

BOULE DE CRISTAL

Créer une boule cristal pour prédire votre avenir

Découverte : Afficheurs LCD, instructions `switch/case`, `random()`

Durée : 1 heure

Difficulté : ■■■■■■

Basé sur les projets : 1,2,3

Une boule de cristal peut vous "aider" à prédire le futur. Vous posez une question à la boule qui sait tout, et vous la retournez pour connaître la réponse. Les réponses seront prédéterminées, mais vous pourrez inscrire ce que vous voudrez. Vous utiliserez votre Arduino pour choisir parmi un total de 8 réponses. L'interrupteur à bascule aidera à simuler le mouvement à donner à la véritable boule pour faire apparaître les réponses.

Les afficheurs LCD peuvent être utilisés pour afficher des caractères alphanumériques. Celui fournit avec votre kit a 16 colonnes et 2 lignes, pour un total de 32 caractères. Il y a un grand nombre de broches sur la plaque de l'afficheur. Ces broches sont utilisées pour l'alimentation et la communication, ainsi l'afficheur sait ce qu'il faut afficher sur l'écran. Mais dans ce projet, vous n'aurez pas besoin de toutes les connecter. Regardez la *Fig.1* pour connaître la liste des broches que vous allez utiliser.

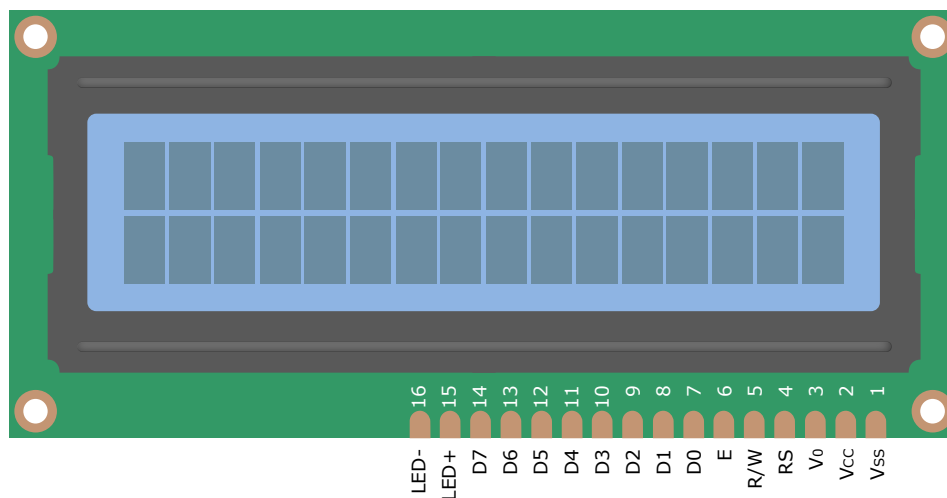


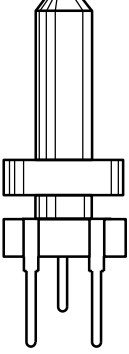

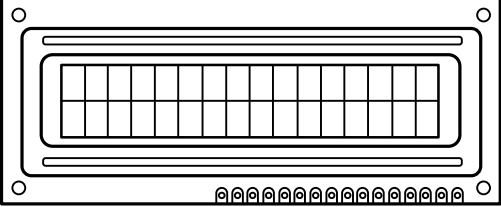


Fig. 1 – Les broches de l'écran LCD qui sont utilisées dans ce projet et leurs libellés

INGREDIENTS

				
Résistance 10 kΩ	Résistance 220 Ω	Potentiomètre	Interrupteur à bascule	Afficheur LCD
x1	x1	x1	x1	x1

CONSTRUIRE LE CIRCUIT

Fig.2 - [Circuit]

