple, une LED pourrait être endommagée par un courant trop fort, mais si tu ajoutes une résistance de valeur correcte au circuit pour limiter le courant qui le traverse, la LED est alors protégée. La résistance se mesure en ohms et est représentée par le symbole  $\Omega$ . Tu dois choisir une résistance de valeur correcte pour limiter le courant au travers d'un circuit. La valeur de la résistance est indiquée par des anneaux de couleur qui se lisent de gauche à droite.

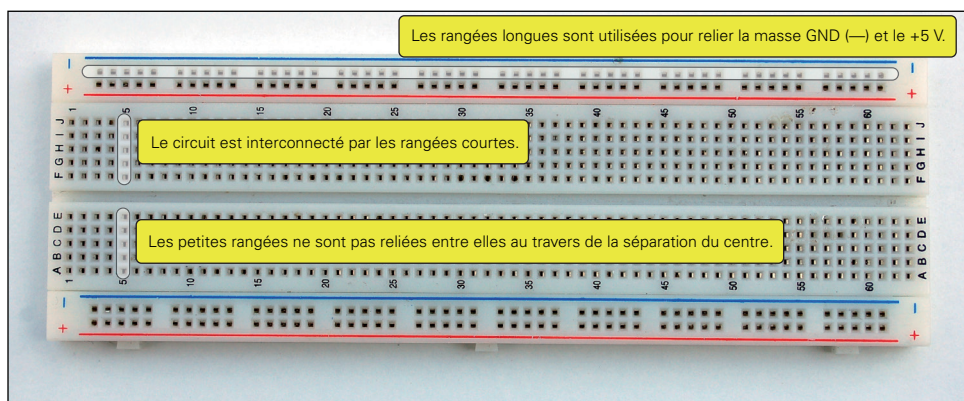


## Utilisation d'une plaque de montage

Maintenant que tu connais le circuit que tu vas réaliser, tu as besoin d'un élément appelé **plaque de montage** pour t'aider à le construire. Une plaque de montage est une plaque qui possède des rangées de trous (voir la figure 1-14). Si tu pouvais voir l'intérieur d'une telle plaque, tu verrais que les trous de chaque rangée font face à la même partie métallique. Relier deux fils dans la même rangée signifie qu'ils sont reliés à la même partie métallique et qu'un courant électrique peut donc les traverser.



Une **plaque de montage** est un composant réutilisable qui permet de créer des circuits sans avoir besoin de souder les composants. Une plaque de montage possède une grande quantité de trous dans lesquels on peut insérer des fils et des composants formant un circuit.

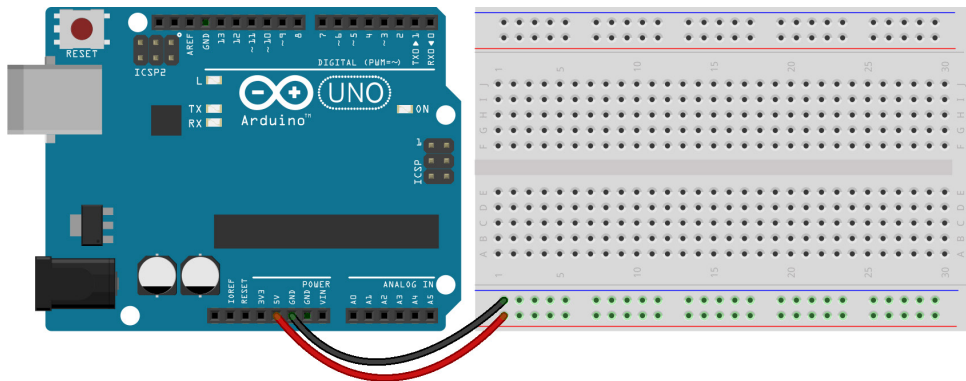


**FIGURE 1-14** Une plaque de montage possède une série de trous qui sont reliés par rangées, dont deux paires de rangées longues près des bords et des rangées perpendiculaires, plus courtes, situées au centre de la plaque.

Les rangées les plus longues, situées sur le bord externe de la plaque, sont celles où tu dois connecter le +5 V et la masse (ou 0 V, GND sur la carte Arduino). Certaines plaques peuvent comporter des inscriptions comme + ou -, ou des couleurs comme du rouge avec du noir ou du bleu. Le rouge représente la tension positive, qui est de 5 V dans le cas de la Uno ; le noir (ou le bleu) représente sa masse GND.

