THE BASIC STARTER KIT TUTORIAL

FOR UNO

- EN FRANCAIS -

V1.0.19.7.24

Préface

Notre companie

Etablie en 2001, Elegoo inc. est une entreprise de technologie qui s'efforce de fournir des contenus open source pour la recherche, l'industrie, mais aussi l'éducation.

Nos locaux sont à Shenzhen, qui est connue comme la "Silicon Valley" chinoise. Nous sommes plus de 150 employés.

Notre production s'étend des câbles de connexion, cartes UNO jusqu'aux starters kits, conçus pour tous les niveaux de savoir. Nous produisons aussi des accessoires pour les cartes Raspberry comme les écrans TFT 2.8'' et les STM32.

Dans le futur, nous envisageons de nous investir dans le domaine des imprimantes 3D.

Tous nos produits sont conformes aux normes de qualité internationales et sont reconnus sur les marchés où ils sont commercialisés.

Notre site web: http://www.elegoo.com

Nos boutiques Amazon :

US Amazon storefront: http://www.amazon.com/shops/A2WWHQ25ENKVJ1

CA Amazon storefront: http://www.amazon.ca/shops/A2WWHQ25ENKVJ1

UK Amazon storefront: http://www.amazon.co.uk/shops/AZF7WYXU5ZANW

DE Amazon storefront: http://www.amazon.de/shops/AZF7WYXU5ZANW

FR Amazon storefront: http://www.amazon.fr/shops/AZF7WYXU5ZANW

ES Amazon storefront: http://www.amazon.es/shops/AZF7WYXU5ZANW

IT Amazon storefront: http://www.amazon.it/shops/AZF7WYXU5ZANW

Ce tutoriel

Ce tutoriel est conçu pour les débutants. Vous y apprendrez les informations essentielles sur la manière d'utiliser une carte UNO. Mais si vous souhaitez allez plus loin, nous vous conseillons de faire acquisition de livres comme il y en a plusieurs références en vente sur des sites comme Amazon.

Beaucoup de codes contenus dans ce tutoriel sont de Simon Monk qui est un auteur reconnu de livre concernant l'Arduino.

Service consommateur

Nous nous efforçons de produire nos kits avec le plus grand soin et un souci permanent de qualité.

Nous sommes à votre écoute. N'hésitez pas à prendre contact avec nous à l'adresse suivante : service@elegoo.com ou EUservice@elegoo.com.

Nous sommes impatients de lire vos remarques et/ou suggestions.

Nous mettrons tout en œuvre pour qu'un ingénieur expérimenté vous réponde dans les 12h ou 24h pendant les périodes de congés.

Packing list

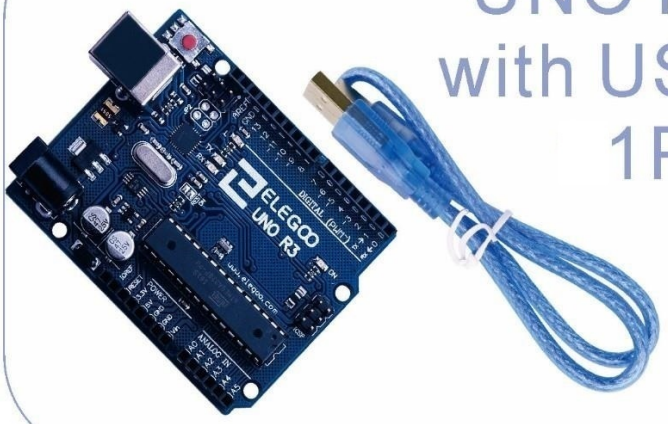
www.elegoo.com

RGB LED
2PCS

Photoresistor
(photocell)
1PC

Resistor
120PCS

UNO R3
with USB
1PC



Tilt Ball Switch
1PC

LED
30PCS

Button(Small)
5PCS

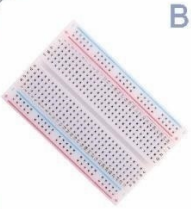
Active Buzzer
1PC

74HC595 IC
1PC

F-M Dupont Wire
5PCS

Breadboard
Jumper Wire
65PCS

Breadboard
1PC



Contact us : service@elegoo.com

Content

Leçon 0 Installation de l'environnement de programmation

Introduction

L'IDE Arduino (Arduino Integrated Development Environment (ou IDE)) est un logiciel dédié de la plateforme Arduino.