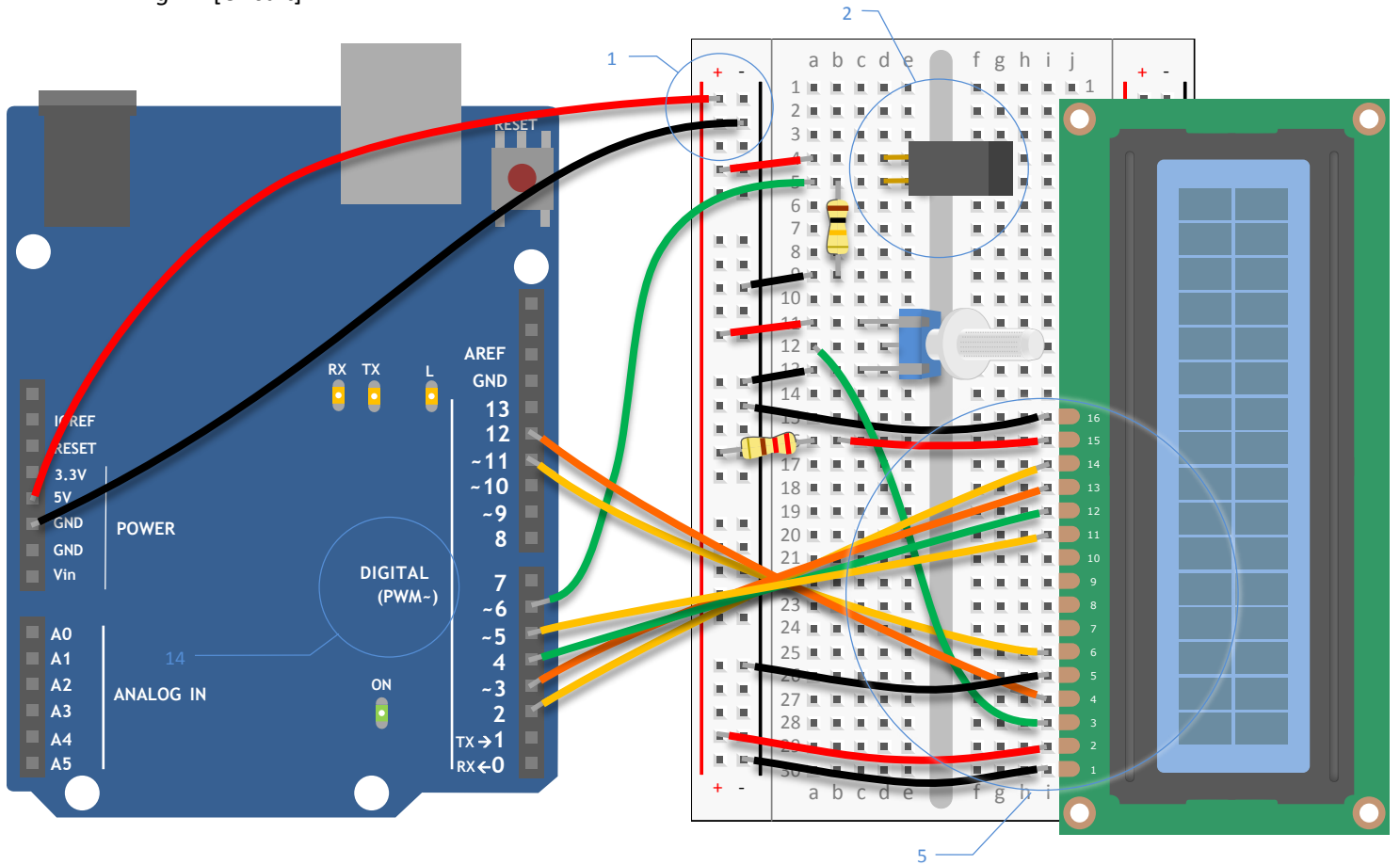