Tu peux considérer les deux rangées longues pour le +5 V et GND comme une extension des broches +5 V et GND de la Uno. Il n'y a qu'une broche +5 V sur l'Arduino Uno, alors que se passera-t-il, si tu as plus d'un composant qui doit y être relié ? En utilisant un fil de pontage, tu pourras relier la broche +5 V à une rangée de la plaque de montage. Tu disposeras ainsi de beaucoup de trous où tu pourras connecter des composants à

cette broche. Il en est de même pour la masse GND. Il y a davantage de broches GND sur l'Arduino Uno (il y en a trois et elles sont identiques, tu peux utiliser n'importe laquelle pour relier les composants à la masse), mais tu peux aussi utiliser un fil de pontage pour augmenter le nombre de trous de connexion à la masse en le connectant à la plaque de montage. Cette façon de faire est représentée sur la figure 1-15.



fritzing

**FIGURE 1-15** Connexion des lignes d'alimentation d'une plaque de montage.

Les rangées situées au milieu de la plaque de montage, entre les grandes rangées des bords, sont celles que tu dois utiliser pour relier tes composants entre eux. Si tu pouvais voir l'intérieur de la plaque, tu verrais que ces rangées sont perpendiculaires aux grandes rangées d'alimentation placées au bord. Les trous sont groupés par cinq et ne sont pas reliés au travers de l'espace au milieu de la plaque.

## Construction de ton premier circuit

Te voilà maintenant prêt à réaliser ton circuit à LED utilisant une résistance de 220 ohms ( $\Omega$ ) et une LED. Une LED possède un sens (on dit qu'elle est polarisée), cela signifie que tu peux, par erreur, la relier à l'envers dans un circuit. Si tu observes une LED de près, tu verras que ses deux fils ne sont pas de la même longueur. Le fil le plus long doit être relié au +5 V du circuit et le plus court à la masse. La résistance, elle, n'est pas polarisée, ses deux fils sont donc de longueur identique et peu importe lequel est relié aux différentes parties du circuit.

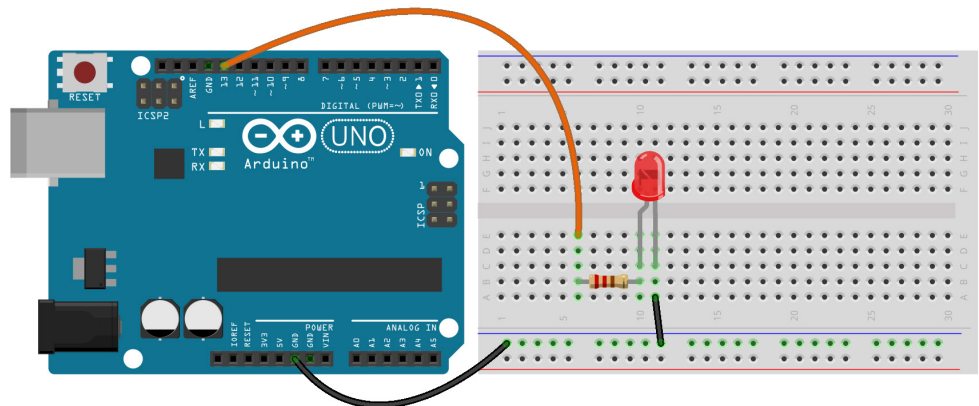
Suis les étapes suivantes pour construire le circuit de la figure 1-16.

1. Relie une extrémité d'un fil de pontage à une des broches GND d'Arduino et l'autre extrémité à une des grandes rangées de la plaque de montage. C'est le fil noir sur la figure 1-16, mais ton fil peut être de n'importe quelle couleur.
2. Insère un fil de la résistance dans une des petites rangées au milieu de la plaque.

3. Insère l'autre fil de la résistance dans une autre petite rangée au milieu de la plaque de montage. Ce doit être une rangée différente de celle utilisée pour le premier fil.
4. Insère le fil le plus long de la LED dans un trou d'une des rangées utilisées par la résistance. Elles sont maintenant reliées à la même partie métallique, le courant électrique pourra donc circuler au travers de la résistance puis dans la rangée de la plaque de montage et dans la LED.
5. Insère le fil le plus court de la LED dans un trou de la grande rangée où le fil de pontage GND est relié (directement ou par l'intermédiaire d'une rangée courte et d'un fil de pontage).
6. Utilise un deuxième fil de pontage pour relier la broche 13 à la petite rangée sur laquelle est relié le fil de la résistance non relié à la LED.