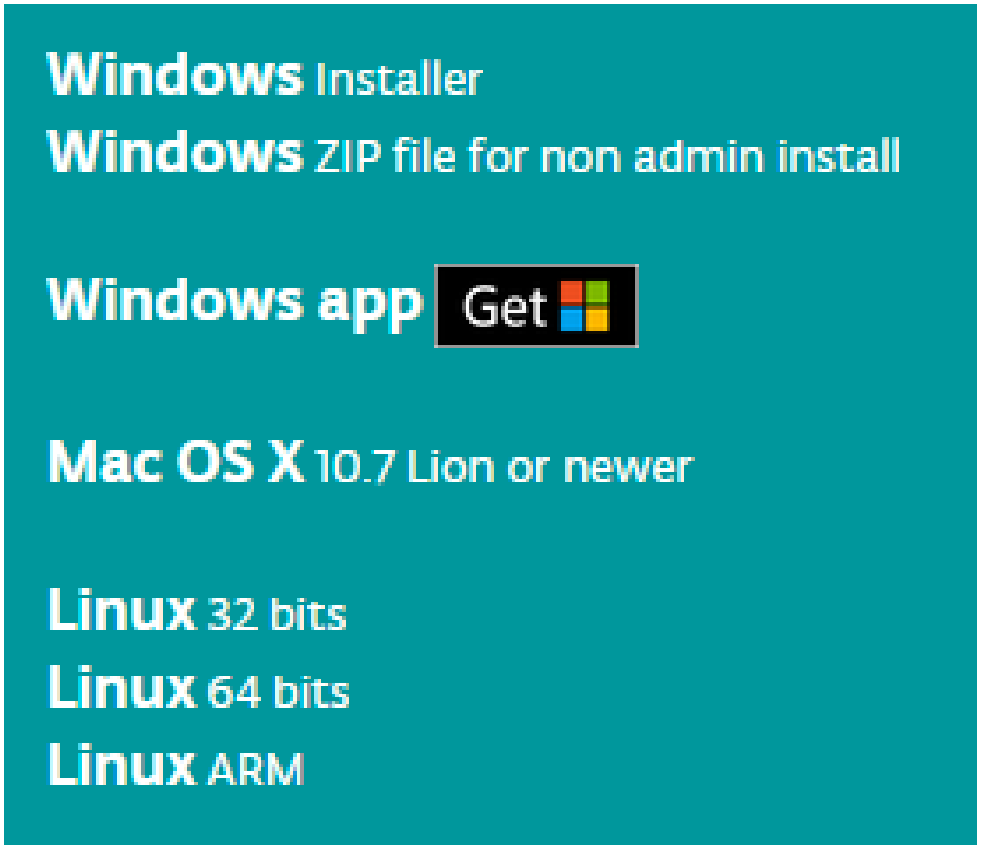
Dans cette leçon, vous allez procéder à l'installation de celui-ci.

Le logiciel est disponible pour Windows, MAC, LINUX.

STEP 1: Rendez-vous sur https://www.arduino.cc/en/Main/Software

Téléchargez la dernière version disponible (qui n'est plus forcément celle présente sur la capture d'écran).

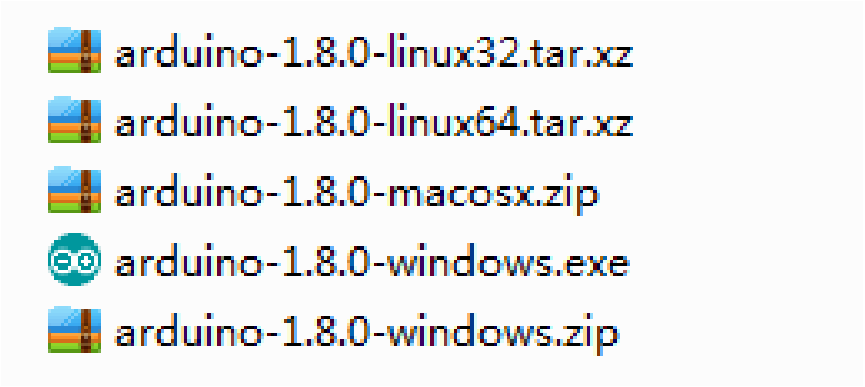
STEP2：Téléchargez la version correspondant à votre ordinateur



Si vous souhaitez faire un don pour soutenir la communauté, cliquez sur “contribute…”