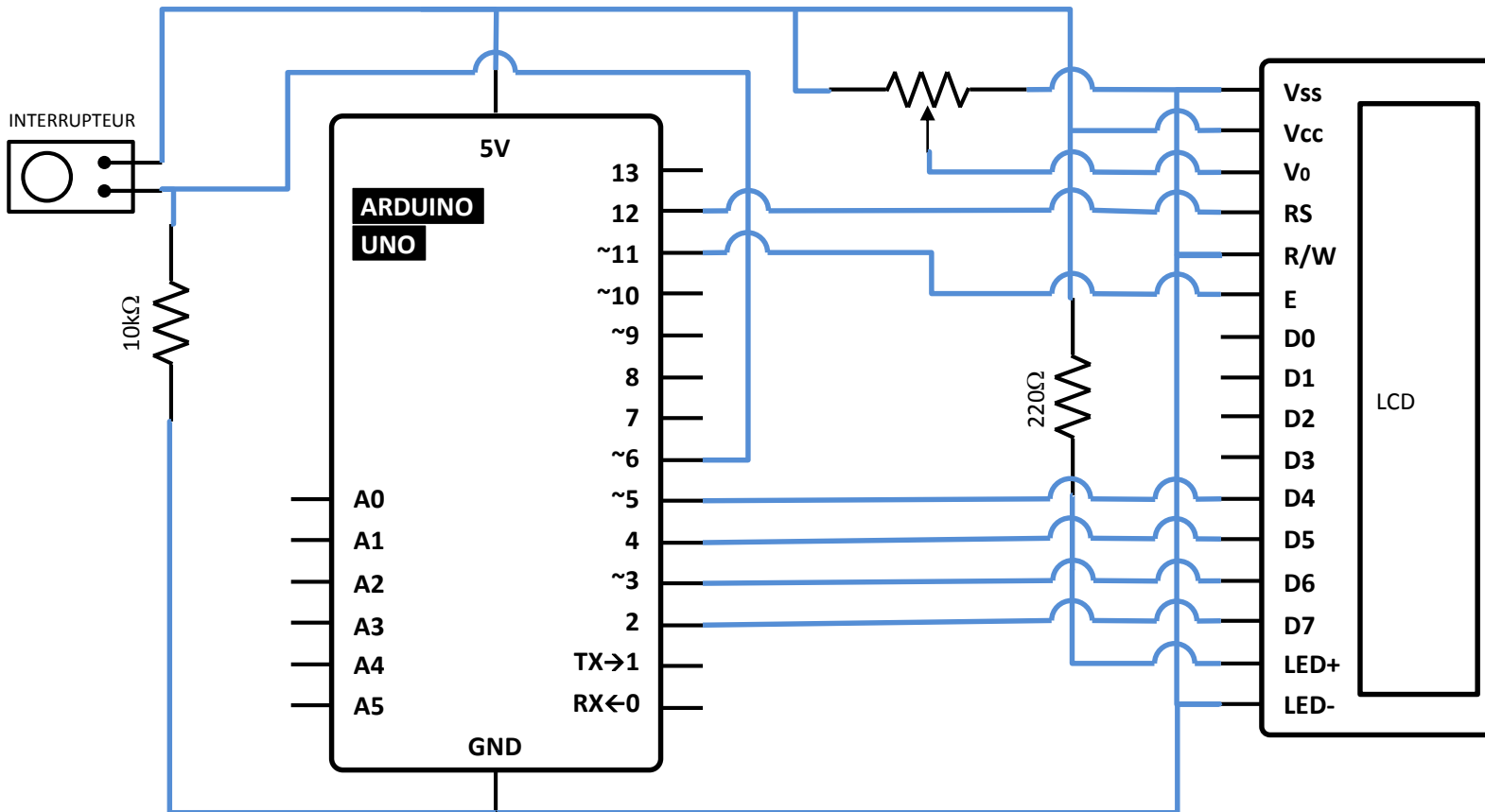


