



SUPER STARTER KIT

MODE D'EMPLOI POUR L'UNO

Avant-propos

Notre entreprise

Notre entreprise, Elegoo Inc. établi en 2011, est un fabricant et exportateur professionnel qui s'occupe à la conception, au développement, à la production et à la commercialisation d'Arduino, imprimantes 3D, Framboise Pi et STM32. Nous sommes situés à Shenzhen, une ville connue sous le nom de Silicon Valley chinois. Tous les produits fabriqués par notre entreprise sont conformes aux normes de qualité internationales et sont grandement appréciés dans une variété de différents marchés dans le monde entier.

Site officiel: <http://www.elegoo.com>

Vitrine à l'Amazon FR: <http://www.amazon.fr/shops/A1780XYQ9DFQM6>

Notre mode d'emploi

Ce mode d'emploi est pour les débutants. Vous pouvez apprendre toutes les informations de base sur l'utilisation de la carte de contrôleur Arduino, des capteurs et des composants. Si vous voulez apprendre Arduino plus en profondeur, nous vous remercions de lire Arduino Cookbook écrit par Michael Margolis.

Notre Service

Si vous avez n'importe quelle question, commentaire ou suggestion sur notre entreprise, nos produits ou notre mode d'emploi, veuillez ne pas hésiter à envoyer un email à EUservice@elegoo.com. Votre rétroaction est inestimable, afin que nous nous engagions à une amélioration continue.

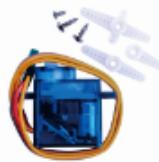
Liste des colisage

www.elegoo.com

Moteur pas-à-pas



Servo (SG90)



Module de récepteur IR



Elegoo UNO R3
microcontrôleur



Relais



Écran LCD1602



Module de conducteur
du moteur pas-à-pas
ULN2003



Module d'alimentation
de la Carte de prototypage



Prototype carte d'extension



Contactez nous: Euservice@elegoo.com

 www.elegoo.com

Affichage de 4-chiffres
7-segments



Affichage de segment



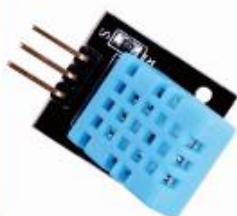
Contacteur à bille



6v DC moteur



Module DHT11



Module de capteur
à ultrasons



Contactez nous: Euservice@elegoo.com

www.elegoo.com

Commutateurs à bouton-poussoir
5 pcs



Potentiomètre



Buzzer passif



batterie 9v et le connecteur



Buzzer Actif



Module de Joystick



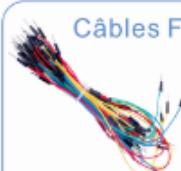
USB câble



Câbles F-M 10 pcs



Câbles F-F 65 pcs



Télécommande infrarouge



Carte de prototypage

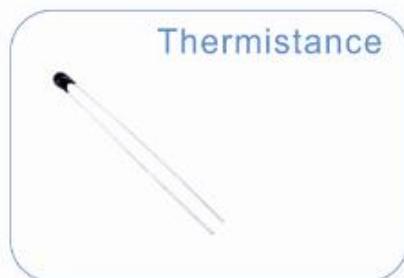


74HC595 IC



Contactez nous: Euservice@elegoo.com

www.elegoo.com



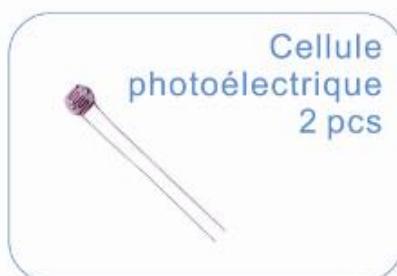
Thermistance



Diode redresseur
2pcs



LED
25 pcs



Cellule
photoélectrique
2 pcs



Résistance
100pcs



RGB
1PC



NPN Transistor
(PN2222)
2 pcs

Contactez nous: Euservice@elegoo.com

Table des matières

Leçon 0 : Installation d'IDE	8
Leçon 1: Ajouter les Bibliothèques.....	17
Leçon 2: Clignotant.....	22
Leçon 3: LED	29
Leçon 4: RGB LED	36
Leçon 5: Entrées Numériques	43
Leçon 6: Huit LEDs avec 74HC595	47
Leçon 7: Moniteur Sériel	54
Leçon 8: Cellule photoélectrique	60
Leçon 9: Prise de sons	64
Leçon 10: Buzzer Passif	68
Leçon 11: Contacteur à bille.....	71
Leçon 12: Relais.....	74
Leçon 13: 74HC595 et Affichage de segment	77
Leçon 14: Affichage de 4-chiffres 7-segments	82
Leçon 15: Servo	83
Leçon 16: Écran LCD	89
Leçon 17: Thermomètre.....	94
Leçon 18: Module de capteur à ultrasons.....	98
Leçon 19: Capteur de température et d'humidité DHT11	102
Leçon 20: Module de joystick analogique.....	107
Leçon 21: Module de récepteur IR.....	110
Leçon 22: Moteurs à courant continu.....	114
Leçon 23: Moteur pas-à-pas.....	119
Leçon 24: Contrôler le moteur pas-à-pas avec la télécommande	121

Leçon 0 : Installation d'IDE

Introduction

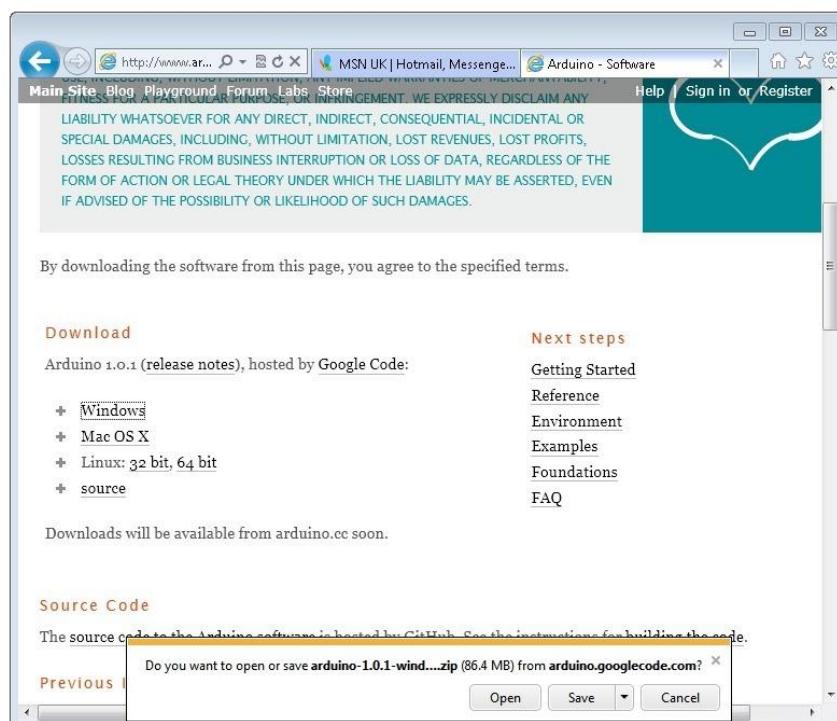
Vous allez apprendre à configurer votre ordinateur pour pouvoir utiliser Arduino et comment suivre les leçons dans cette leçon. Le logiciel Arduino qui sera utilisé pour programmer vos Arduino est disponible pour Windows, Mac et Linux. C'est différent sur le processus d'installation pour les trois plates-formes et, malheureusement, il existe une certaine quantité de travail manuel pour installer le logiciel. Il n'existe pas de programme d'installation; par contre, vous devez décompresser le fichier zip avec le dossier d'Arduino composant le programme d'Arduino et quelques autres éléments.

Dans une étape distincte, vous devez alors installer les pilotes USB.

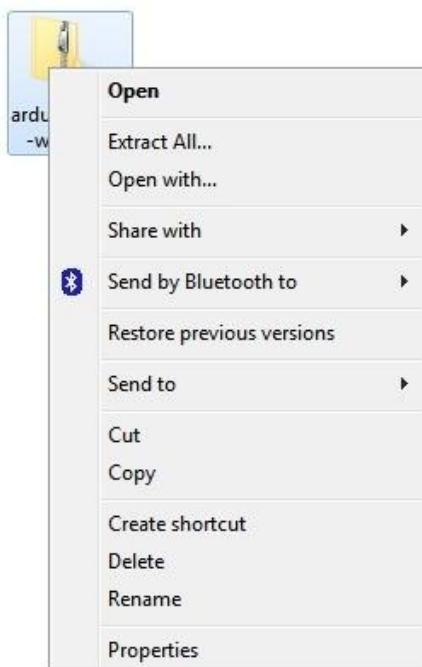
Installation d'Arduino (Windows)

Vous pourrez commencer par la visite du site web Arduino.cc. En avril 2014, nous vous suggérons d'utiliser v1.05 que 1.5 est encore en version bêta. Si la version 1.5 n'est plus en version bêta au moment où vous lisez ce mode d'emploi, être pouvez l'essayer!

Téléchargez le fichier zip tout d'abord pour Windows. Il n'y a qu'une version de logiciel pour Windows XP à Windows 7.



Lorsque le fichier zip est téléchargé, pour extraire le contenu sur le bureau, cliquez le bouton droit sur le fichier et sélectionnez "Extract All..." dans le menu pop-up.



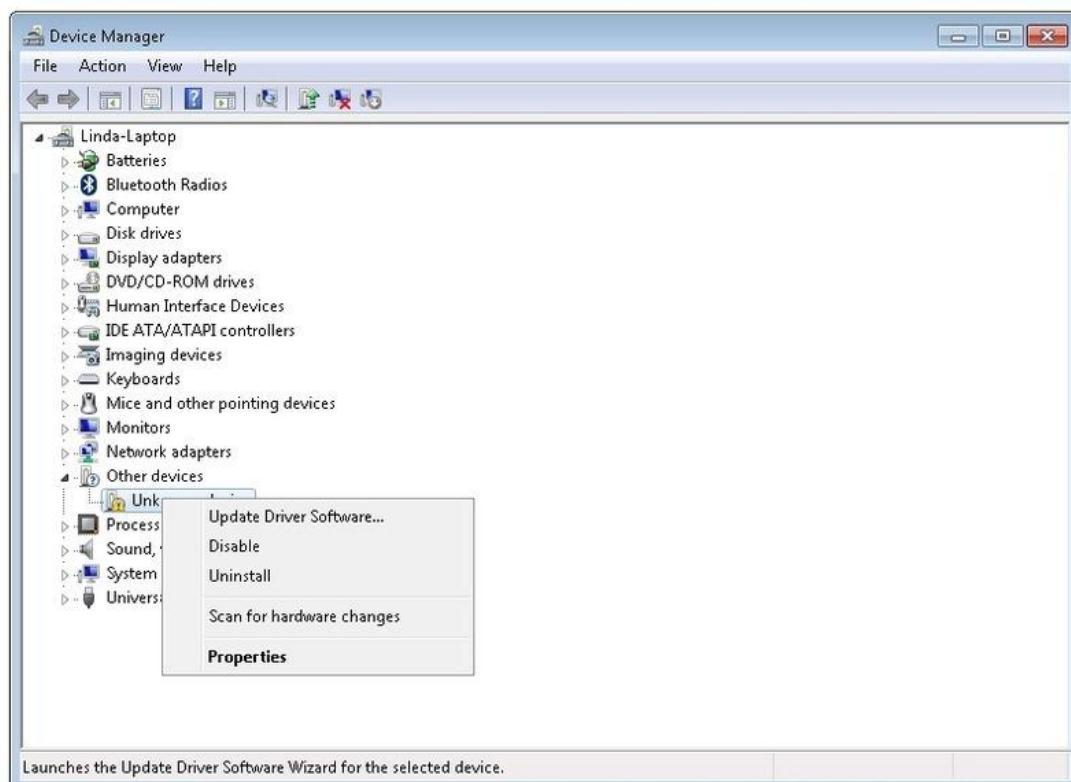
Et puis, sélectionnez votre Bureau et cliquez "Extract". Vous pouvez déplacer les articles extraits à un autre endroit sur votre ordinateur plus tard, juste en déplaçant le dossier, mais pour l'instant, il suffit de garder sur le bureau.

Dans le dossier Arduino, il existe non seulement le programme d'Arduino lui-même, mais aussi les pilotes permettant l'Arduino à être connecté à votre ordinateur par un câble USB. Avant de lancer le logiciel d'arduino, vous devez installer les pilotes USB.

Insérez une extrémité du câble USB à l'Arduino et l'autre exterminé à la prise USB dans votre ordinateur. Le voyant d'alimentation sur LED s'allume et il s'affichera un message "Found New Hardware" sur la fenêtre. Ignorer ce message et annuler toutes les tentatives que le système d'exploitation Windows faire essayer et installer automatiquement les pilotes pour vous.

La méthode la plus fiable d'installer les pilotes USB est d'utiliser le Gestionnaire de Périphériques. C'est possible d'avoir différentes manières basées sur votre version de Windows. Pour Windows 7, vous devez tout d'abord ouvrir Panneau de Contrôle, et puis sélectionner l'option pour afficher les icônes, et vous devrait trouver Gestionnaire de Périphériques dans la liste.

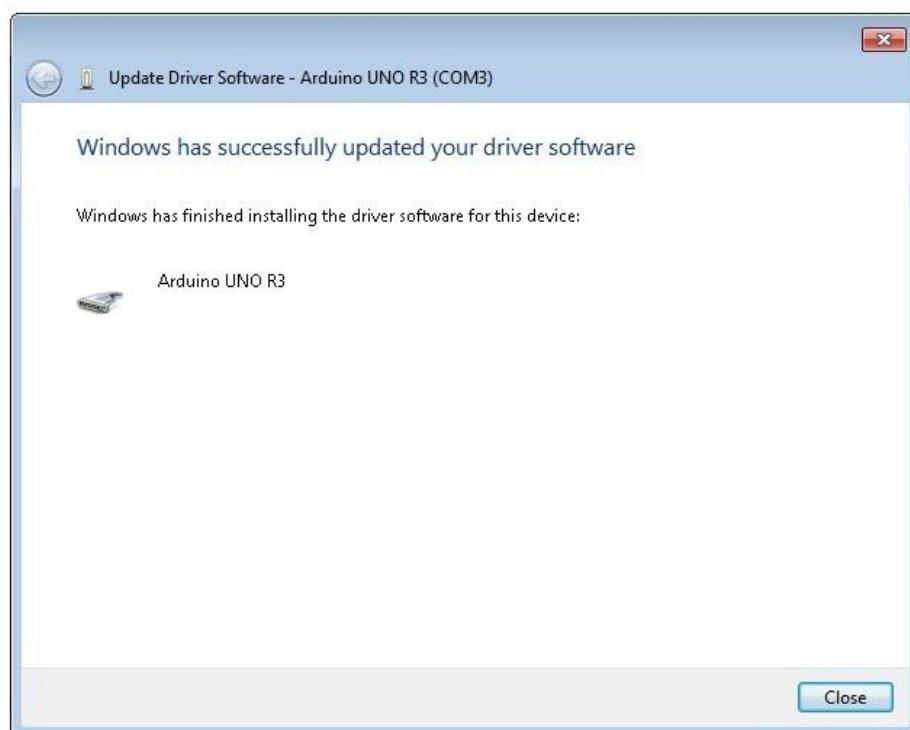
Sous 'Other Devices', vous pouvez trouver une icône pour 'unknown device' avec un petit triangle d'avertissement jaune à côté de lui. C'est votre Arduino.



Faites un clic droit sur le périphérique et sélectionnez l'option de menu supérieur (Update Driver Software...). Alors le système vous invitera de sélectionner soit "Search Automatically for updated driver software" ou 'Browse my computer for driver software'. Sélectionnez l'option pour parcourir et naviguer jusqu'à arduino-1.0.2-windows\arduino1.0.2\drivers.



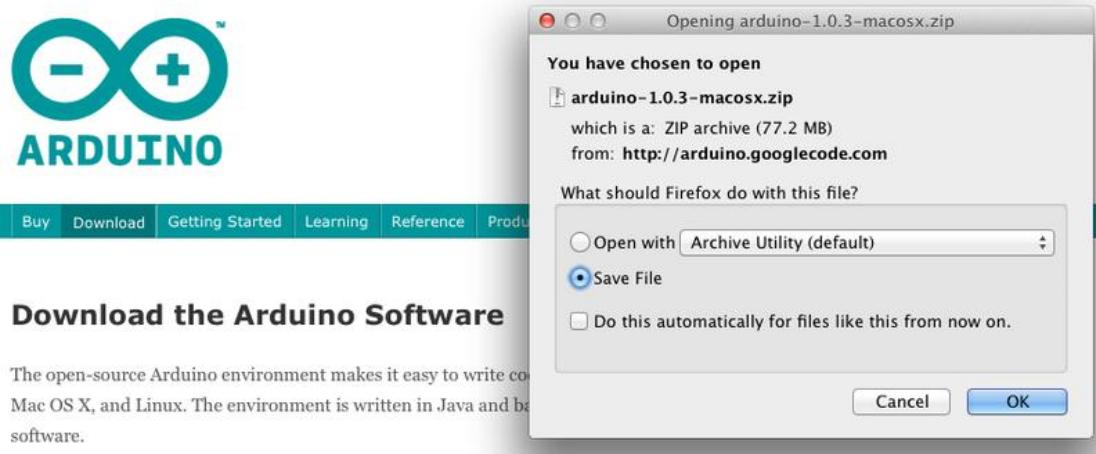
Cliquez sur le bouton 'Next' et un message d'avertissement sécuritaire s'affiche, le cas échéant, permettez le logiciel à installer. Lorsque l'installation du logiciel est achevée, un message de confirmation s'affiche.



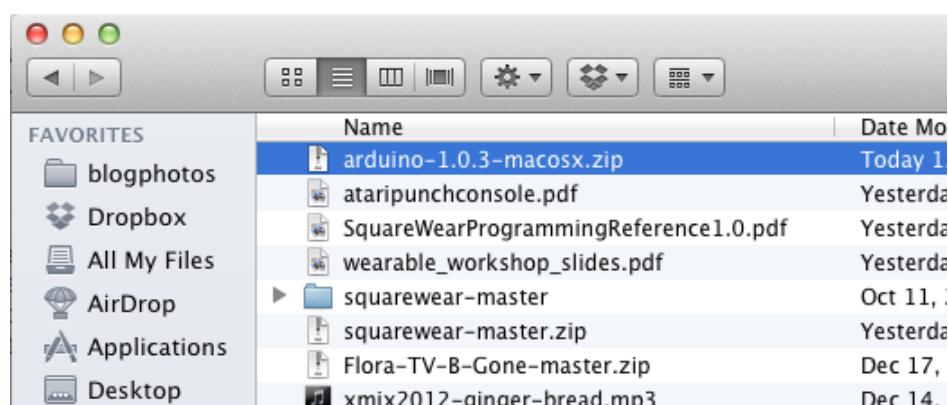
C'est ça. Actuellement vous êtes prêt pour l'opération, alors Sauter à la section prochaine de l'installation sur Mac et Linux et passer directement à "Boards and Ports".

Installation d'Arduino (Mac et Linux)

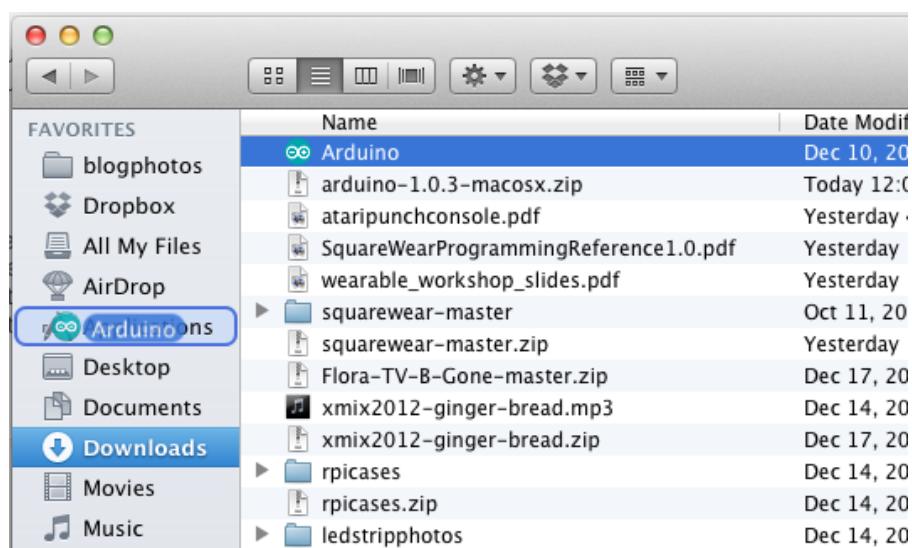
Vous pouvez commencer par la visite du site web Arduino.cc et télécharger l'IDE correspondant pour votre système d'exploitation. En avril 2014, nous vous suggérons d'utiliser v1.05 que 1.5 est encore en version bêta. Si la version 1.5 n'est plus en version bêta au moment où vous lisez ce mode d'emploi, être pouvez l'essayer!



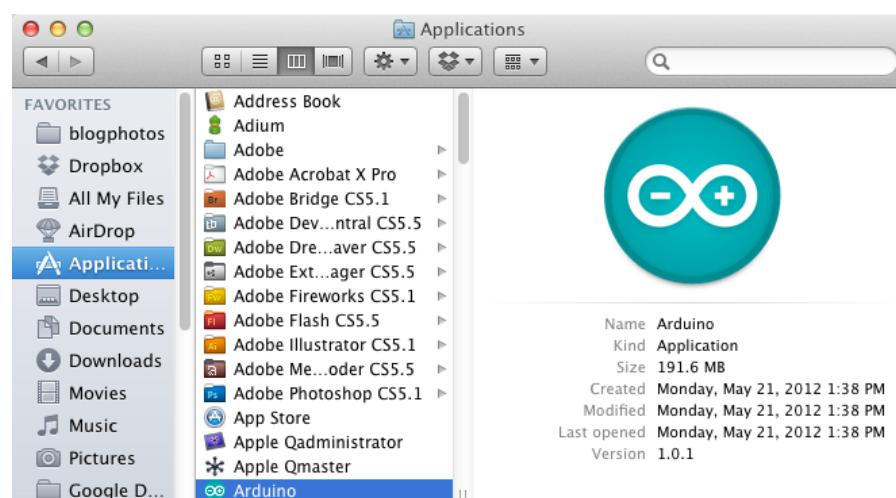
Enregistrez le logiciel d'installation sur votre ordinateur, soit au bureau ou n'importe où.

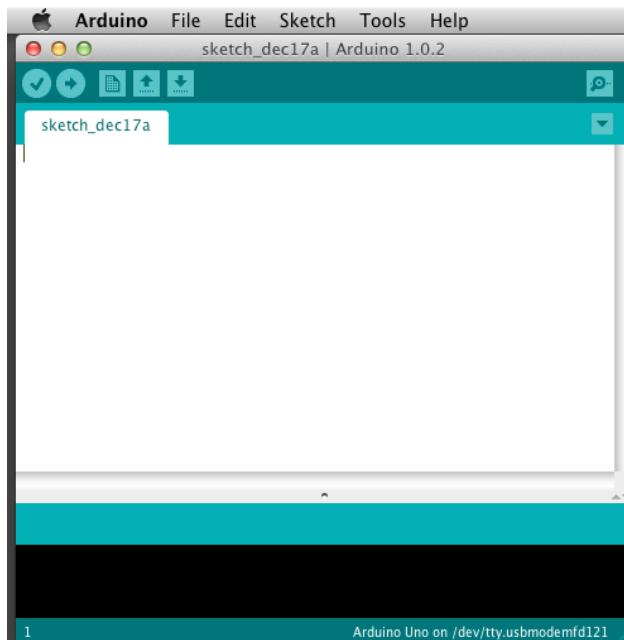


Le processus pour installer le logiciel d'arduino sur Mac est beaucoup plus facile que sur PC. Comme auparavant, la première étape est à télécharger le fichier. Pour Mac, c'est un fichier zip.



Lorsque le fichier zip est téléchargé, double-cliquez sur le fichier zip, où il va extraire un seul fichier 'Arduino.app'. Il s'agit d'une application complète de l'Arduino; juste faites-le glisser dans votre dossier Applications.



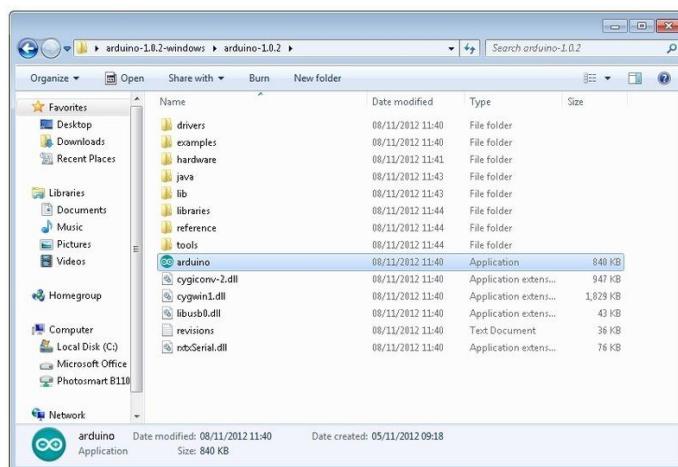


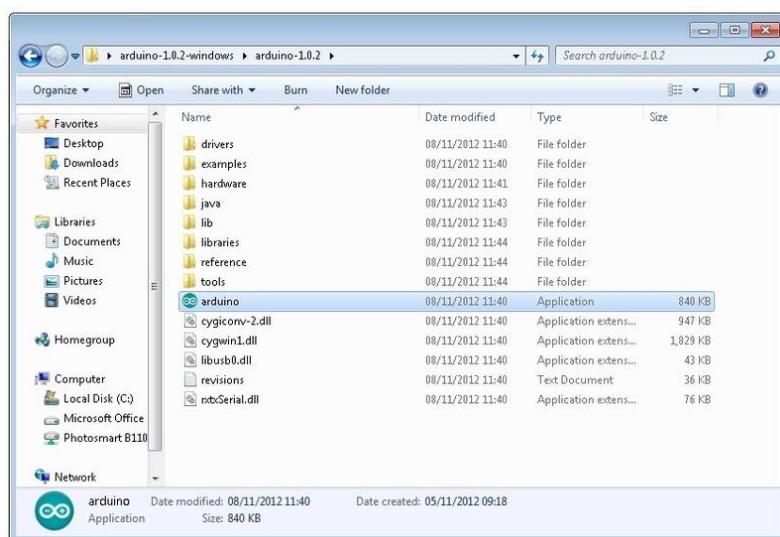
Alors, on peut trouver et lancer le logiciel d'Arduino dans votre dossier d'Applications. Lorsque vous voulez l'utiliser fréquemment, vous pouvez faire un clic droit sur son icône dans le dock pour le mettre à "Keep In Dock".

Il existe de nombreuses distributions de Linux et les instructions pour chaque distribution sont un peu différentes. La communauté Arduino a fait un excellent travail en regroupant des séries d'instructions pour chaque distribution.

Cartes et Ports

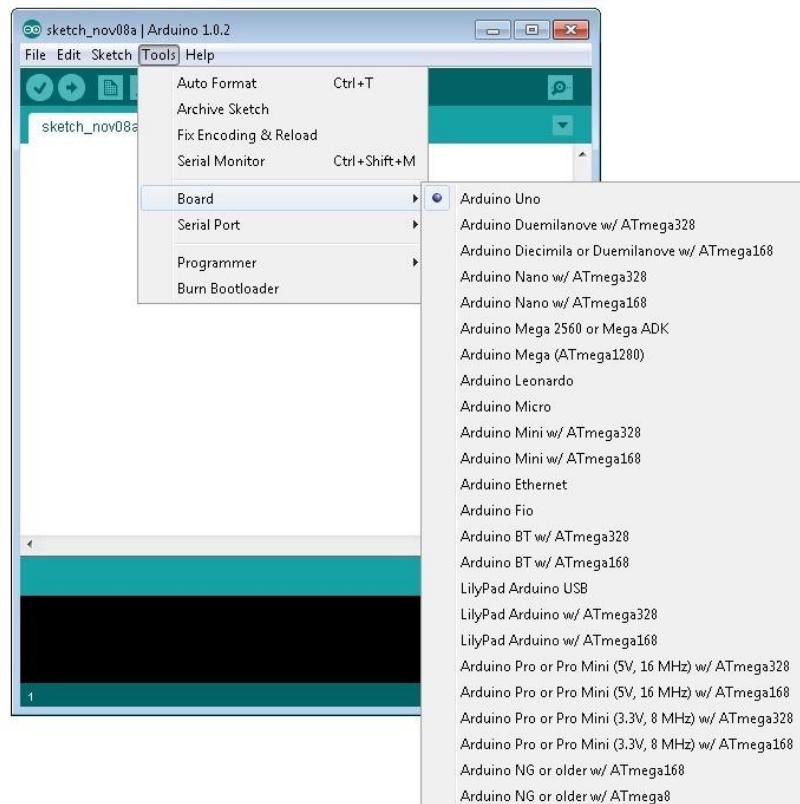
Actuellement, il est prêt de démarrer le logiciel d'Arduino, You are now ready to start the Arduino Software, quels que soient la plate-forme que vous utilisez, ouvrez le dossier Arduino et ouvrez l'application Arduino contenue dedans.



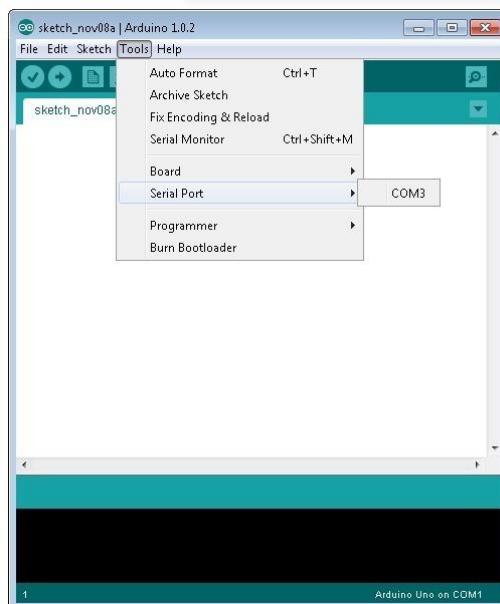


Il va démarrer l'IDE d'Arduino, mais, avant de lancer le programme, vous devez indiquer au logiciel d'Arduino quel type de carte d'Arduino vous voulez utiliser, ainsi sélectionner le port à brancher.

Pour indiquer à l'IDE d'Arduino quel type de carte vous voulez utiliser, sélectionnez Board du menu "Tools" et puis "Arduino Uno" ou "Leonardo" selon le cas.

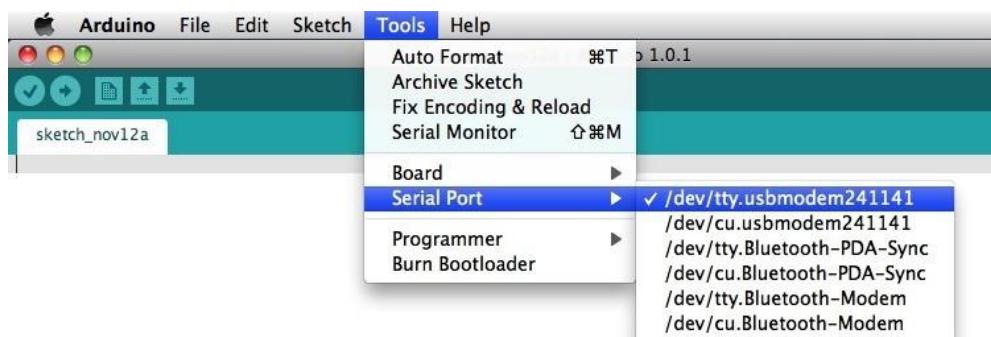


En plus, dans le menu 'Tools', vous pouvez trouver l'option 'Serial Port'. Sélectionnez cette option.



Lors d'utiliser Windows, c'est possible d'avoir une seule option ici et c'est-à-dire COM3 ou COM4. Même si une option, on doit la sélectionner.

Pour Mac ou Linux, c'est possible d'avoir plusieurs options, mais il est généralement la première option dans la liste, puisqu'il peut être le périphérique le plus récemment branché. C'est utile, parce que la dénomination du port ne peut pas ressembler à l'emploi d'Arduino. Il est possible d'être appelé comme /dev/tty.usbmodemXXXX or /dev/ttyUSBn.



Leçon 1: Ajouter les Bibliothèques

Lorsque vous êtes à l'aise avec le logiciel Arduino et en utilisant les fonctions intégrées, vous pouvez vouloir étendre ses fonctionnalités avec des bibliothèques supplémentaires.

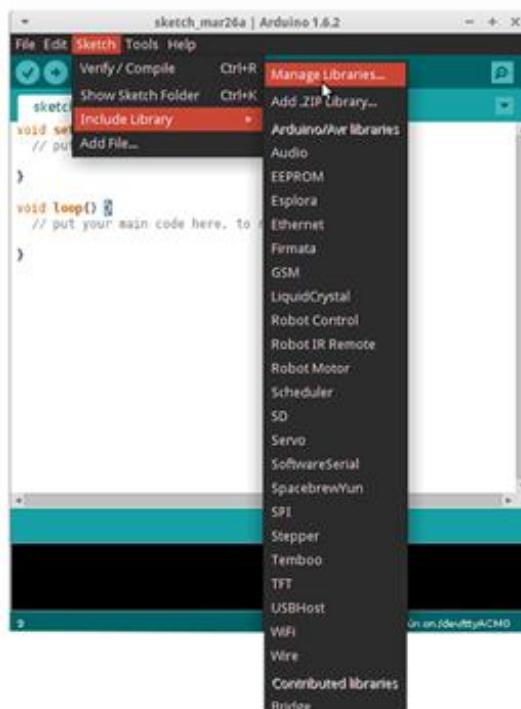
Quelles sont les bibliothèques?

Les bibliothèques sont un ensemble de codes qui peut le rendre facile pour vous connecter à un capteur, affichage, module, etc. Par exemple, la bibliothèque built-in LiquidCrystal est facile à faire un dialogue avec l'affichage LCD à caractère. Il existe des centaines de bibliothèques supplémentaires disponibles sur Internet à télécharger. Les bibliothèques intégrées et certaines de ces bibliothèques supplémentaires sont énumérés dans la référence. Vous devez installer ces bibliothèques supplémentaires avant d'utiliser.

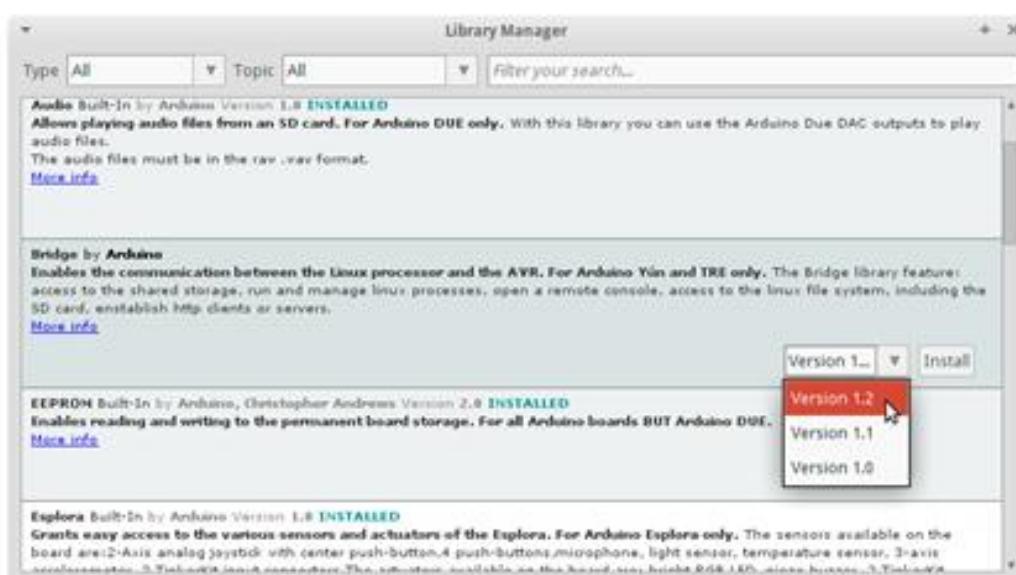
Comment installer une bibliothèque

Utiliser le Gestionnaire de Bibliothèques

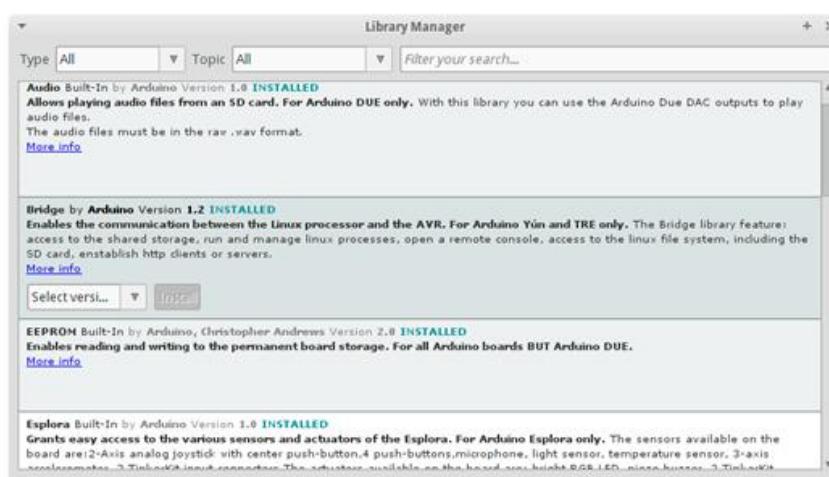
Vous pouvez utiliser le Gestionnaire de Bibliothèques pour installer une nouvelle bibliothèque dans votre IDE d'Arduino (disponible à partie de la version 1.6.2). Ouvrez IDE et cliquez Sketch > Include > Library > Manage Libraries.



Le gestionnaire de bibliothèques sera ouvert et vous pouvez trouver une liste de bibliothèques qui sont déjà installés ou prêts à installer. Dans cet exemple, nous voulons installer la bibliothèque Bridge. Défilez la liste et trouvez cette bibliothèque, puis sélectionnez la version de la bibliothèque que vous souhaitez installer. Parfois, il n'y a qu'une seule version de la bibliothèque disponible. Lorsque le menu de la sélection de version ne s'affiche pas, ne vous inquiétez pas; c'est normal.



Enfin, cliquez sur le bouton Install et attendez IDE à installer cette nouvelle bibliothèque. Le téléchargement peut prendre du temps en fonction de votre vitesse de connexion. Lorsqu'il a terminé, un dialogue Installed s'affichera à côté de la bibliothèque Bridge. Vous pouvez fermer le gestionnaire de bibliothèques.



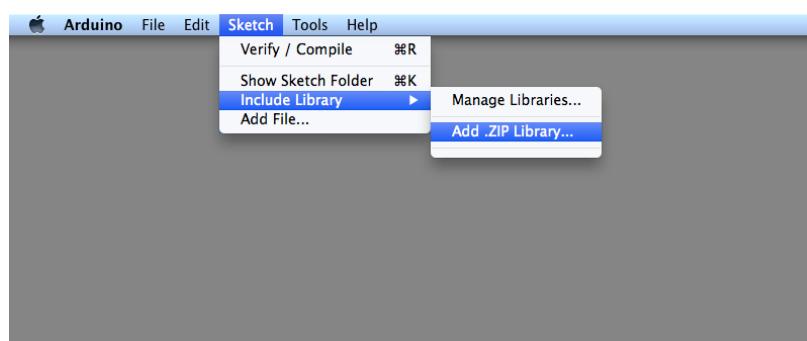
Maintenant, vous pouvez trouver une nouvelle bibliothèque disponible dans la liste de Library. Si vous voulez ajouter votre propre bibliothèque, veuillez ouvrir une nouvelle émission sur GitHub.

Importer une bibliothèque .zip

Les bibliothèques sont généralement dans un fichier ou un dossier sous une forme .ZIP. Le nom du dossier est le nom de bibliothèque. Le contenu du dossier sera comme suivant: fichier .cpp, fichier .h, aussi un fichier keywords.txt, dossier d'exemples, et d'autres fichiers requis par la bibliothèque. À partir de la version 1.0.5, vous pouvez installer les bibliothèques tierces dans IDE. Ce n'est pas la peine de décompresser la bibliothèque téléchargée; Laissez-la.

Dans IDE d'Arduino,

In the Arduino IDE, naviguez jusqu'à Sketch > Include Library > Add .ZIP Library.



Vous serez invité à sélectionner la bibliothèque que vous souhaitez ajouter. Accédez à l'emplacement du fichier .zip et ouvrez-le.

Retournez à Sketch > Import Library menu. Actuellement vous pouvez trouver la bibliothèque au bas de la liste déroulante. Il est déjà prêt d'être utilisé dans votre sketch. Le fichier zippé aura été élargi dans le dossier de Bibliothèques dans votre répertoire de sketches d'Arduino.

NB: La bibliothèque sera disponible d'utiliser dans les sketches, mais les exemples pour la bibliothèque ne s'affichent pas dans File > Examples avant de redémarrer IDE.

Installation manuelle

Pour installer la bibliothèque, tout d'abord, quittez l'application d'Arduino. Puis décompressez le fichier ZIP contenant la bibliothèque. Par exemple, si vous voulez installer une bibliothèque "ArduinoParty", décompressez ArduinoParty.zip. Il doit contenir un dossier appelé ArduinoParty, avec les fichiers tels que ArduinoParty.cpp et ArduinoParty.h. (Lorsque les fichiers .cpp et .h ne sont pas dans le dossier, vous devez les créer. Dans ce cas, vous ferez un dossier appelé "ArduinoParty" et déplacez tous les fichiers qui étaient dans le fichier ZIP dedans, comme ArduinoParty.cpp et ArduinoParty.h.)

Déplacez le dossier ArduinoParty dans ce dossier (votre dossier de bibliothèques). Pour Windows, il sera probablement appelé "My Documents\Arduino\libraries". Pour les utilisateurs de Mac, il sera probablement appelé "Documents/Arduino/libraries". Pour Linux, il sera le dossier "libraries" dans votre carnet.

Alors, votre dossier de bibliothèques d'Arduino devrait ressembler à (sous Windows) :

My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.cpp

My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.h

My Documents\Arduino\libraries\ArduinoParty\examples

....

Ou (sous Mac et Linux):

Documents/Arduino/libraries/ArduinoParty/ArduinoParty.cpp

Documents/Arduino/libraries/ArduinoParty/ArduinoParty.h

Documents/Arduino/libraries/ArduinoParty/examples

....

Il y a peut-être plus de fichiers que les fichiers .cpp et .h afin de s'assurer qu'ils sont tous là. (La bibliothèque ne travaillera pas si vous mettre directement les fichiers .cpp et .h dans le dossier de bibliothèques ou s'ils sont imbriqués dans un dossier supplémentaire. Par exemple: Documents\Arduino\libraries\ArduinoParty.cpp et Documents\Arduino\libraries\ArduinoParty\ArduinoParty.cpp ne fonctionnera pas.)

Redémarrez l'application d'Arduino. Assurez-vous la nouvelle bibliothèque s'affichent dans Sketch > Import Library menu. Voilà. Vous avez installé une bibliothèque!

Résumé

Dans cette leçon, nous allons installer tous les bibliothèques que nous utiliserons dans le mode d'emploi. Ouvrez le dossier de bibliothèques et installez les fichiers ZIP un par un, nous ne sommes donc pas besoin de faire cette étape dans les leçons suivantes. Nous connectons simplement le composant selon le schéma et téléchargez le code fourni. Puis le kit fonctionnera.

Leçon 2: Clignotant

Vue d'ensemble

Dans cette leçon, vous allez apprendre comment programmer votre carte de contrôleur UNO R3 pour intégrer l'Arduino au Clignotant LED.