Ne relie jamais la broche +5 V et une broche de masse (GND) ensemble sans intercaler une résistance ou une LED entre elles : cela créerait un court-circuit et pourrait endommager ton Arduino Uno. Si ça arrive par erreur, ton ordinateur s'apercevra probablement que quelque chose d'anormal se passe dans ta carte Arduino et coupera son alimentation électrique vers la carte. Si cela arrive, déconnecte simplement la carte Arduino de l'ordinateur puis reconnecte-la.



fritzing

**FIGURE 1-16** Le circuit à LED sur la plaque de montage.

La LED sur la plaque de montage devrait maintenant clignoter exactement comme le faisait la LED sur la carte Arduino. Félicitations ! Tu as construit ton premier circuit Arduino ! Ton programme contrôle l'électricité et le clignotement de la lumière. Mais ce n'est que le début de tes aventures... Beaucoup d'autres projets sympas t'attendent !

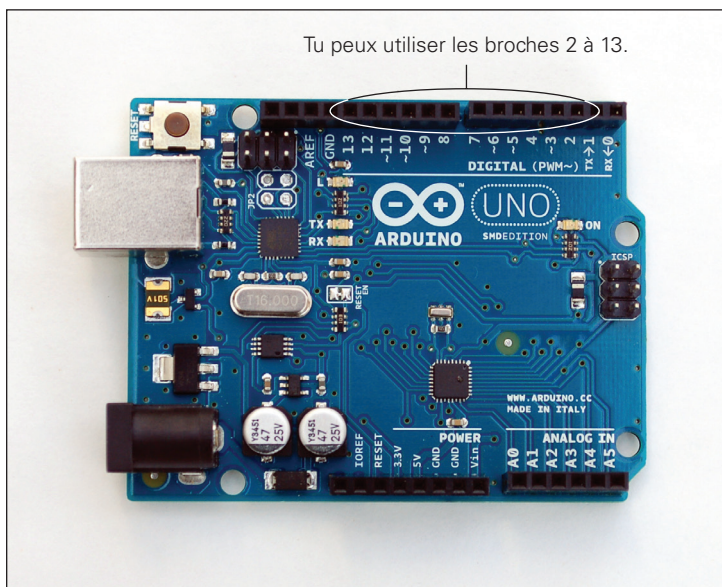
Rends-toi sur le site web d'accompagnement [www.editions-eyrolles.com/go/arduino](http://www.editions-eyrolles.com/go/arduino) pour voir une vidéo (en anglais) montrant comment construire ce circuit.



## DÉFI

Change le numéro de la broche à laquelle est relié ton circuit LED. Tu peux utiliser n'importe laquelle des broches 2 à 13 de la zone Digital, montrée sur la figure 1-17. Les autres broches ont une fonction particulière que tu étudieras dans la prochaine aventure.

Souviens-toi que tu dois faire une modification dans ton code (et téléverser le programme modifié dans ta carte Arduino) et que tu dois aussi modifier ton circuit.



**FIGURE 1-17** Les broches numériques de la carte Arduino. Les broches 0 et 1 sont des broches spéciales que tu étudieras plus tard.

## Plus d'aventures en Arduino

Félicitations ! Tu as été très fort. Tu trouves peut-être que ça représente beaucoup de travail pour simplement faire clignoter une LED, mais en réalité cela signifie que tu es maintenant prêt pour toutes sortes d'aventures.

Regarde les projets que d'autres ont réalisés pour te faire une idée de ce qui est possible :

- <http://makezine.com/category/electronics/arduino/>
- [www.creativeapplications.net/tag/arduino/](http://www.creativeapplications.net/tag/arduino/)