Fig.3 - [Schéma]



Sur un schéma, l'ordre des broches du LCD ne correspond pas nécessairement à l'emplacement physique des broches sur la Fig. 2. Sur un schéma, les broches sont regroupées par groupe logique afin de rendre le schéma aussi clair que possible. C'est un petit peu déstabilisant pour les débutants jusqu'à ce qu'ils en aient l'habitude.

Ce circuit n'est pas extrêmement complexe, mais il y comporte beaucoup de fils. Prêtez une grande attention au câblage afin de vous assurer que tout est correct.

- 1) Branchez l'alimentation et la masse sur un coté de votre platine d'expérimentation.
- 2) Placez l'interrupteur à bascule sur la platine d'expérimentation et connectez une broche au 5V. Connectez l'autre broche d'une part à la masse via une résistance de 10 k Ω , d'autre part à la borne 6 de votre Arduino. Vous la connectez ainsi à une entrée numérique, comme vous l'avez déjà fait dans de nombreux autres projets.
- 3) La broche 'Register Select' (**RS**) (*Selection du registre*) contrôle où les caractères vont apparaître à l'écran. La broche 'Read/Write' (**R/W**) (*Lecture/Ecriture*) place l'écran en mode 'Ecriture' (**Write**) ou 'Lecture' (**Read**). Dans ce projet, vous allez utiliser le mode 'Ecriture'. La broche 'Enable' (**EN**) (*Active*) indique au LCD qu'il va recevoir une commande. Les broches 'Data' (**D0-D7**) (*Donnée*) sont utilisées pour envoyer les données relatives à un caractère. Vous allez utiliser 4 d'entre elles (**D4-D7**). Enfin, il y a une broche qui permet d'ajuster le contraste de l'affichage. Vous allez utiliser un potentiomètre pour le contrôler.
- 4) La bibliothèque **LiquidCrystal** fournie avec le logiciel de votre Arduino s'occupe des opérations d'écriture sur ces broches, et simplifie l'écriture de programmes qui permettent d'afficher des caractères.
Les deux broches aux extrémités du LCD (**Vss** et **LED-**) doivent être connectées à la masse. Connectez aussi la broche **R/W** à la masse. Cela met l'écran en mode 'Ecriture'. La broche d'alimentation du LCD (**Vcc**) doit être connectée directement au 5V. La broche LED+ de l'afficheur doit être connectée à l'alimentation via une résistance de 220 Ω .
- 5) Connectez
 - la borne numérique **2** de l'Arduino à la broche **D7** du LCD,
 - la borne numérique **3** de l'Arduino à la broche **D6** du LCD,
 - la borne numérique **4** de l'Arduino à la broche **D5** du LCD,
 - la borne numérique **5** de l'Arduino à la broche **D4** du LCD,Ce sont les broches de donnée qui indiquent à l'écran quel caractère doit être affiché.
- 6) Connectez la broche **EN** de l'écran à la borne **11** de votre Arduino. La broche **RS** du LCD se connecte à la borne **13** de l'Arduino. Cette broche permet l'écriture sur le LCD.
- 7) Placez le potentiomètre sur la platine d'expérimentation, en connectant les broches situées dans les coins, l'une à l'alimentation et l'autre à la masse. La broche centrale doit être connectée à la broche **V0** sur le LCD. Cela permettra de régler le contraste sur l'écran LCD.

LE PROGRAMME

Déclaration de la bibliothèque LiquidCrystal

Tout d'abord, vous avez besoin d'importer la bibliothèque `LiquidCrystal`.

Ensuite, vous allez initialiser la bibliothèque, de manière similaire à ce que vous avez fait précédemment avec la bibliothèque `Servo`, en lui indiquant quelles broches seront utilisées pour communiquer.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

Maintenant que vous avez configuré la bibliothèque, il est temps de créer quelques variables et constantes. Créez une constante pour stocker le numéro de broche de l'interrupteur, une variable pour l'état courant de l'interrupteur, une variable pour l'état précédent de l'interrupteur, et une de plus pour choisir quelle réponse devra être affichée à l'écran.

```
const int brocheInterrupteur = 6;
int etatInterrupteur = 0;
int etatInterrupteurPrecedent = 0;
int reponse = 0;
```

Ecrivez votre première ligne

Déclarez la broche de l'interrupteur grâce à `pinMode()` dans votre `setup()`. Démarrez la bibliothèque LCD, et indiquez les dimensions de l'écran.

```
void setup() {
  lcd.begin(16, 2);
  pinMode(brocheInterrupteur, INPUT);
}
```

Documentation de référence sur la bibliothèque LCD: <http://www.arduino.cc/lcdlibrary>

Déplacez le curseur

Maintenant il est temps d'écrire un petit message d'introduction vous invitant à utiliser votre boule de cristal, telle une véritable "Boule n°8". La fonction `print()` écrit sur l'écran LCD.

Vous allez écrire les mots "Interrogez la" sur la ligne du haut de l'écran. Le curseur est automatiquement positionné en début de ligne.

```
lcd.print("Interrogez la");
```

Afin d'écrire sur la ligne suivante, vous devez indiquer à l'écran où doit être placé le curseur. Les coordonnées de la première colonne sur la seconde ligne sont (0,1) (Rappelons que les ordinateurs sont indexés à partir de 0. (0,0) sont les coordonnées de la première colonne de la première ligne). Utilisez la fonction `lcd.setCursor()` pour déplacer le curseur au bon endroit, et dites que vous voulez écrire "Boule de Cristal".

```
lcd.setCursor(0, 1);
lcd.print("Boule de Cristal");
```

Maintenant, lorsque vous démarrez le programme, l'écran affiche "Interrogez la Boule de Cristal".