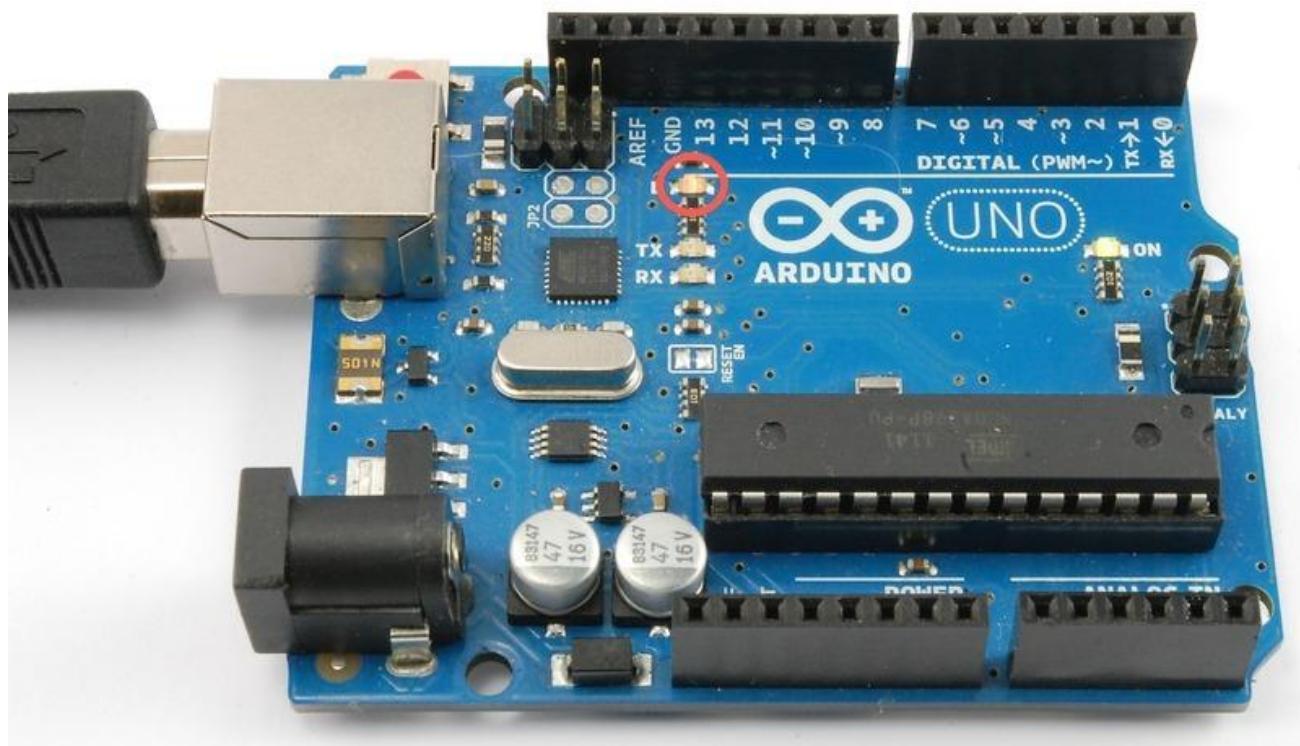
Composant requis

(1) Elegoo UNO R3

Principe

La carte UNO R3 a des lignes de connecteurs le long des deux côtés qui sont utilisés pour se connecter à plusieurs appareils électroniques et le plug-in "shields" qui étend sa capacité.

Il dispose également d'un LED unique que vous pouvez contrôler à partir de vos sketches. Ce LED est construit sur la carte UNO R3, et est souvent appelé LED "L" comme c'est comment il est étiqueté sur la carte.



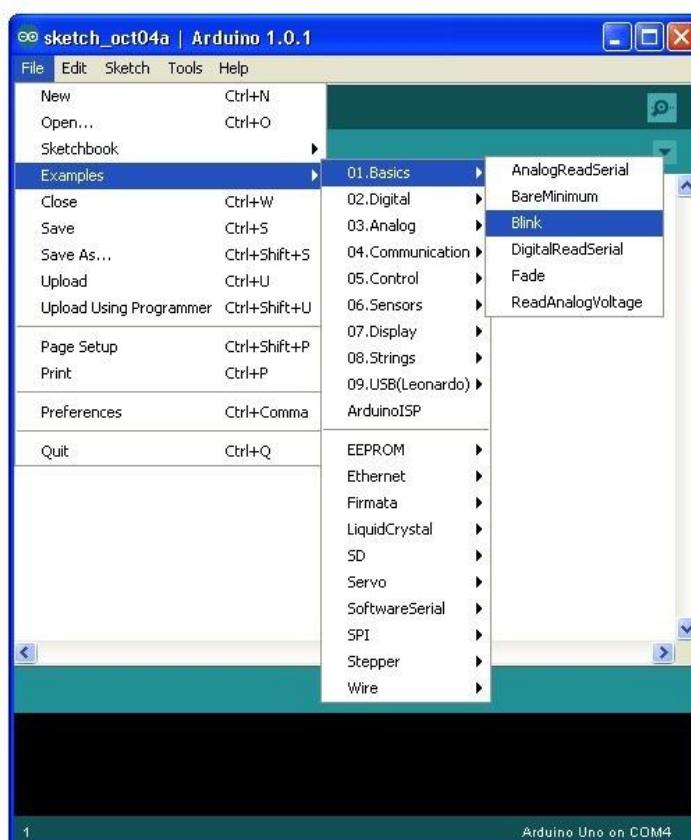
Vous pouvez trouver que LED "L" sur la carte UNO R3 clignote déjà lorsque vous le connectez à une prise USB. Puisque les cartes sont généralement livrés avec une sketche "Blink" pré-installé.

Dans cette leçon, on peut reprogrammer la carte UNO R3 avec notre propre sketche et puis modifier sa vitesse clignotante.

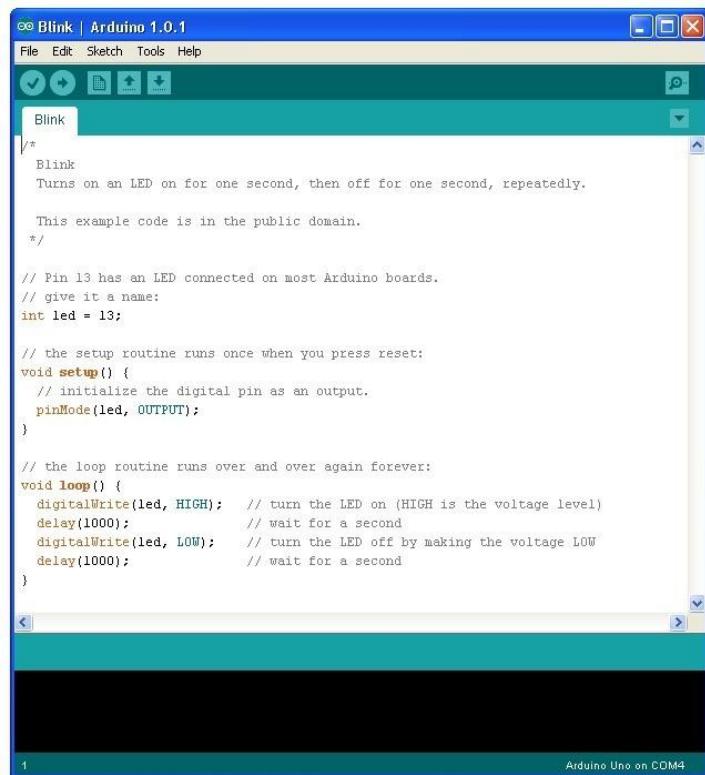
Dans leçon 0, vous avez configuré votre IDE d'Arduino et vous avez fait en sorte que vous pouvez trouver le port en série correct pour connecter à votre carte UNO R3. C'est le temps à mettre le franchement au test et programmes votre carte UNO R3.

L'IDE d'Arduino comprend une grande collection d'sketches en exemple ce que vous pouvez télécharger et utiliser. Il comprend une sketche en exemple pour faire LED "L" clignoter.

Chargez la sketche "Blink" ce que vous pouvez trouver dans le système de menu de l'IDE sous File > Examples > 01.Basics



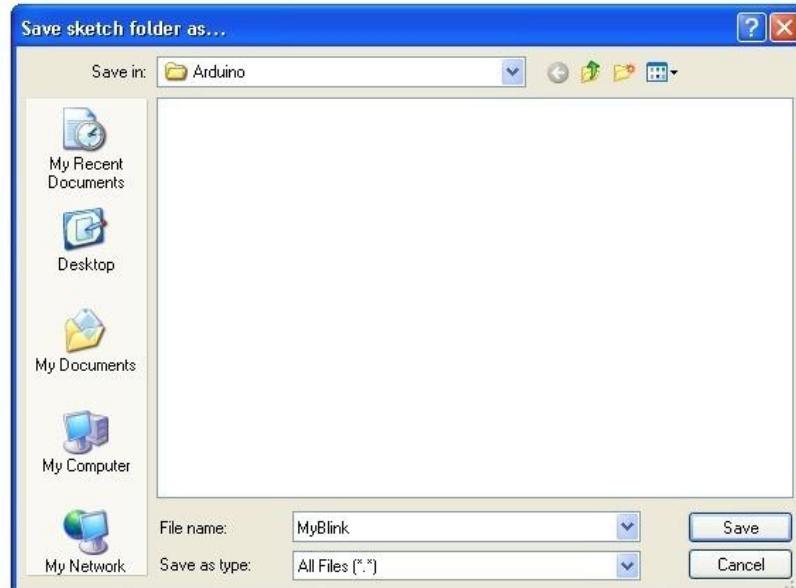
Lorsque la fenêtre de sketche est ouverte, agrandissez cette fenêtre pour voir l'ensemble de sketches dans la fenêtre.



Les sketches en exemple inclus avec l'IDE d'Arduino sont 'read-only'. Il s'agit que vous pouvez les télécharger à une carte UNO r3, mais si vous changez ces sketches, vous ne pouvez pas les sauvegarder comme un même fichier.

Puisque nous allons modifier cette sketche, la première chose que vous devez faire est de sauvegarder votre propre copie.

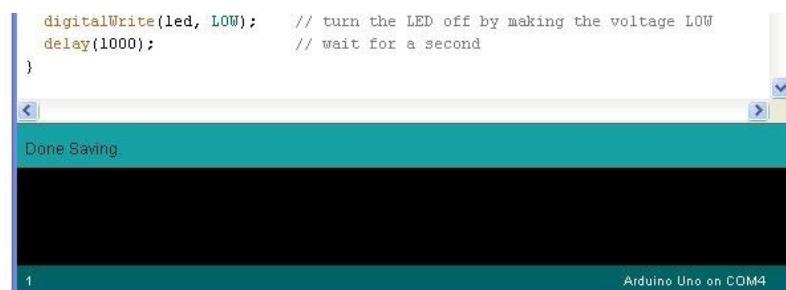
Depuis le menu File sur IDE d'Arduino, sélectionnez 'Save As..' et puis sauvegardez la sketche sous un nom 'MyBlink'.



Vous avez sauvegardé votre copie de "Blink" dans votre sketchbook. C'est-à-dire que si vous voulez le retrouver, vous l'ouvrez uniquement en utilisant l'option de menu File > Sketchbook.



Branchez votre carte d'Arduino à votre ordinateur par le câble USB et vérifiez que "Board Type" et "Serial Port" sont configuré correctement. Vous devrez peut-être revenir à la leçon 0. Il s'affichera la configuration actuelle sur l'IDE d'Arduino pour la carte au bas de la fenêtre.

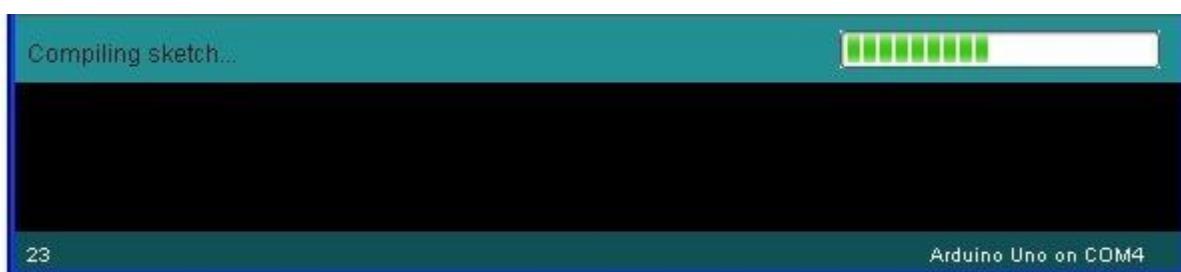


Cliquez sur le bouton 'Upload'. C'est le deuxième bouton à partir de la gauche sur la barre d'outils.

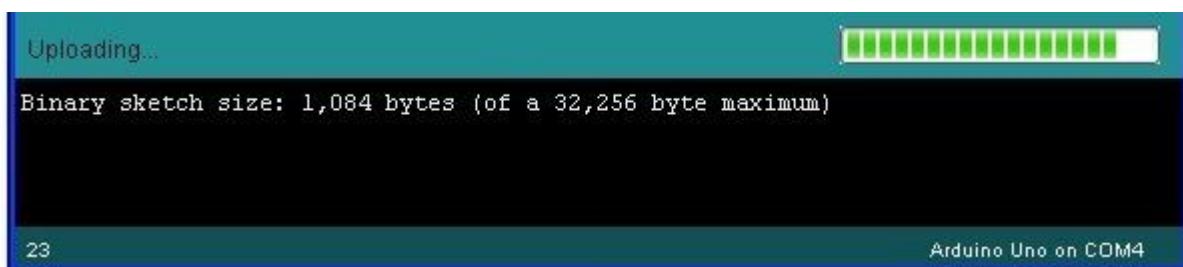


Si vous regardez la zone d'état de l'IDE,

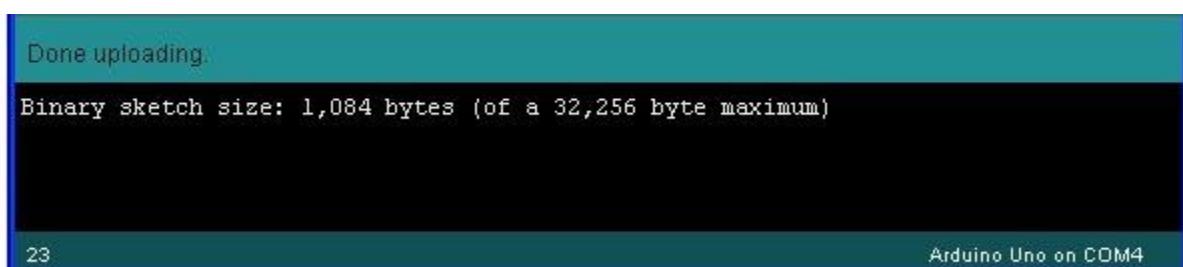
If you watch the status area of the IDE, Vous verrez une barre de progression et une série de messages. Tout d'abord, il s'affichera 'Compiling Sketch...'. Il convertit la sketche en un format approprié pour télécharger à la carte.



Ensuite, l'état passe à 'Uploading'. À ce moment, la LED sur la carte d'Arduino doit commencer à clignoter comme la sketche est transférée.



Enfin, l'état passe à 'Done uploading'. Un message s'affichera pour indiquer que la sketche utilise 1,084 octets de la 32,256 octets disponibles.



Veuillez noter que vous pouvez obtenir un message d'erreur suivant pendant l'étape 'Compiling Sketch...':



Il s'agit que votre carte n'est pas bien connectée, ou que les pilotes n'ont pas été installés (si nécessaire) ou que le mauvais port série est sélectionné.

Si vous rencontrez ce problème, rappelez la leçon 0 et vérifiez votre installation.

Une fois le téléchargement a terminé, la carte devrait redémarrer et commencer à clignoter.

Ouvrez le code

Notez qu'une partie considérable de cette sketche est composé de commentaires. Ils ne sont pas des instructions du programme; ils ont plutôt simplement d'expliquer comment le programme fonctionne. Ils sont là pour votre bénéfice.

N'importe quel entre /* et */ en haut de l'sketch est un commentaire de bloc ; il explique ce que le sketch est pour.

Les commentaires de ligne unique commence par // et tout jusqu'à la fin de cette ligne est considérée comme un commentaire.

La première ligne de code est:

Copier le code

1. `int led = 13;`

Comme un commentaire expliqué ci-dessus, c'est un nom à la borne que le LED est attachée.

Il s'agit 13 sur la plupart des Arduinos, y compris l'ONU et Leonardo.

Ensuite, nous avons la fonction 'setup'. Encore une fois, que le commentaire dit, il est exécuté lorsque la touche Reset est enfoncée. Il est également exécuté chaque fois que la carte se réinitialise pour quelque raison que ce soit, tels que la première mise sous tension, ou après qu'une sketch a été téléchargé.

Copier le code

1. `void setup() {`
2. `// initialize the digital pin as an output.`
3. `pinMode(led, OUTPUT);`
4. `}`

Chaque sketch d'Arduino doit avoir la fonction 'setup', et l'emplacement où vous voulez ajouter vos propres instruction est entre { et }.

Dans ce cas, il n'y a qu'une seule commande, qui, comme le commentaire raconte la carte d'Arduino que nous allons utiliser la broche LED comme une sortie.

Il est également obligatoire pour un sketch à avoir une fonction "Loop". **Il est contrairement à la fonction 'Setup' qui ne s'exécute qu'une fois, après une remise à zéro, la fonction 'loop' va immédiatement commencer à nouveau après qu'il ait fini d'exécuter ses commandes.**

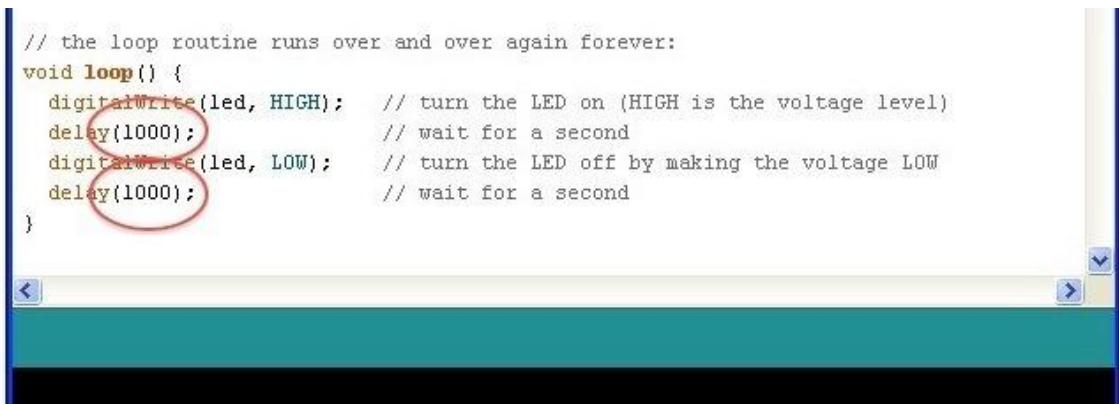
Copier le code

1. `void loop() {`
2. `digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)`
3. `delay(1000); // wait for a second`

```
4. digitalWrite(led, LOW);      // turn the LED off by making the voltage LOW
5. delay(1000);                // wait for a second
6. }
```

Dans la fonction loop, Les commandes tout d'abord s'allument la broche LED (HIGH), et puis 'delay' Pour 1000 millisecondes (1 second), ensuite s'éteignent la broche LED et donnent une pause pendant une autre seconde.

Maintenant, pour mettre votre clignotant LED plus rapide. Comme vous l'avez deviné, la clé réside dans le changement dans le paramètre () pour la commande 'delay'.



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

La période de délai est en milliseconde, donc si vous souhaitez que la LED clignote deux fois plus vite, modifier la valeur de 1000 à 500. Il s'agit de pauser This would then pause pour une demi-seconde chaque retard plutôt que toute une seconde.

Chargez la sketche à nouveau et vous pouvez trouver que LED clignote plus rapidement.

Leçon 3: LED

Vue d'ensemble

Dans cette leçon, vous pouvez apprendre à changer la luminosité d'une LED avec les différentes valeurs de résistance.

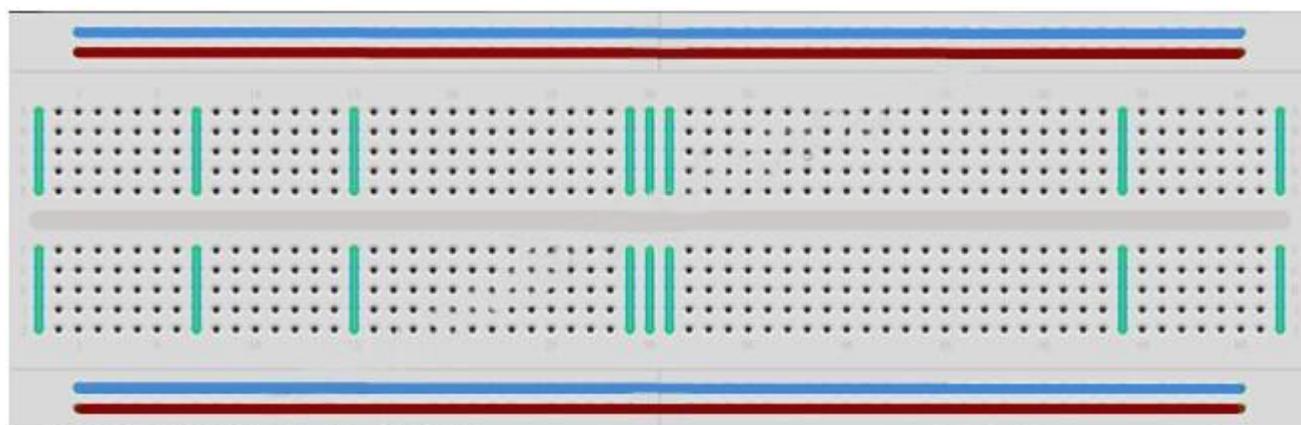
Composants requis

- (1) Elegoo UNO R3
- (1) 5mm LED rouge
- (1) 220 ohm résistance
- (1) 1k ohm résistance
- (1) 10k ohm résistance
- (2) M-M câbles

Introduction de composant

CARTE DE PROTOTYPAGE MB-102:

La carte de prototypage vous permet de circuits prototypes rapidement, sans avoir à souder les connexions. Un exemple est comme suivant.



Les cartes de prototypage sont en différentes tailles et configurations. Le type le plus simple est juste une grille de trous dans un bloc en plastique. Les bandes de métal qui fournissent la connexion électrique entre les trous dans les lignes plus courtes, sont à l'intérieur. Pousser les jambes des deux composants différents dans la même ligne se joint ensemble électriquement. Un chenal profond qui descend au milieu indique qu'il y a une rupture dans les connexions, il

s'agit que vous pouvez pousser une puce avec les jambes de chaque côté du canal sans les relier ensemble. Certaines cartes de prototypage ont deux bandes de trous le long des bords longitudinaux de la carte d'administration qui sont séparés de la grille principale. Ces disposent de bandes s'étendant sur toute la longueur du bord intérieur et fournissent un moyen de connecter un voltage courant. Ils sont habituellement par paires pour +5 volts et mis à la terre. Ces bandes sont appelées comme rails et ils vous permettent de raccorder aux nombreux composants ou points au sein de la carte.

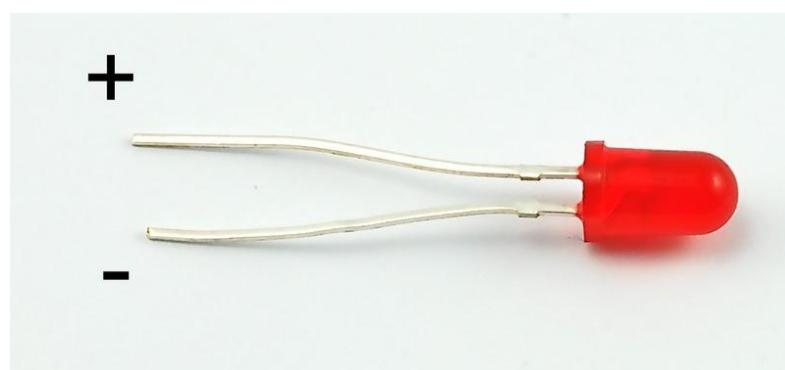
Lorsque les cartes de prototypage sont idéales pour le prototypage, ils ont certaines limites. Parce que les connexions sont à emmanchement et temporaires, ils ne sont pas aussi fiables que les connexions soudées. Si vous rencontrez des problèmes intermittents avec un circuit, ça pourrait être dû à une mauvaise connexion sur une carte de prototypage.

LED:

LEDs font de grands voyants lumineux. Ils utilisent très peu d'électricité et ils ont assez bien à durer éternellement.

Dans cette leçon, vous pouvez prendre peut-être le plus commun de tous les voyants LEDs: un 5mm LED rouge. 5mm désigne le diamètre de la LED. D'autres tailles courantes sont 3mm et 10mm.

Il est interdit de connecter directement une LED sur une batterie ou une source de tension parce que 1) LED possède une borne positive et une borne négative, et ne s'allumera plus si la disposition est incorrecte, et 2) Une LED doit être utilisée avec une résistance à limiter où "Choke" le montant du courant le traversant; sinon, il grillera!



Si vous n'utilisez aucune résistance pour une LED, il peut bien être détruit presque immédiatement, comme trop grande quantité de courant s'écoule, en chauffant et détruisant

"Junction" où la lumière est produite.

Il existe deux façons de savoir qui est la borne positive et qui est la borne négative de la LED.

Tout d'abord, la borne positive est plus longue.

Deuxièmement, sur l'emplacement où la borne négative pénètre dans le corps de la LED, il y a un bord plat au coffret de la LED.

Si vous avez une LED qui présente un bord plat à côté de la broche plus longue, vous devez supposer que la borne plus longue est positive.

RÉSISTANCES:

Montré comme le nom, les résistances sont utilisées à résister l'écoulement de l'électricité. La valeur de la résistance plus élevée, la résistance d'électricité plus forte, et le courant à traverser la résistance plus faible. On va utiliser les résistances pour contrôler la quantité d'électricité du circuit à traverser la LED, et par conséquent, comment il resplendit.

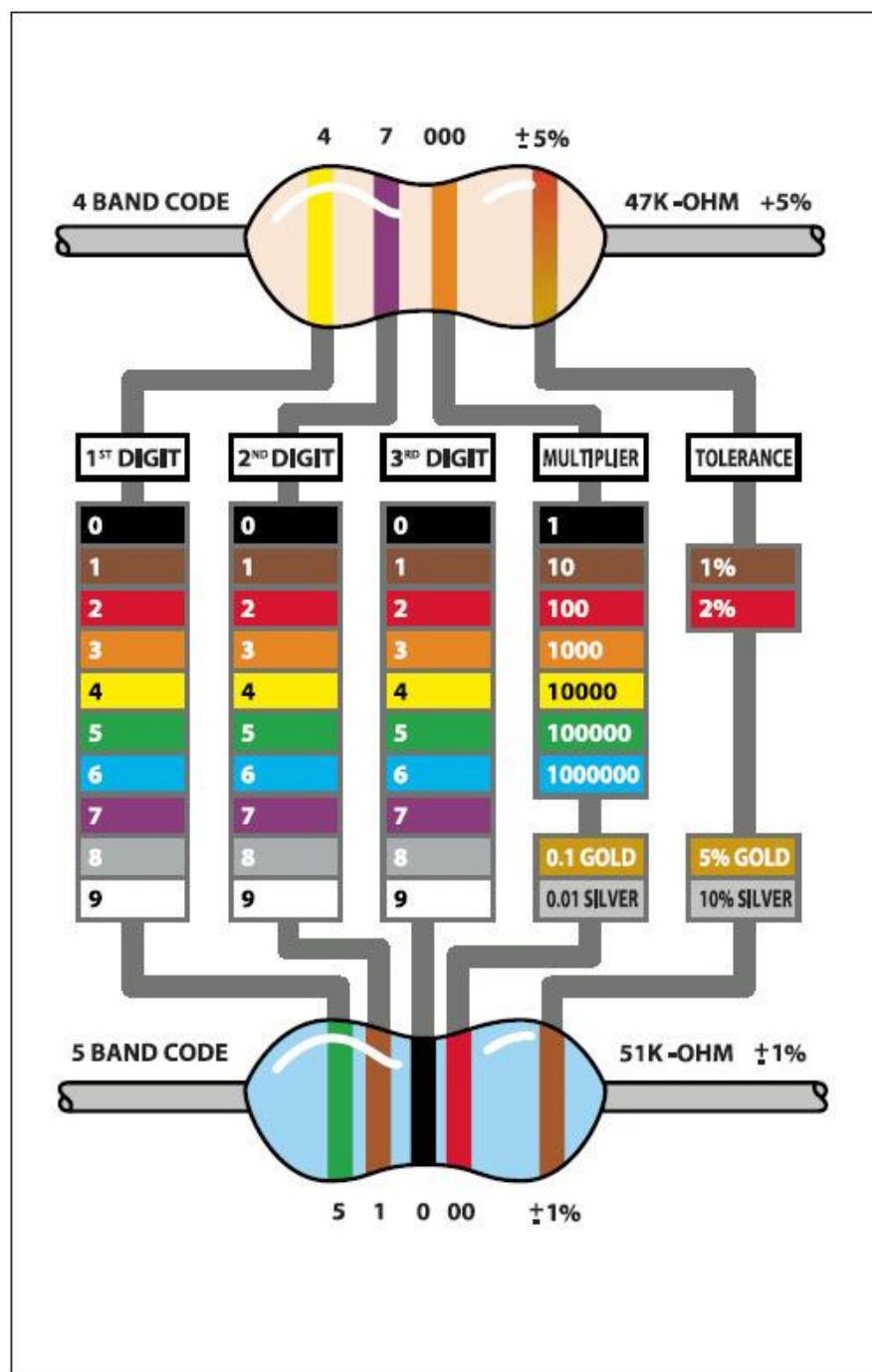


Mais tout d'abord, plus informations sur les résistances...

L'unité de résistance est en Ohm, qui est habituellement raccourci à ω la lettre grecque oméga. Parce qu'un ohm est une faible valeur de résistance (Il n'a pas résisté plus du tout), Nous avons également désigné les valeurs de résistances en $k\Omega$ (1,000 Ω) et $M\Omega$ (1,000,000 Ω). On appelle ces kilo-ohms et méga-ohms.

Dans cette leçon, les trois valeurs différentes de la résistance seront utilisées: 220Ω , $1k\Omega$ et $10k\Omega$. Ces résistances toutes la même apparence, sauf qu'elles ont différentes bandes colorées sur eux. Ces bandes vous informent la valeur de la résistance.

Le code de couleur de résistance donne trois bandes colorées et puis une bande d'or à une extrémité.



Contrairement à LED, il n'existe pas la borne négative ou la borne positive sur les résistances.

Ils peuvent être connectés dans un sens comme dans l'autre.

Raccordement

Schéma

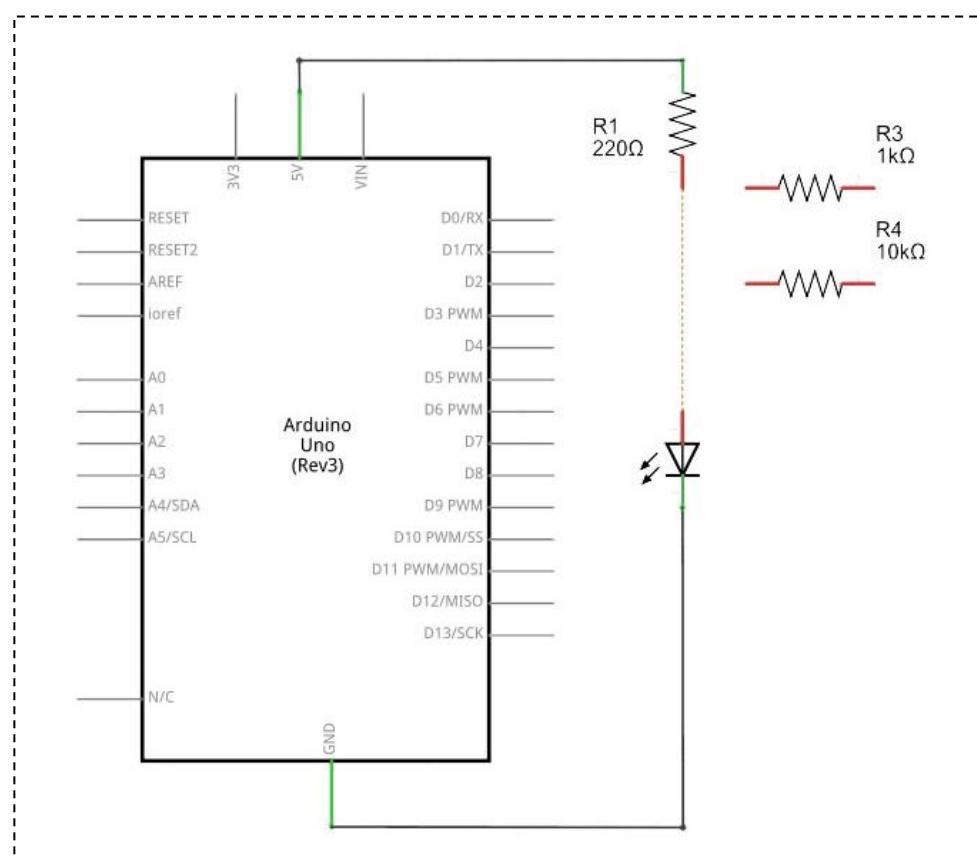
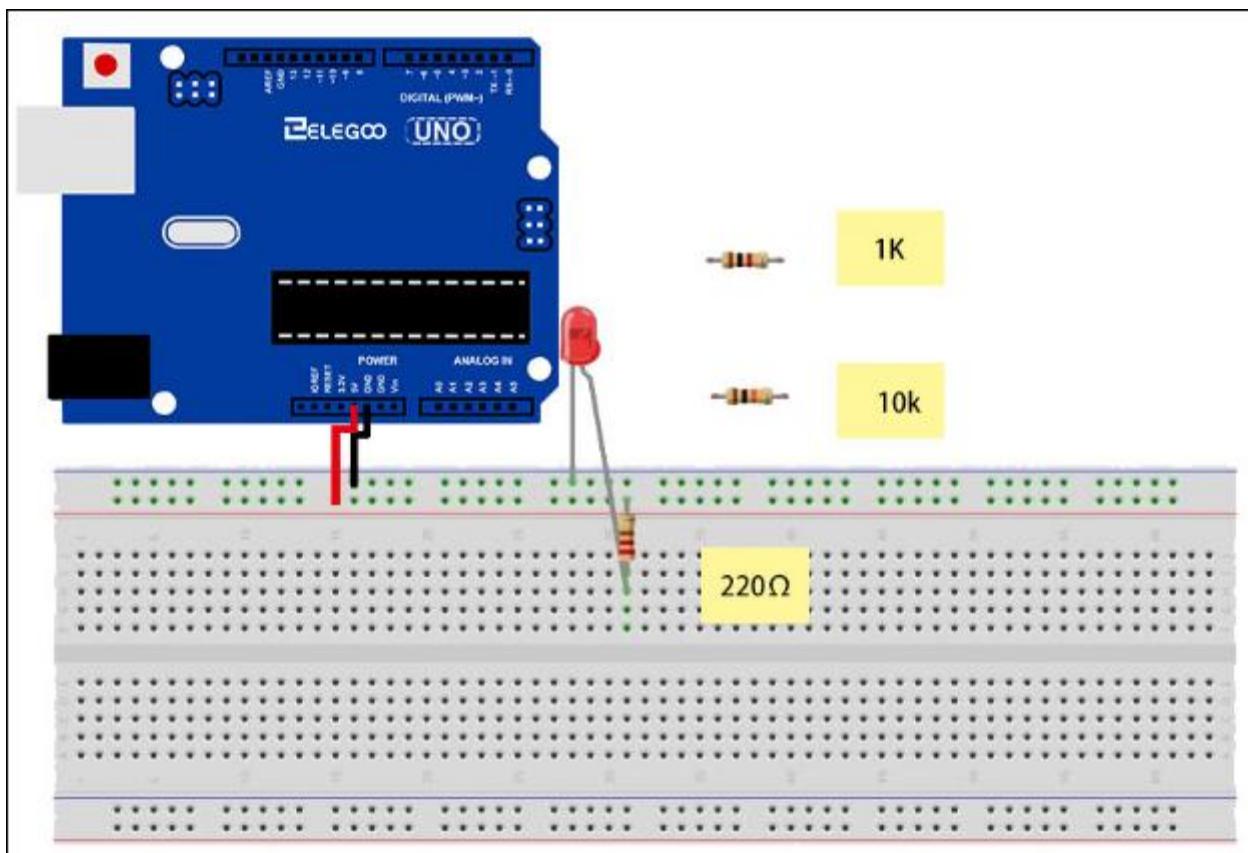


Schéma de câblage



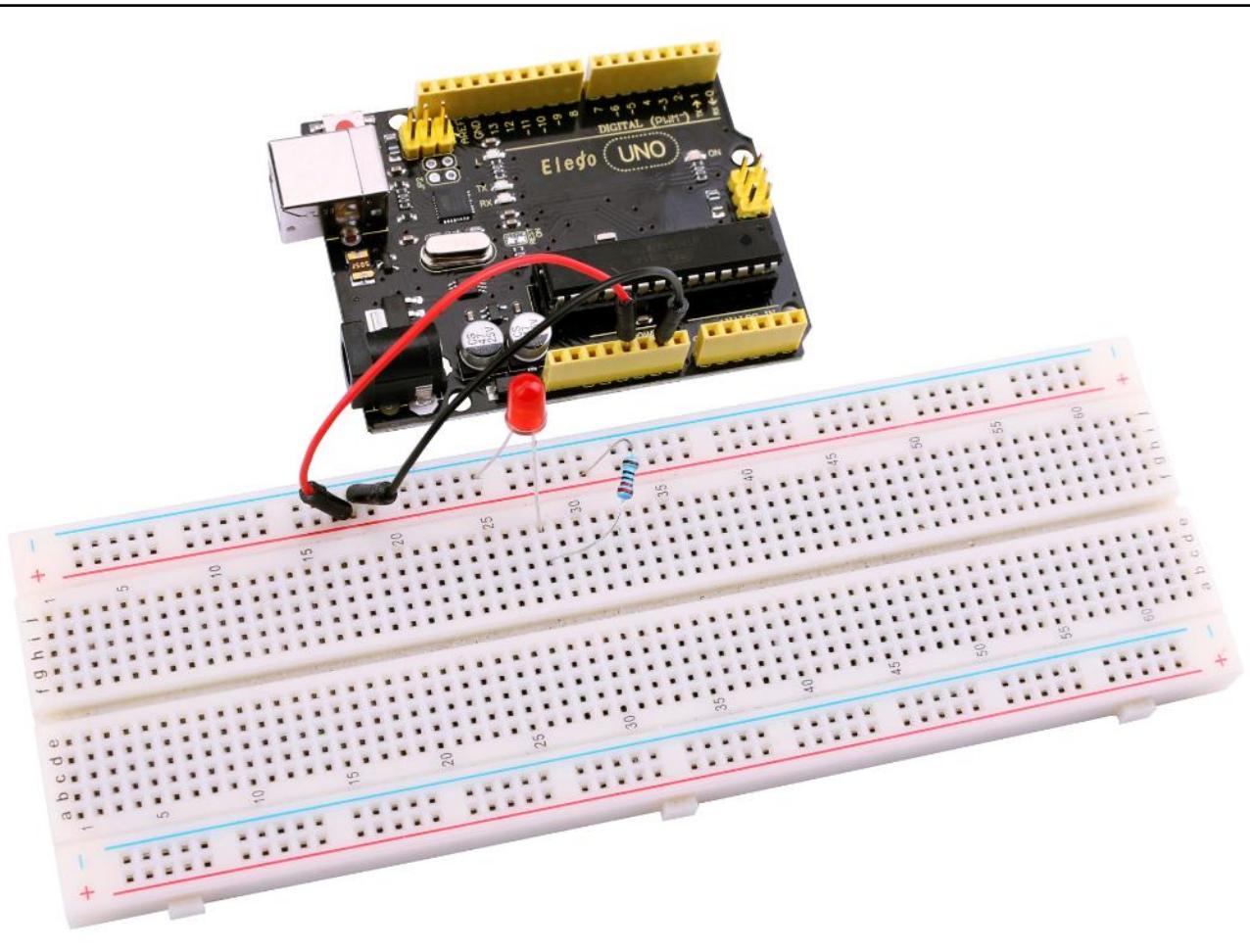
L'UNO est une source commode de 5 volts, que l'on l'utilise pour fournir la puissance de la LED et la résistance. Vous ne devez faire aucune opération sur votre UNO que le brancher un câble USB.

Avec la résistance $220\ \Omega$ en place, la LED doit être assez lumineux. Si vous permutez la résistance $220\ \Omega$ par la résistance $1k\Omega$, le voyant apparaît un peu sombre. Enfin, avec la résistance $10\ k\Omega$ resistor en place, le voyant doit être à peu près visible. Tirez le fil de pontage rouge hors de la carte de prototypage et touchez-le dans le trou et retirez-le, de sorte qu'il agit comme un interrupteur. Vous devez juste être capable de remarquer la différence.

Pour le moment, vous avez 5V allant à une patte de la résistance, l'autre patte de la résistance va être vers le côté positif de la LED et l'autre côté de la LED vers au GND. Toutefois, si l'on déménage la résistance de sorte qu'il est arrivé après la LED, comme illustré ci-dessous, la LED s'allument toujours.

Vous voudrez probablement remettre la résistance $220\ \Omega$ en place.

Il n'importe pas de quel côté de la LED nous mettons la résistance, tant qu'il est à quelque part.



Leçon 4: RGB LED

Vue d'ensemble

RGB LEDs sont un moyen amusant et facile d'ajouter la couleur à vos projets. Parce qu'ils sont comme 3 voyants réguliers dans l'un, comment les utiliser et connecter n'est pas très différent. Ils sont principalement en 2 versions: Common Anode ou Common Cathode.

Common Anode utilise 5V sur la broche commune, lorsque Common Cathode est mis à la terre.

Pour n'importe quelle LED, nous devons connecter quelques résistances en ligne (3 en total) afin que nous puissions limiter le courant.

Dans notre sketch, nous débutera avec la LED dans l'état de couleur rouge, puis fondu au vert, puis fondu au bleu et finalement de retourner à la couleur rouge. Nous allons parcourir plus de la couleur qui peut être atteint en faisant cela.