Rappel des commandes Arduino de cette aventure	
Commande	Description
<code>setup()</code>	Fonction qui ne s'exécute qu'une fois au premier démarrage de l'Arduino Uno. Voir aussi <a href="http://arduino.cc/en/Reference/Setup">http://arduino.cc/en/Reference/Setup</a> .
<code>loop()</code>	Fonction qui s'exécute de façon répétitive après que la fonction <code>setup()</code> ait terminé et jusqu'à ce que Arduino soit éteinte. Voir aussi <a href="http://arduino.cc/en/Reference/Loop">http://arduino.cc/en/Reference/Loop</a> .
<code>pinMode()</code>	Définit si la broche dont le numéro est fourni en argument est utilisée en sortie d'électricité ou en entrée. Voir aussi <a href="http://arduino.cc/en/Reference/PinMode">http://arduino.cc/en/Reference/PinMode</a> .
OUTPUT	Mot-clé, second argument de la fonction <code>pinMode()</code> qui indique que la broche va délivrer une tension électrique. Voir aussi <a href="http://arduino.cc/en/Reference/Constants">http://arduino.cc/en/Reference/Constants</a> .
<code>digitalWrite()</code>	Provoque la sortie d'une tension électrique ou sa disparition sur la broche indiquée. Voir aussi <a href="http://arduino.cc/en/Reference/DigitalWrite">http://arduino.cc/en/Reference/DigitalWrite</a> .
HIGH	Mot-clé utilisé pour délivrer une tension électrique dans la fonction <code>digitalWrite()</code> . Voir aussi <a href="http://arduino.cc/en/Reference/Constants">http://arduino.cc/en/Reference/Constants</a> .
LOW	Mot-clé utilisé pour interrompre la sortie d'une tension dans <code>digitalWrite()</code> . Voir aussi <a href="http://arduino.cc/en/Reference/Constants">http://arduino.cc/en/Reference/Constants</a> .
<code>delay()</code>	Provoque une pause de l'Arduino Uno pendant le nombre de millisecondes spécifié en argument. Voir aussi <a href="http://arduino.cc/en/Reference/Delay">http://arduino.cc/en/Reference/Delay</a> .



**Étape franchie :** Tu sais faire les bons branchements maintenant !

### Dans la prochaine aventure

Tu commenceras à ajouter de l'interactivité dans ton code Arduino et à contrôler des LED en utilisant un bouton rotatif !







## Aventure 2

# Utilise des capteurs

**TU AS MAINTENANT** installé et fait fonctionner ton logiciel, et tu y as téléchargé ton premier sketch pour t'assurer que tout était correctement installé. (Si tu ne l'as pas fait, il serait bon que tu retournes à l'aventure 1 et que tu le fasses maintenant !) C'est super de pouvoir dire à Arduino de contrôler quelque chose, comme faire clignoter une LED ; mais c'est encore plus amusant quand les projets deviennent interactifs. Il y a deux choses qui doivent arriver pour qu'un projet le devienne : tout d'abord, Arduino doit savoir ce qui se passe dans son environnement ; ensuite, son programme doit exécuter une action basée sur les informations recueillies.

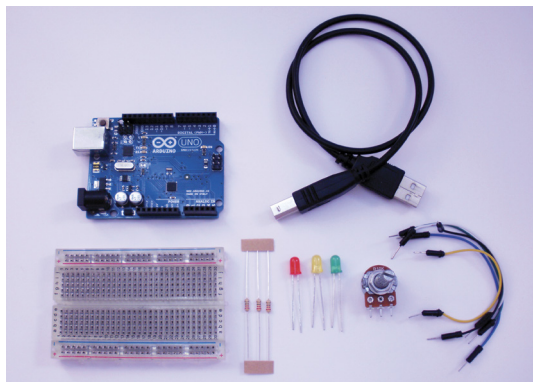
Tu vas parcourir un long chemin dans cette aventure ! Tu vas commencer par contrôler plusieurs LED, puis tu vas afficher des messages envoyés à ton ordinateur depuis la carte Arduino. Ensuite, tu vas lire des informations fournies par un capteur et les envoyer à ton ordinateur pour qu'il les affiche. Pour terminer, tu vas mettre à profit toutes tes connaissances pour réaliser un afficheur de messages effrayants, que tu choisiras à l'aide d'un bouton rotatif. Parfait pour accueillir ou éloigner un visiteur en fonction de ton humeur !

## Ce dont tu as besoin

Tu vas d'abord apprendre à ajouter plusieurs LED à ton circuit puis à utiliser un capteur appelé potentiomètre. La liste suivante indique ce dont tu as besoin, la figure 2-1

montre les composants électroniques. Souviens-toi que l'annexe B contient des suggestions d'endroits où tu peux acheter ces éléments :

- un ordinateur ;
- une carte Arduino Uno ;
- un câble USB ;
- une plaque de montage ;
- 3 LED (une verte, une jaune et une rouge) ;
- un potentiomètre de 10 k $\Omega$  ;
- 4 fils de pontage.