Défiez le hasard

La séquence aléatoire générée par Arduino, bien que très longue, et aléatoire, est toujours la même. L'instruction `randomSeed()` initialise le générateur de nombre pseudo-aléatoire, permettant de débiter la séquence aléatoire en n'importe quel endroit. Sans cette instruction, la boule de cristal donnerait invariablement les mêmes réponses dans le même ordre après chaque remise à zéro (RESET ou mise sous tension de Arduino).

```
randomSeed(analogRead(0));
```

```
}
```

Dans la boucle **loop()**, vous allez commencer par vérifier la position de l'interrupteur, et mettre la valeur dans la variable **etatInterrupteur**.

```
void loop() {  
    etatInterrupteur = digitalRead(brocheInterrupteur);
```

Choisissez une réponse au hasard

Utilisez une instruction **if()** (*si*) afin de déterminer si l'interrupteur est dans une position différente que précédemment. Si la position n'est plus la même qu'avant, et qu'elle est actuellement à **LOW** (*bas*), alors c'est le moment de choisir une réponse au hasard.

La fonction **random()** (*aléatoire*) renvoie un nombre basé sur la valeur de l'argument que vous lui avez fourni. Pour commencer, vous aurez au total 8 réponses différentes pour votre boule. Lorsque l'instruction **random(8)** est appelée, elle retournera une valeur comprise entre 0 et 7. Stockez cette valeur dans la variable **reponse**.

```
if (etatInterrupteur != etatInterrupteurPrecedent) {  
    if (etatInterrupteur == LOW) {  
        reponse = random(8);
```

Documentation de référence sur Random: <http://www.arduino.cc/random>

Effacez l'écran grâce à la fonction **lcd.clear()**. Cela remplace aussi le curseur à sa position initiale (0,0); la première colonne sur la première ligne de l'écran LCD. Ecrivez sur la ligne "La boule répond:" et déplacez le curseur sur la ligne suivante pour écrire la réponse.

```
lcd.clear();  
lcd.setCursor(0, 0); // Déjà fait par la fonction lcd.clear()  
lcd.print("La boule répond:");  
lcd.setCursor(0, 1);
```

Prédisez le futur

L'instruction **switch()** (*aiguillage*) exécute différents blocs de code en fonction de la valeur que vous lui donnez. Chacun de ces blocs de code est appelé **case** (*cas*). **switch()** vérifiera le contenu de la variable **reponse**; La valeur contenue dans **reponse** déterminera quel bloc d'instruction **case** sera exécuté.

A l'intérieur des blocs **case** le code sera le même, seuls les messages seront différents. Par exemple, dans le cas 0 le code sera **lcd.print("Oui")**. Après la fonction **lcd.print()**, il y a une autre commande : **break** (*interruption*). Elle indique à Arduino où se situe la fin du bloc **case**. Lorsqu'il rencontre l'instruction **break**, il saute directement à la fin de l'instruction **switch()**. Pour commencer, vous allez créer 8 cas, donc 8 réponses. Quatre de ces réponses seront positives, 2 seront négatives, et enfin 2 vous demanderons de réessayer.

```
switch(reponse) {  
    case 0:  
        lcd.print("Oui");  
        break;  
    case 1:  
        lcd.print("Plutôt");  
        break;  
    case 2:  
        lcd.print("Certainement");  
        break;  
    case 3:  
        lcd.print("Cela semble bon");  
        break;  
    case 4:  
        lcd.print("J'ai un doute...");  
        break;
```

```
        case 5:
            lcd.print("Redemandez !");
            break;
        case 6:
            lcd.print("Surement pas !");
            break;
        case 7:
            lcd.print("Non");
            break;
    }
}
}
```

La dernière chose que vous ferez dans la **loop()** sera d'assigner la valeur de **etatInterrupteur** à la variable **etatInterrupteurPrecedent**. Cela vous permettra de déceler un éventuel changement d'état de l'interrupteur à la prochaine exécution de la boucle **loop()**.

```
    etatInterrupteurPrecedent = etatInterrupteur ;
}
```

Documentation de référence sur Switch Case : <http://www.arduino.cc/switchcase>

UTILISEZ LE MONTAGE

Pour utiliser la boule magique, allumez Arduino. Vérifiez l'écran afin de vous assurer que le message "Interrogez la Boule de Cristal" est bien affiché. Si vous ne pouvez pas voir ces caractères, essayez de faire varier le potentiomètre. Cela permettra d'ajuster le contraste de l'écran.

Posez alors une question à votre boule de cristal, et faites basculer votre platine d'expérimentation vers le bas, puis remettez votre montage dans sa position initiale. Vous devriez obtenir une réponse à votre question. Si cette réponse ne vous convient pas, demandez encore !