Composants requis

- (1) Elegoo UNO R3
- (1) Carte de prototypage
- (4) M-M câbles
- (1) RGB LED
- (3) 220 ohm résistances

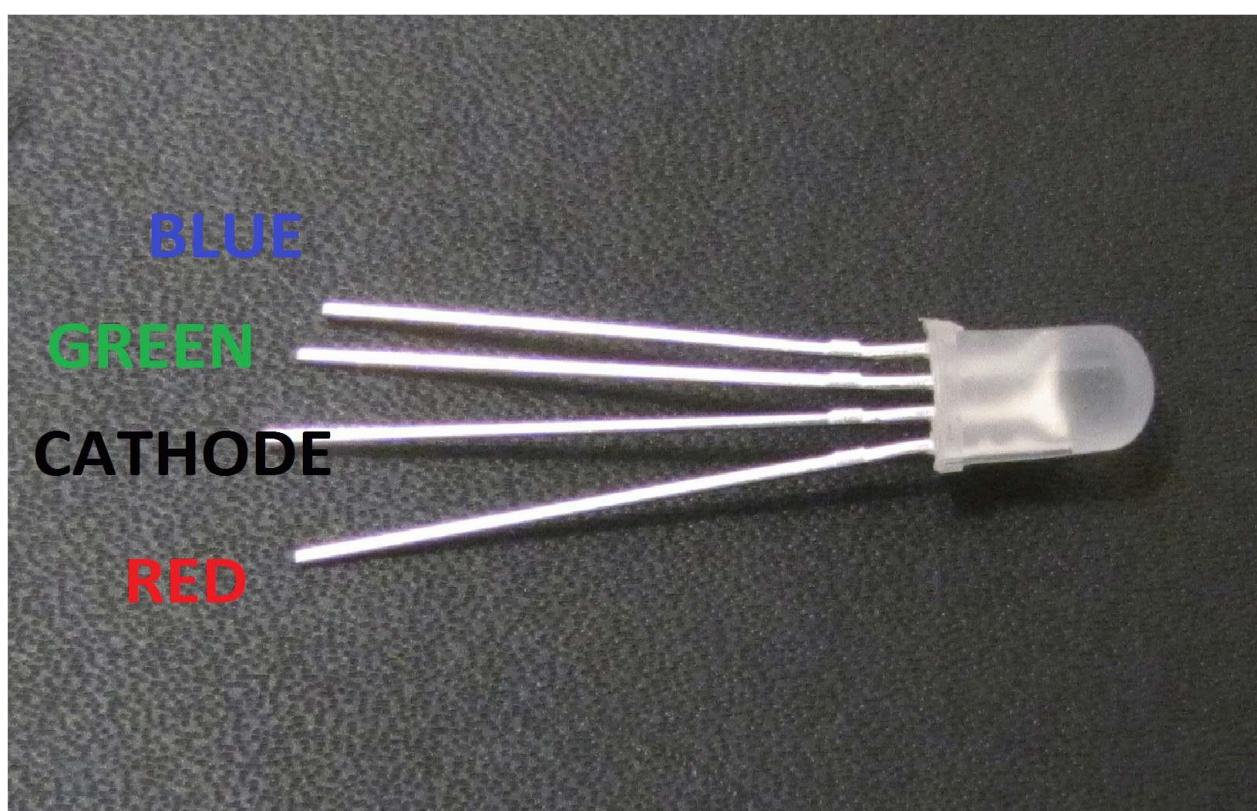
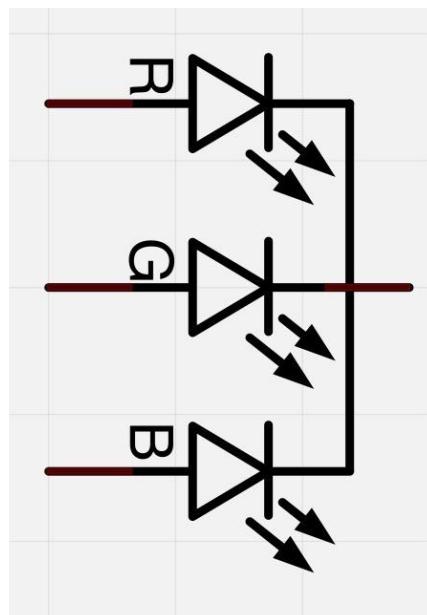
Introduction de composant

RGB:

Au premier coup d'oeil, RGB (Rouge, Vert, Bleu) LEDs ressemblent juste à LED ordinaire. Toutefois, à l'intérieur de l'emballage habituel de LED, il y a effectivement des trois LEDs, rouge, verte et bleue. En contrôlant la luminosité de chacune des LEDs individuelles vous pouvez mélanger à peu près n'importe quelle couleur vous voulez.

Nous mélangeons les couleurs de la même façon que vous le feriez pour mélanger la peinture sur une palette - en ajustant la luminosité de chacun des trois voyants. Le moyen plus dur pour ce faire serait d'utiliser des résistances de valeur différente (ou résistances variables) comme nous l'avons fait dans la leçon 2, mais c'est beaucoup de travail ! Heureusement pour nous, la

carte UNO R3 dispose une fonction analogWrite que vous pouvez utiliser avec les bornes marqués avec a ~ pour sortir une quantité variable de puissance pour les voyants appropriés. La LED RGB présente quatre fils. Un fil va aller à la connexion positive de chacune des LED unique au sein du package et un fil unique est connecté à tous trois points négatifs des voyants.



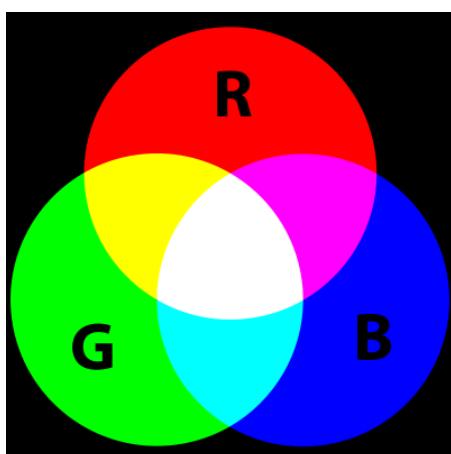
La connexion négative commune du package LED est la deuxième borne à partir du côté plat. Il est également le fil plus long dans ces quatre fils, et il sera mis à la terre.

Chacune LED dans le package demande sa propre résistance 270Ω . Each LED inside the package requires its own 270Ω resistor afin d'éviter trop de courant le traversant. Les trois fils positifs de LEDs (une rouge, une verte et une bleue) sont branchés à la broche de sortie UNO en utilisant les résistances.

COULEUR:

La raison pour laquelle vous pouvez mélanger n'importe quelle couleur que vous aimez en faisant varier les quantités de rouge, verte et bleue, c'est que votre oeil a trois types de récepteurs de lumière dedans (rouge, vert et bleu). Vos yeux et votre cerveau traitent les quantités de rouge, vert et bleu et le convertissent en une couleur du spectre.

En quelque sorte, en utilisant les trois LEDs, nous jouons un truc sur les yeux. Il est de même idée en utilisant sur les TVs, où l'écran LCD a des points de couleur rouge, vert et bleu à côté de l'autre pour composer chaque pixel.



Si nous définissons la luminosité de tous les trois LEDs d'être la même, et puis la couleur générale de la lumière sera blanche. Si on éteint la LED bleue, et que les voyants vert et rouge ne sont la même luminosité, alors la lumière apparaîtra jaune.

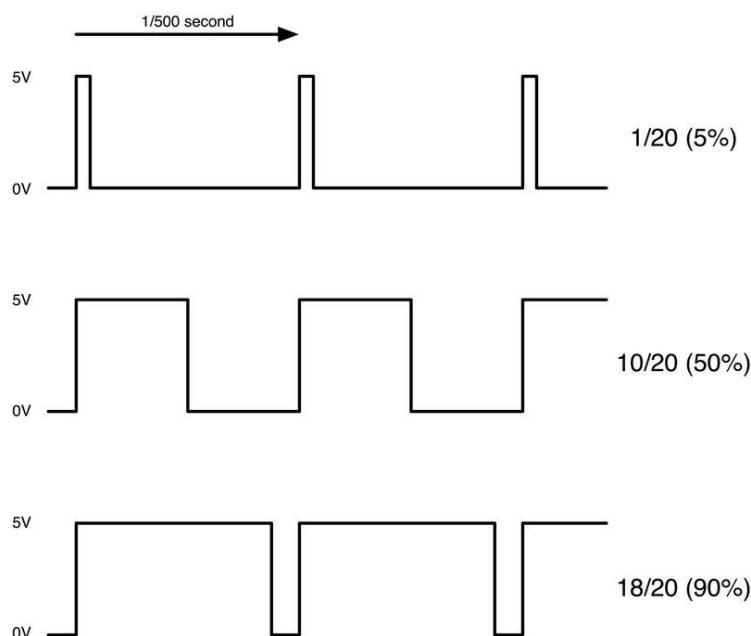
On peut contrôler la luminosité de chacune des couleurs entre rouge, vert et bleu de la LED séparément, afin d'être possible de mixer toute couleur que nous aimons.

Noir n'est pas tellement une couleur comme une absence de lumière. Nous pouvons donc venir le plus proche au noir lorsque notre LED est de désactiver les trois couleurs.

Théorie de PWM

Pulse Width Modulation (PWM) est une technique pour contrôler l'alimentation. Nous avons aussi de l'utiliser ici pour contrôler la luminosité de chacune couleur des LEDs.

Le diagramme ci-dessous montre le signal de l'une des broches de PWM sur l'UNO.



La sortie de PWM produira une impulsion à peu près tous les 1/500 de seconde. La longueur de cette impulsion est contrôlée par la fonction 'analogWrite'. Alors 'analogWrite(0)' ne produira pas des tout impulsions et 'analogWrite(255)' produira une impulsion qui persiste tout le chemin jusqu'à l'impulsion suivante est due, pour que la sortie est en fait sur tout le temps. Si nous spécifions une valeur dans l'analogWrite qui est quelque part entre 0 et 255, une impulsion sera produite. Si l'impulsion de sortie est seulement élevés pour 5% du temps, n'importe quel nous conduisons ne recevra que 5% de la pleine puissance. Toutefois, si la sortie est à 5 V pour 90 % du temps, puis le chargement recevra 90 % de l'électricité livrée. Nous ne pouvons pas voir les voyants en état actif ou non actif à cette vitesse, alors pour nous, il ressemble uniquement que la luminosité change.

Raccordement

Schéma

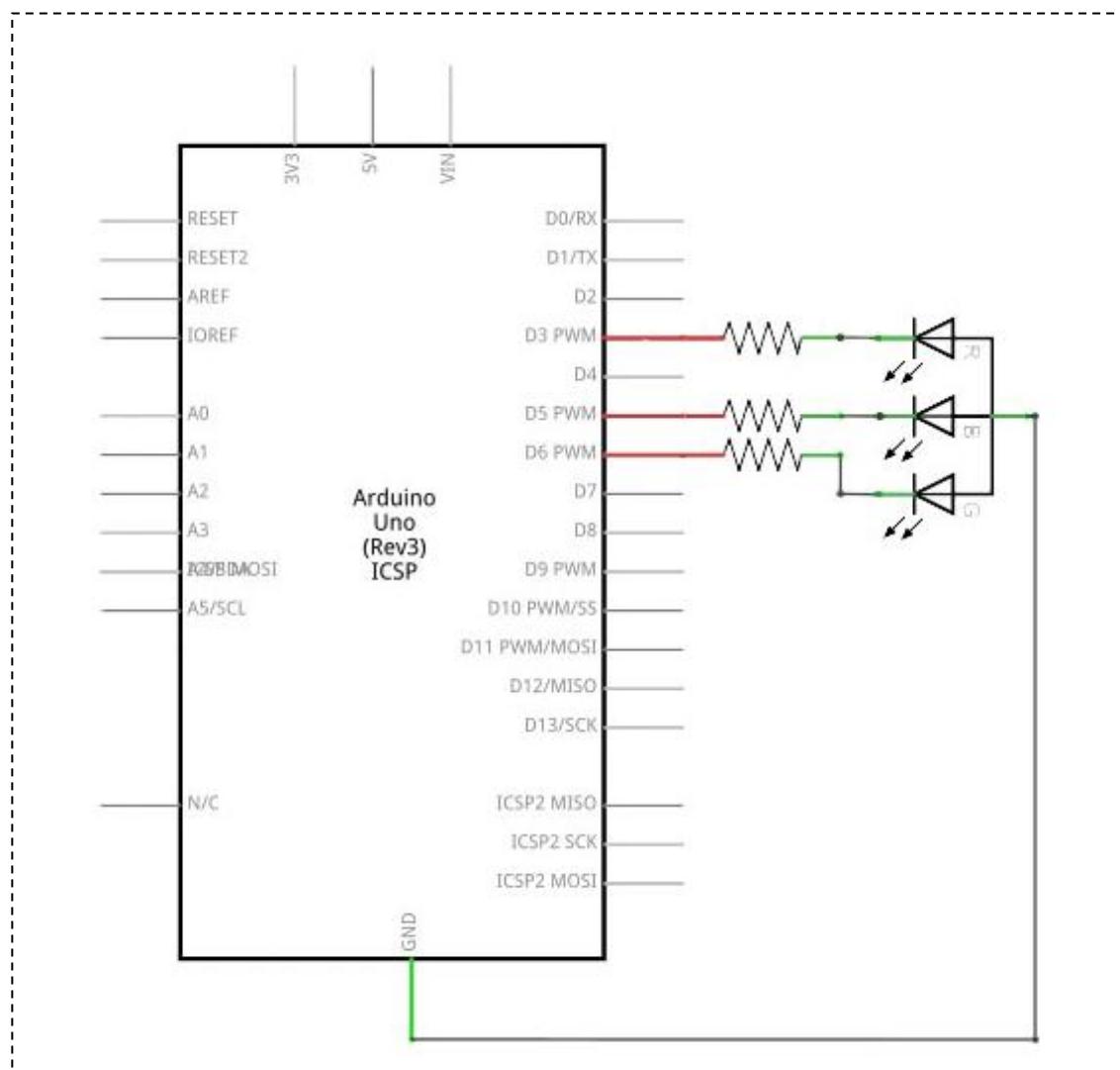
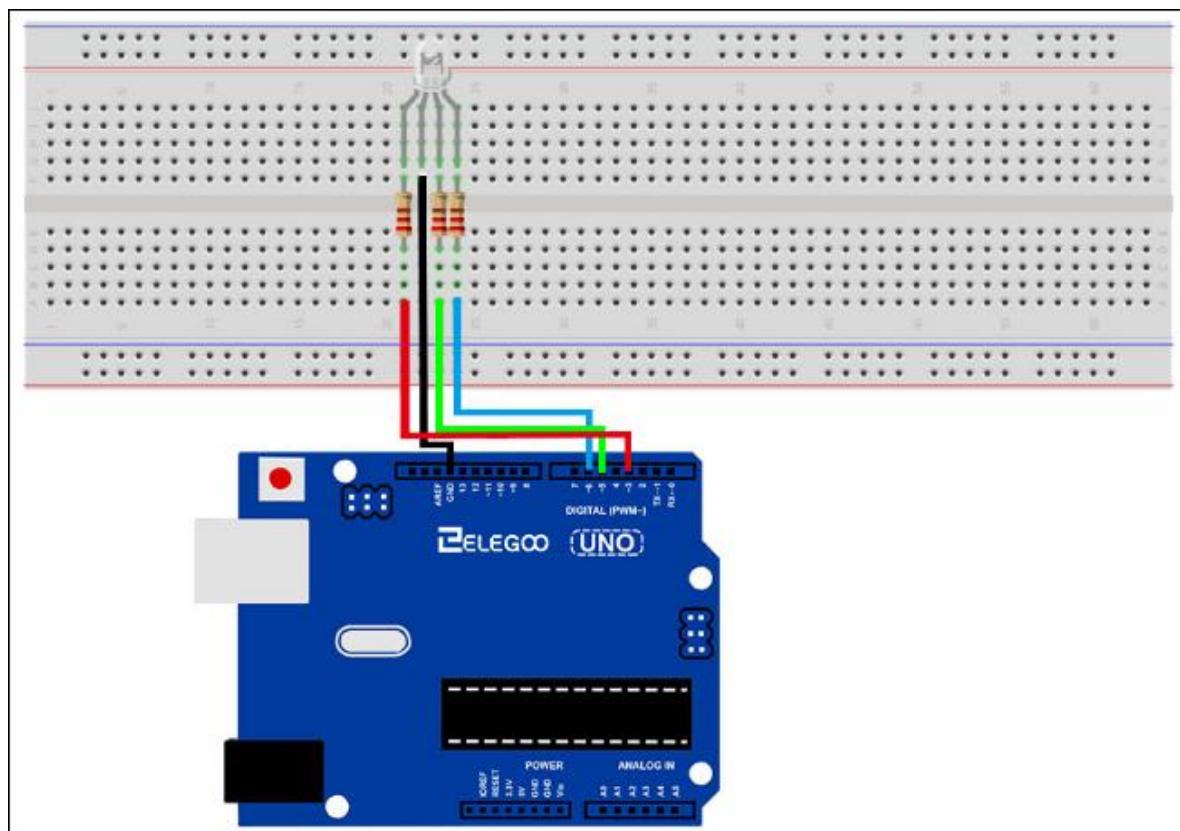


Schéma de câblage



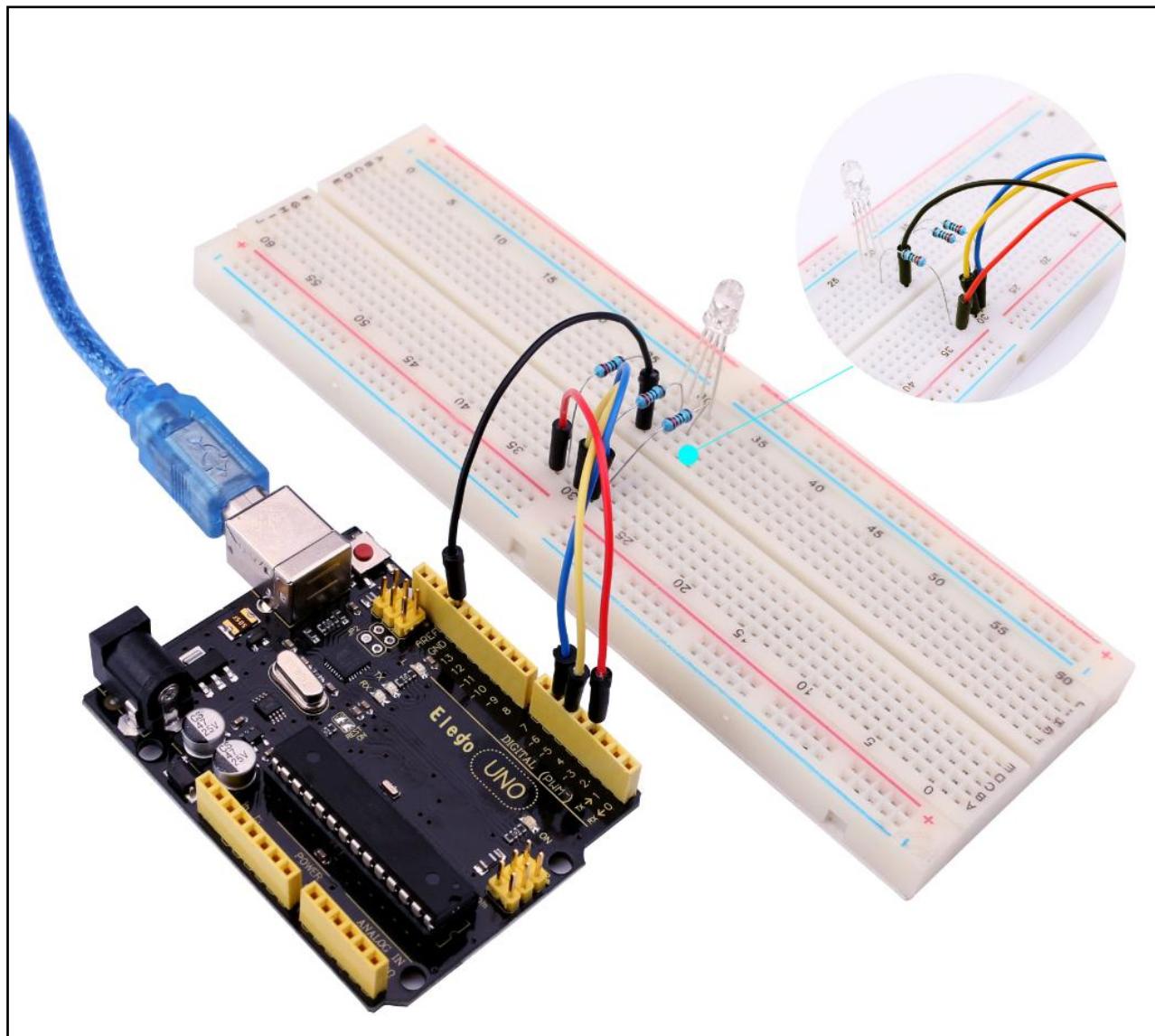
Code

Notre code utilisera le code de boucle FOR à défiler les couleurs.

La première boucle FOR va passer du Rouge au Vert.

La deuxième boucle FOR va passer du Vert au Bleu.

La dernière boucle For va passer du Bleu au Rouge.



Leçon 5: Entrées Numériques

Vue d'ensemble

Dans cette leçon, vous apprendrez à utiliser des boutons poussoirs avec entrées numériques pour allumer une LED ou l'éteindre.

En appuyant sur le bouton près du haut de la carte de prototypage va allumer LED; En appuyant sur l'autre bouton va éteindre LED.

Composants requis

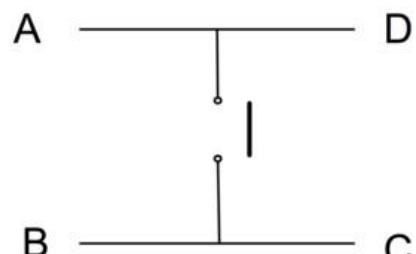
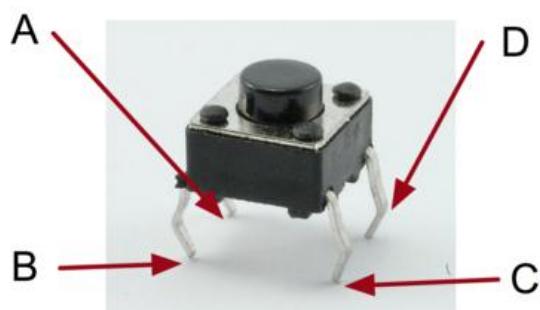
- (1) Elegoo UNO R3
- (1) Carte de prototypage
- (1) 5mm LED rouge
- (1) 220 ohm résistance
- (2) Commutateurs à bouton-poussoir
- (6) M-M câbles

Introduction de composant

COMMUTATEURS À BOUTON-POUSSOIR:

Les commutateurs sont des composants très simples. Lorsque vous appuyez sur un bouton ou pivotez un levier, ils peuvent brancher deux contacts ensemble afin que l'électricité peut circuler à travers eux.

Les petits commutateurs tactiles utilisés dans cette leçon ont autre connexions, qui peut être un peu déroutant.



Actuellement, il n'y a que deux connexions électriques. Dans le package d'commutateurs, les broches B et C sont branchées ensemble comme A et D.

Raccordement

Schéma

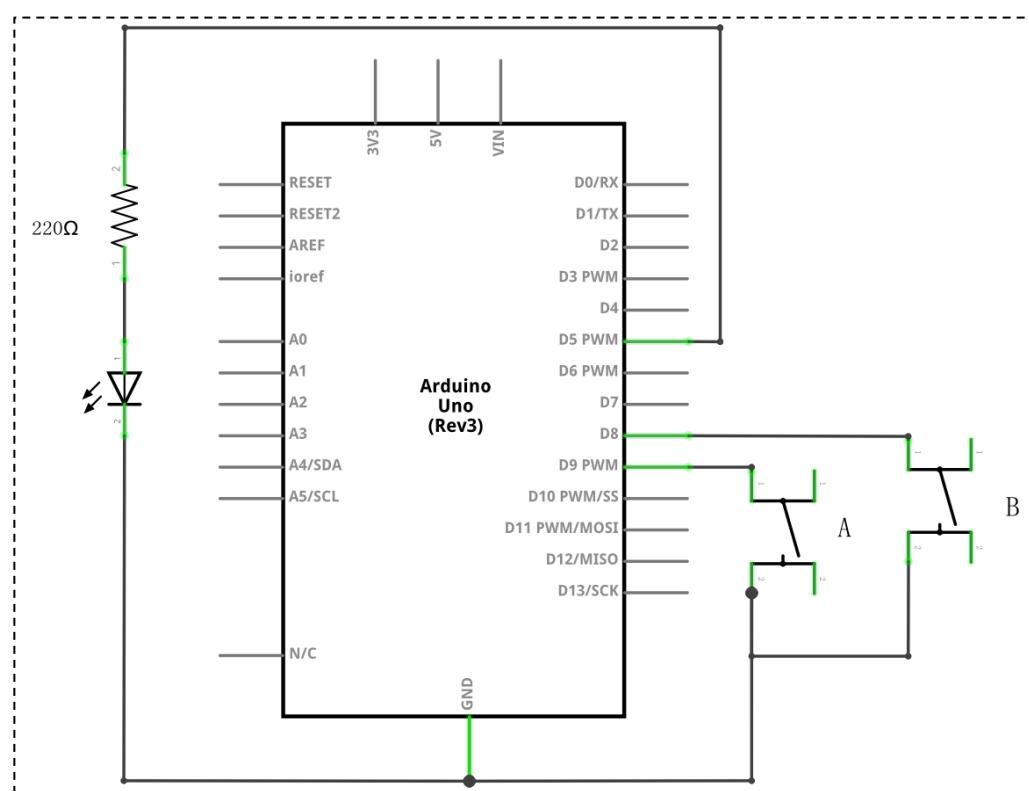
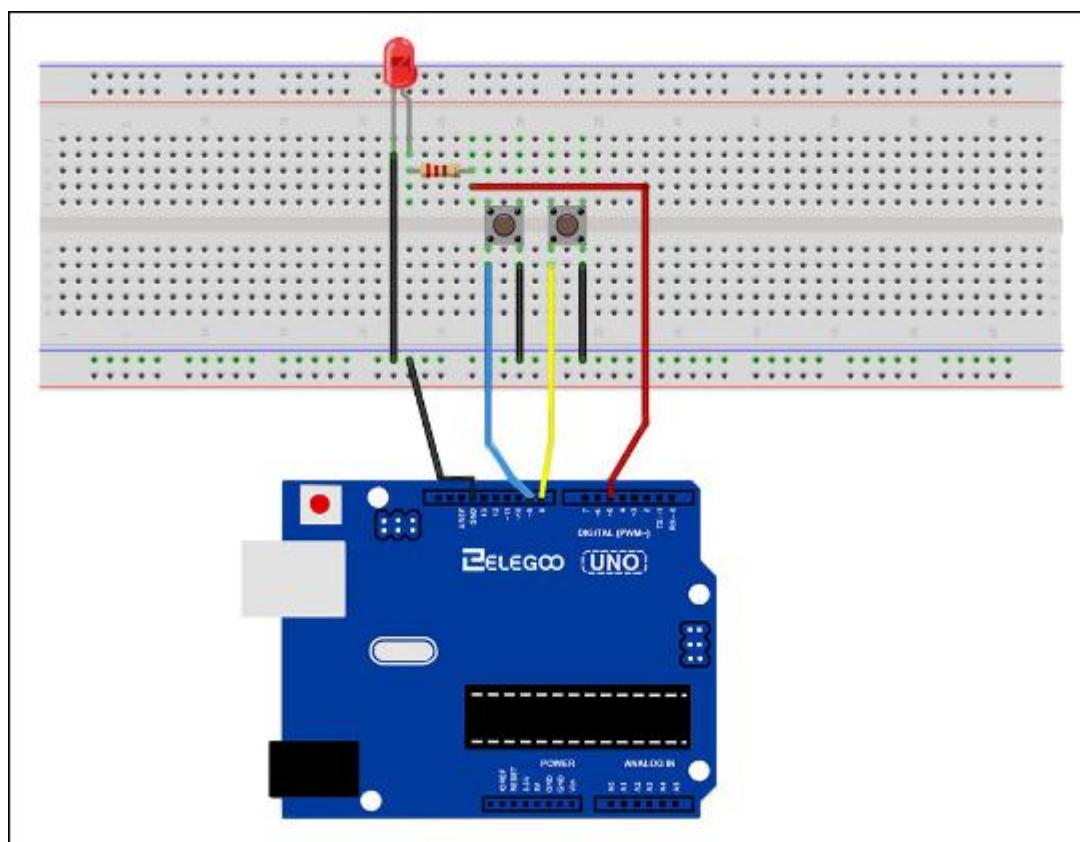


Schéma de câblage



Bien que les organismes des commutateurs soient carrés, les broches ressortent à partir des côtés opposés du commutateur. Il s'agit que les broches seront suffisamment éloignés lorsqu'ils sont mis en place comportement sur la carte de prototypage.

N'oubliez pas que la LED doit avoir la borne négative plus courte vers la droite.

Code

Chargez la sketche à votre carte UNO. En appuyant sur le bouton supérieur va allumer LED lors qu'en appuyant sur le bouton inférieur va l'éteindre.

La première partie de l'sketche définit trois variables pour les trois broches à utiliser. "ledPin" est la broche de sortie et "ButtonApin" va consulter le commutateur près du haut de la carte de prototypage et 'buttonBpin' à l'autre interrupteur.

La fonction 'setup' définit ledPin étant une sortie normale, mais maintenant, nous avons les deux entrées à traiter. Dans ce cas, on utilise de l'ensemble pinMode comme 'INPUT_PULLUP', comme suivant:

```
pinMode(buttonApin, INPUT_PULLUP);  
pinMode(buttonBpin, INPUT_PULLUP);
```

Le pinMode d'INPUT_PULLUP signifie que la broche doit être utilisée comme une entrée, mais si rien d'autre n'est connecté à l'entrée, il devrait être 'pulled up' à HIGH. En d'autres termes, la valeur par défaut pour l'entrée est HIGH, sauf qu'il est mis à LOW par l'action en appuyant sur le bouton.

C'est pourquoi les commutateurs sont connectés à la masse. Lorsque le commutateur est enfoncé, il connecte la broche d'entrée à la masse, de sorte qu'il n'est plus HIGH.

Parce que l'entrée est générale en HIGH et elle ne va être LOW que quand le bouton est enfoncé, la logique est un peu à l'envers. Nous allons traiter ceci dans la fonction "Loop"

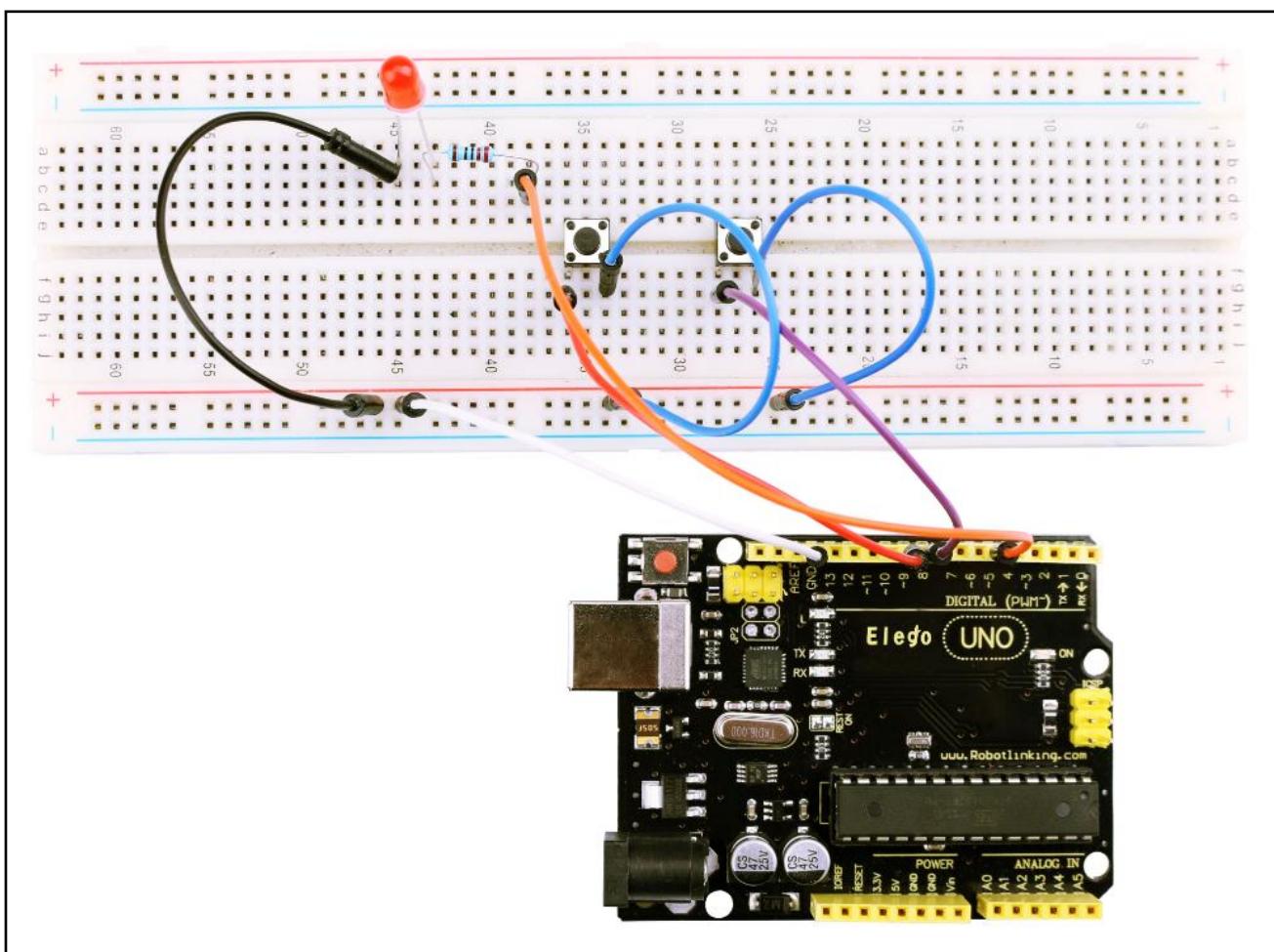
```
void loop()  
{  
    if (digitalRead(buttonApin) == LOW)  
    {  
        digitalWrite(ledPin, HIGH);  
    }  
}
```

```
if (digitalRead(buttonBpin) == LOW)
{
    digitalWrite(ledPin, LOW);
}
}
```

Dans la fonction "Loop", il y a deux déclarations "if", une pour chaque bouton. Chaque un fait un "digitalRead" sur l'entrée appropriée.

N'oubliez pas que si le bouton est enfoncé, l'entrée correspondante sera en LOW. Lorsque le bouton A est en LOW, et "digitalWrite" dans ledPin active.

De la même façon, lorsque le bouton B est enfoncé, LOW est écrit à ledPin.



Leçon 6: Huit LEDs avec 74HC595

Vue d'ensemble

Dans cette leçon, vous pouvez apprendre comment utiliser huit LEDs en rouge de grande taille avec un UNO sans devoir renoncer à 8 broches de sortie!

Bien que vous pourriez brancher huit voyants de chaque avec une résistance à une broche d'UNO que vous pouvez rapidement commencer à activer les broches sur votre UNO. Si vous n'avez pas beaucoup de trucs connectés à votre UNO. C'est OK pour le faire - mais souvent, nous voulons les boutons, les capteurs, les servos, etc. et vous ne laissez aucune broche avant que vous le savez. Alors, au lieu de cela, vous allez utiliser une puce appelée la série 74HC595 au convertisseur parallèle. Cette puce dispose huit sorties (parfait) et trois entrées que vous utilisez pour passer des données dans un bit à la fois.

Cette puce rend un peu plus lent à conduire les voyants (vous pouvez uniquement changer les LEDs environ 500,000 fois par seconde seconde au lieu de 8,000,000 par seconde), mais c'est toujours un moyen vraiment plus rapide que les humains peuvent détecter, il est donc en vaut vraiment la peine!

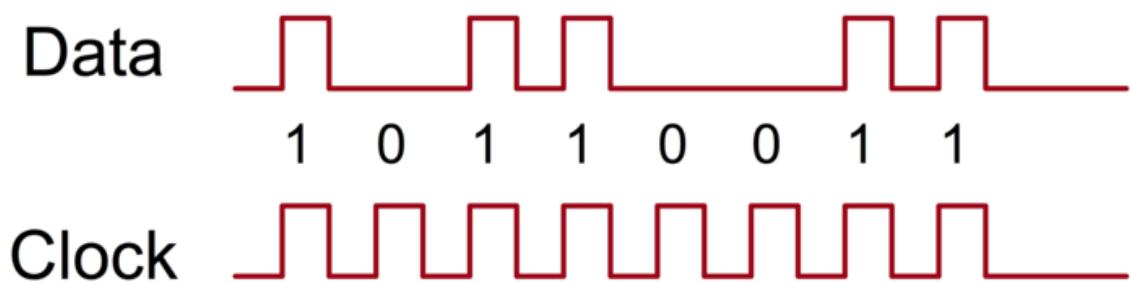
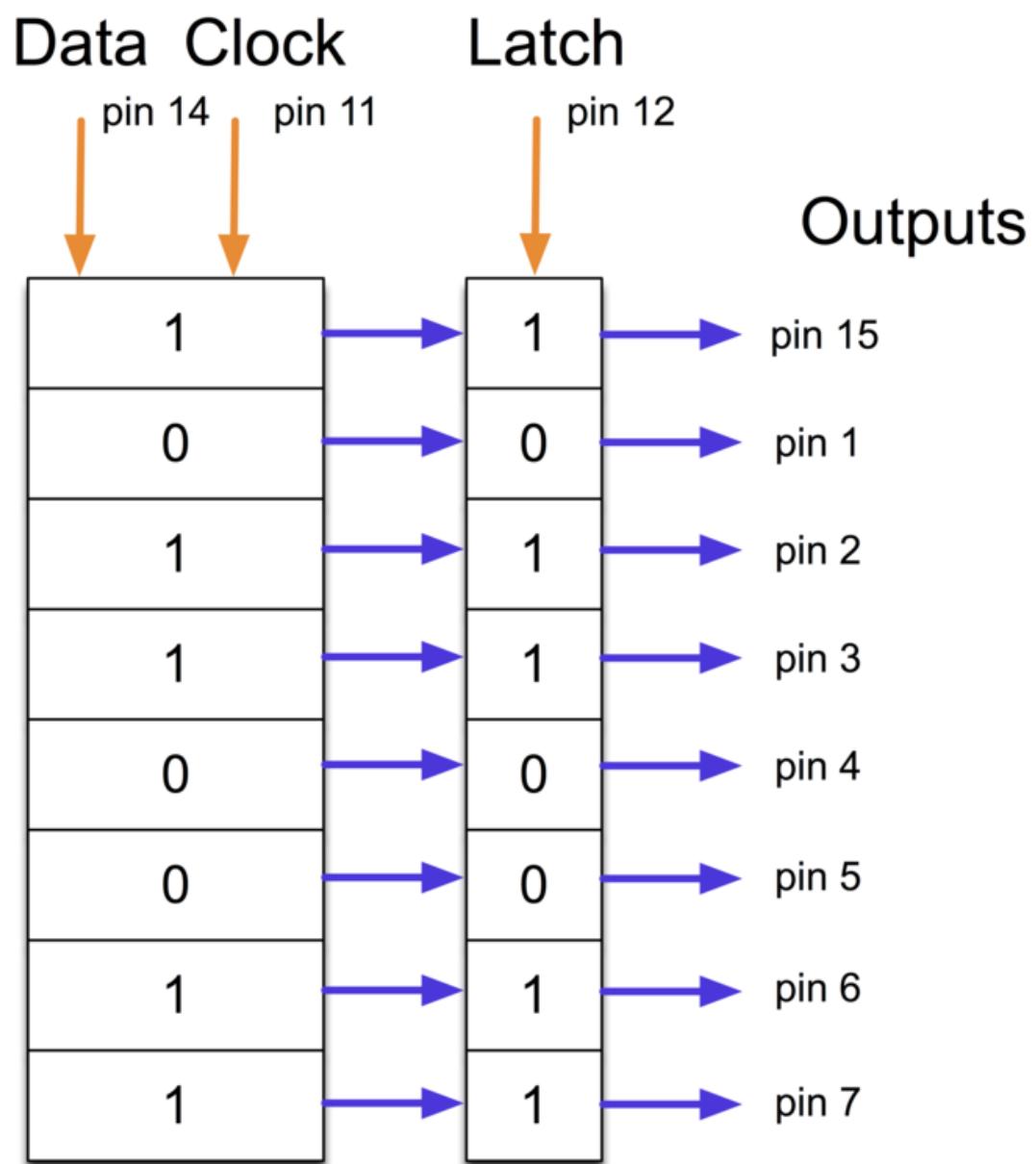
Composants requis

- (1) Elegoo UNO R3
- (1) Carte de prototypage
- (8) LEDs
- (8) 220 ohm résistances
- (1) 74HC595 IC
- (14) M-M câbles

Introduction de composant

Registre à décalage 74HC595:

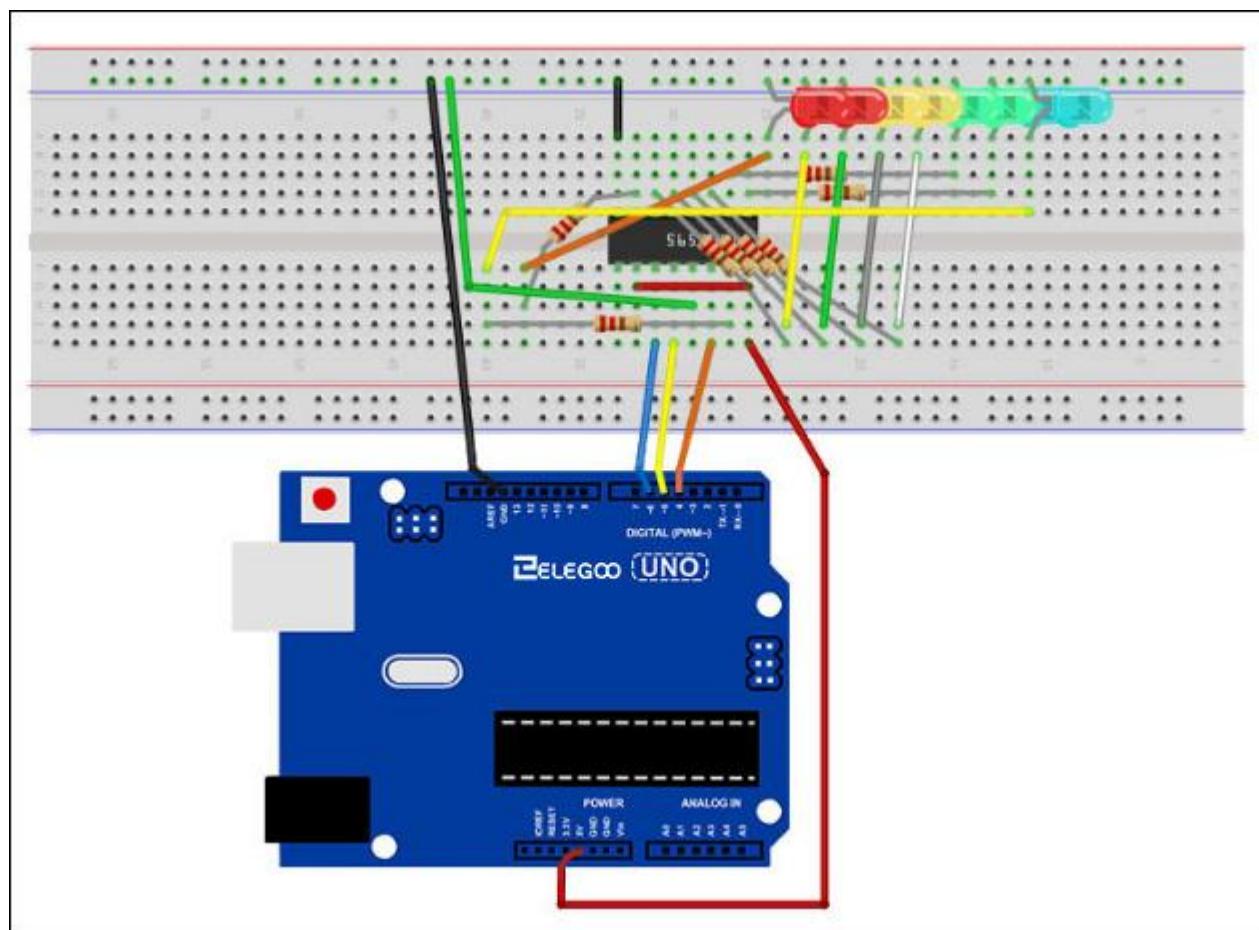
Le registre à décalage est un type de puce qui détient ce qui peut être considéré comme huit emplacements de mémoire. Chacun peut être un 1 ou un 0. Pour mettre chacun des valeurs en actif ou inactif, nous fournissons les données en utilisant les broches 'Data' et 'Clock' de la puce.



La broche horloge doit recevoir huit impulsions. À chaque impulsion, si la broche de données est élevée, un 1 est poussé dans le registre à décalage; sinon, un 0. Lorsque les huit impulsions ont été reçues, il permet de copier ces huit valeurs au registre à verrouillage. C'est nécessaire; sinon, la mauvaise LED va clignoter lorsque les données sont chargées au registre à décalage. La puce est également équipée d'une broche de sortie disponible(OE), qui est utilisé pour activer ou désactiver les sorties tous à la fois. Vous pouvez le raccorder à une PWM-capable UNO broche, et utiliser "analogWrite" pour contrôler la luminosité de LED. Cette broche est active en LOW, alors on le met à la terre.

Raccordement

Schéma de câblage



Car on a huit LED et huit résistances à brancher, il y a actuellement un certain nombre de connexions qui doivent être effectuées.