**FIGURE 2-1** Les composants électroniques dont tu as besoin pour la première partie de cette aventure.

## Ajout de plusieurs LED

Dans la première aventure, tu as réalisé un circuit sur une plaque de montage pour contrôler une seule LED avec ta carte Arduino. Une LED c'est bien, plusieurs c'est encore mieux ! De quoi as-tu besoin pour ajouter plusieurs LED ?

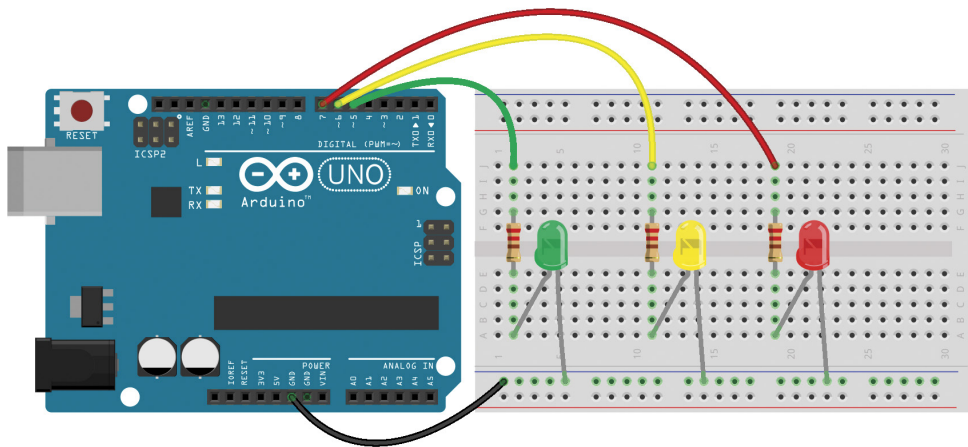
Tu as d'abord besoin de plusieurs LED : cette aventure en utilise trois. Tu as également besoin de trois résistances de 220  $\Omega$ , car chaque LED a besoin d'une résistance pour limiter le courant qui la traverse. Les résistances permettent de contrôler le courant électrique dans un circuit afin de protéger les LED contre la destruction qui résulterait d'un courant trop fort. Tu peux lire plus d'informations au sujet du courant électrique et des résistances dans l'aventure 1.

Commence par construire le circuit présenté sur la figure 2-2.

1. Introduis un fil d'une résistance dans une rangée courte de la moitié haute de la plaque de montage, près du bord gauche de la plaque. Insère l'autre fil de la

résistance dans la rangée courte de la moitié basse de la plaque, dans le prolongement de la rangée où tu as inséré le premier fil de la résistance. Les liaisons électriques des rangées courtes sont interrompues par l'espace au milieu qui les sépare. Ainsi, les deux fils de la résistance que tu viens de brancher ne sont pas reliés électriquement et possèdent chacun leur propre rangée de contacts.

2. Fais de même avec les deux autres résistances. Place la deuxième au centre de la plaque et la troisième vers la droite. Chaque résistance doit chevaucher l'espace de séparation du milieu de la plaque : avoir un fil inséré dans une rangée au-dessus de cette séparation et l'autre fil en dessous.
3. Ajoute maintenant les LED. Le fil le plus long d'une LED doit être relié à une résistance et l'autre fil à la masse (GND). Insère le fil long de chaque LED dans la rangée qui se situe au-dessous d'une résistance. Place la LED verte à gauche, la jaune au centre et la rouge à droite de la plaque.
4. Insère le fil court de chaque LED dans une des grandes rangées de la longueur inférieure de la plaque. Si ta plaque de montage comporte une ligne bleue ou noire ou le signe -, insère les trois fils courts des LED dans cette rangée. Si elle ne possède pas de marquage, assure-toi simplement que les trois fils sont bien insérés dans la même rangée.



fritzing

**FIGURE 2-2** Construction du circuit contrôlant plusieurs LED.

5. Ton circuit est maintenant réalisé sur ta plaque de montage : il ne reste plus qu'à le relier à ta carte Arduino Uno. Utilise un fil de pontage pour relier une broche GND (il y en a trois sur ta carte Arduino Uno, mais tu peux choisir celle que tu veux, car elles sont identiques) à la rangée à laquelle tu as relié les fils courts des trois LED.