Essayez d'insérer vos propres phrases dans les instructions **print()**, mais gardez bien à l'esprit que vous ne pourrez afficher que 16 caractères par ligne. Vous pouvez aussi essayer d'ajouter plus de réponses. Lorsque vous ajouterez d'autres **case** dans le **switch**, vous devrez aussi ajuster le nombre qui sera retourné au hasard par la fonction **random** dans la variable **reponse**.



Les écrans LCD fonctionnent en modifiant les propriétés électriques d'un liquide prisonnier entre 2 plaques de verre polarisé. Ce verre ne permet qu'à un certain type de lumière de passer au travers. Lorsque le liquide pris entre les plaques de verre est chargé, il passe dans un état semi-solide. Ce nouvel état est ordonné et orienté dans une direction différente de celle des verres polarisés, empêchant la lumière de passer au travers, créant ainsi, point par point, les caractères que vous voyez à l'écran.



Les fonctions utilisées ici pour modifier le texte sur l'écran LCD sont vraiment simples. Lorsque vous aurez compris comment elles fonctionnent, allez jeter un œil aux autres fonctions de la bibliothèque. Essayer de faire défiler le texte, ou de le modifier continuellement. Pour trouver de plus amples informations sur le fonctionnement de la bibliothèque **LiquidCrystal**, visitez la page : <http://www.arduino.cc/lcd>

*Un affichage LCD permet d'afficher du texte à l'écran, en utilisant la bibliothèque **LiquidCrystal**. Les instructions **switch...case** contrôlent l'exécution de parties de programme en comparant le contenu d'une variable à des valeurs prédéfinies.*

EXTRAIT DU LIVRE DE PROJETS ARDUINO

EDITEURS

Projets et texte par Scott Fitzgerald et Michael Shiloh

Revue de texte complémentaire par Tom Igoe

DESIGN ET DIRECTION ARTISTIQUE

TODO

Giorgio Olivero, Mario Ciardulli, Vanessa Poli, Michelle Nebiolo

todo.to.it

EDITION NUMERIQUE ET GESTION DE PROJET

Officine Arduino Torino

Katia De Coi, Enrico Bassi

CONSEILLERS ET SUPPORTERS

Massimo Banzi, Gianluca Martino, Smart Projects

TESTEURS DES PROJETS ET RELECTEURS

Michael Shiloh, Michelle Nebiolo, Katia De Coi, Alessandro Buat, Frederico Vanzati, David Mellis

REMERCIEMENTS

Un grand merci à toute la communauté des utilisateurs Arduino pour leurs contributions continues, leur soutien, et leurs retours.

Nous remercions particulièrement l'équipe Fritzing: Quelques-unes des illustrations de composants électroniques utilisés dans le livre sont issues ou inspirées par le projet open-source de Fritzing (www.fritzing.org).

Un grand merci à Paul Badger pour la bibliothèque *CapacitiveSensor* utilisée dans le projet 13.

Le texte du Livre de Projets Arduino est distribué sous licence Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License 2012 par Arduino LLC. Cela signifie que vous pouvez copier, réutiliser, adapter et vous appuyer sur le texte de ce livre en citant l'œuvre originale (mais pas d'une manière qui suggérerait que nous souscrivons à vous ou votre utilisation de l'œuvre) et seulement si le résultat de votre travail est transmis sous la même licence Creative Commons.

Les termes complets de la licence sont disponibles à : creativecommons.org/licenses/by-nc-sa/3.0/

© 2012 Arduino LLC. Le nom *Arduino* et le logo sont des marques de Arduino, déposées aux États-Unis et dans le reste du monde. Les autres noms de produits et de sociétés mentionnés dans ce document sont des marques commerciales de leurs sociétés respectives.

Les informations contenues dans ce livre sont distribuées «telles quelles» sans aucune garantie supplémentaire. Bien que toutes les précautions aient été prises dans la conception de ce livre, ni les auteurs ni Arduino LLC ne pourraient endosser une quelconque responsabilité envers toute personne ou entité à l'égard de toutes pertes ou dommages causés ou déclarés causés directement ou indirectement par les instructions contenues dans ce livre ou par le logiciel et le matériel qu'il décrit.

Ce livre ne peut être vendu séparément du 'Kit de Démarrage Arduino'.

Conçu, imprimé et relié à Turin, Italie

Septembre 2012

Première réimpression, Décembre 2012

Traduit de l'anglais par Nicolas PONCET