Il est probablement plus facile de mettre la puce 74HC595 tout d'abord, parce qu'à peu près tout le reste s'y connecte. Placez-le de sorte que la petite encoche en forme U est vers le haut de la carte de prototypage. La broche 1 de la puce est à la gauche de cette encoche.

- Numérique 4 à partir de l'UNO vers la borne #14 du registre à décalage
- Numérique 5 à partir de l'UNO vers la borne #12 du registre à décalage
- Numérique 6 à partir de l'UNO vers la borne #11 du registre à décalage

Il n'existe qu'une des sorties de l'IC sur le côté gauche de la puce. Par conséquent, pour faciliter la connexion, c'est aussi l'endroit où les LEDs sont.

À l'arrière de la puce, mettez les résistances en place. Vous devez veiller à ce qu'aucun fil des résistances se touche les uns les autres. Vous devez le vérifier à nouveau avant que vous connectez le câble d'alimentation à votre UNO. Si vous éprouvez des difficultés à organiser les résistances sans leur fils de toucher, puis il permet de raccourcir les fils de façon à ce qu'ils mentent plus près de la surface de la carte de prototypage.

Ensuite, placez les voyants sur la carte de prototypage. Les fils positifs plus longues de LED doivent tous être vers la puce, pour n'importe quel côté de la carte de prototypage qu'ils soient.

Fixez les câbles de raccordement comme illustré ci-dessus. N'oubliez pas celui qui va de la broche 8 de l'IC à la colonne de GND de la carte de prototypage.

Chargez la sketche énumérée un peu plus tard et essayez-la. Chaque LED doit s'allumer à son tour jusqu'à ce que tous les voyants soient allumés, et puis ils s'éteignent en tout et le cycle se répète.

Code

La première chose à faire est de définir les trois broches que nous allons utiliser. Ils sont les sorties numériques d'UNO qui seront connectées aux broches à verrouillage, à horloge et de données de 74HC595.

```
int latchPin = 5;  
int clockPin = 6;  
int dataPin = 4;
```

Et puis, une variable appelée "leds" est défini. Cela servira à tenir le modèle dont les LEDs sont actuellement activés ou désactivé. Les données de type « byte » représentent des nombres à l'aide de huit bits. Chaque bit peut être activé ou désactivé, donc c'est parfait pour garder une trace que nos huit voyants sont actifs ou inactifs.

```
byte leds = 0;
```

La fonction 'setup' Il définit simplement les trois broches que nous utilisons pour être sorties numériques.

```
void setup()
```

```
{
```

```
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
```

```
}
```

La fonction 'loop' désactive initialement tous les voyants, en donnant la valeur 0 à la variable « leds ». Et puis il appelle 'updateShiftRegister' qui enverra le modèle 'leds' au registre à décalage pour que tous les voyants DEL s'éteignent. Nous allons traiter comment fonctionner 'updateShiftRegister' le plus tard.

La fonction Loop fait une pause pendant une demi-seconde et commence alors à compter de 0 à 7 à l'aide de la boucle 'for' et la variable 'i'. Chaque fois, il utilise la fonction d'Arduino 'bitSet' pour activer le bit qui contrôle si le LED dans la variable 'leds'. Et il appelle alors aussi 'updateShiftRegister' afin que les leds mise à jour pour refléter ce qui est dans la variable 'leds'. Alors, il existe une demi-seconde de regard avant que 'i' est incrémenté et le voyant suivant s'allume.

```
void loop()
```

```
{
```

```
  leds = 0;
  updateShiftRegister();
  delay(500);
```

```
for (int i = 0; i < 8; i++)  
{  
    bitSet(leds, i);  
    updateShiftRegister();  
    delay(500);  
}  
}
```

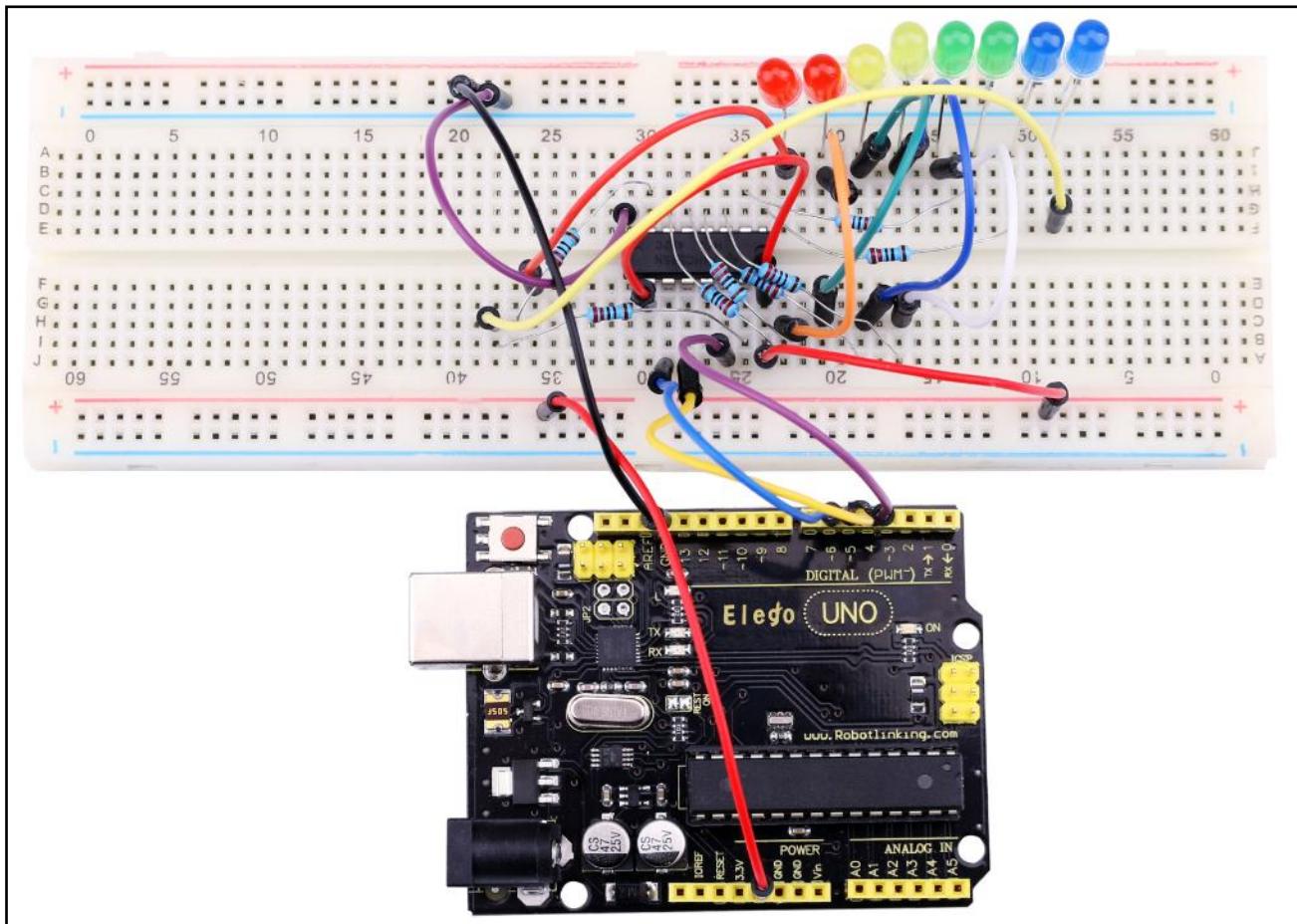
Tout d'abord, la fonction 'updateShiftRegister' met le latchPin en LOW, et puis la fonction 'shiftOut' d'UNO est appelée avant de mettre 'latchPin' en HIGH à nouveau. Cela prend quatre paramètres. Les deux premiers sont les broches à utiliser pour les données et horloge, respectivement.

Le troisième paramètre spécifie à quelle fin des données que vous souhaitez démarrer. Nous allons commencer avec le bit le plus à droite, qui est appelé comme 'Least Significant Bit' (LSB).

Le dernier paramètre correspond aux données réelles pour être déplacé dans le registre à décalage, c'est-à-dire, dans ce cas, qui est 'leds'.

```
void updateShiftRegister()  
{  
    digitalWrite(latchPin, LOW);  
    shiftOut(dataPin, clockPin, LSBFIRST, leds);  
    digitalWrite(latchPin, HIGH);  
}
```

Si vous avez voulu tourner l'une des DEL à éteindre plutôt qu'à allumer, vous devez appeler la fonction d'Arduino (bitClear) avec la variable 'leds'. Cela mettra le bit de 'leds' à 0 et vous devez alors juste de suivre avec un appel à « updateShiftRegister » pour mettre à jour les voyants actuels.



Leçon 7: Moniteur Sériel

Vue d'ensemble

Dans cette leçon, basée sur la leçon 6, en ajoutant le périphérique pour contrôler les LEDS à partir de votre ordinateur par le Moniteur Sériel d' Arduino. Le Moniteur Sériel est une 'attache' entre votre ordinateur et votre UNO. Il vous permet d'envoyer et de recevoir des messages texte, pratique pour le débogage et contrôler l'UNO depuis un clavier ! Par exemple, vous serez en mesure d'envoyer des commandes de votre ordinateur pour allumer les LEDs.

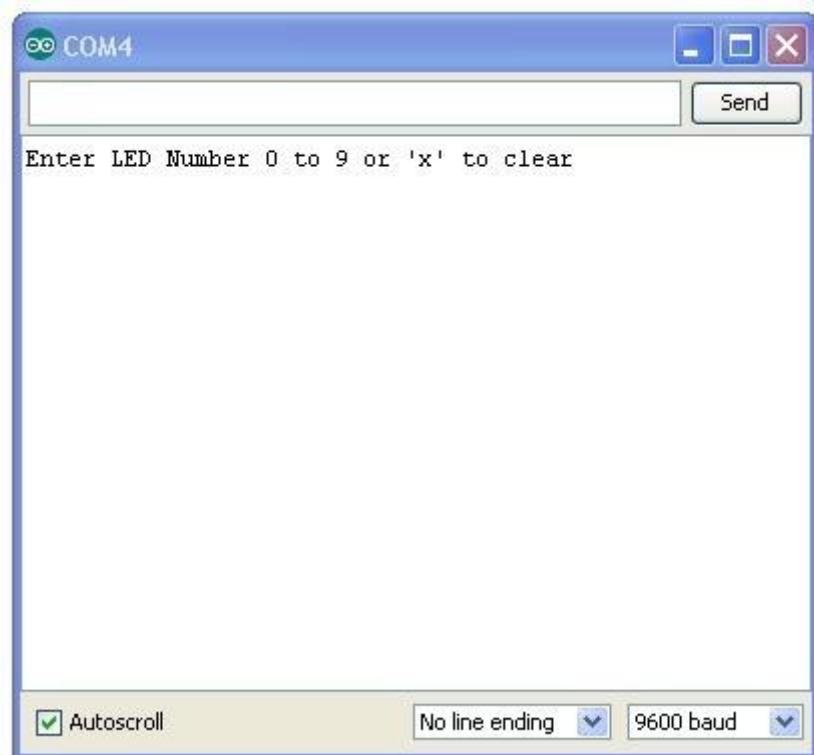
Dans cette leçon, vous allez utiliser exactement les mêmes pièces et une disposition similaire de la carte à prototypage montrée comme dans la leçon 6. Donc, si vous ne l'avez pas déjà fait, suivez maintenant leçon 6.

Étapes

Après que vous avez téléchargé cette sketche sur votre UNO, cliquez sur le bouton à droite sur la barre d'outils dans l'IDE d'Arduino. Le bouton est encerclé ci-dessous.



La fenêtre suivante s'ouvre.

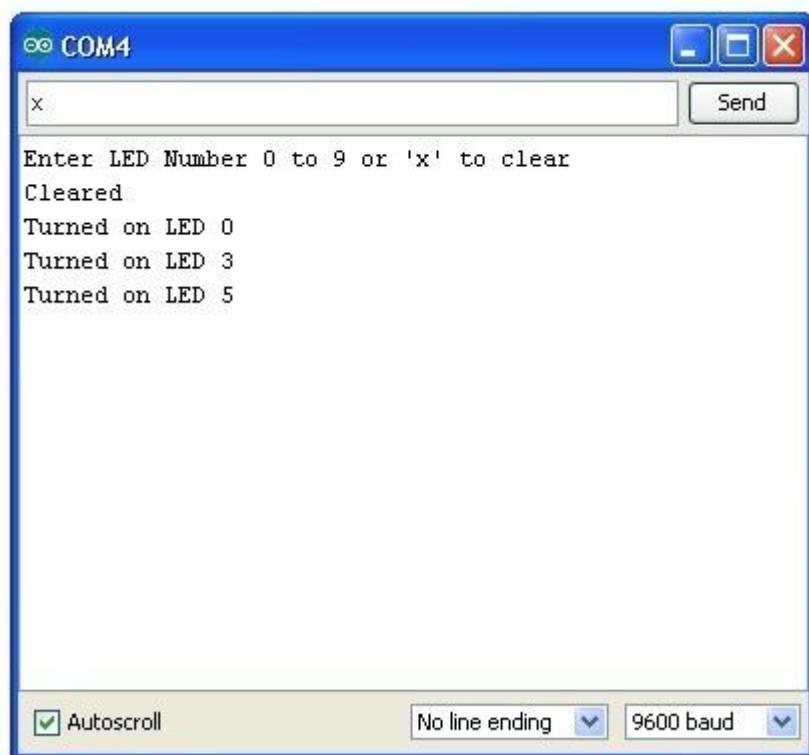


Cette fenêtre est appelée le Moniteur Sériel et il fait une partie du logiciel de l'IDE d'Arduino. Son travail est de permettre d'envoyer les messages depuis votre ordinateur à une carte UNO (via USB) et aussi à recevoir les messages de l'ONU.

Le message "Enter LED Number 0 to 9 or 'x' to clear" a été envoyé par l'Arduino. Il peut nous donner quelles commandes nous pouvons envoyer à l'Arduino : soit envoyer le 'x' (pour désactiver toutes les LEDs), soit le numéro de la LED vous voulez allumer (où 0 est le fond de LED, 1 est le prochain vous le haut, jusqu'à 7 pour la LED plus haut).

Essayez de taper les commandes suivantes dans la zone supérieure du Moniteur Sériel qui est de niveau au bouton 'Send'. Appuyez sur 'Send', après la saisie de chaque de ces caractères :
x 0 3 5

En tapant x sera indisponible si toutes les LEDS s'éteignent déjà, mais en raison d'entrer chaque numéro, la LED correspondante s'allume et vous pouvez avoir un message de confirmation à partir de la carte d'UNO. Le Moniteur Sériel s'affiche comme indiqué ci-dessous.



Tapez x à nouveau et appuyer sur 'Send' pour éteindre toutes les LEDs.

Code

Comme vous vous en doutez, la sketche est basé sur l'sketche utilisés dans la leçon 6. Alors, nous allons couvrir juste les nouveaux bits ici. Il vous sera utile de se référer à la sketche complet dans votre IDE d'Arduino.

Dans la fonction 'setup', il y a trois nouvelles lignes à la fin:

```
void setup()
{
    pinMode(latchPin, OUTPUT);
    pinMode(dataPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
    updateShiftRegister();
    Serial.begin(9600);
    while (! Serial); // Wait until Serial is ready - Leonardo
    Serial.println("Enter LED Number 0 to 7 or 'x' to clear");
}
```

Tout d'abord, nous avons la commande 'Serial.begin(9600)'. Cela démarre une communication en série, de sorte que l'UNO peut envoyer des commandes via la connexion USB. La valeur 9600 est appelé le 'baud rate' de la liaison. C'est la rapidité avec laquelle les données doivent être envoyées. Vous pouvez la modifier à une valeur plus élevée, mais vous devrez également modifier le Moniteur Sériel d'Arduino à la même valeur. Nous allons discuter sur cette section plus tard; pour le moment, laissez-le à 9600.

La ligne commence avec 'while' pour garantie qu'il existe quelque chose à l'autre extrémité de la liaison USB pour l'Arduino à appeler avant de commencer à envoyer les messages. Sinon, le message peut être envoyé, mais ne s'affiche pas. Actuellement, cette ligne est nécessaire si vous utilisez un Leonardo Arduino, puisque l'UNO d'Arduino initialise automatiquement la carte d'Arduino lorsque vous démarrez le Moniteur Sériel, alors que ce n'est pas le cas avec le programme Leonardo.

La fin de ces nouvelles lignes dans 'setup' envoie le message ce que on a vu en haut du Moniteur Sériel.

La fonction 'loop' existe où toute l'action se passe:

```
void loop()
{
    if (Serial.available())
    {
        char ch = Serial.read();
        if (ch >= '0' && ch <= '7')
        {
            int led = ch - '0';
            bitSet(leds, led);
            updateShiftRegister();
            Serial.print("Turned on LED ");
            Serial.println(led);
        }
        if (ch == 'x')
```

```
{  
    leds = 0;  
    updateShiftRegister();  
    Serial.println("Cleared");  
}  
}  
}
```

Tout ce qui se passe à l'intérieur de la boucle est contenue dans la déclaration " si ". Et à moins que l'appel à la fonction intégrée de l'Arduino 'Serial.available()' est ' true ', puis rien n'arrivera. Serial.available() va retourner 'true' si les données sont bien envoyées à l'UNO, et s'il existe des données prêtes à être traitées. Les messages entrants sont conservés dans ce qu'on appelle un tampon et Serial.available() renvoie true si le tampon n'est pas vide.

Si le message a été reçu, alors il déplace à la prochaine ligne de code:

```
char ch = Serial.read();
```

Cette ligne de code lit le caractère prochain du tampon, et l'enlève à partir du tampon. Il également assigne à la variable ' ch '. La variable ' ch ' est en type ' char ' en étant l'acronyme de ' caractère '. Montré comme son nom, il est un caractère unique.

Si vous avez suivez les instructions dans l'invite de commande en haut du Moniteur Sériel, ce caractère sera soit un chiffre entre 0 et 7 ou la lettre ' X '.

Et la déclaration ' if ' dans la ligne prochaine vérifie s'il est un chiffre unique en voyant si " ch " est supérieur ou égal au caractère " 0 " et inférieur ou égal au caractère ' 7 '. Il semble un peu étrange de comparaison des caractères de cette façon, mais c'est parfaitement acceptable.

Chaque caractère est représenté par un chiffre unique, appelé sa valeur ASCII. Il s'agit que lorsque l'on compare les caractères en utilisant <= et >=, c'est actuel les valeurs ASCII à comparer.

Lorsqu'il passe le test, on va arriver la ligne prochaine:

```
int led = ch - '0';
```

Maintenant, on effectue les opérations arithmétiques sur les caractères ! Nous sommes en

soustrayant le chiffre '0' de n'importe quel chiffre a été saisi. Alors, si vous entrez '0' et puis '0' – '0' va être égal à 0. Si vous entrez '7' et puis '7' – '0' va être égal à 7, puisqu'il est actuel les valeurs ASCII qui vont être utilisés dans la soustraction.

Parce que l'on sait le numéro de LED ce qui on va allumer, on a uniquement besoin de définir le bit dans la variable 'leds' et mettre à jours le registre à décalage.

```
bitSet(leds, led);
```

```
updateShiftRegister();
```

Les deux lignes suivantes retournent un message de confirmation au Moniteur Sériel.

```
Serial.print("Turned on LED ");
```

```
Serial.println(led);
```

La première ligne utilise Serial.print plutôt que Serial.println. La différence entre les deux lignes est que Serial.print ne commence pas à partir d'une nouvelle ligne après d'imprimer à tout ce qui est en son paramètre. Nous l'utilisons dans la première ligne parce que nous imprimons le message en deux parties: le bit général 'Turned on LED' et puis le numéro de la LED.

Le numéro de la LED est mis dan une variable 'int' plutôt que d'être une chaîne de texte. Serial.print peut prendre une chaîne de texte entre guillemets, 'int, ou à peu près n'importe quel type de variable.

Après la déclaration 'if' qui gère le cas, lorsque un chiffre unique a été mis dedans, il existe une deuxième déclaration 'if' pour vérifier si 'ch' est une mettre 'x'.

```
if (ch == 'x')  
{  
    leds = 0;  
    updateShiftRegister();  
    Serial.println("Cleared");  
}
```

Si oui, il va effacer toutes les LEDs et envoyer un message de confirmation.

Leçon 8: Cellule photoélectrique

Vue d'ensemble

Dans cette leçon, Vous pouvez apprendre comment mesurer l'intensité de la lumière à l'aide d'une entrée analogique. Vous se fonderez sur le circuit et utiliserez le niveau de lumière permettent de contrôler le nombre de LED à allumer.

La cellule photoélectrique est en bas de la carte de prototypage, où le pot est à la partie supérieure.

Composants requis

- (1) Elegoo UNO R3
- (1) Carte de prototypage
- (8) LEDs
- (8) 220 ohm résistances
- (1) 1k ohm résistances
- (1) 74HC595 IC
- (1) Cellule photoélectrique
- (14) M-M câbles

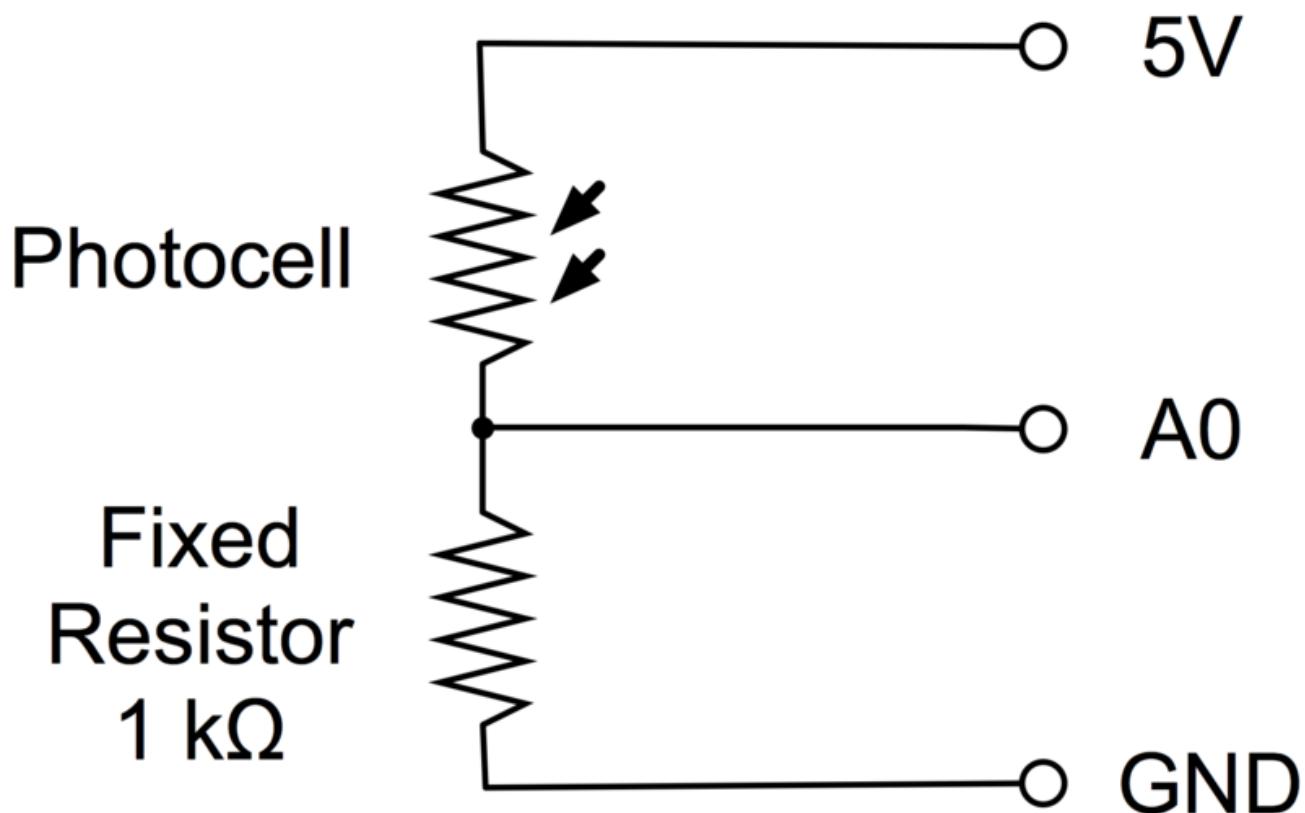
Introduction de composant

CELLULE PHOTOÉLECTRIQUE:

Cette cellule photoélectrique utilisée est d'un type appelé la photorésistance(LDR). Montré comme son nom, ce composent joue un rôle comme une résistance, sauf que la résistance peut changer en réponse à la quantité de lumière qui tombe sur elles.

Il a une résistance d'environ $50\text{ k}\Omega$ dans près de ténèbres et de $500\ \Omega$ en pleine lumière. Pour convertir cette valeur variable de résistance en quelque chose, nous pouvons mesurer sur les entrées analogiques de la carte de l'UNO R3, elle doit être convertie en une tension.

La façon plus facile à faire est à combiner avec une résistance fixe.



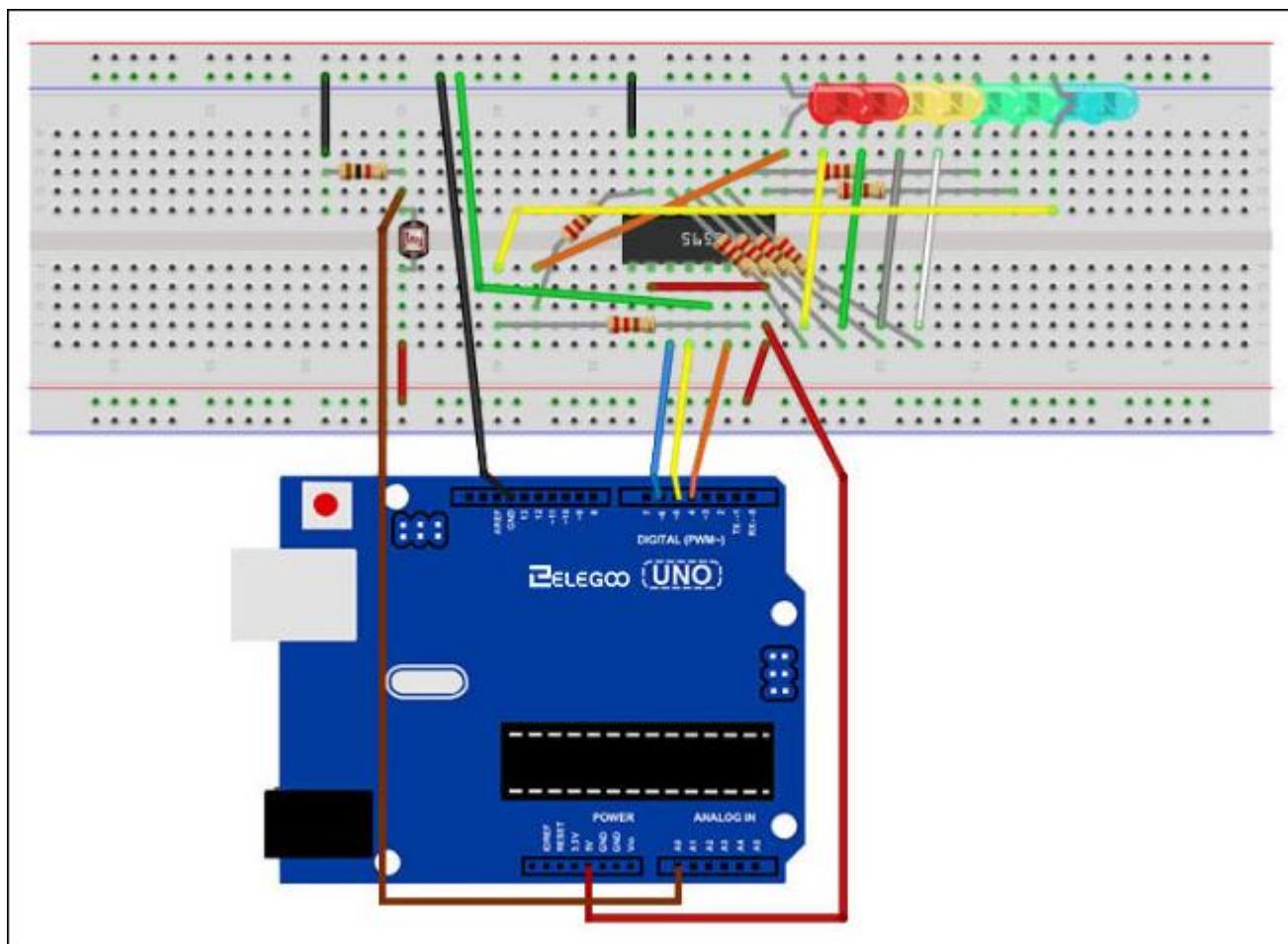
La résistance et la Cellule photoélectrique ensemble sont comportées comme un pot. Lorsque la lumière est très brillante, puis la résistance de la Cellule photoélectrique est très faible par rapport à la résistance de valeur fixe, et alors c'est comme si le pot ont été transformés au maximum.

Lorsque la cellule photoélectrique est en lumière terne, la résistance est plus grande que la résistance fixe 1 kΩ et c'est comme si le pot a été tourné vers la masse.

Chargez la sketche montrée dans la prochaine section, essayez de couvrir la cellule photoélectrique avec votre doigt, et puis maintenez-la à approcher la source de lumière.

Raccordement

Schéma de câblage



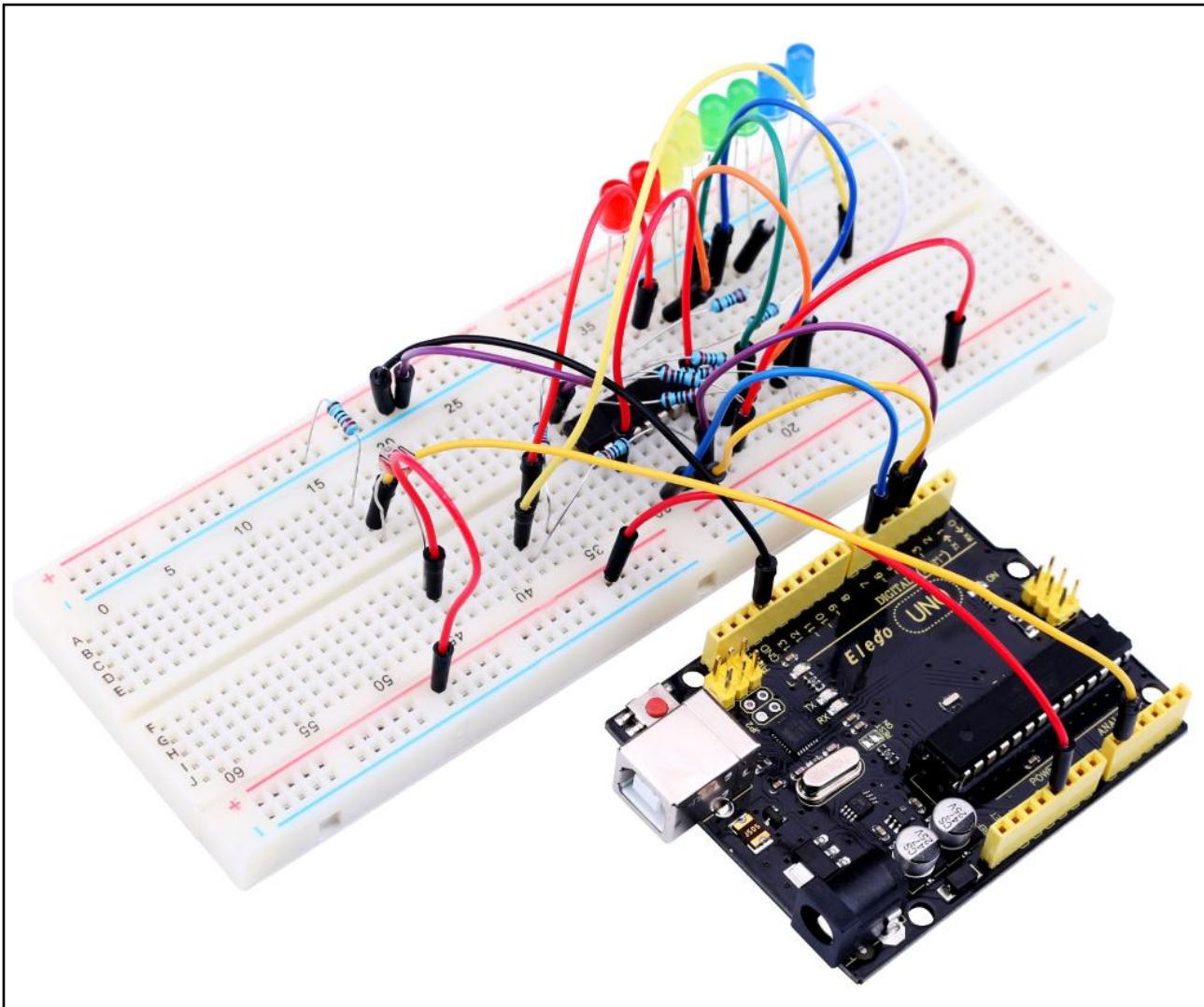
Code

Il doit noter tout d'abord que nous avons modifiés le nom de la broche analogique comme 'lightPin' plutôt que 'potPin', puisque plus d'un pot n'est connecté.

Un seul changement substantiel à la sketche est la ligne à calculer combien de LEDs à s'allumer:

```
int numLEDSLit = reading / 57; // all LEDs lit at 1k
```

À ce moment, on divise la lecture brute par 57 plutôt que 114. C'est-à-dire que nous le divisons par moitié comme autant que nous l'avons fait avec le pot à diviser en neuf zones, d'aucune LED allumée à huit allumé. Ce facteur supplémentaire doit tenir compte de la résistance fixe d'1 kΩ. Il s'agit que lorsque la cellule photoélectrique possède une résistance d'1 kΩ (De la même façon qu'une résistance fixe), la lecture brute sera $1023 / 2 = 511$. Ceci correspondra à toutes les LED s'allumants et puis un bit (numLEDSLit) sera 9.



Leçon 9: Prise de sons

Vue d'ensemble

Dans cette leçon, vous pouvez apprendre à générer un son à un buzzer actif.

Composants requis

- (1) Elegoo UNO R3
- (1) Buzzer Actif
- (2) F-M câbles

Introduction de composant

BUZZER:

Les buzzers électroniques sont alimentés en courant continu et équipés avec un circuit intégré. Ils sont largement utilisés dans les ordinateurs, les imprimantes, les photocopieurs, les alarmes, les jouets électroniques, les dispositifs électroniques automobiles, les téléphones, les temporiseurs et d'autres produits électroniques pour les périphériques voix. Tournez les broches de deux buzzers vers le haut. Un avec une carte verte de circuit est le buzzer passif, en plus, l'autre clôturé avec une bande noire est le buzzer actif.

La différence entre deux buzzers est que le buzzer actif possède une source oscillante intégrée, donc il va produire une voix lors d'être électrifié. Le buzzer passif ne possède pas de source comme ça, alors il ne peut pas sonner si les signaux DC sont utilisés; Par contre, vous devez utiliser les ondes carrées avec la fréquence comprise entre 2K et 5K pour son entraînement. Le buzzer actif est souvent plus cher que le buzzer passif en raison des multiples circuits oscillants intégrés.

Raccordement

Schéma

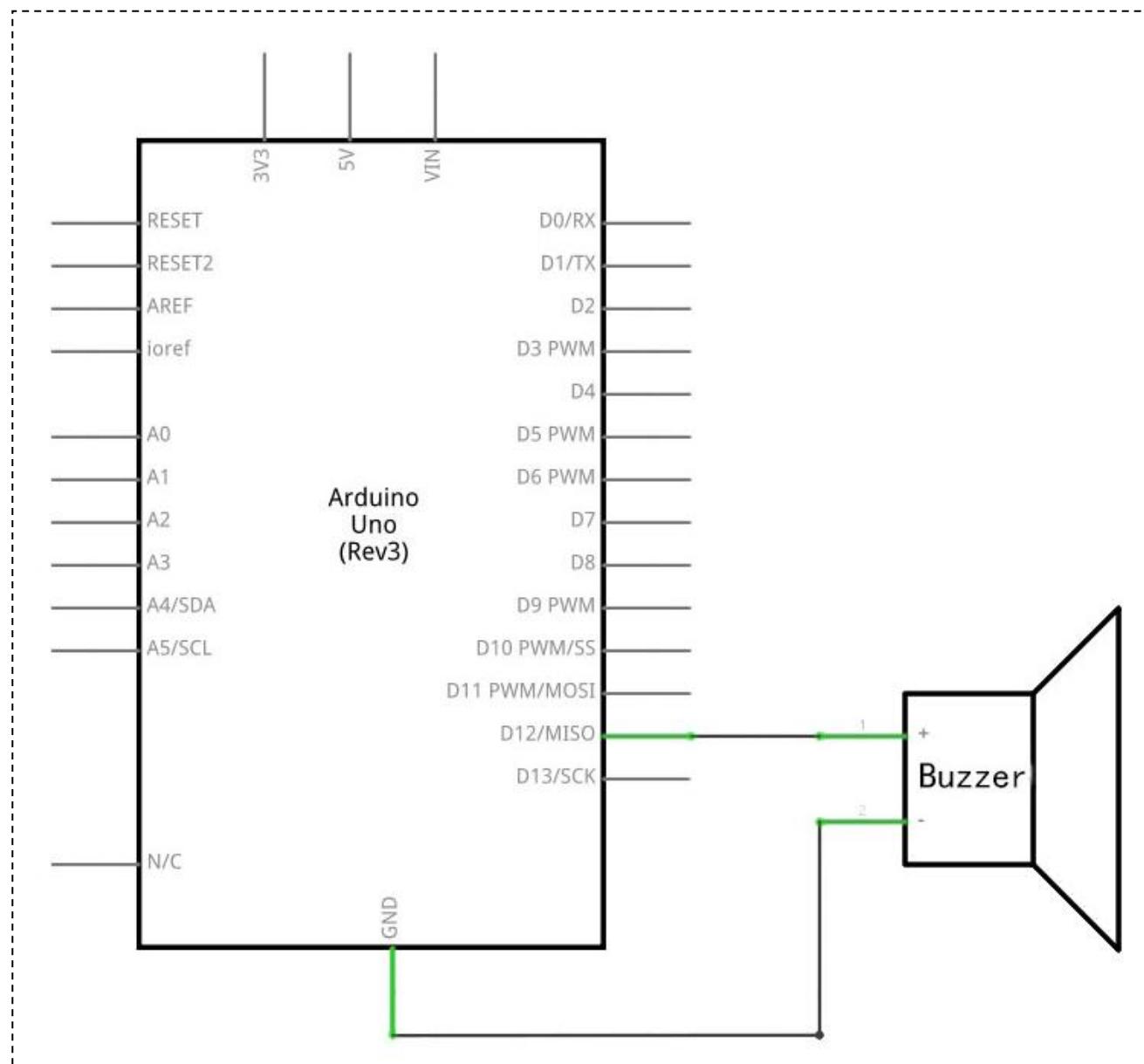
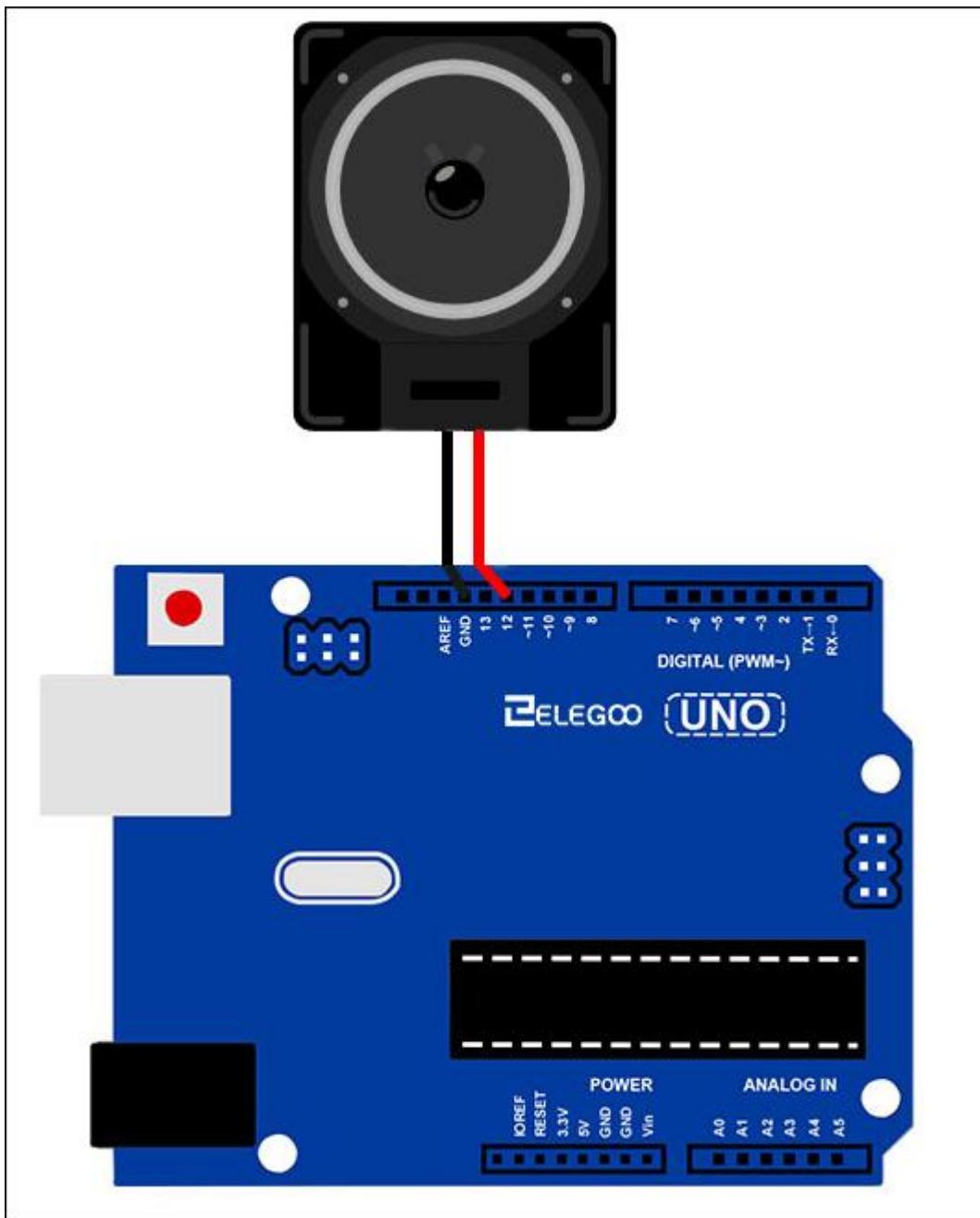
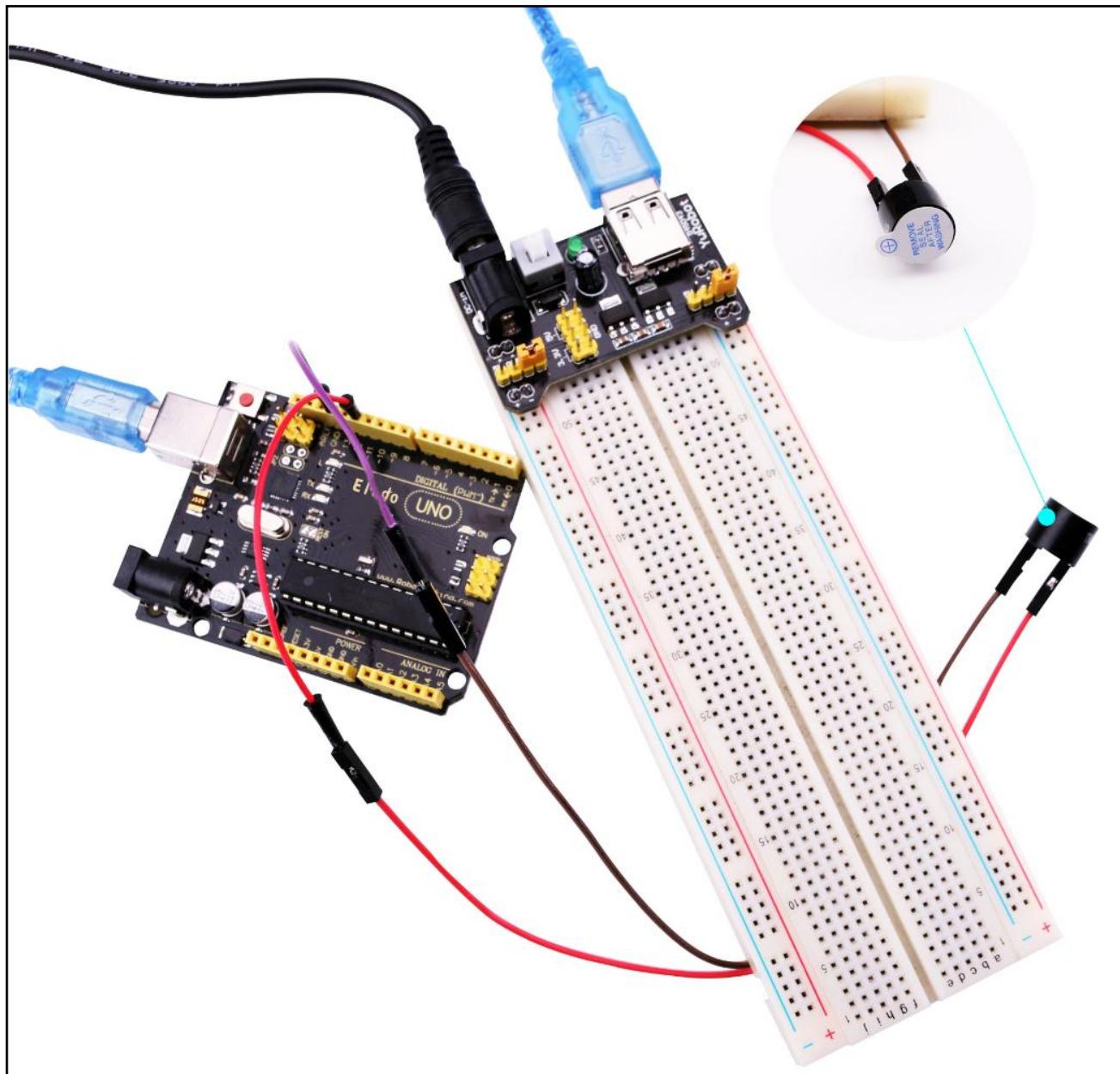


Schéma de câblage



Code

Veuillez voir le fichier de Code.



Leçon 10: Buzzer Passif

Vue d'ensemble

Dans cette leçon, vous pouvez apprendre comment utiliser le buzzer passif.

Le but de l'expérience est de générer huit sons différents, chaque son ayant une durée de 0,5 secondes:

À partir d'Alto Do (523Hz), Re (587Hz), Mi (659Hz), Fa (698Hz), So (784Hz), La (880Hz), Si (988Hz) à Treble Do (1047Hz).

Composants requis

(1) Elegoo UNO R3

(1) Buzzer passif

(2) F-M câbles

Introduction de composant

Buzzer Passif:

Le principe de fonctionnement du Buzzer Passif est à utiliser PWM à générer un audio pour rendre l'air de vibrer.

Modifié de façon appropriée tant que la fréquence de vibration, il peut générer plusieurs voix différents. Par exemple, il peut générer Alto Do en transmettant une impulsion de 523Hz, et midrange Re avec l'impulsion de 587 Hz, et midrange Mi avec l'impulsion de 659Hz. Vous pouvez jouer une chanson par le buzzer.

Nous devrions être attentifs à ne pas utiliser la fonction analog Write () de la carte d'UNO R3 à générer une impulsion sur le buzzer, puisque la sortie d'impulsion de la fonction analog Write () est fixée (500 Hz).

Raccordement

Schéma

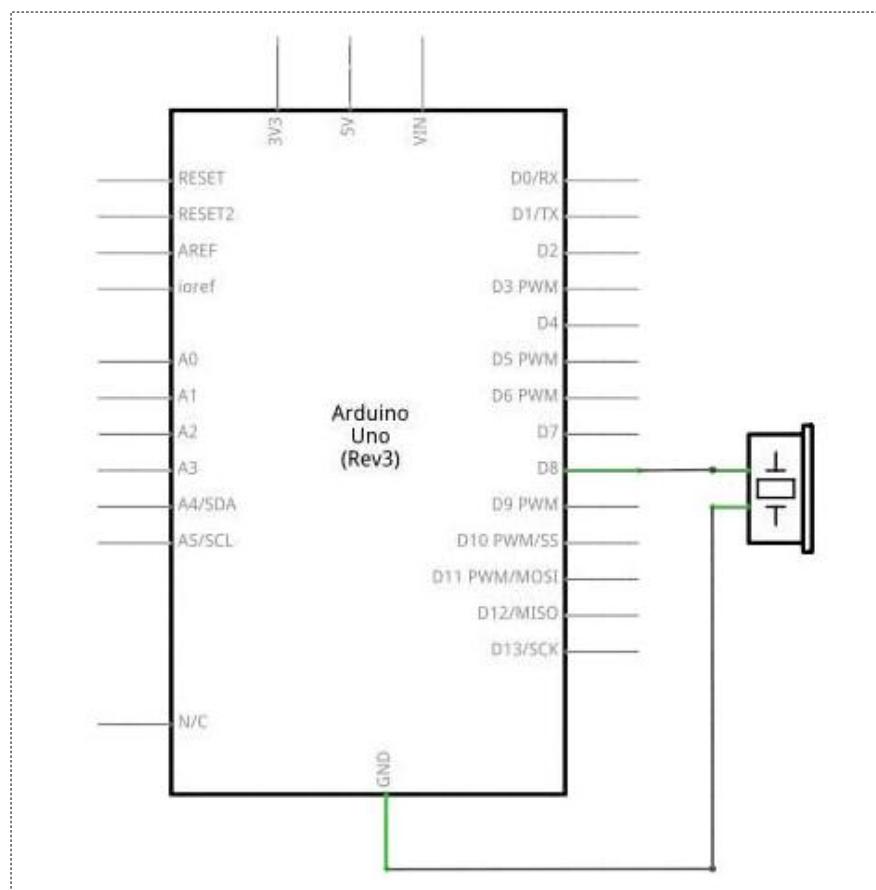
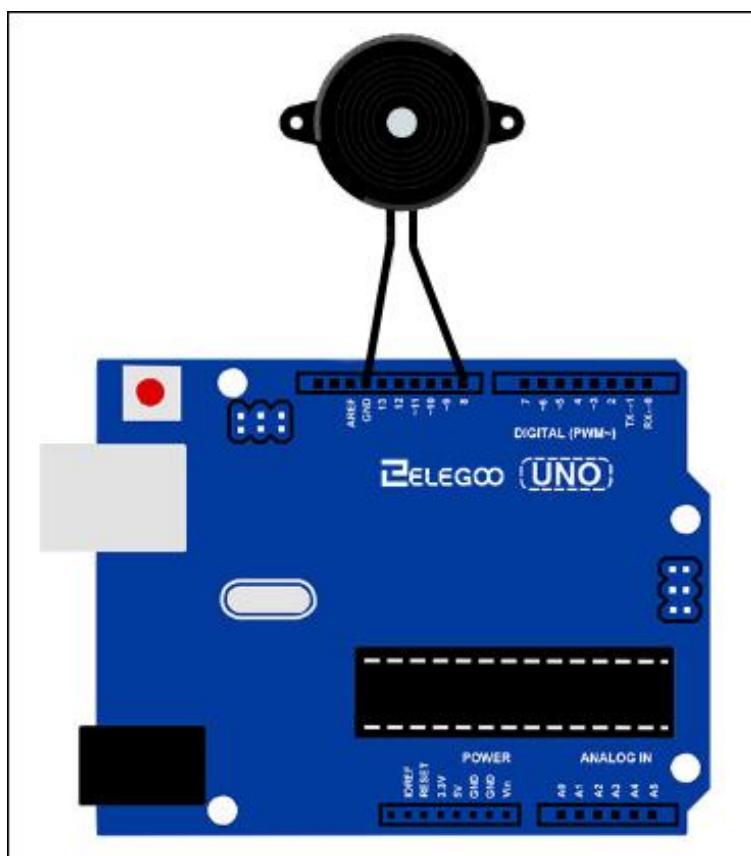


Schéma de câblage



Câblez le buzzer à la carte d'UNO R3, et le câble rouge (positif) à la broche 8, le câble noir (négatif) mis à la terre.

Code

Avant d'effectuer le programme, veuillez s'assurer que vous avez bien installé la bibliothèque <pitches>, et réinstallez-la en cas nécessaire. Sinon, votre code n'exécute pas.

Leçon 11: Contacteur à bille

Vue d'ensemble

Dans cette leçon, vous pouvez apprendre comment utiliser un contacteur à bille pour détecter le petit angle d'inclinaison.

Composants requis

(1) Elegoo UNO R3

(1) Contacteur à bille

(2) F-M câbles

Introduction de composant

Capteur d'inclinaison:

Les capteurs d'inclinaisons vous permettent de détecter l'orientation ou l'inclinaison. Ils sont petits, pas cher, faible consommation d'énergie et facile à utiliser. S'il est utilisé correctement, ils ne vont pas à l'usure. Leur simplicité les rend populaires pour les jouets, les gadgets et les équipements. Parfois, ils sont dénommés "interrupteurs au mercure ", " commutateurs d'inclinaison" ou " capteurs de balle roulant " pour des raisons évidentes.

Ils sont souvent constitués d'une cavité quelconque (souvent sous une forme de cylindrique, mais pas toujours) avec une masse conductrice libre à l'intérieur, comme un blob de mercure ou une boule de roulement. Une extrémité de la cavité possède deux éléments conductrices (pôles). Lorsque le capteur est orienté afin que cette extrémité soit vers le bas, la masse roule au pôle et met plus courte, comme une un jet de commutateur.

Tandis que pas aussi précis ou flexible comme un accéléromètre complet, les commutateurs d'inclinaison peuvent détecter la motion ou l'orientation.

D'autre bénéfice est que les plus grands peuvent basculer le pouvoir eux-mêmes. D'autre part, les accéléromètres peuvent sortir le chiffre ou la tension analogique qui doit alors être analysé en utilisant un circuit supplémentaire.

Raccordement

Schéma

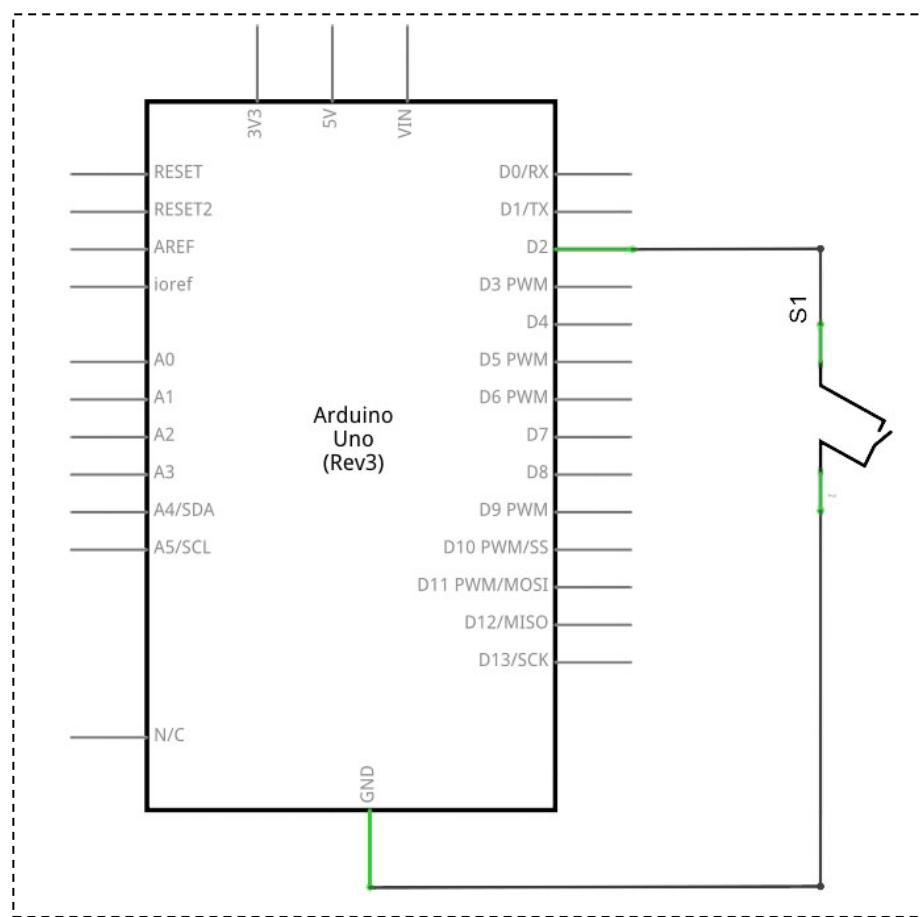
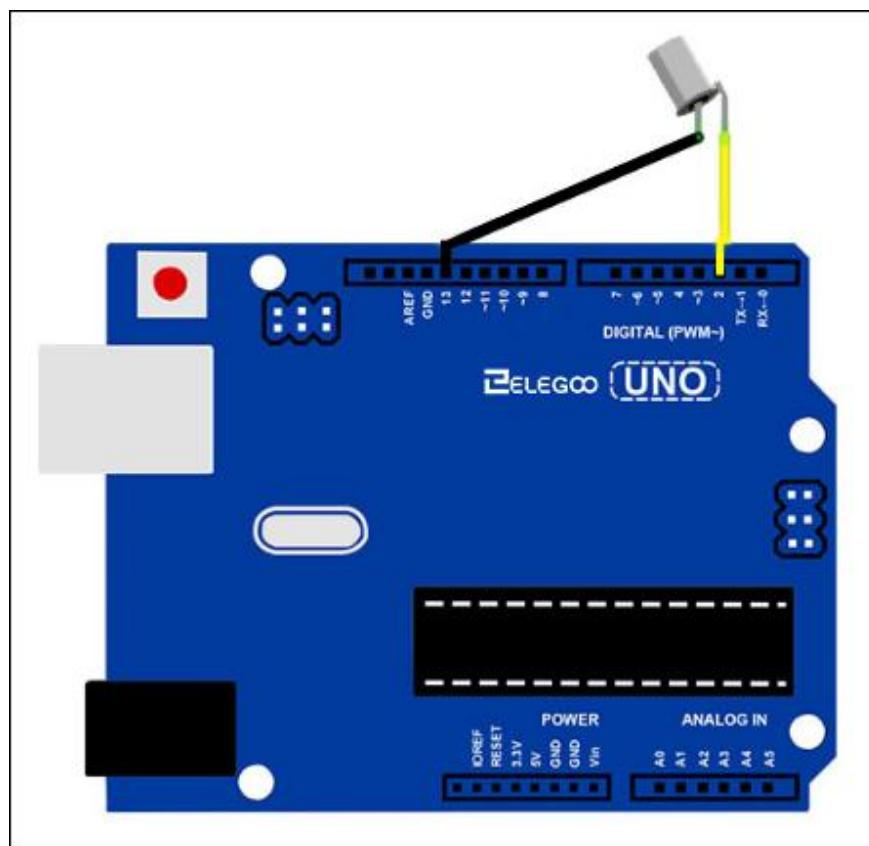
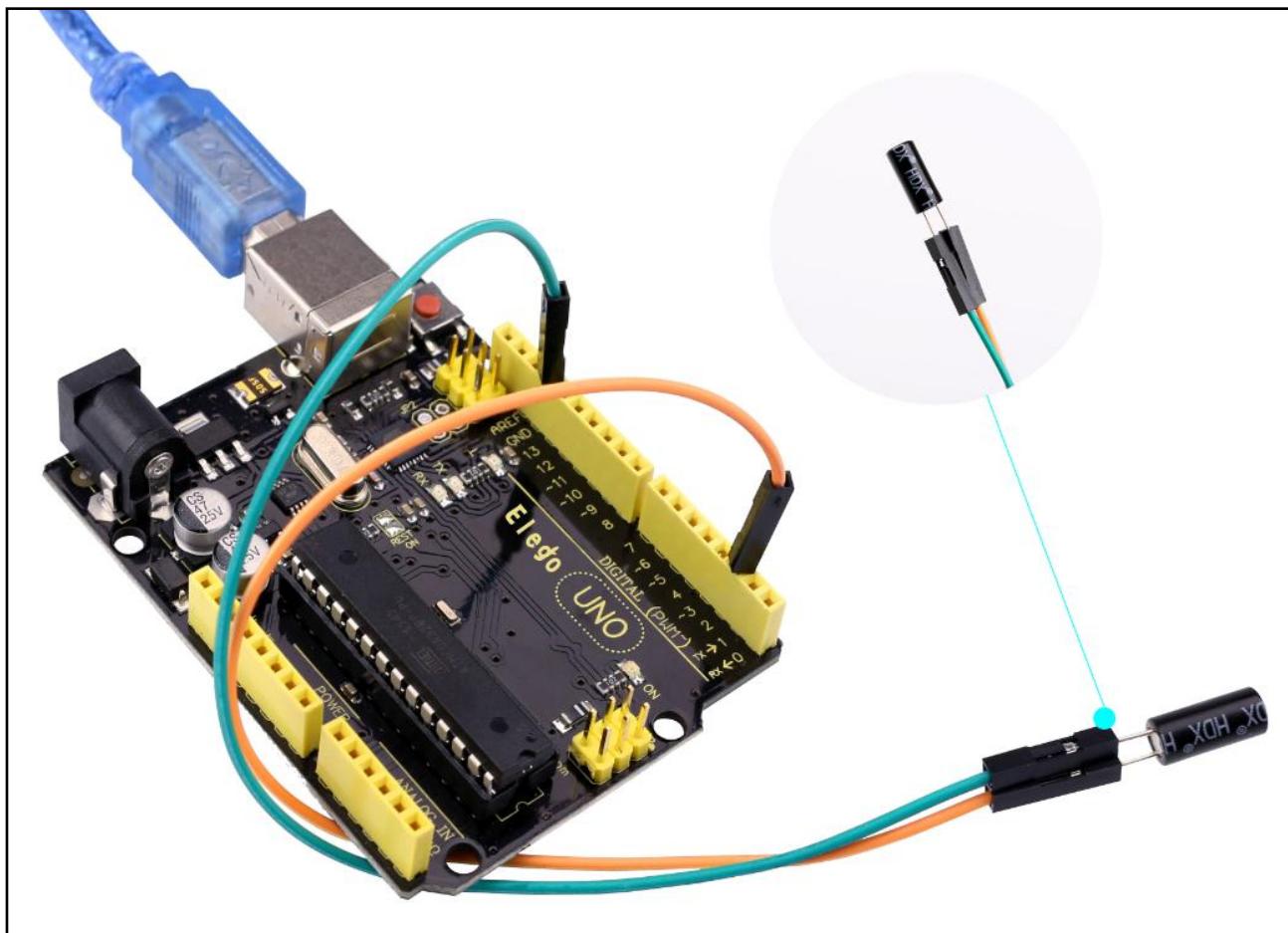


Schéma de câblage



Code

Veuillez voir le fichier de code.



Leçon 12: Relais

Vue d'ensemble

Dans cette leçon, vous pouvez apprendre comment utiliser un relais.

Composants requis

- (1) Elegoo UNO R3
- (1) Carte de prototypage
- (1) PN2222
- (1) 1N4007
- (1) 220 ohm résistance
- (1) 6v DC moteur
- (1) Relais
- (1) module d'alimentation de la Carte de prototypage
- (1) 9v Adaptateur d'alimentation
- (8) M-M câbles

Introduction de composant

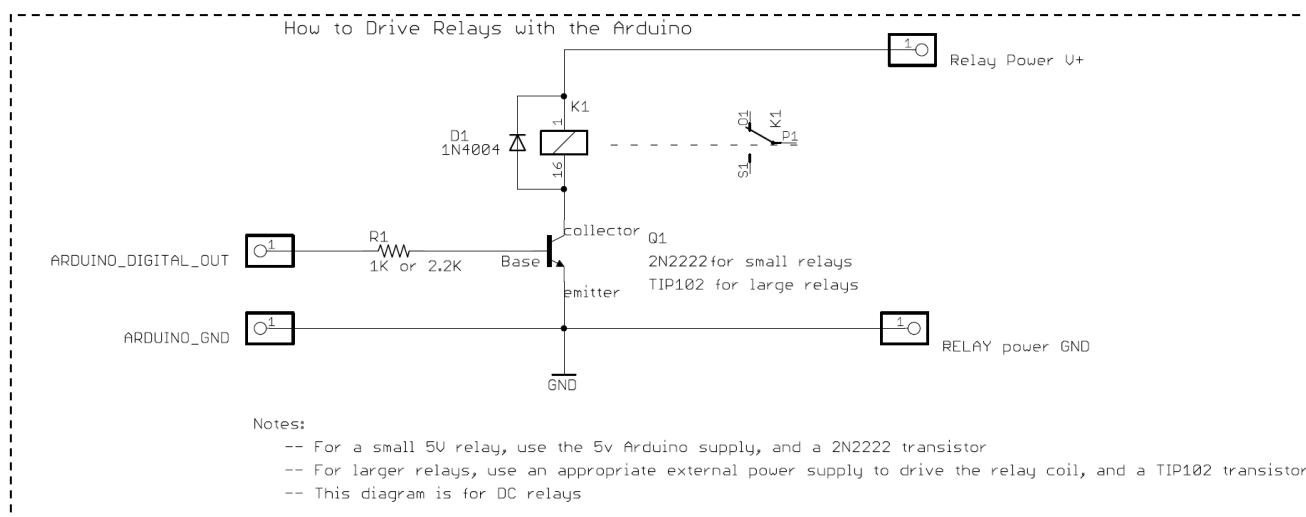
Relais:

Un relais est un commutateur actionné électriquement. Certains Relais utilisent un électro-aimant pour actionner mécaniquement un commutateur, mais les autres principes de fonctionnement sont également utilisés comme les relais à l'état solide. Les relais sont utilisés où il est nécessaire de contrôler un circuit par un signal de faible puissance (avec l'isolation électrique complète entre les circuits de commande et contrôlés), ou où plusieurs circuits doivent être commandés par un signal. Les premiers relais ont été utilisés dans les circuits de télégraphe sur de longues distances comme les amplificateurs. Ils rappellent le signal en provenance d'un circuit et retransmettent ce signal à l'autre circuit. Les relais ont été utilisés largement dans les centraux téléphoniques et les ordinateurs précédents pour exécuter les opérations du logiciel.

Le type d'un relais qui peut gérer la haute puissance requise à contrôler directement un moteur électrique ou d'autres changement est dénommé un contacteur. Les relais à l'état solide contrôlent les circuits d'alimentation avec aucune pièce mobile, au lieu d'utiliser un

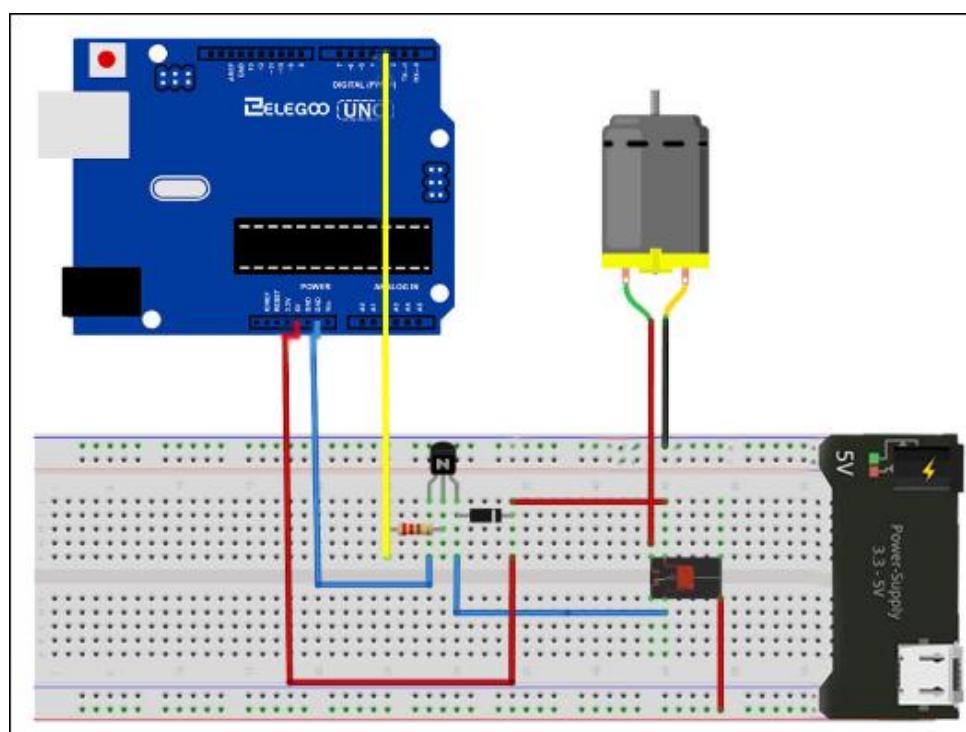
dispositif semi-conducteur à gérer le commutateur. Les relais avec les caractéristiques de fonctionnement calibré et parfois les bobines d'exploitation multiples sont utilisés à protéger les circuits électriques dans la surcharge ou les erreurs. Dans les systèmes d'alimentation électriques moderne, ces fonctions sont gérées par les instruments numériques dénommés "Relais de protection".

Le schéma ci-dessous indique comment conduire un relais avec Arduino (télécharger à partir de 'arduino.cc').



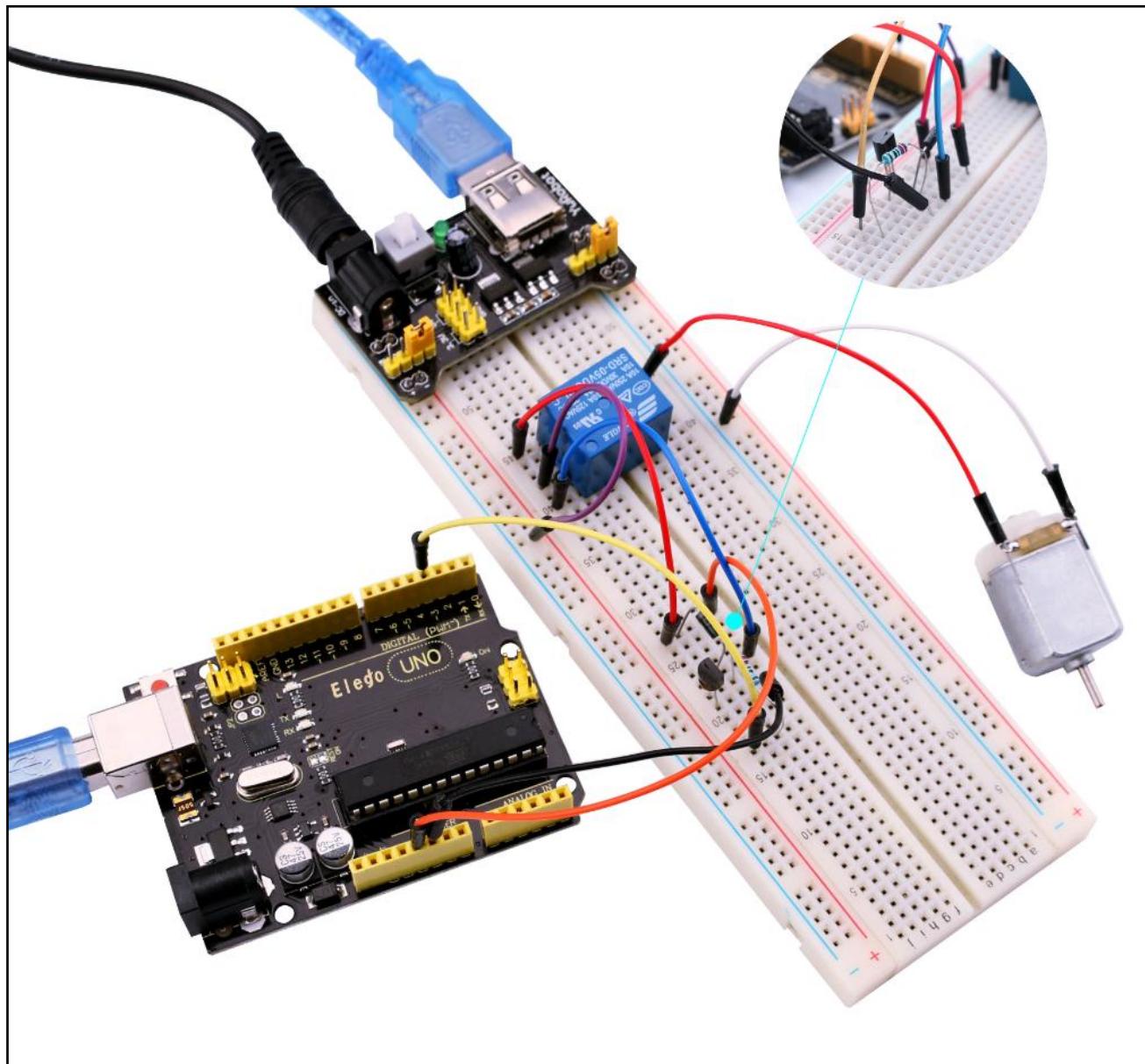
Raccordement

Schéma de câblage



Code

Veuillez voir le fichier de code.



Leçon 13: 74HC595 et Affichage de segment

Vue d'ensemble

Après d'apprendre Leçon 6 et Leçon 7, on peut utiliser la tessiture à décalage 74HC595 à contrôler l'affichage de segment. L'affichage de segment s'affiche les chiffres de 9 à 0.

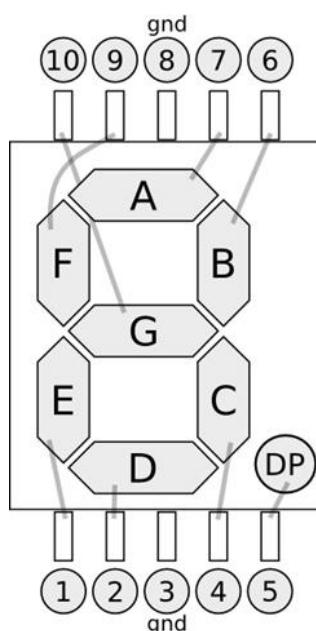
Composants requis

- (1) Elegoo UNO R3
- (2) Carte de prototypage
- (1) 74HC595
- (1) Affichage de segment
- (8) 220 ohm résistance
- (20) M-M câbles

Introduction de composant

Affichage de 7-segments:

Il montre le diagramme sur la broche de 7-segments.



0-9 dix chiffres correspondants à chaque segment sont comme suivants (Le tableau suivant s'applique à un dispositif d'affichage de 7-segments à la cathode commune, si vous utilisez une anode commune, le tableau doit être remplacé tous les 1 à 0, et 0 doit être remplacé par 1.):

Affichage	dp	a	b	c	d	e	f	g
numérique	dp	a	b	c	d	e	f	g
e								
0	0	1	1	1	1	1	1	0
1	0	0	1	1	0	0	0	0
2	0	1	1	0	1	1	0	1
3	0	1	1	1	1	0	0	1
4	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1
6	0	1	0	1	1	1	1	1
7	0	1	1	1	0	0	0	0
8	0	1	1	1	1	1	1	1
9	0	1	1	1	1	0	1	1

Raccordement

Schéma

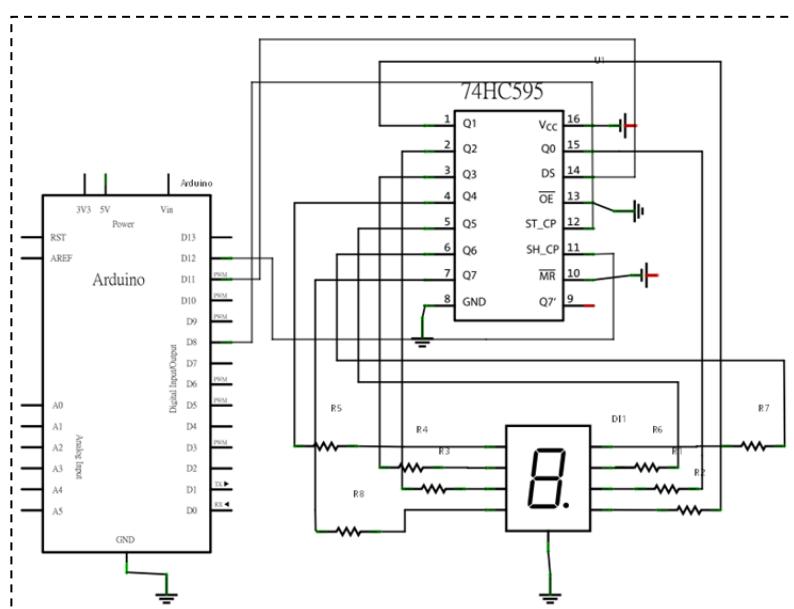
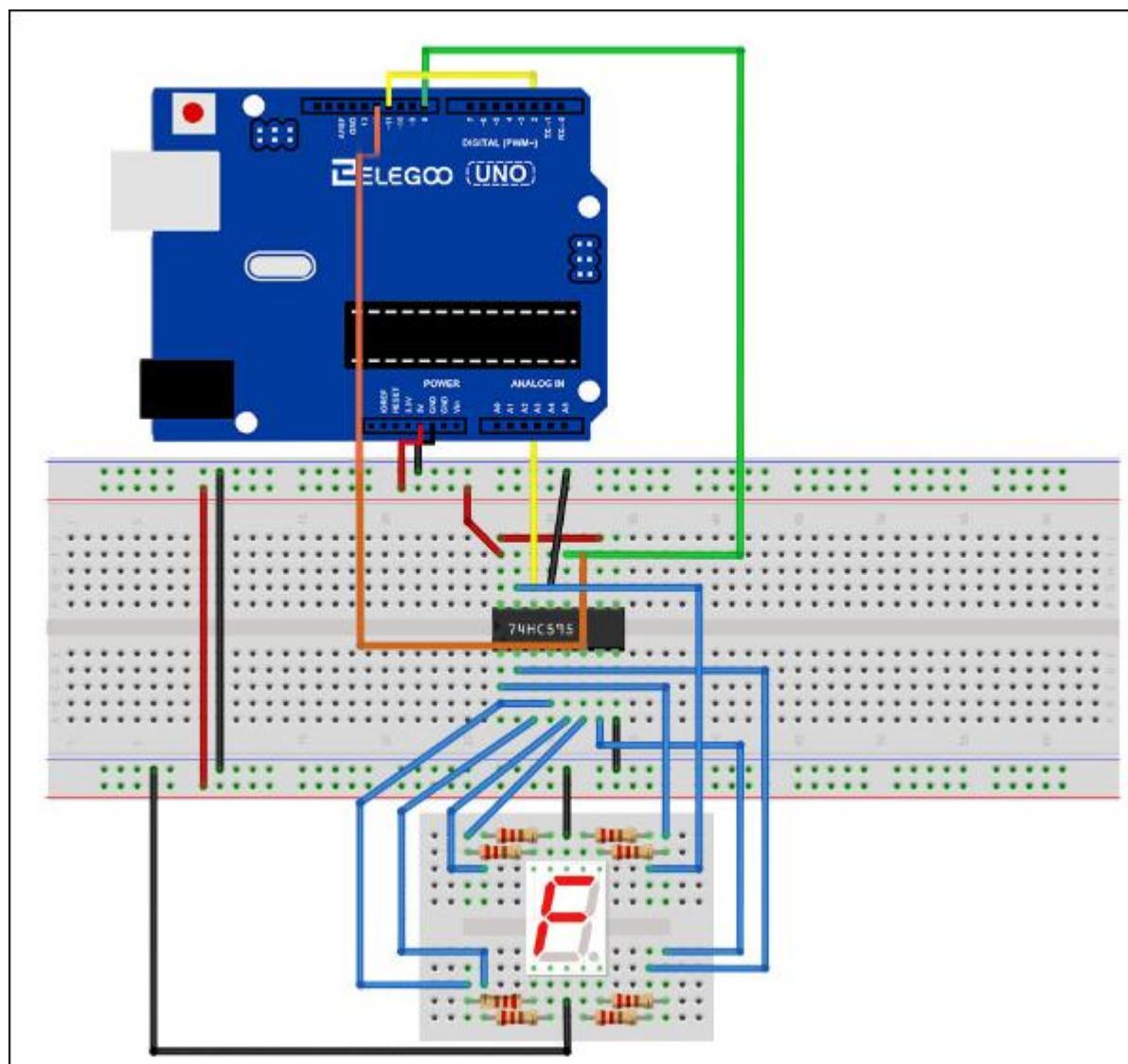


Schéma de câblage



Le tableau suivant montre les broches correspondantes de l'affichage de 7-segments 74HC595:

Broche 74HC595	Broche de contrôle remarquable à 7 spectacles (cours)
Q0	7 (A)
Q1	6 (B)
Q2	4 (C)
Q3	2 (D)
Q4	1 (E)
Q5	9 (F)
Q6	10 (G)
Q7	5 (DP)

Étape 1: Brancher 74HC595

Tout d'abord, branchez le câble à la source d'alimentation et mettez le câble à la terre:

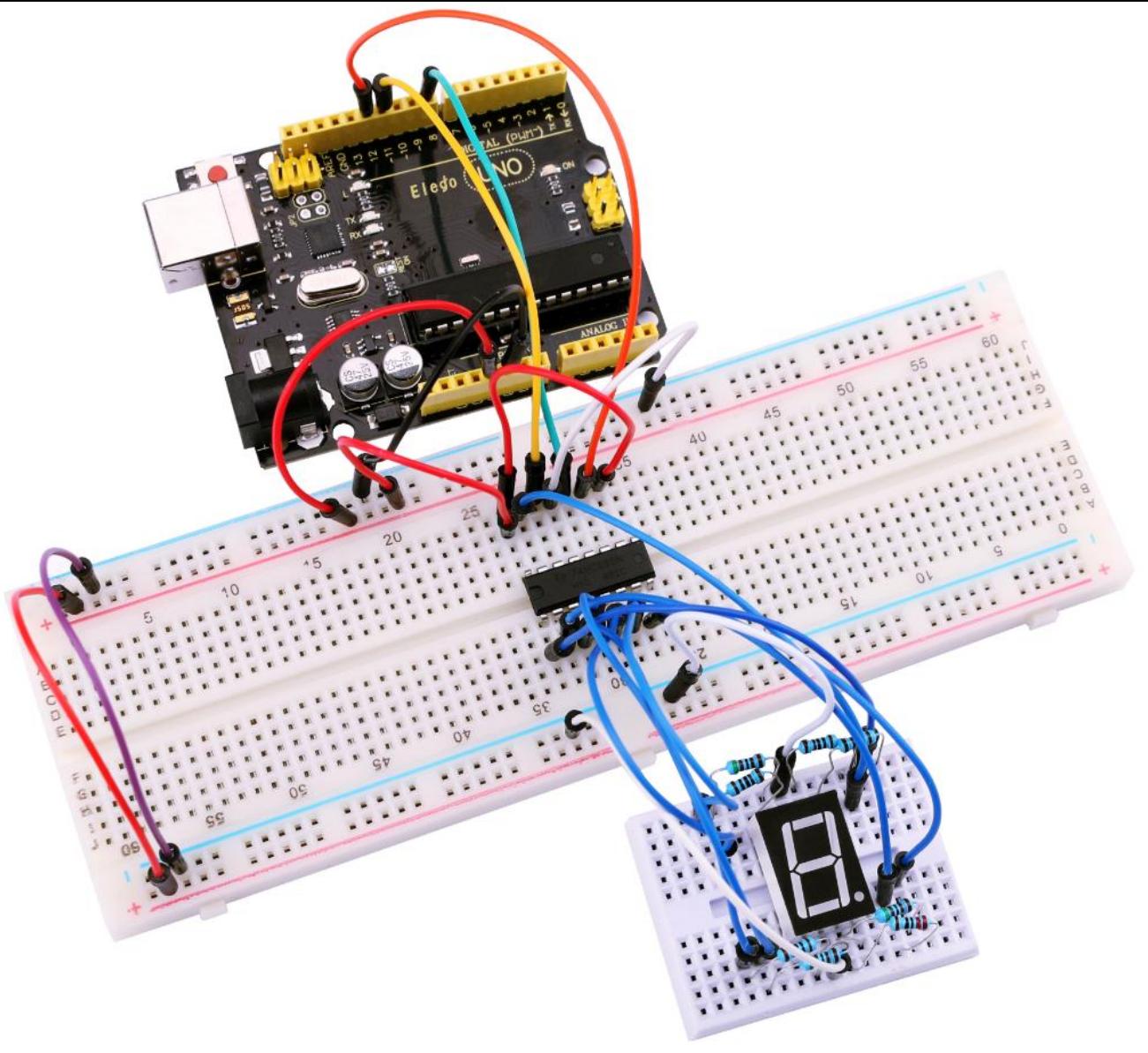
- Vcc (broche 16) et MR (broche 10) connectés à 5V
- GND (broche 8) et OE (broche 13) à la terre
- Branchez les broches DS, ST_CP et SH_CP:
- Branchez DS (broche 14) à la broche 11 sur la carte d'UNO R3 (ligne bleue)
- Branchez ST_CP (broche 12, broche de verrouillage) à la broche 8 sur la carte d'UNO R3 (ligne verte)
- Branchez SH_CP (broche 11, broche de horloge) à la broche 12 sur la carte d'UNO R3 (jaune)

Étape 2: Brancher l'affichage de 7-segments

- Branchez les broches 3 et 8 de l'affichage de 7-segments à GND sur la carte d'UNO R3 (cet exemple utilise la cathode commune, si vous voulez utiliser l'anode commune, veuillez brancher les broches 3 et 8 à +5V sur la carte d'UNO R3).
- En fonction du tableau ci-dessus, connectez 74HC595 Q0 ~ Q7 à la broche correspondante de l'affichage de 7-segments (A-G et DP), et puis chaque pied dans la résistance 220 Ohm en série.

Code

Veuillez voir le fichier de code.



Leçon 14: Affichage de 4-chiffres 7-segments

Vue d'ensemble

Dans cette leçon, vous pouvez apprendre comment utiliser un affichage de 4-chiffres 7-segments.

Lors d'utiliser l'affichage d'1-chiffre 7 segments avec une anode commune, la broche d'anode commune est connectée à la source d'alimentation; Si il est la cathode commune, la broche de cathode commune est connectée à GND.

Lors d'utiliser l'affichage de 4-chiffres 7 segments, la broche d'anode commune ou de cathode comme est utilisée à contrôler quel chiffre à afficher. Même s'il n'y a qu'un chiffre travaillant, le principe de persistance de la vision vous permet de voir tous les chiffres affichés puisque chaque vitesse de balayage est si rapide que vous ne remarquerez guère les intervalles.

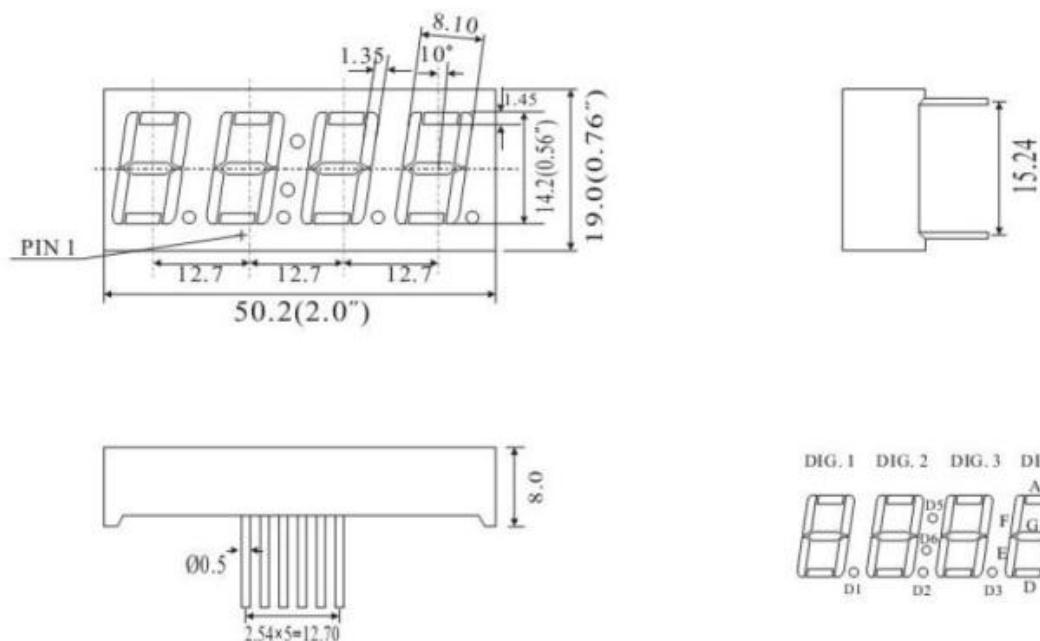
Composants requis

- (1) Elegoo UNO R3
- (1) Carte de prototypage
- (1) 74HC595
- (1) Affichage de 4-chiffres 7-segments
- (1) 220 ohm résistance
- (20) M-M câbles

Introduction de composant

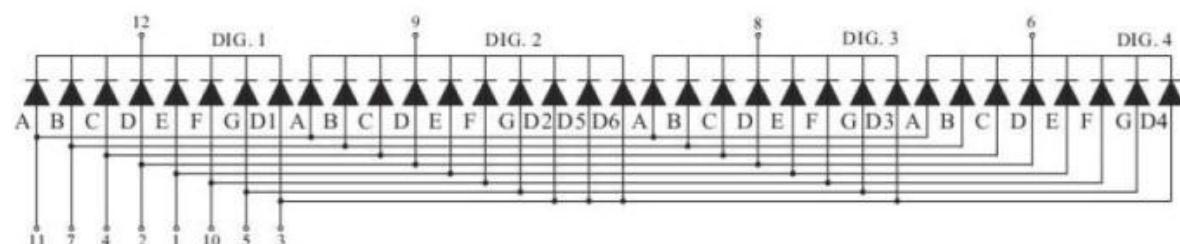
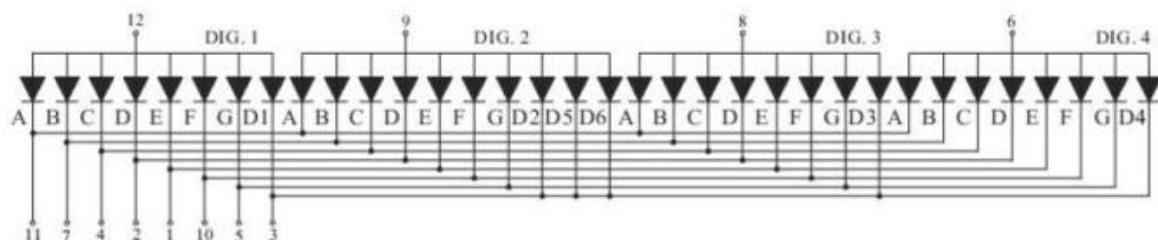
Affichage de 4-chiffres 7-segments:

Package Dimensions

CPS05643AB

UNIT: MM(INCH) TOLERANCE: $\pm 0.25(0.01")$

Internal Circuit Diagram

**5643A****5643B**

Four Digits Displays Series

Raccordement

Schéma

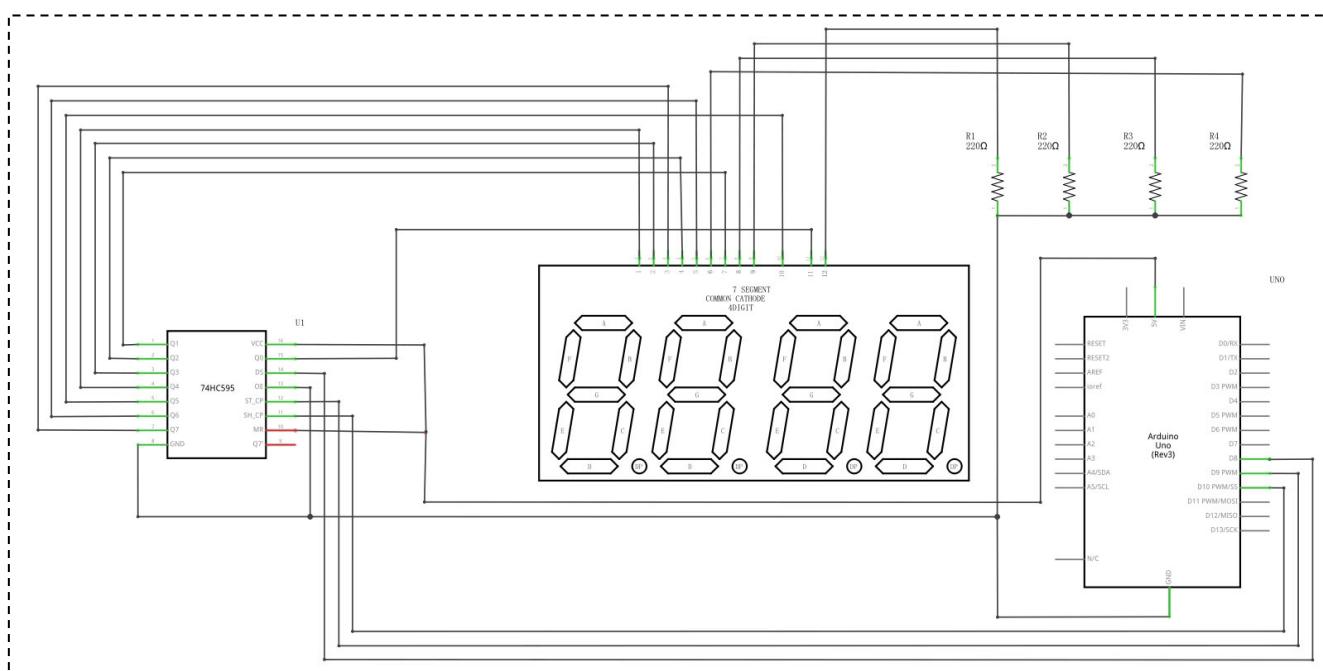
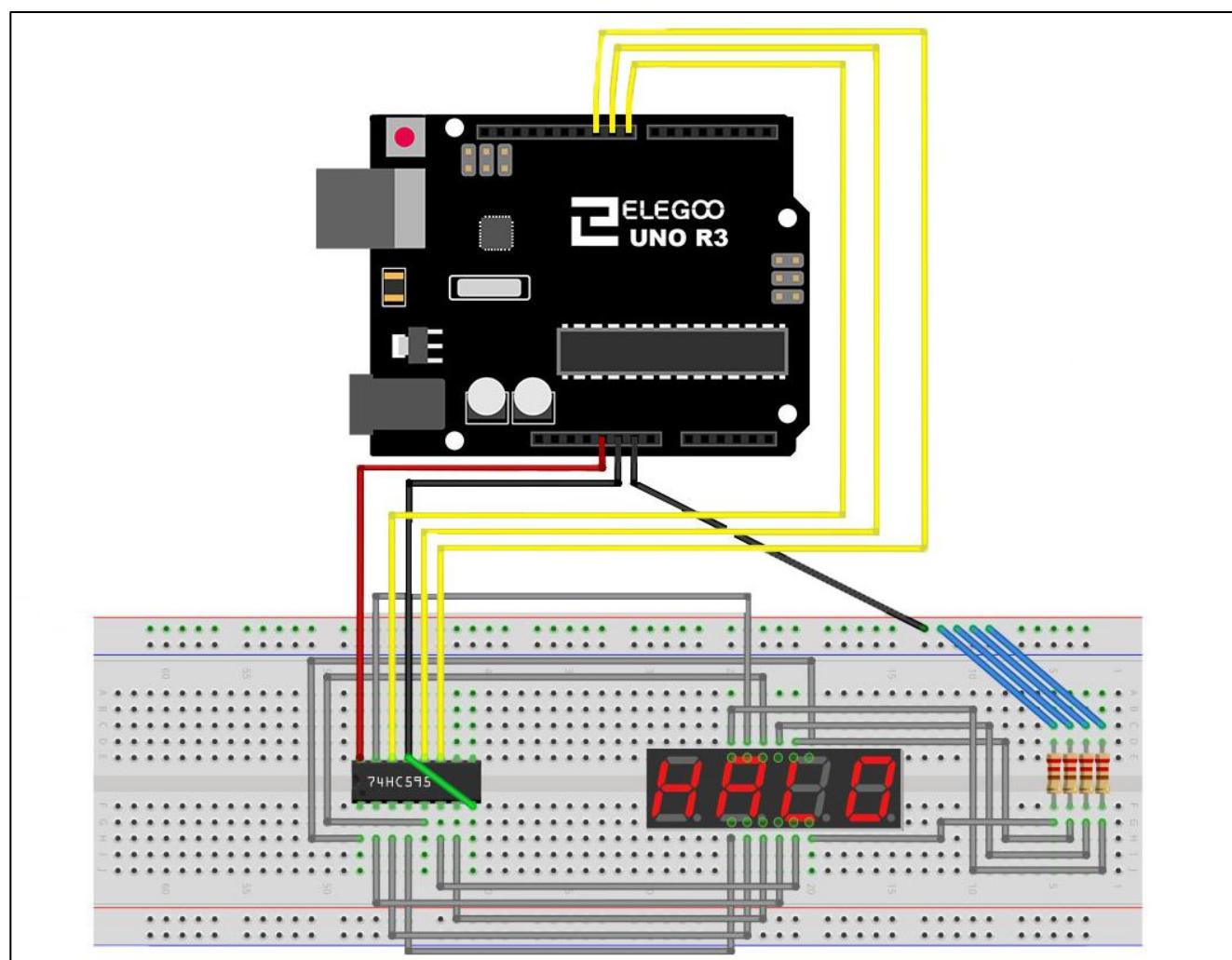
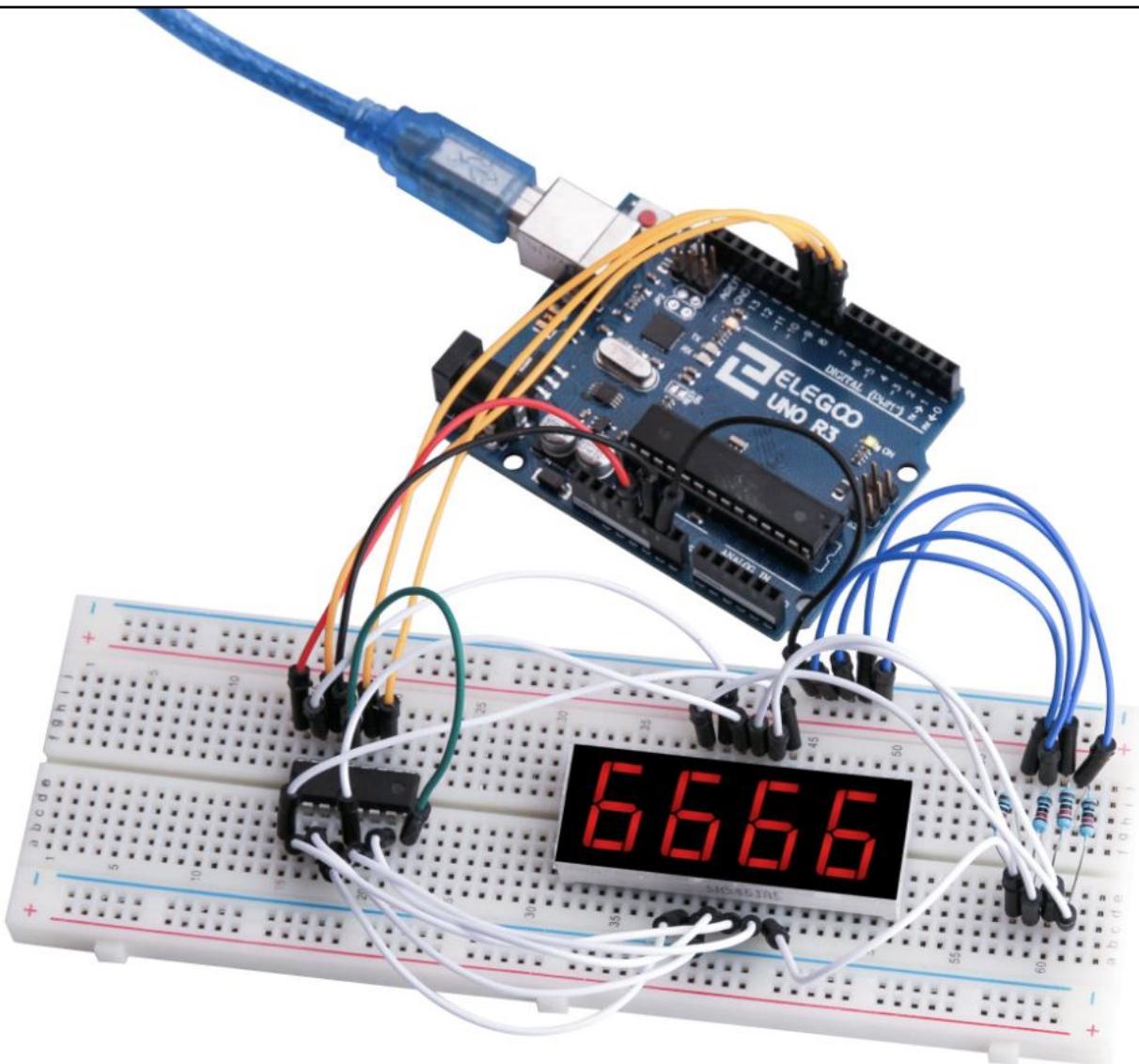


Schéma de câblage



Code

Veuillez voir le fichier de code.



Leçon 15: Servo

Vue d'ensemble

Servo est un type de motoréducteur qui peut seulement tourner de 180 degrés. Il est contrôlé en envoyant des impulsions électriques de votre carte d'UNO R3. Ces impulsions disent le servo quelle position il devrait passer à. Un servo possède trois fils, le fil marron est GND, le fil rouge est VCC et le fil orange est la ligne de signal.

Composants requis

- (1) Elegoo UNO R3
- (1) Servo
- (2) F-M câbles

Raccordement

Schéma

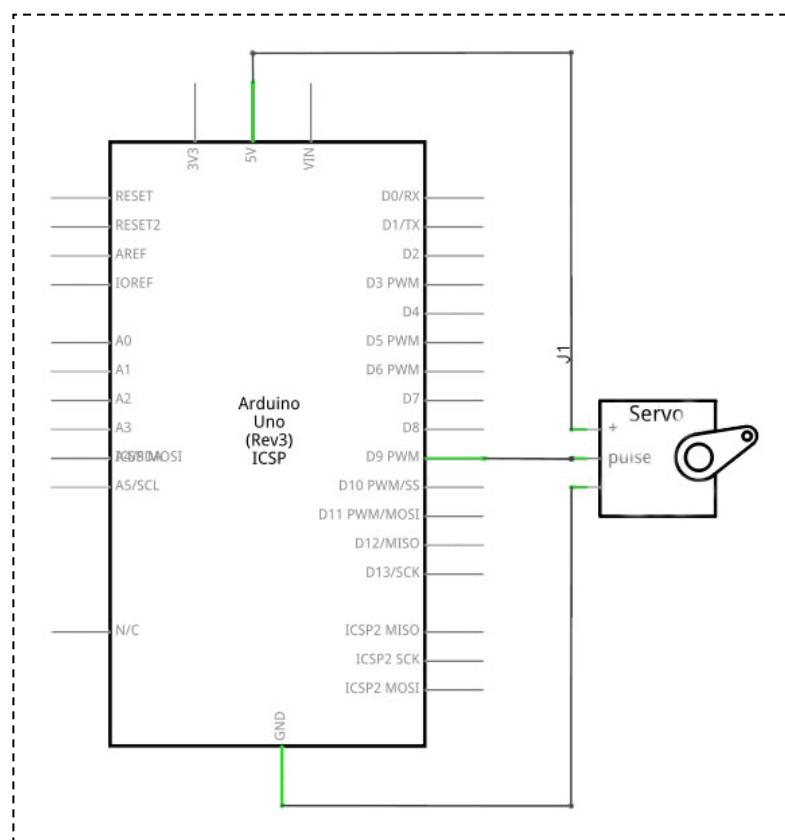
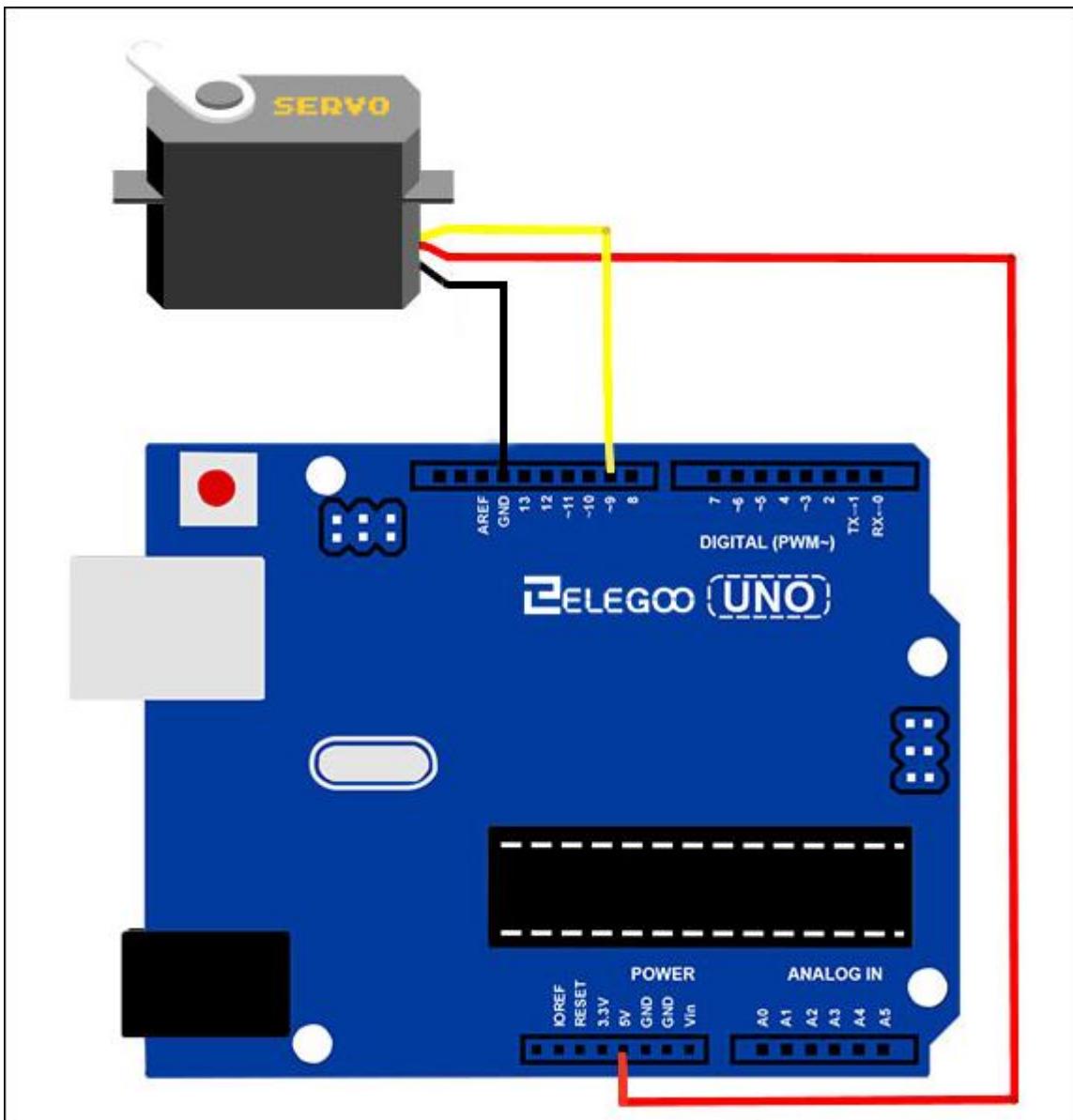
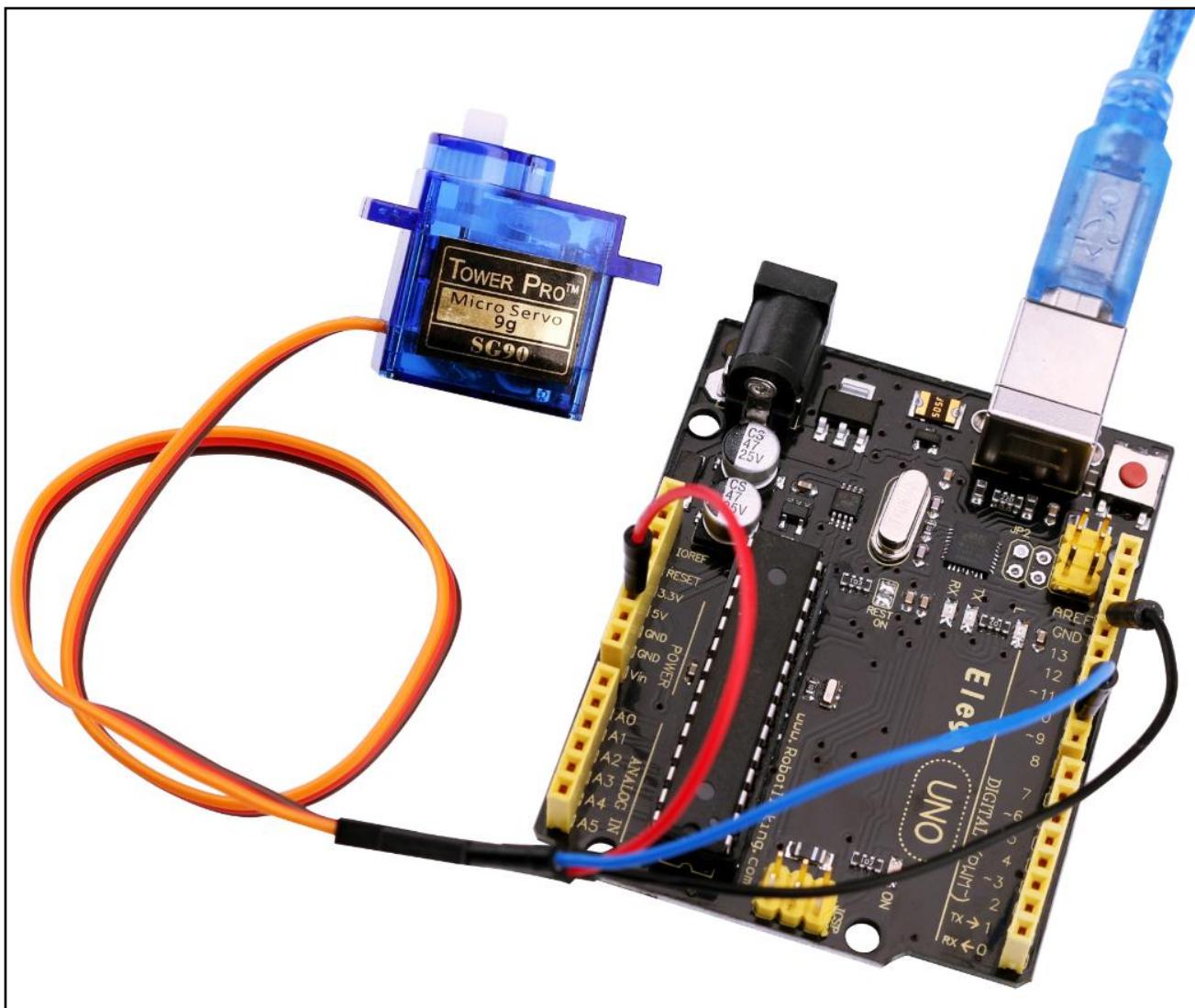


Schéma de câblage



Code

Veuillez voir le fichier de code.



Leçon 16: Écran LCD

Vue d'ensemble

Dans cette leçon, vous pouvez apprendre comment câbler et utiliser l'écran LCD alphanumérique.

L'écran possède un rétro-éclairage LED et peut afficher deux rangées avec jusqu'à 16 caractères sur chaque ligne. Les rectangles peuvent être trouvés pour chaque caractère à l'écran et les pixels qui composent chaque caractère.

Cet écran est tout blanc sur bleu et est destiné à l'affichage de texte.

Dans cette leçon, vous pouvez exécuter le programme en exemple d'Arduino pour la bibliothèque LCD, mais dans la leçon suivante, c'est possible d'afficher la tempête et le niveau de lumière en utilisant les capteurs sur l'écran.

Composants requis

- (1) Elegoo UNO R3
- (1) Écran LCD1602
- (1) Potentiomètre
- (1) Carte de prototypage
- (16) F-M câbles

Introduction de composant

LCD1602:

VSS: Une broche mise à la terre

VDD: Une broche connectée à la source d'alimentation +5V

VO: Une broche à régler le contraste de LCD1602

RS: Le registre sélectionne la broche qui contrôle à quel mémoire de l'écran LCD vous écrivez les données. Vous pouvez selectionnez également le registre de données, qui enregistre les données sur l'écran, ou un registre des instructions pour permettre le contrôleur d'LCD à rechercher les instructions comment faire ensuite.

R/W: La broche Read/Write pour sélectionner le mode de lecture ou le mode d'écriture.

E: Lorsqu'ils sont fournis avec un faible niveau d'énergie, une broche Enabling qui provoque le module LCD d'exécuter les instructions pertinentes.

D0-D7: Broches à lire et écrire les données

A and K: Broches à contrôler la rétro-éclairage LED

Raccordement

Schéma

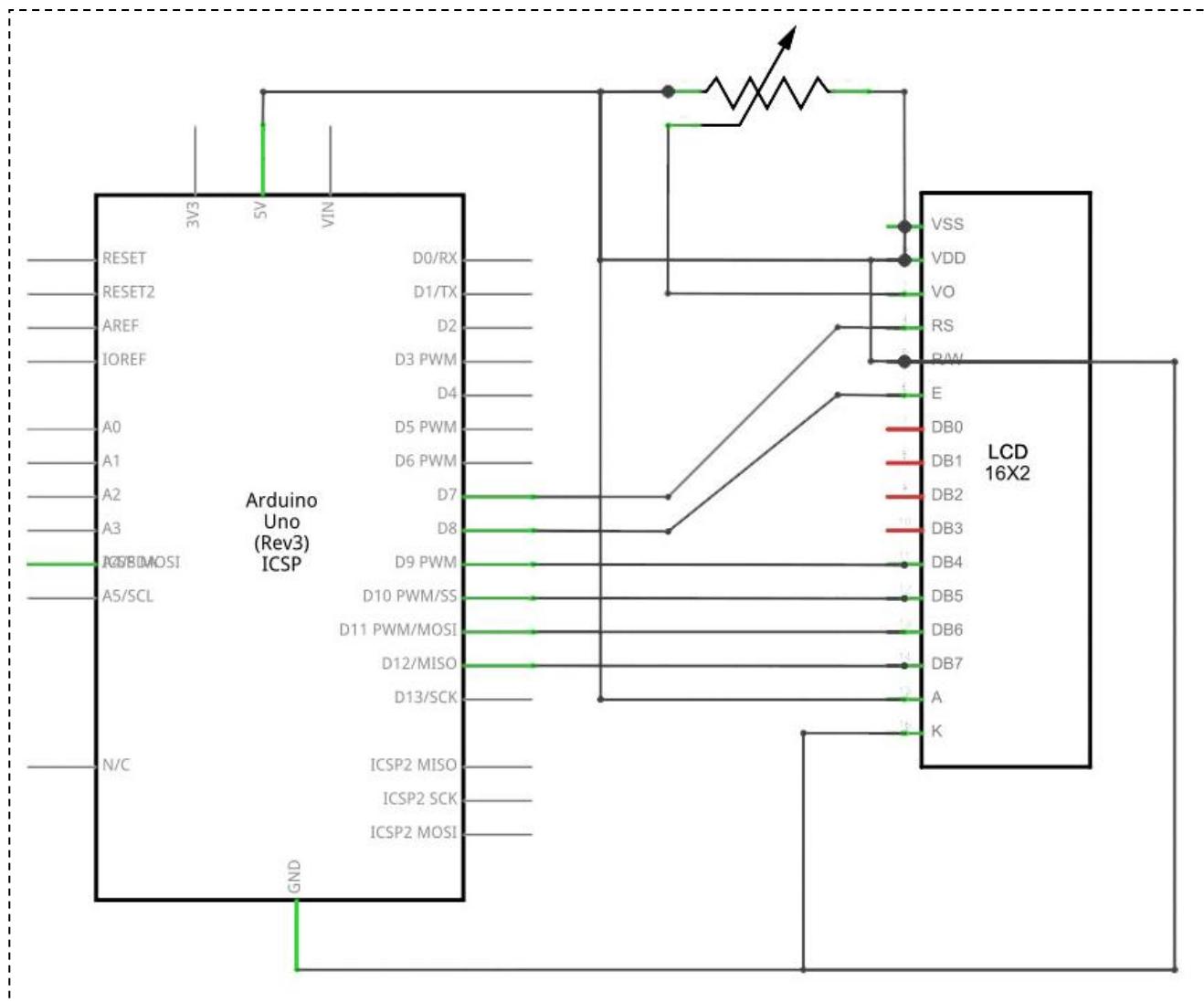
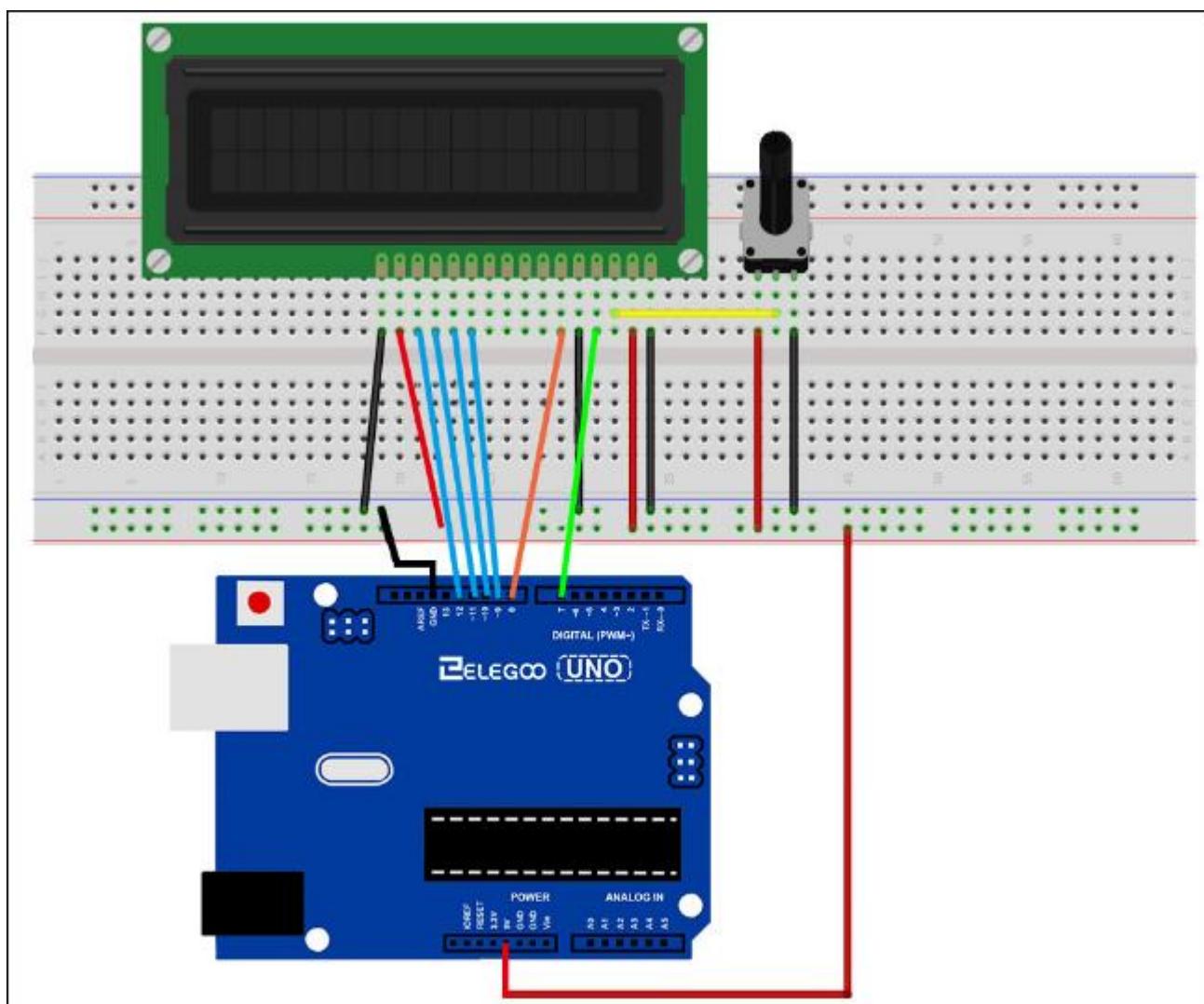


Schéma de câblage



L'écran LCD absoin de 6 broches Arduino, tous sont définis comme les sorties numériques. Il a également besoin des connexions 5V et GND.

Il y a un certain nombre de connexions à effectuer. En alignant l'écran avec le haut de la carte de prototypage permet d'identifier ses broches sans trop compter, surtout si la carte de prototypage a ses lignes numérotées avec la ligne 1 comme étant la rangée supérieure de la carte. N'oubliez pas que le fil jaune longue relie le curseur du pot à la broche 3 de l'écran. Le 'pot' est utilisé à contrôler le contraste de l'écran.

Vous pouvez trouver que votre écran est alimenté sans broche d'en-tête à attacher. Si tel est le cas, suivez les instructions de la section suivante.

Code

Actuellement, vous avez fait la configuration physique, tout ce que nous avons besoin est le code.

Avant de l'exécuter, assurez-vous que vous avez installé la bibliothèque <LiquidCrystal>, ou le cas échéant, vous devez la réinstaller. Sinon, votre code ne fonctionne pas.

Cet exemple utilise les différentes broches pour ceux que nous utilisons, alors la ligne de code est comme suivante:

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

Et modifiez à:

LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

Chargez le code à votre carte d'Arduino et vous pouvez lire le message 'hello, world' affiché, suivi par un nombre qui compte à partir de zéro.

La première ligne de la note dans les sketches est la ligne:

#include <LiquidCrystal.h>

Ceci indique à Arduino que nous souhaitons utiliser la bibliothèque Liquid Crystal.

Ensuite, c'est la ligne que nous devons à modifier. Il définit quelles broches de l'Arduino sont reliés à ce qui les broches de l'écran.

LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

Les arguments pour cela sont les suivantes:

Display Pin Name Display Pin Number Arduino Pin (dans cet exemple) RS 4 7 E 6 8 D4 11 9 D5 12
10 D6 13 11 D7 14 12

Après d'avoir téléchargé ce code, assurez-vous que le rétro-éclairage s'allume, et réglez le potentiomètre tout le chemin jusqu'à ce que vous voyez le message texte.

Dans la fonction 'setup', nous avons deux commandes:

lcd.begin(16, 2);

lcd.print("hello, world!");

La première ligne indique à la bibliothèque Liquid Crystal combien de colonnes et de lignes l'écran s'affiche. La deuxième ligne s'affiche le message que l'on voit sur la première ligne de l'écran.

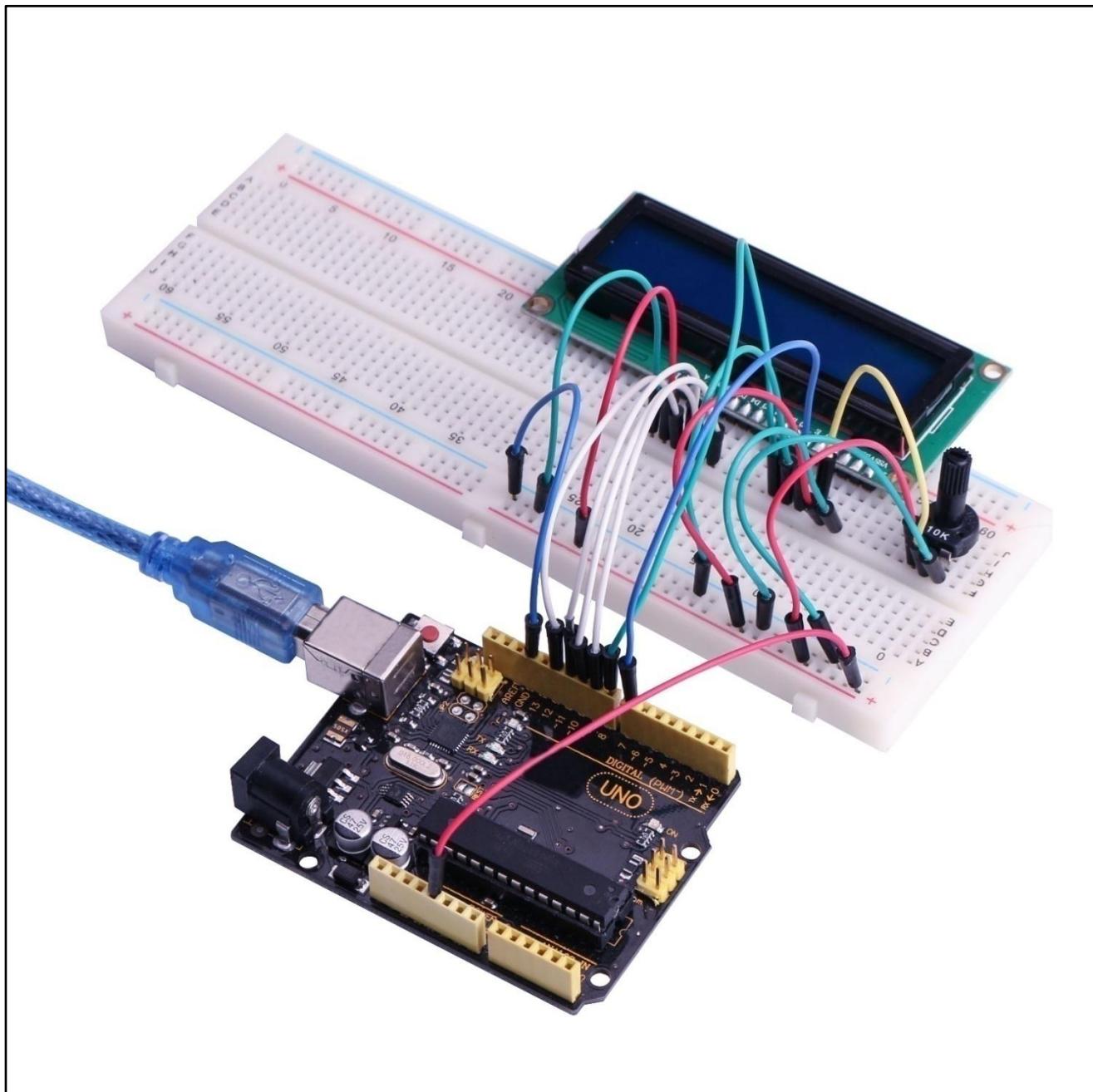
Dans la fonction 'loop', nous avons également deux commandes:

lcd.setCursor(0, 1);

lcd.print(millis()/1000);

La première ligne définit la position du curseur (où le prochain texte apparaîtra) à la colonne 0 & ligne 1. Les chiffres de colonnes et lignes commencent à 0 plutôt qu'1.

La deuxième ligne s'affiche le nombre de millisecondes puisque l'Arduino a été réinitialisé.



Leçon 17: Thermomètre

Vue d'ensemble

Dans cette leçon, vous pouvez utiliser l'écran LCD pour afficher la température.

Composants requis

- (1) Elegoo UNO R3
- (1) Écran LCD1602
- (1) 10k ohm résistance
- (1) Thermistance
- (1) Potentiomètre
- (1) Carte de prototypage
- (16) F-M câbles

Raccordement

Schéma

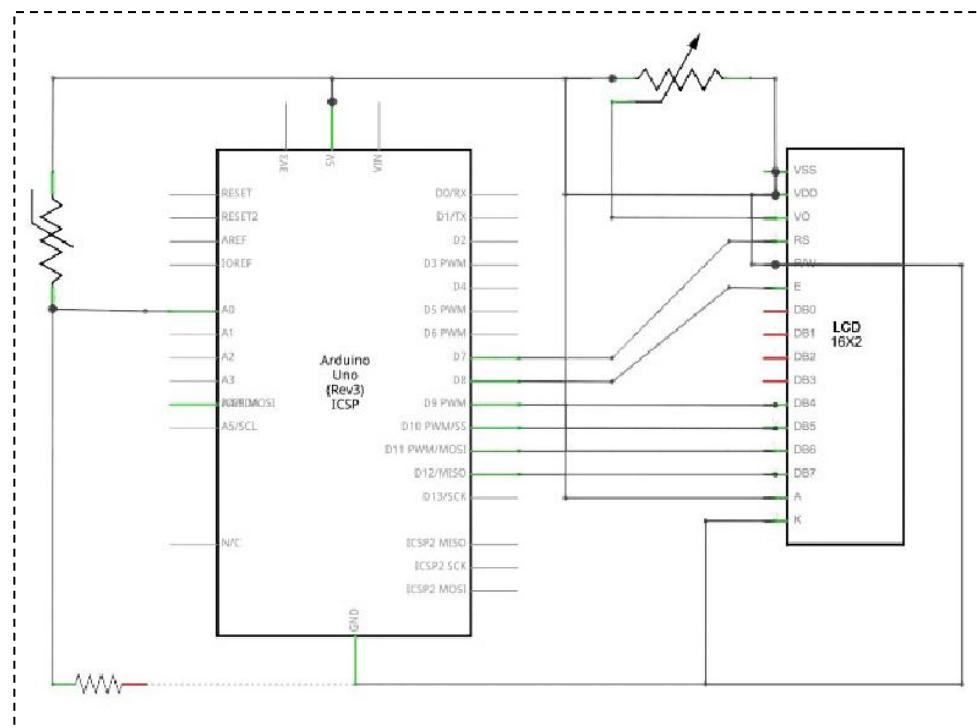
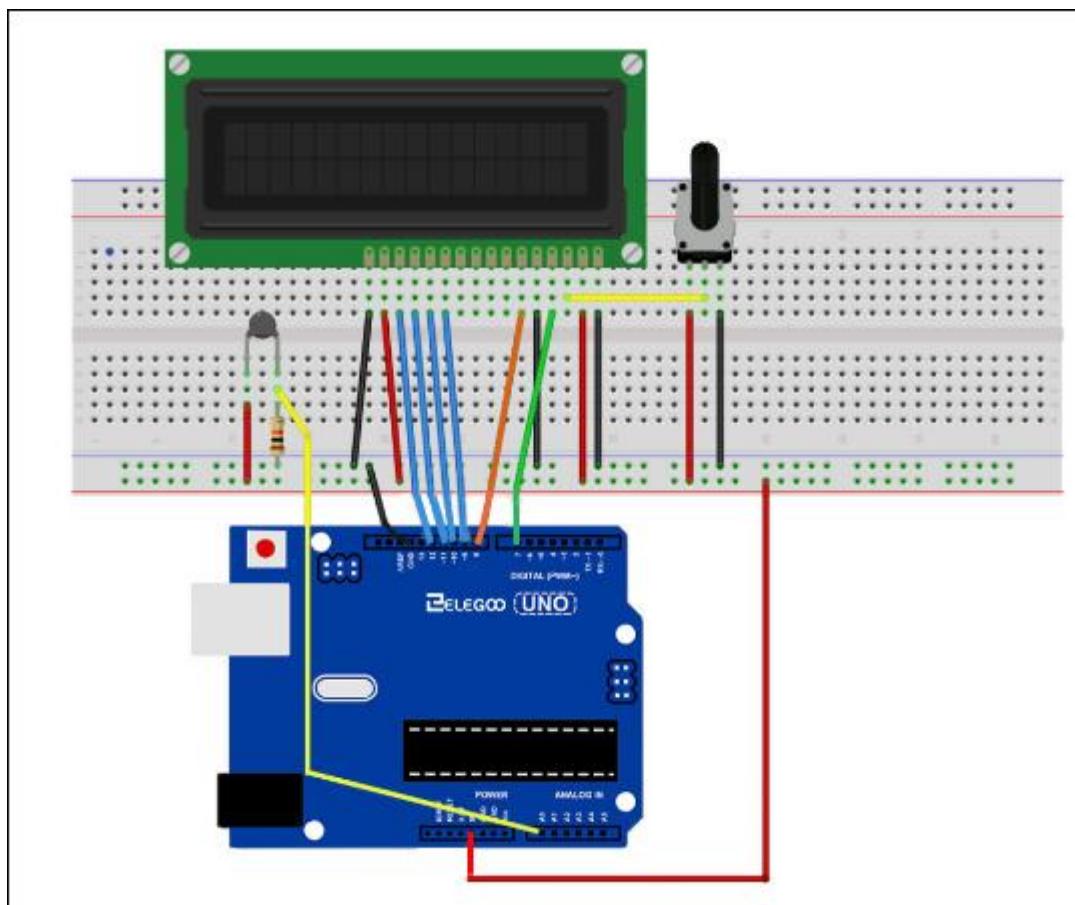


Schéma de câblage



La mise en page de la carte de prototypage est basée sur la disposition dans la Leçon 6, alors il va simplifier les choses si vous l'avez toujours sur votre carte de prototypage.

Il existe quelques fils volants près du pot qui a été déplacé légèrement sur cette disposition.

La cellule photoélectrique, la résistance 1 kΩ et la thermistance sont tous les nouveaux ajouts à la carte.

Code

Avant de l'exécuter, assurez-vous que vous avez installé la bibliothèque <LiquidCrystal>, ou le cas échéant, vous devez la réinstaller. Sinon, votre code ne fonctionne pas.

La sketche pour cela est basée sur celui dans la Leçon 14, faites-le télécharger à votre Arduino et vous devriez trouver que si vous placez votre doigt sur le capteur afin d'effectuer un réchauffement, il va augmenter la lecture de température.

Alors on trouve qu'il est utile de mettre une ligne de commentaire dessus de la commande 'lcd'.

```
//           BS  E  D4 D5  D6 D7
```

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

C'est plus facile pour vous si vous voulez modifier une broche ce que vous avez utilisé.

Dans la fonction 'loop', il existe maintenant deux choses intéressantes. La première chose est que nous devons convertir l'analogique du capteur de température dans une température réelle, et la seconde est que nous devons trouver comment les afficher.

Tout d'abord, penchons-nous sur le calcul de la température.

```
int tempReading = analogRead(tempPin);
float tempVolts = tempReading * 5.0 / 1024.0;
float tempC = (tempVolts - 0.5) * 100.0;
float tempF = tempC * 9.0 / 5.0 + 32.0;
```

La lecture brute du capteur de température est tout d'abord multipliée par 5 et ensuite divisée par 1024 pour nous donner la tension (entre 0 et 5) à l'entrée analogique 'tempPin'.

Pour convertir la tension provenant de le TMP36 dans une température en degrés C, vous devez soustraire 0.5V de la mesure et puis multiplier par 100.

Pour le convertir la température à Fahrenheit, vous devez multiplier par 9/5, et puis ajouter 32.

Les lectures modifiantes affichées sur l'écran LEC peuvent être délicates. Le problème principal est que la lecture peut être ne pas de même chiffre toujours. Alors, si la température passe de 101.50 à 99.00, le chiffre supplémentaire venu de la lecture ancienne peut avoir un danger de laisser le chiffre à gauche de l'écran.

Pour l'éviter, écrivez l'ensemble de lignes de l'écran LCD chaque fois autour de la boucle.

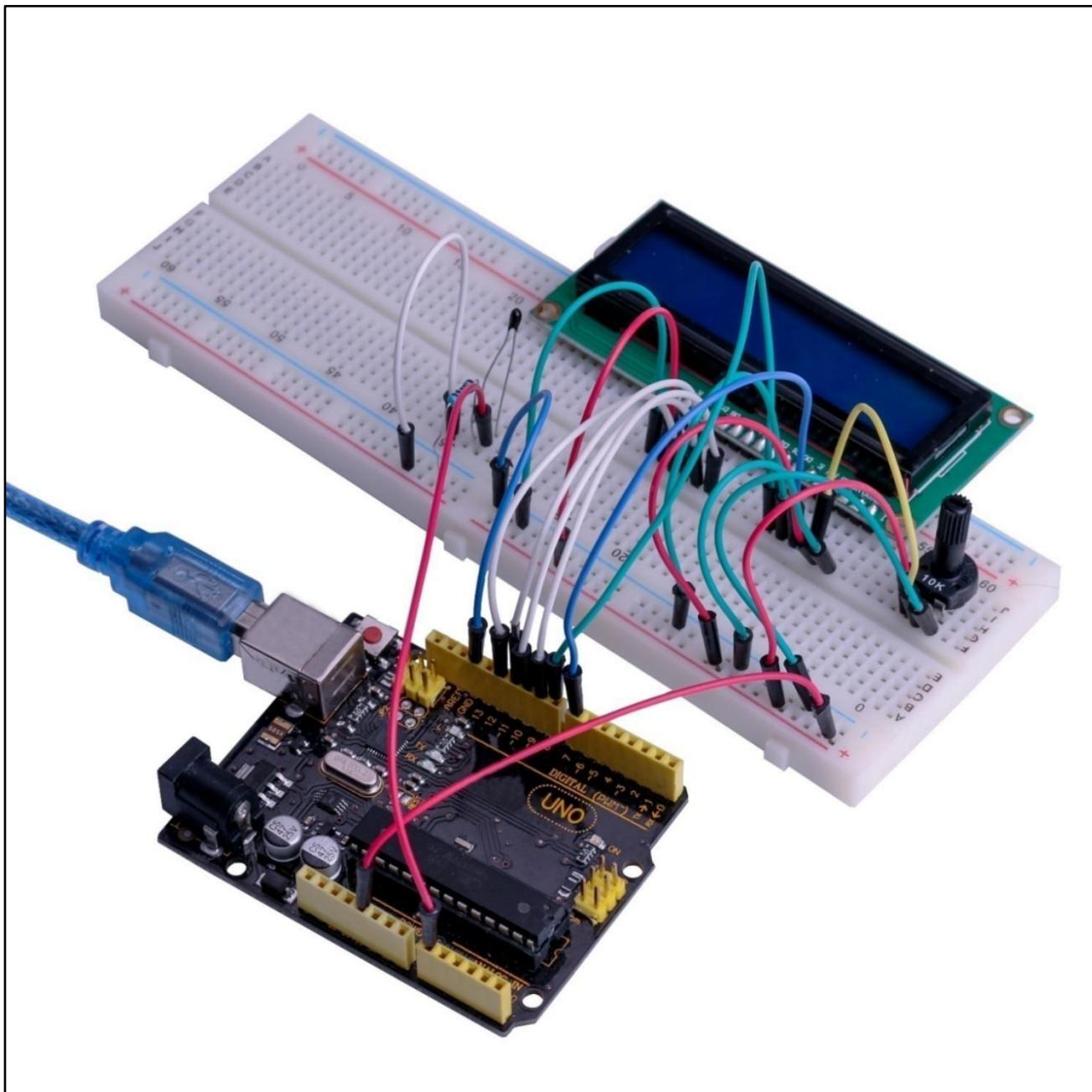
```
// -----
lcd.print("Temp      F  ");
lcd.setCursor(6, 0);
lcd.print(tempF);
```

Les commentaires plutôt étrangers permettent de vous rappeler les 16 colonnes de l'écran.

Alors vous pouvez entrer une chaîne de caractères de cette longueur avec des espaces où elle ira la lecture réelle.

Pour remplir les blancs, définissez la position du curseur pour l'emplacement où la lecture doit apparaître et imprimez-la.

En effet, la même approche est utilisée pour afficher le niveau de luminosité. Il n'existe aucun unit pour le niveau de luminosité, on affiche uniquement la lecture brute à partir de la lecture analogique.



Leçon 18: Module de capteur à ultrasons

Vue d'ensemble

Le capteur à ultrasons est parfait pour tous les types de projets qui ont besoin de mesures de la distance, en évitant les obstacles montrés comme les exemples.

HC-SR04 n'est pas cher et est facile à utiliser puisque l'on va utiliser une bibliothèque spécialement conçue pour ces capteurs.

Composants requis

- (1) Elegoo UNO R3
- (1) Module de capteur à ultrasons
- (4) F-M câbles

Introduction de composant

Capteur à ultrasons

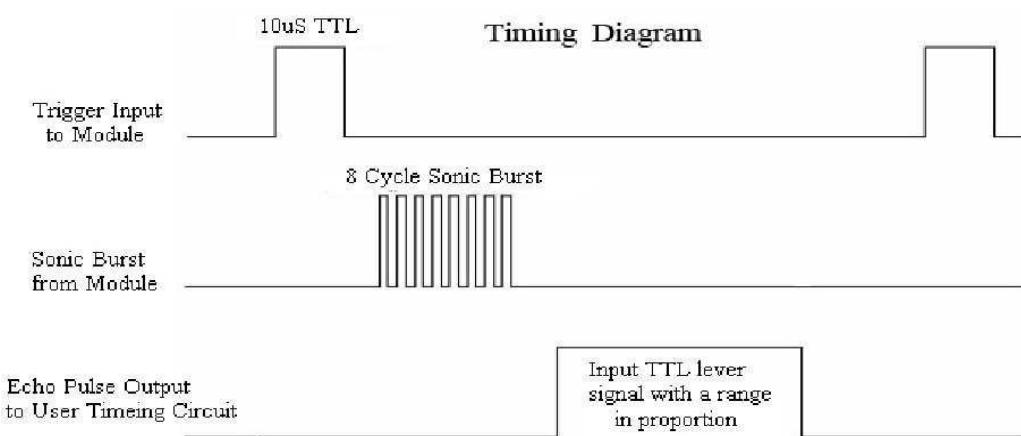
Le module de télémétrie à ultrasons HC - SR04 fournit la fonction 2cm - 400cm non-contact measurement, et la précision de télémétrie peut arriver à 3mm. Le module se compose les émetteurs à ultrasons, le récepteur et le circuit de contrôle. Le principe de base:

- (1) utilisant trigger IO pour le signal de haut niveau pendant au moins 10us;
- (2) le module envoie huit 40 kHz automatiquement et détecte s'il existe un signal d'impulsion à retourner;
- (3) si le signal retourne, à travers du niveau élevé, la durée de sortie élevée IO est le moment à partir de l'envoi de tournant de tore à ultrasons.

Distance d'essai = (Durée de niveau élevé×Vitesse du son (340M/S) /2

L'oscillogramme est illustré ci-dessous. Vous devez seulement fournir une impulsion courte 10us à l'entrée de trigger pour déclencher, et puis le module envoyera un éclatement des ultrasons en 8 cycle à 40 Hz, et élèvera son écho. L'écho est un objet à distance qui est à la largeur d'impulsion et la gamme en proportion. Vous pouvez calculer la gamme à partir de l'intervalle de temps entre le signal de trigger envoyé et le signal d'écho reçu. La formule:

$uS / 58 = \text{Centimètres}$ ou $uS / 148 = \text{pouce}$; ou: la gamme = durée de niveau élevé * Vélocité(340M/S) / 2; nous vous sugerons d'utiliser un cycle de mesure de 60ms, afin d'empêcher le signal de déclenchement au signal d'écho.



Connection

Schéma

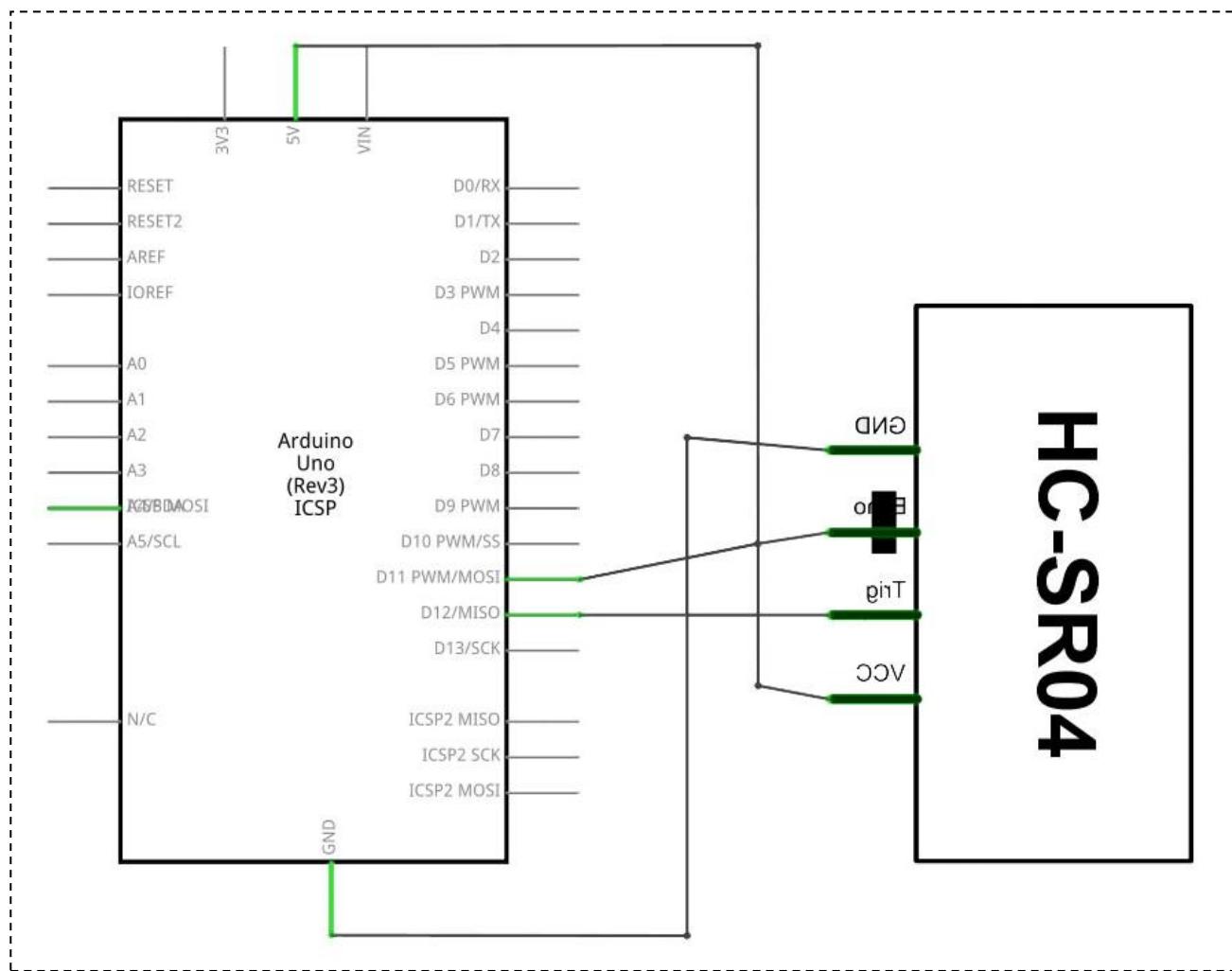
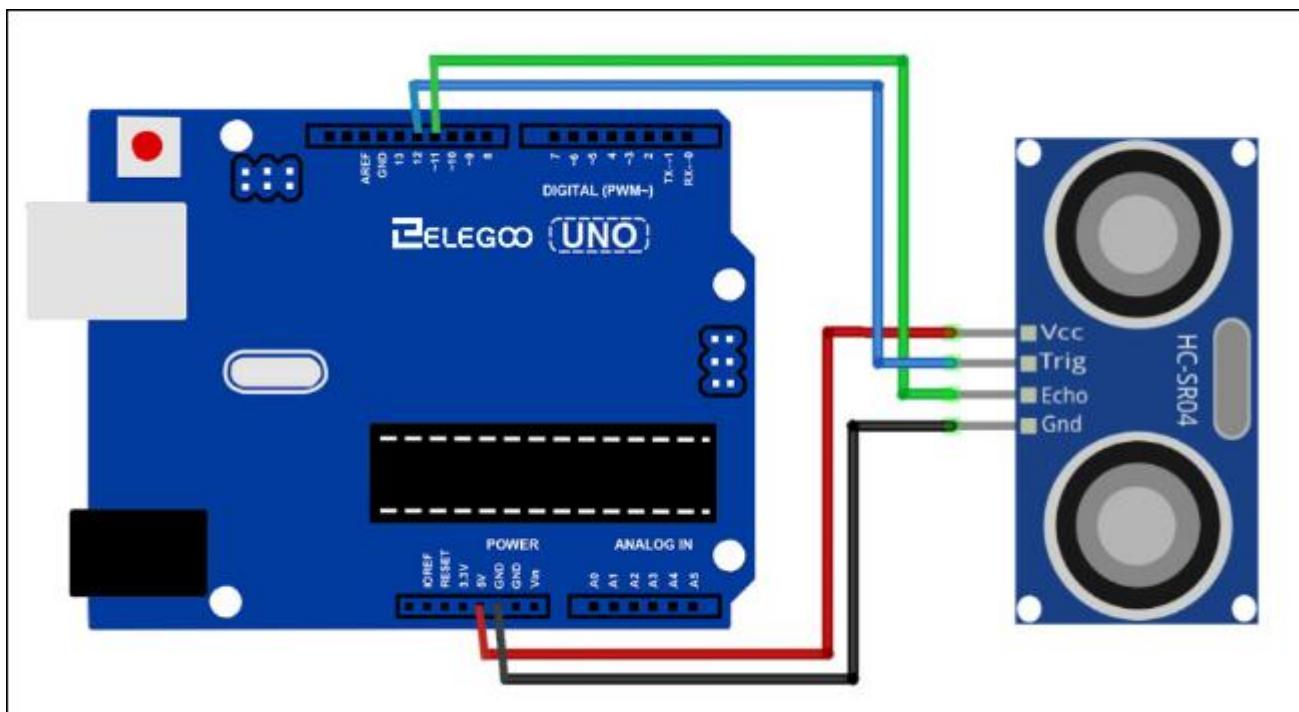


Schéma de câblage



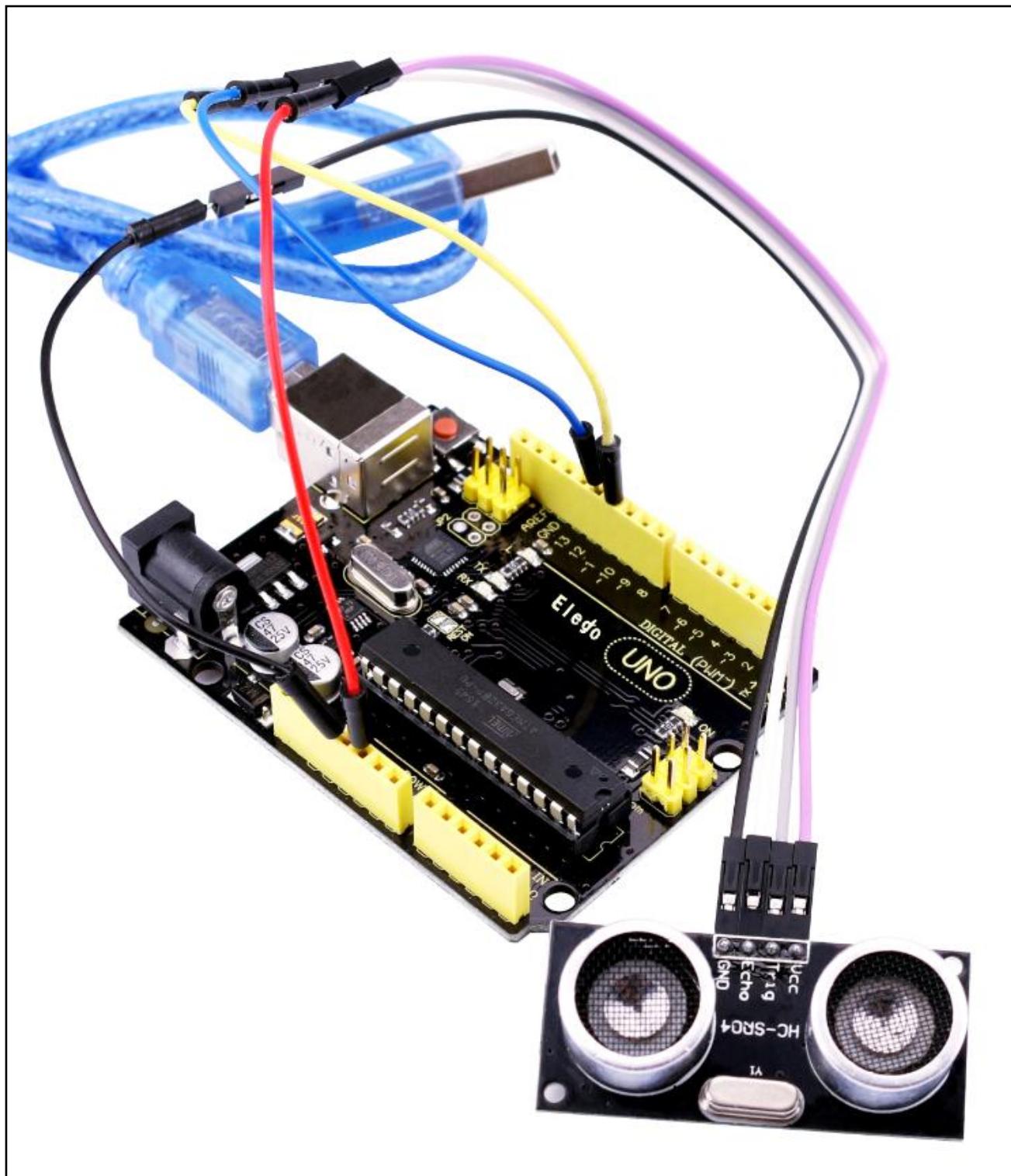
Code

En utilisant une bibliothèque conçue pour ces capteurs vont rendre les codes courts et simples.

La bibliothèque se compose au début de notre code, et puis à l'aide des commandes simples pour nous à contrôler le comportement du capteur.

Maintenant, nous avons fait une installation physique, et ce que nous avons besoins est le code.

Avant de l'exécuter, vous devez vous assurer que vous avez bien installé la bibliothèque <HC-SR04_Library>. Sinon, vous devez la réinstaller. Si vous ne le faites pas, votre code ne fonctionnera pas.



Leçon 19: Capteur de température et d'humidité DHT11

Vue d'ensemble

Dans ce mode d'emploi, on peut également utiliser comment utiliser un capteur de température et d'humidité DHT11.

C'est suffisamment précis pour la plupart des projets qui ont besoin de suivre des relevés de température et d'humidité.

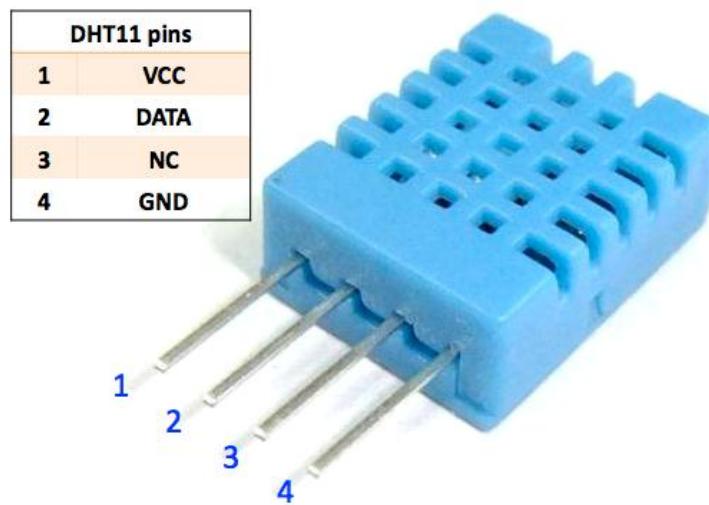
En plus, on va utiliser une bibliothèque spécifique conçue pour ces capteurs qui vont laisser le code plus court et facile à écrire.

Composants requis

- (1) Elegoo UNO R3
- (1) Module DHT11
- (3) F-M câbles

Introduction de composant

Capteur de température et d'humidité:



Le capteur numérique de température et d'humidité DHT11 est un capteur composite composant une sortie de signal numérique étalonné de la température et l'humidité. L'application de la technologie de collecte des modules numériques dédiés et la technologie de détection de la température et d'humidité est utilisée pour assurer que le produit possède une haute fiabilité et une excellente stabilité à long terme. Le capteur se compose un

sentiment résistif de composants mouillés et un dispositif de mesure de température NTC, et est ce capteur est connecté avec un microcontrôleur de haute performance à 8-bit.

Applications: HVAC, déshumidificateur, équipement d'essai et d'inspection, biens de consommation, automobile, autocommande, enregistreurs de données, stations météorologiques, appareils électroménagers, régulateur de l'humidité, médicaux et autres contrôle et mesure de l'humidité.

Paramètres du produit

Humidité relative:

Résolution: 16Bit

Répétabilité: $\pm 1\%$ RH

Exactitude: à 25°C $\pm 5\%$ RH

Interchangeabilité : entièrement interchangeables

Temps de réponse: 1 / e (63%) de 25°C 6s

1m / s air 6s

Hystérésis : $<\pm 0,3\%$ HR

Stabilité à long terme: $<\pm 0,5\%$ RH / yr in

Température:

Résolution: 16Bit

Répétabilité: $\pm 0,2^{\circ}\text{C}$

Gamme: à 25°C $\pm 2^{\circ}\text{C}$

Temps de réponse: 1 / e (63%) 10s

Caractéristiques électriques

Alimentation: DC 3.5~5.5V

Courant: 0.3mA en mesure et 60 μ A en veille

Période d'échantillonnage : plus de 2 secondes

Description sur les broches:

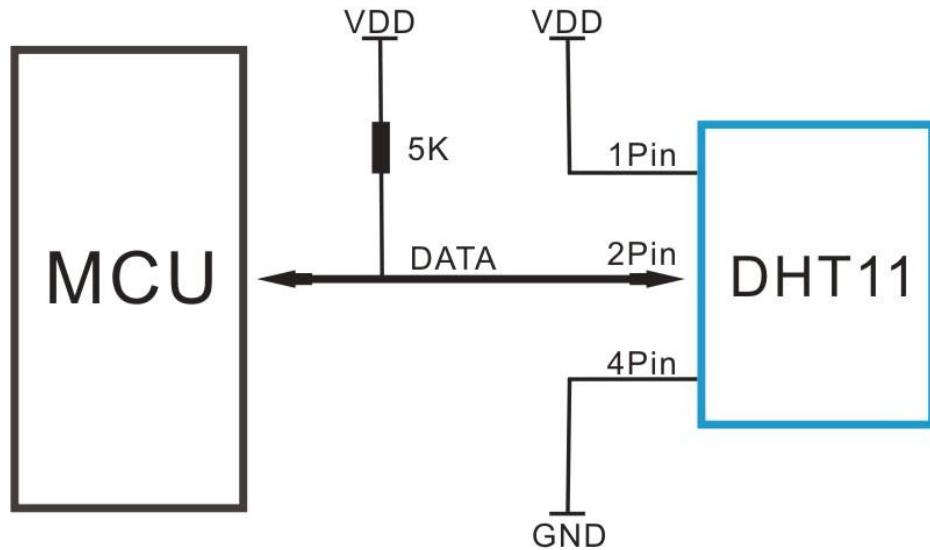
1, Alimentation VDD 3.5~5.5V DC

2 DATA données en série, un single bus

3, NC, broche vide

4, GND mise à la terre, la puissance négative

Application typique



Raccordement

Schéma

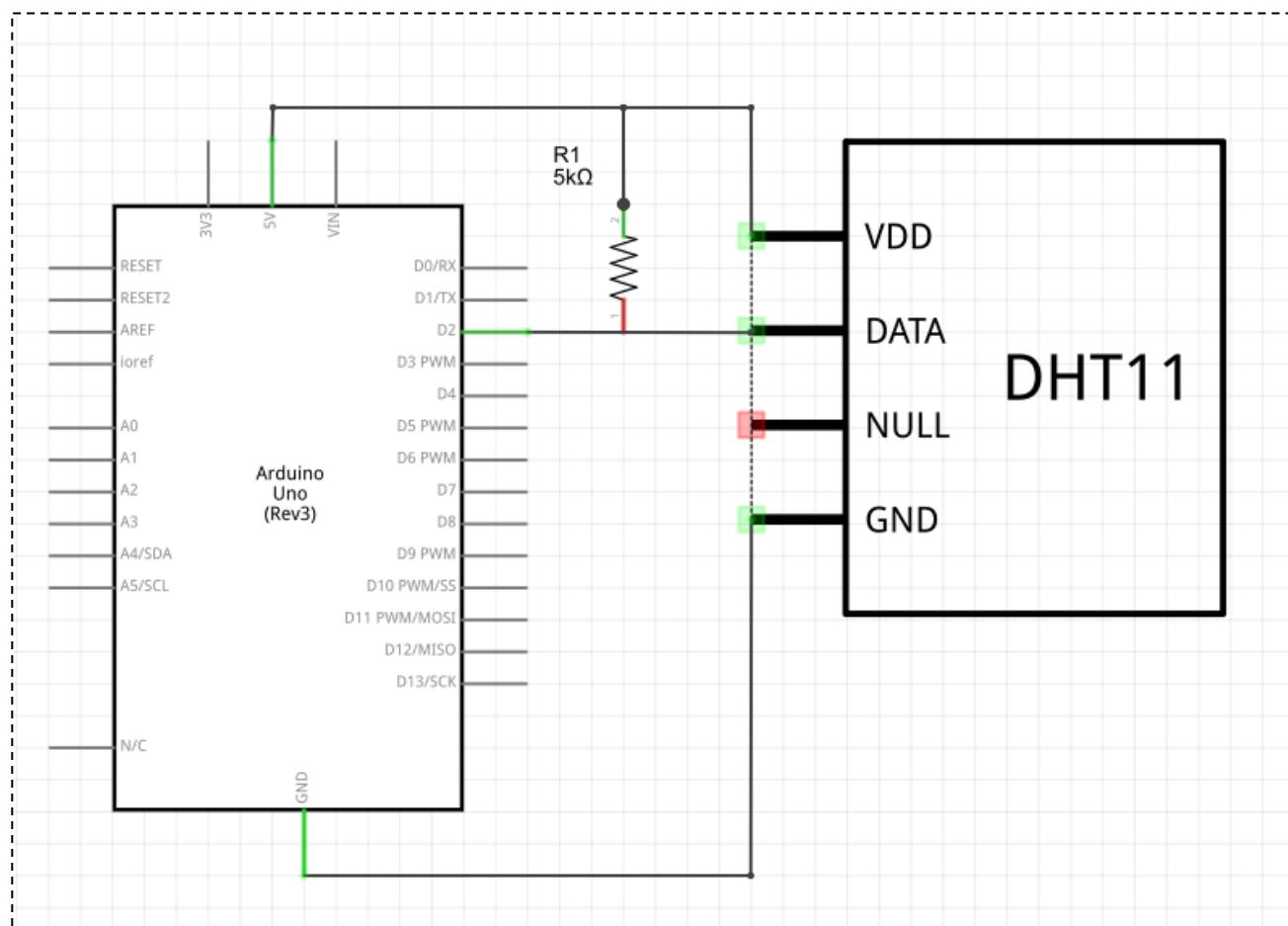
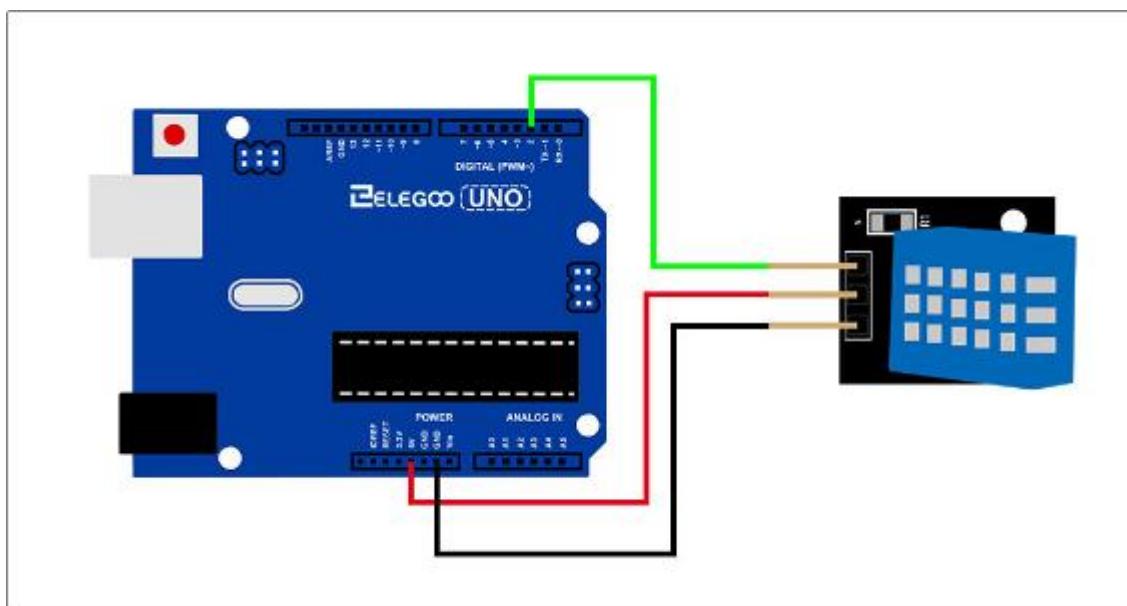


Schéma de câblage



Comme vous voyez, on utilise uniquement 3 connexions au capteur parce que l'une des broches n'est pas utilisée.

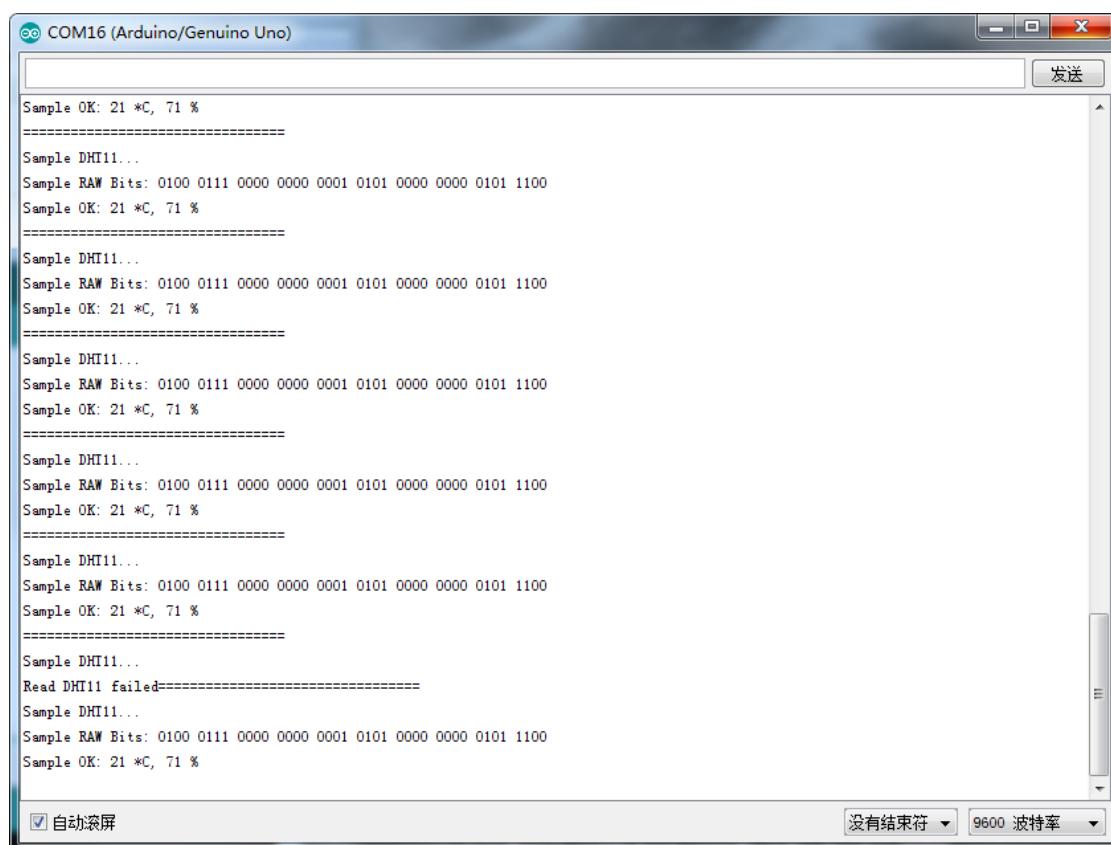
Les connexions sont: Voltage, Ground et Signal qui peuvent être connectés à n'importe quelle broche analogique sur notre UNO.

Code

Maintenant, nous avons fait une installation physique, et ce que nous avons besoin est le code.

Avant de l'exécuter, vous devez vous assurer que vous avez bien installé la bibliothèque <simpleDHT>. Sinon, vous devez la réinstaller. Si vous ne le faites pas, votre code ne fonctionnera pas.

Chargez le programme et puis rouvre le moniteur, il s'affichera les données suivantes: (à partir de l'affichage sur la température ambiante, on peut voir qu'il est 21 degrés.)



```
Sample OK: 21 *C, 71 %
=====
Sample DHT11...
Sample RAW Bits: 0100 0111 0000 0000 0001 0101 0000 0000 0101 1100
Sample OK: 21 *C, 71 %
=====
Sample DHT11...
Sample RAW Bits: 0100 0111 0000 0000 0001 0101 0000 0000 0101 1100
Sample OK: 21 *C, 71 %
=====
Sample DHT11...
Sample RAW Bits: 0100 0111 0000 0000 0001 0101 0000 0000 0101 1100
Sample OK: 21 *C, 71 %
=====
Sample DHT11...
Sample RAW Bits: 0100 0111 0000 0000 0001 0101 0000 0000 0101 1100
Sample OK: 21 *C, 71 %
=====
Sample DHT11...
Read DHT11 failed=====
Sample DHT11...
Sample RAW Bits: 0100 0111 0000 0000 0001 0101 0000 0000 0101 1100
Sample OK: 21 *C, 71 %
```

Leçon 20: Module de joystick analogique

Vue d'ensemble

Les joysticks analogiques sont un moyen idéal pour ajouter un certain contrôle dans vos projets.

Dans ce mode d'emploi, on peut apprendre comment utiliser le module de joystick analogique.

Composants requis

- (1) Elegoo UNO R3
- (1) Module de Joystick
- (5) F-M câbles

Introduction de composant

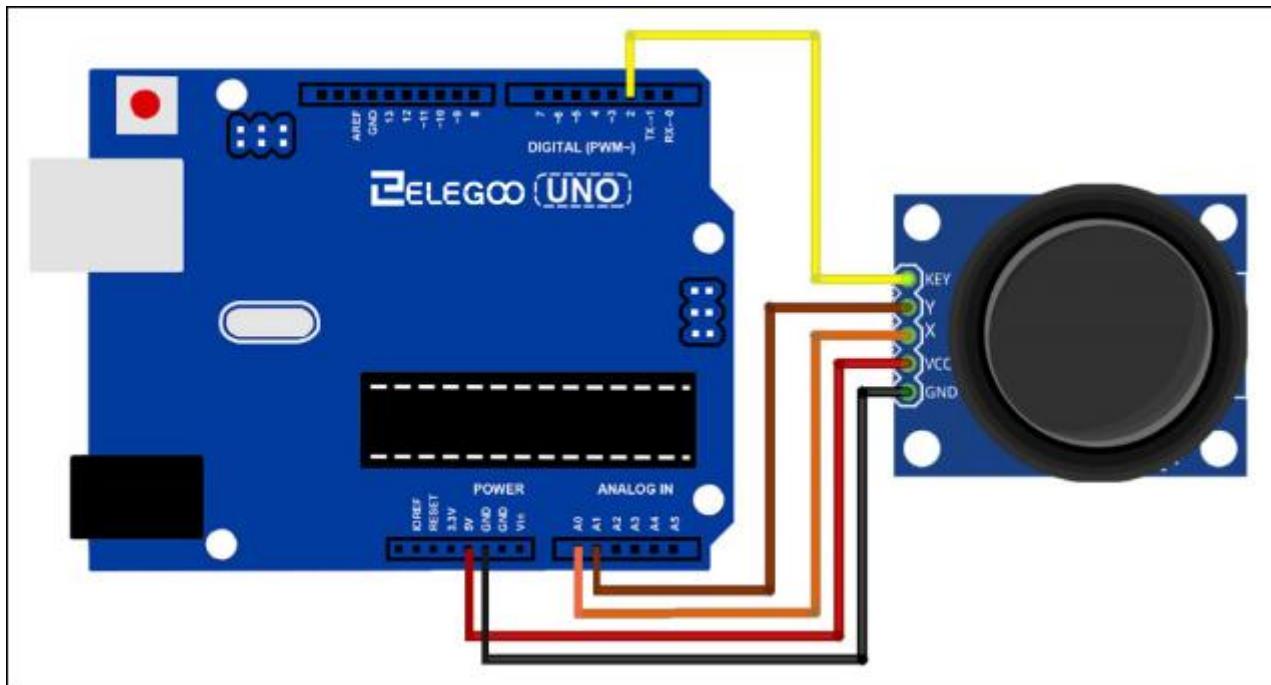
Joystick

Ce module possède 5 broches: Vcc, Ground, X, Y, Key. Il faut bien noter que les étiquettes sur les broches peuvent être légèrement différentes, selon l'endroit à partir d'où vous avez obtenu le module. La molette est analogique et il devrait fournir des relevés plus précis que les joysticks 'directionnel' simples qui utilisent certaines formes de boutons ou les commutateurs mécaniques. En outre, vous pouvez appuyer sur le joystick vers le bas (plutôt difficile sur la mienne) pour activer un bouton-poussoir 'press to select'.

Nous devons utiliser les broches analogiques d'Arduino à lire les données à partir des broches X/Y, et une broche numérique à lire le bouton. La broche Key est mise à la terre, lorsque le joystick est enfoncé, et il est flottant au contraire. Pour obtenir les relevés stables à partir de la broche Key>Select, il est nécessaire d'être connecté à la broche Vcc par une résistance de pull-up. Les résistances intégrées dans les broches numériques d'Arduino peuvent être utilisées. Pour le mode d'emploi sur comment activer les résistances de pull-up pour les broches d'Arduino, il peut être configuré comme les entrées.

Raccordement

Schéma de câblage



On a besoin de 5 connexions au joystick.

Les connexions sont: Key, Y, X, Voltage et GND.

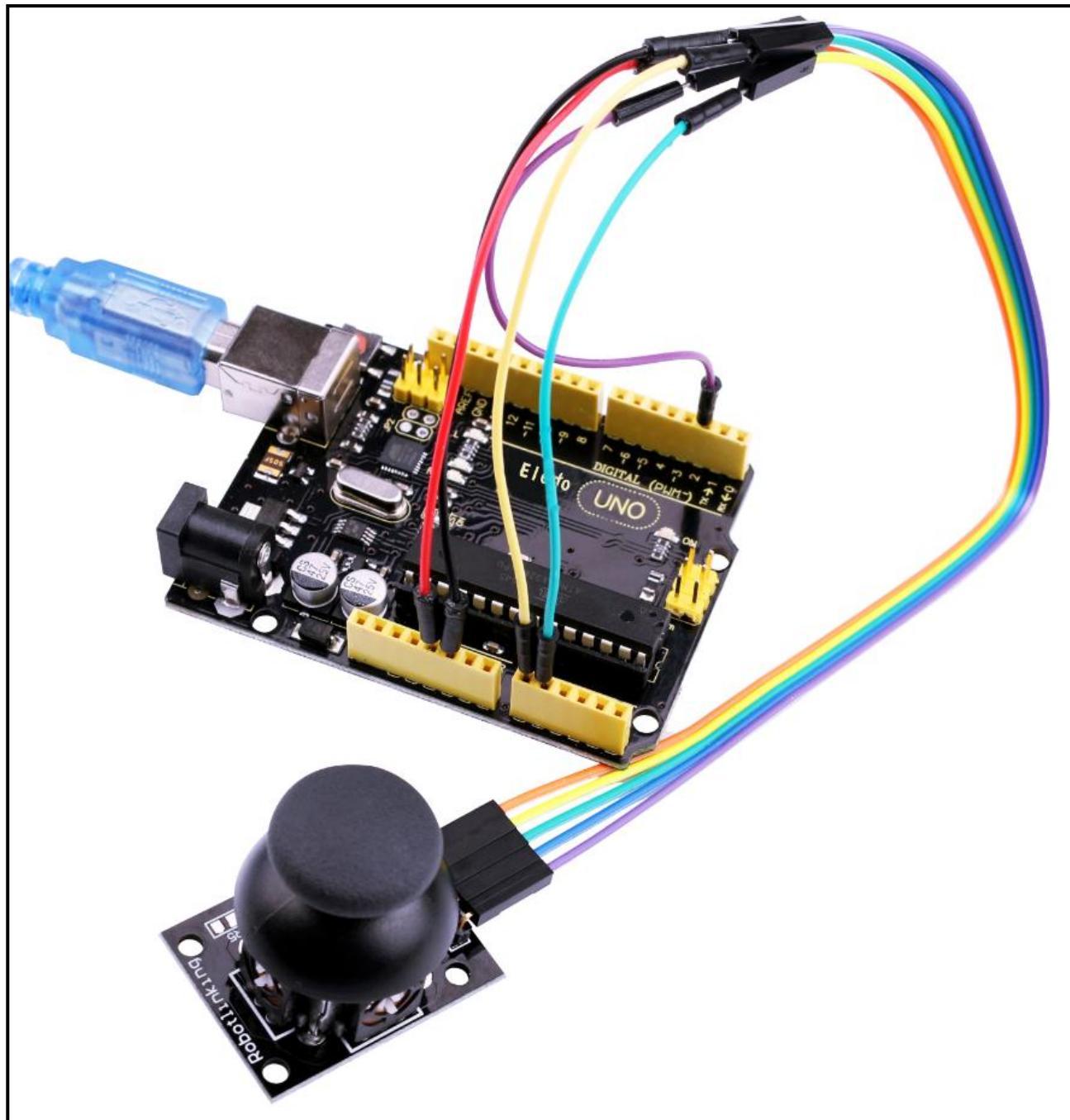
'Y' et 'X' sont analogiques et 'Key' est numérique. Si vous n'avez pas besoin de commutateur, vous pouvez utiliser uniquement 4 broches.

Code

Les joysticks analogiques sont essentiellement basés sur les potentiomètres afin qu'ils retournent des valeurs analogiques.

Lorsque le joystick est à la position de repos ou au milieu, il va retourner une valeur comme 512.

La gamme de valeurs est de 0 à 1024.



Leçon 21: Module de récepteur IR

Vue d'ensemble

À l'aide d'une télécommande à infrarouge est un excellent moyen d'avoir un contrôle sans fil de votre projet.

Les télécommandes infrarouges sont simples et faciles à utiliser. Dans ce mode d'emploi, on peut connecter le récepteur IR à UNO, et puis utiliser une bibliothèque conçue pour ce capteur particulier.

Dans notre sketch, nous aurons tous les codes hexadécimaux IR qui sont disponibles sur cette télécommande, on va détecter également si le code est reconnu et aussi si nous sommes à maintenir une touche enfoncée.

Composants requis

- (1) Elegoo UNO R3
- (1) Module de récepteur IR
- (1) Télécommande infrarouge
- (3) F-M câbles

Introduction de composant

Capteur de récepteur IR:

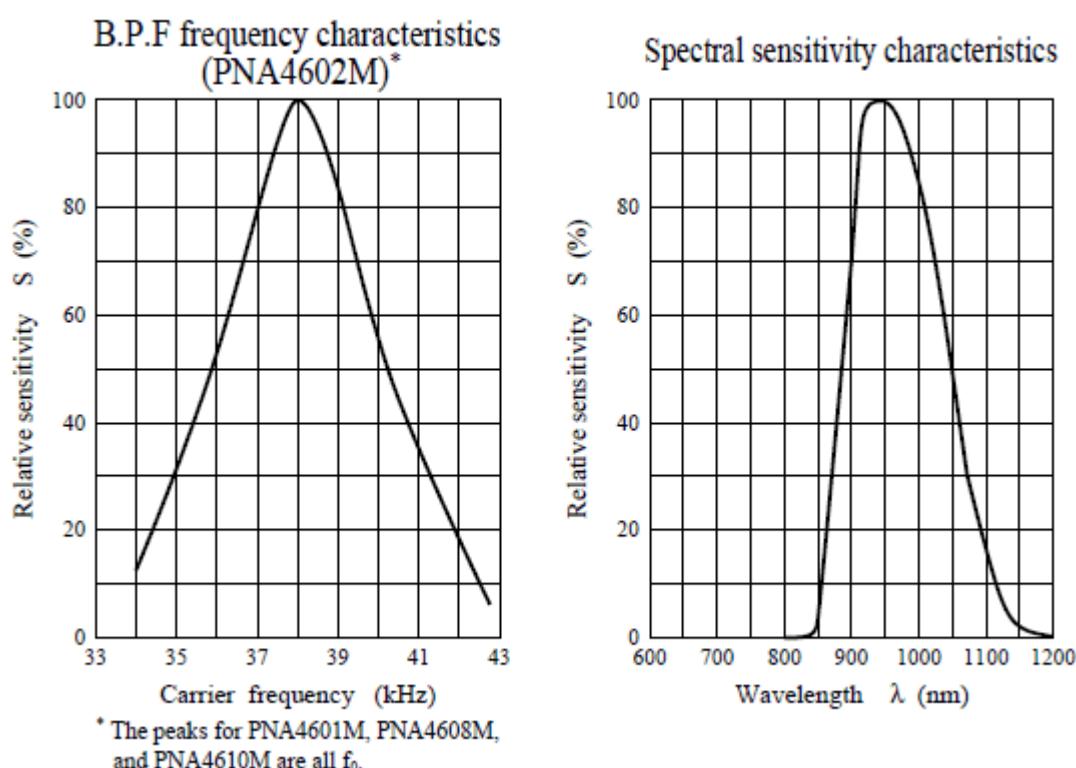
Les détecteurs IR sont les petites micropuces avec la cellule photoélectrique qui sont réglées à recevoir la lumière infrarouge. Elles sont presque toujours utilisées pour la détection de télécommande - chaque TV et lecture DVD possède l'une d'eux dans l'avant pour recevoir le signal IR à partir de la télécommande. Un voyant LED IR existe dans la télécommande, qui émet les impulsions IR à demander la télévision pour activer, désactiver ou changer le canal. La lumière IR n'est pas visible à l'oeil humain, c'est-à-dire qu'elle prend un peu plus de travail pour tester une configuration.

Il existe quelques différences entre eux et à appeler une cellule photoélectriques CdS :

Les détecteurs infrarouges sont spécialement filtrés pour la lumière infrarouge, ils ne sont pas capables de détecter la lumière visible. D'autre part, les cellules photoélectriques sont parfaites pour détecter la lumière visible Jaune/Vert, mais pas bon pour la lumière infrarouge.

- Les détecteurs infrarouges possèdent un démodulateur à l'intérieur qui ressemble pour IR modulée à 38 KHz. Un brillant unique d'une LED IR ne sera pas détecté, il doit être PWM clignotant à 38KHz. La cellule photoélectrique ne possède aucune sorte de démodulateur et elle peut détecter toute fréquence (y compris DC) dans la vitesse de réponse de la cellule photoélectrique (qui est d'environ 1Khz).
- Les détecteurs infrarouges sont d'une sortie numérique – soit qu'ils détectent un signal infrarouge à 38 KHz et une sortie faible (0V), soit qu'ils ne détectent aucun signal et détectent une sortie forte (5V). La cellule photoélectrique joue un rôle comme une résistance, et la résistance varie selon la quantité de lumière qu'il est exposé.

Ce que vous pouvez mesurer



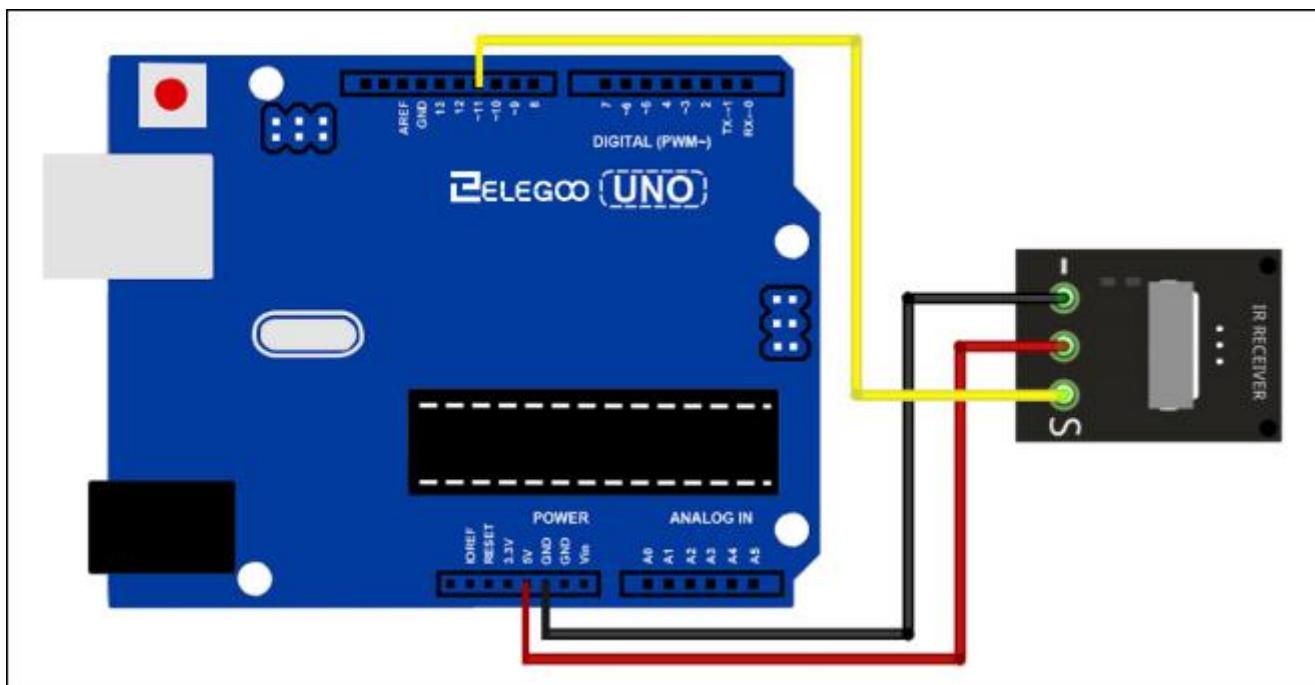
Comme ce que vous pouvez voir à partir de ces graphiques de la fiche technique, la détection sur la fréquence de crête est à 38 KHz et la couleur de LED à la crête est de 940nm. C'est possible de fonctionner d'environ 35Khz à 41Khz, mais sa sensibilité va diminuer afin qu'elle ne détecte pas aussi bien de loin. De même, LEDs de 850 à 1100 nm peuvent être utilisées, mais elles ne fonctionneront pas hors la gamme entre 900-1000nm où il assure

d'obtenir les voyants correspondants! Inspectez la fiche technique de votre LED IR pour vérifier la longueur d'onde.

Pour obtenir 940nm – n'oubliez pas que 940nm est une lumière invisible (Infrarouge)!

Raccordement

Schéma de câblage



Il existe 3 connexions au récepteur IR.

Les connexions sont: Signal, Tension et Mise à la terre.

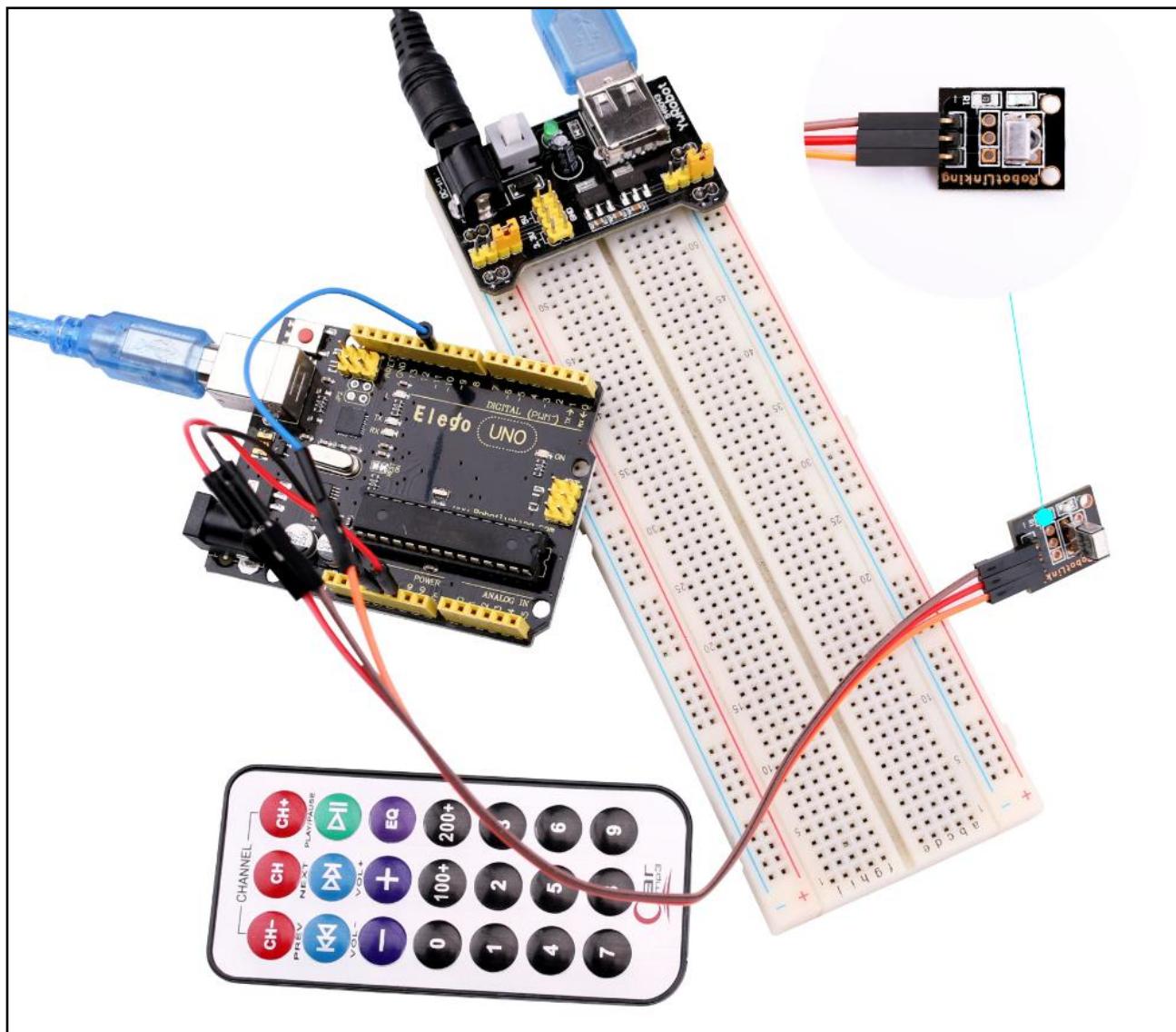
'-' est la mise à la terre, 'S' est le signal et la broche au milieu est la tension 5V.

Code

Avant de l'exécuter, assurez-vous que vous avez installé la bibliothèque <IRremote>, et le cas échéant, vous devez la réinstaller. Sinon, votre code ne fonctionnera pas.

Ensuite, nous devons déplacer la bibliothèque <RobotIRremote> hors du dossier Library, on fait ça puisque la bibliothèque est en conflit avec celui que nous allons utiliser. Lorsque vous avez terminé à programmer votre microcontrôleur, vous pouvez simplement faire glisser cette bibliothèque à nouveau dans le dossier bibliothèque.

Après d'avoir installé la bibliothèque, il suffit d'aller à l'avant et redémarrer votre logiciel IDE.



Leçon 22: Moteurs à courant continu

Vue d'ensemble

Dans cette leçon, vous pouvez apprendre comment contrôler un petit moteur à courant continu à l'aide d'UNO R3 et d'un transistor.

Composants requis

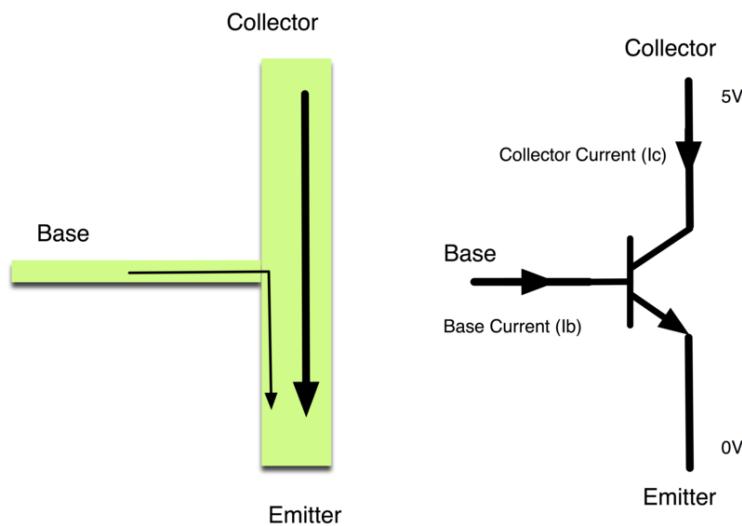
- (1) Elegoo UNO R3
- (1) Carte de prototypage
- (1) Moteur à courant continu 6v
- (1) PN2222
- (1) 1N4007
- (1) 220 ohm résistance
- (3) M-M câbles

Introduction de composant

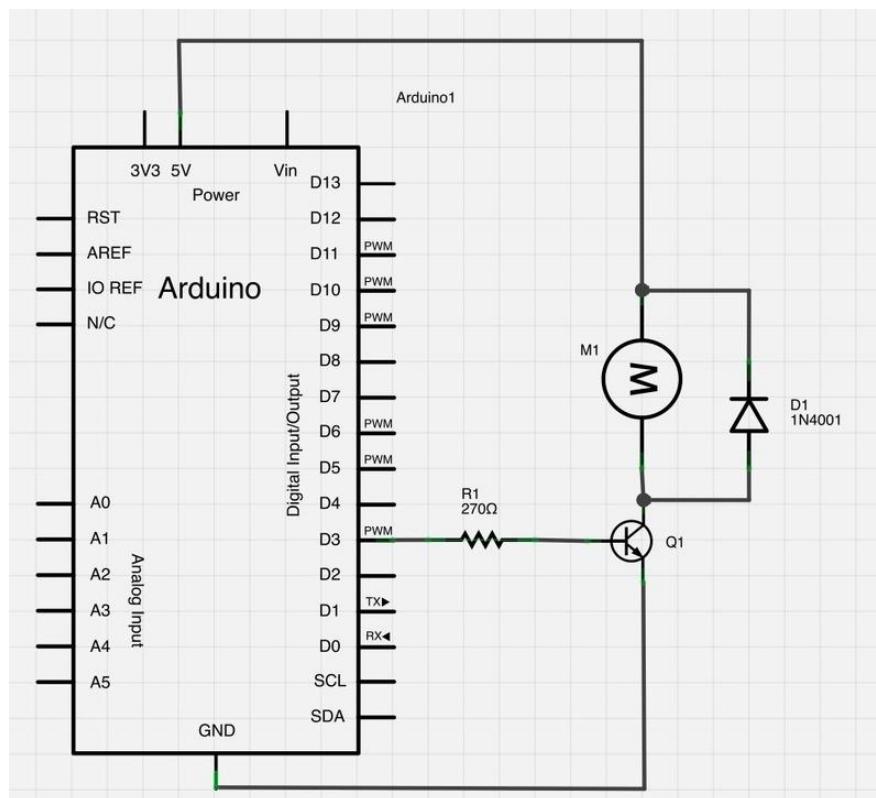
TRANSISTORS:

Le petit moteur à courant continu est susceptible d'utiliser plus d'énergie qu'une sortie numérique de la carte d'UNO R3 peut gérer directement. Si l'on a essayé de brocher le moteur directement à une broche de la carte d'UNO R3, c'est possible d'endommager la carte d'UNO R3.

Un petit transistor comme PN2222 peut être utilisé comme un commutateur pour utiliser uniquement un petit courant à partir de la sortie numérique de la carte d'UNO R3 pour contrôler la plus grande courant du moteur.



Le transistor possède trois conducteurs. La plupart d'électricité s'écoule à partir du collecteur à l'émetteur, mais ce cas ne se produira que si une petite quantité se déverse dans la connexion de base. Ce petit courant est fourni par la sortie numérique de la carte d'UNO R3. Le schéma ci-dessous est un diagramme de schéma de l'Arduino. Comme la disposition de la carte de prototypage, c'est une façon de montrer comment les parties d'un projet électronique sont connectées ensemble.



La broche D3 de la carte d'UNO R3 est connectée à la résistance. Comme l'utilisation d'une LED, il limite le courant qui circule dans le transistor par l'intermédiaire de la base.

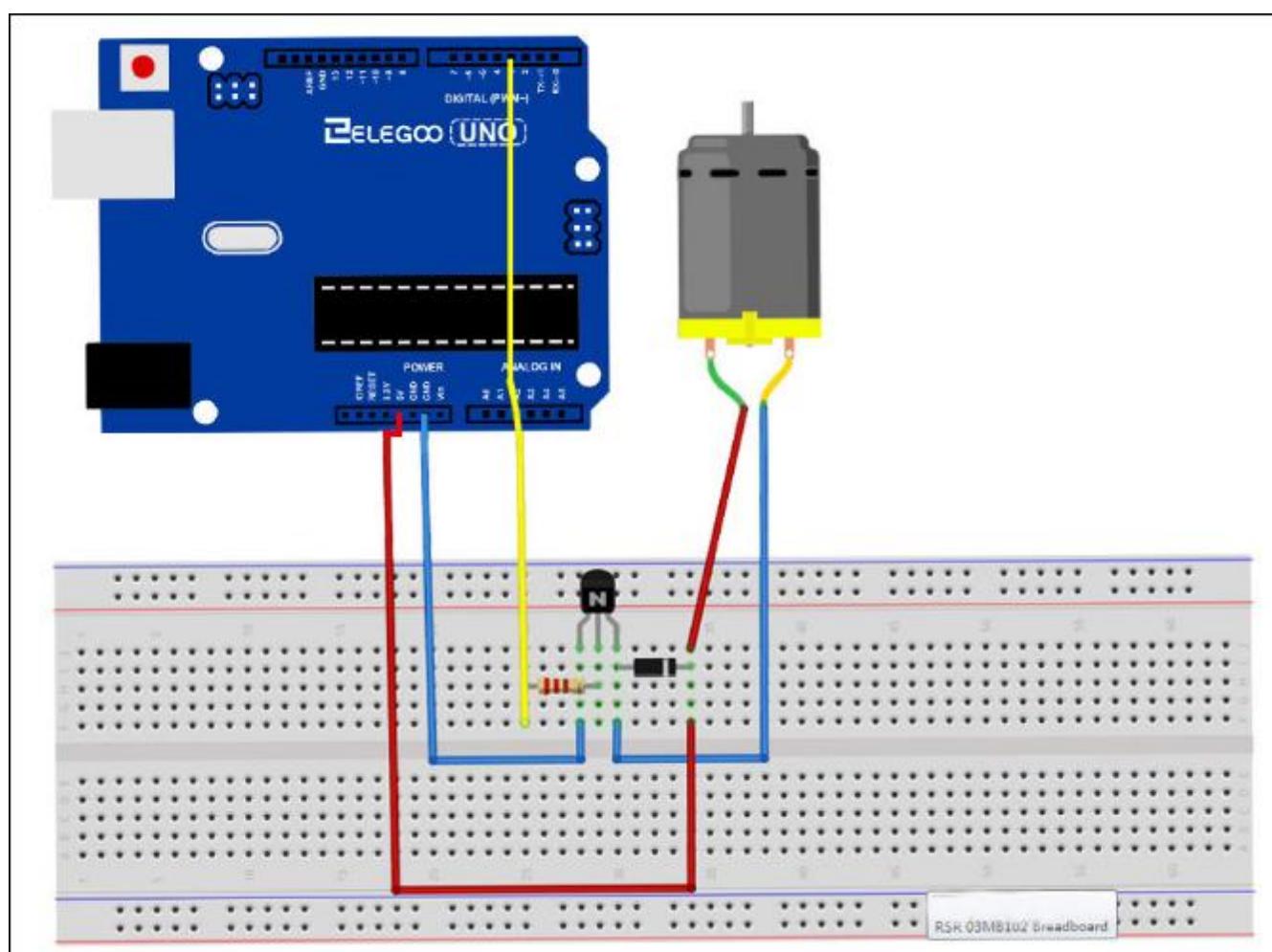
Il existe une diode connectée à travers des connexions du moteur. Les diodes ne permettent que l'électricité à circuler dans une orientation (l'orientation de leur flèche).

Lorsque vous coupez la puissance d'un moteur, vous pouvez avoir un pic négatif de tension, afin de pouvoir endommager votre carte d'UNO R3 ou le transistor. La diode joue un rôle d'éviter ce dommage, par un court-circuitage d'une telle inversion de courant du moteur.

Grâce à cette expérience, vous avez appris comment conduire LCD 1602. Alors vous pouvez créer vos propres messages à s'afficher! Vous pouvez également essayer de mettre votre LCD 1602 à afficher les chiffres.

Raccordement

Schéma de câblage



Vous allez utiliser une sortie analogique de la carte d'UNO R3 (PWM) pour contrôler la vitesse du moteur par l'envoi d'un chiffre entre 0 et 255 à partir du moniteur serial.

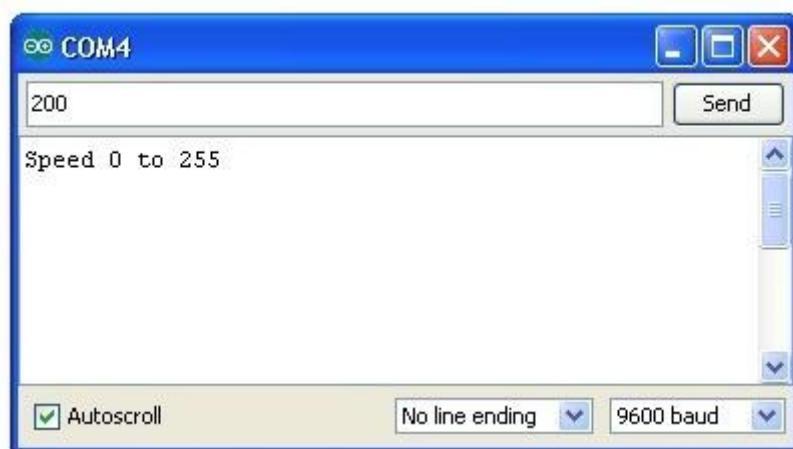
Lorsque vous mettez la carte de prototypage ensemble, il existe deux éléments à rechercher. Tout d'abord, assurez-vous que le côté plat du transistor est au côté droit de la carte de prototypage.

Deuxième, l'extrémité rayée de la diode doit être orientée vers la ligne d'alimentation +5V – veuillez voir l'image ci-dessous!

Le moteur fourni avec les kits ne consomme pas plus de 250 mA, mais si vous avez un moteur différent, il est facile d'atteindre à 1000mA, plus d'un port USB peut en gérer ! Si vous n'assurez pas le courant du moteur, alimentez la carte d'UNO R3 à partir un adaptateur mural, mais qu'USB.

Le transistor joue un rôle comme un commutateur, pour contrôler la puissance du moteur, et la broche 3 de la carte d'UNO R3 est utilisée pour allumer ou éteindre le transistor et aussi dénommée 'motorPin' dans la sketche.

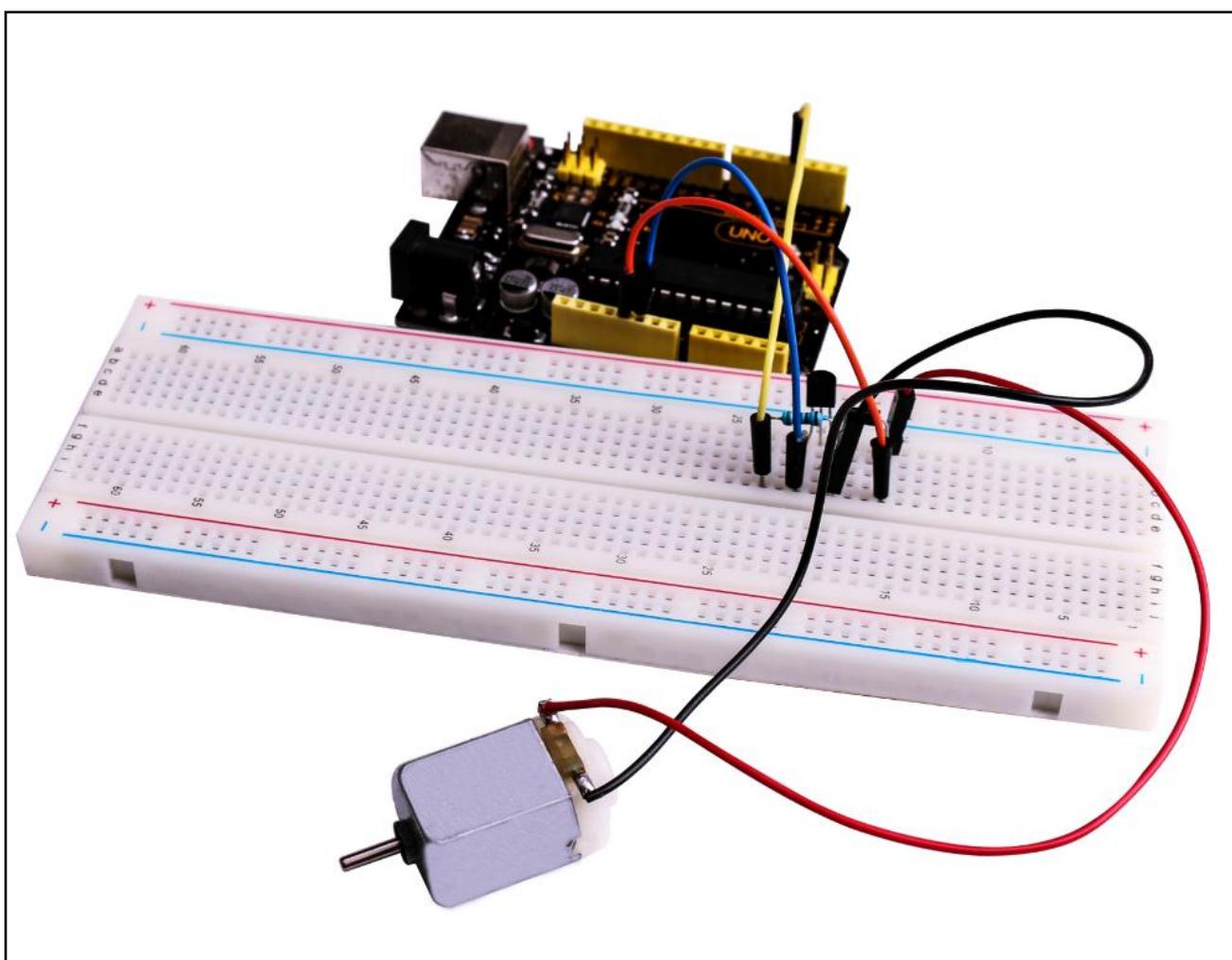
Lorsque la sketche démarre, elle vous invite à vous rappeler que pour contrôler la vitesse du moteur, et vous devez entrée une valeur entre 0 et 255 dans le moniteur serial.



Dans la fonction 'loop', la commande 'Serial.parseInt' est utilisé pour lire les chiffres saisis au texte dans le moniteur serial et les convertir à 'int'.

Vous pouvez taper n'importe quel chiffre ici, alors la déclaration 'if' dans la ligne suivante est seulement une écriture analogique avec ce chiffre lorsque le chiffre est entre 0 et 255.

Code



Leçon 23: Moteur pas-à-pas

Vue d'ensemble

Dans cette leçon, vous pouvez apprendre une façon intéressante et facile à entraîner un moteur pas-à-pas.

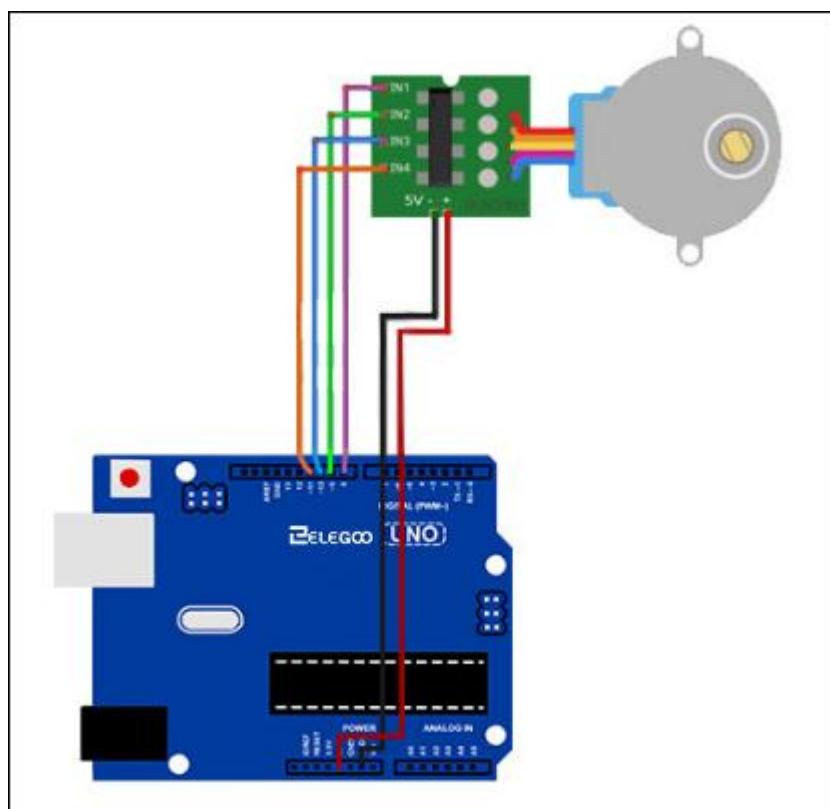
Le stepper à utiliser est livré avec son propre carte de conducteur afin de permettre facilement à connecter à notre UNO.

Composants requis

- (1) Elegoo UNO R3
- (1) Module de conducteur du moteur pas-à-pas Uln2003
- (1) Moteur pas-à-pas
- (9) F-M câbles

Raccordement

Schéma de câblage



On utilise quatre broches à contrôler le Stepper.

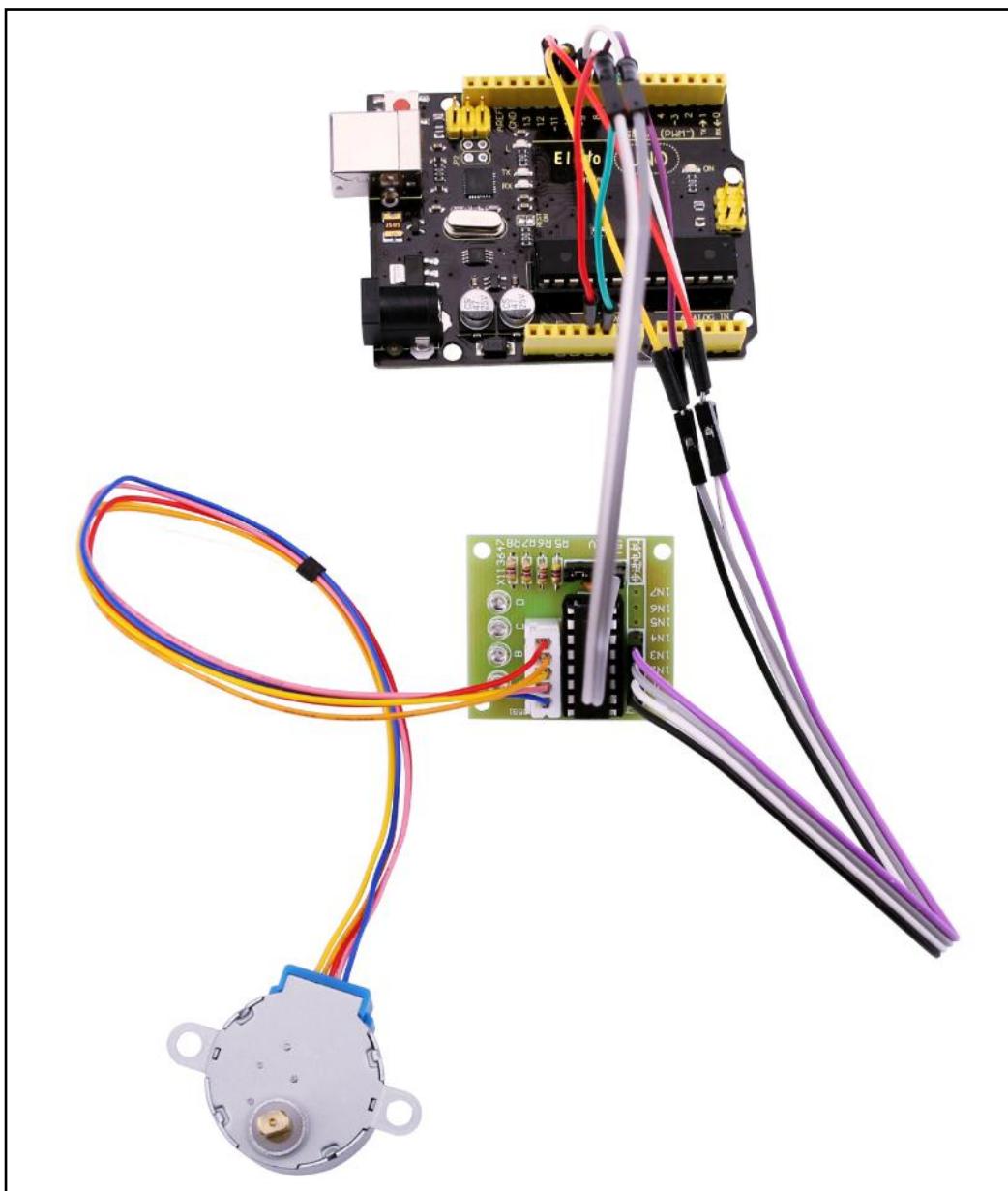
Les broches 8-11 sont utilisées à contrôler le moteur pas-à-pas.

Nous connectons 5V et GND à partir de l'UNO au moteur pas-à-pas.

Code

Avant de l'exécuter, assurez-vous que vous avez installé la bibliothèque

<Stepper_Motor_Control>, et le cas échéant, réinstallez-la. Sinon, votre code ne fonctionnera pas.



Leçon 24: Contrôler le moteur pas-à-pas avec la télécommande

Vue d'ensemble

Dans cette leçon, vous pouvez apprendre une façon intéressante et facile pour contrôler un moteur pas-à-pas à distance à l'aide une télécommande infrarouge.

Le stepper à utiliser est livré avec son propre carte de conducteur afin de permettre facilement à connecter à notre UNO.

Si nous ne voulons pas entraîner le moteur directement à partir de l'UNO, on peut utiliser un petit bloc d'alimentation de la carte de prototypage pas cher à insérer correctement à notre carte de prototypage et l'alimenter avec l'alimentation de 9V 1Amp.

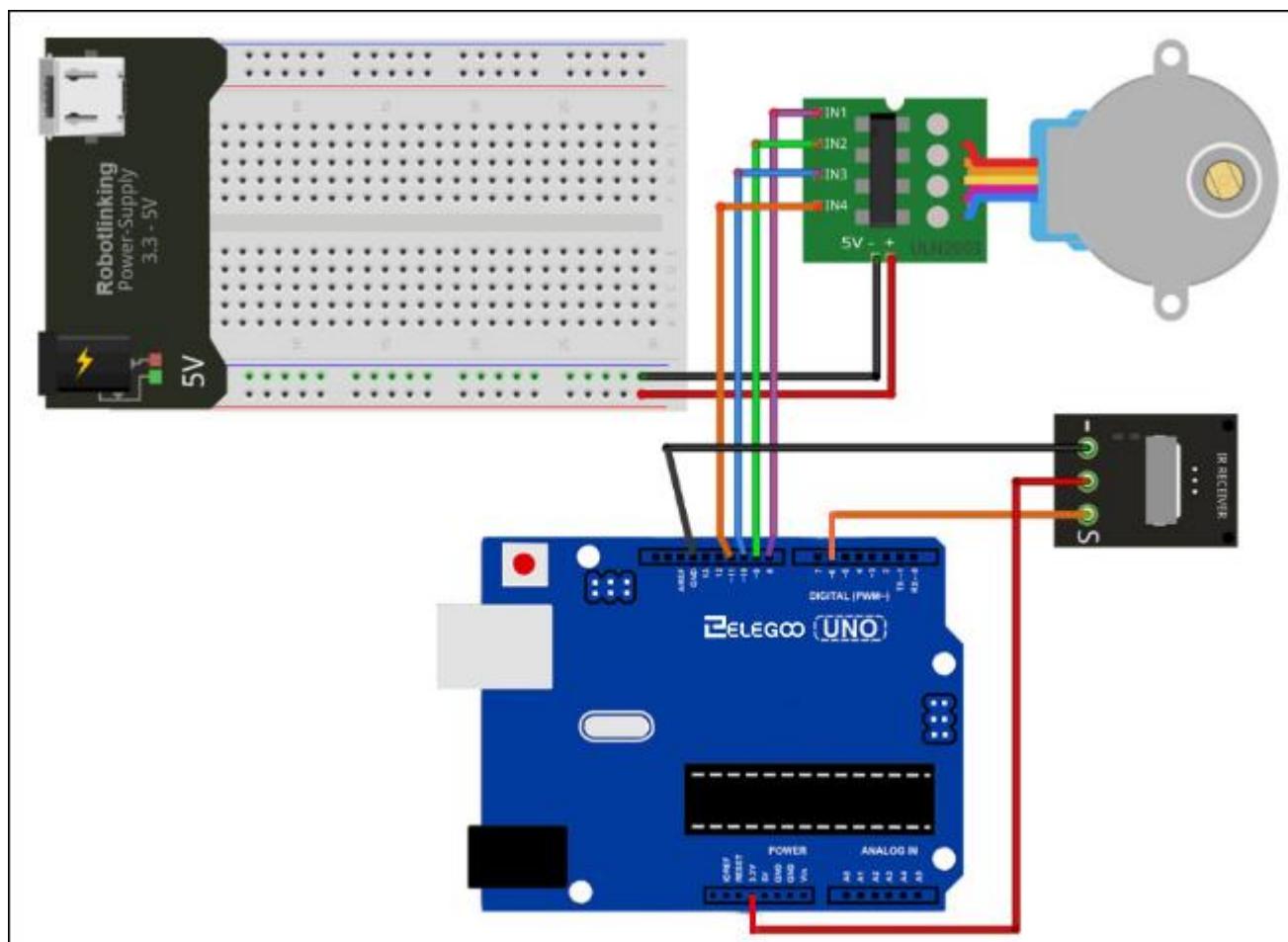
Le capteur IR est connecté à UNO directement puisqu'il n'utilise presque aucune alimentation.

Composants requis

- (1) Elegoo UNO R3
- (1) Carte de prototypage
- (1) Module de récepteur IR
- (1) Télécommande infrarouge
- (1) Module de conducteur du moteur pas-à-pas Uln2003
- (1) Moteur pas-à-pas
- (1) Module d'alimentation
- (1) Adapteur
- (9) F-M câbles

Raccordement

Schéma de câblage



On utilise quatre broches à contrôler le Stepper et 1 broche pour le capteur IR.

Les broches 8-11 sont utilisées à contrôler le moteur pas-à-pas et la broche 6 est utilisée pour recevoir l'information d'IR.

Nous connectons 5V et GND à partir de l'UNO au capteur. Par mesure de précaution, utilisez un bloc d'alimentation de la carte de prototypage à alimenter le moteur pas-à-pas, puisqu'il peut utiliser plus d'énergie et nous ne voulons pas endommager le bloc d'alimentation de l'UNO.

Code

Avant de l'exécuter, assurez-vous que vous avez installé la bibliothèque <IRremote> <Stepper_Motor_Control>, et le cas échéant, réinstallez-la. Sinon, votre code ne fonctionnera pas.