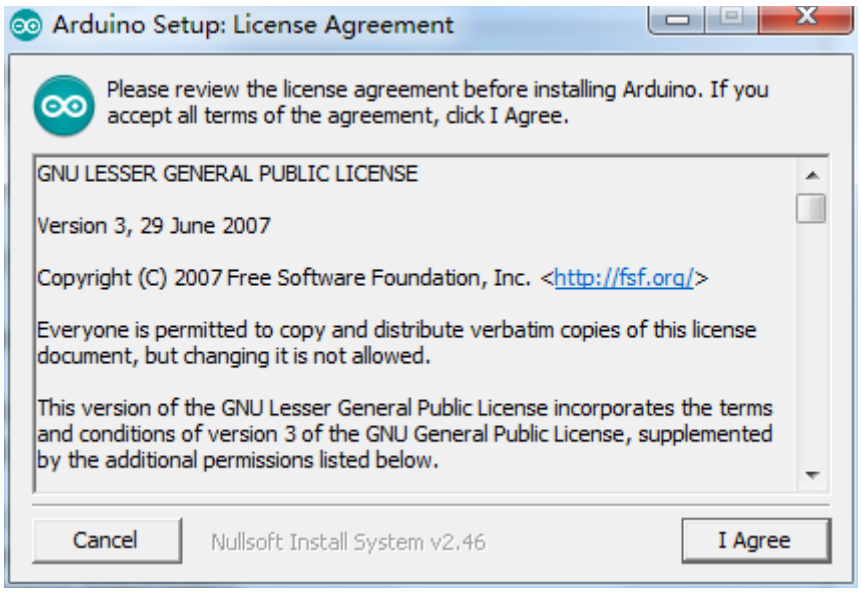
Si vous souhaitez téléchargez simplement, cliquez sur “just download”.

Sélectionnez le fichier correspondant à votre environnement de travail.

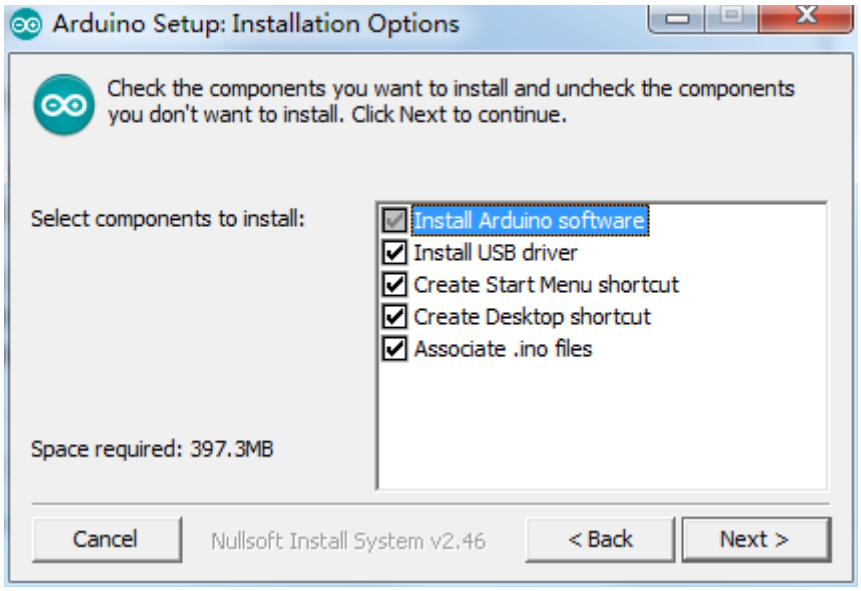


Installation sous Windows

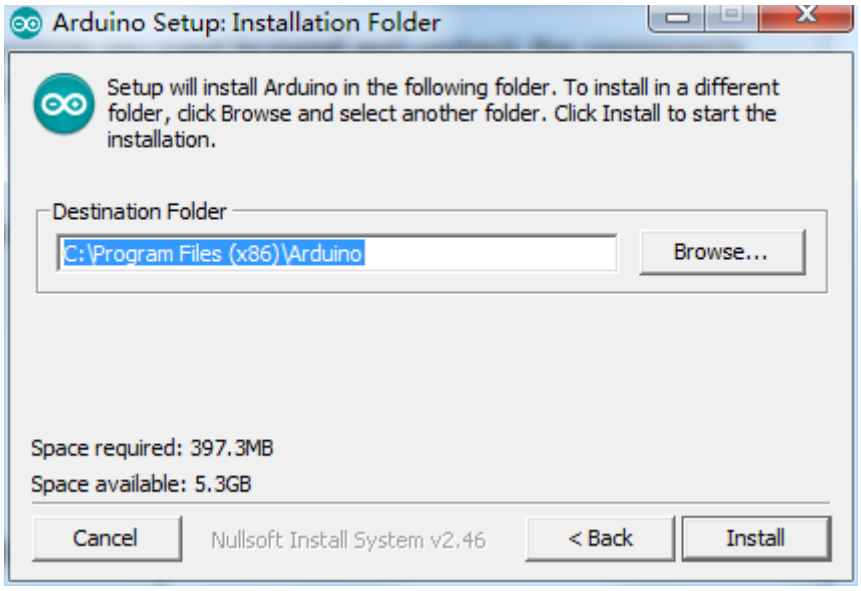
Exécutez le fichier.



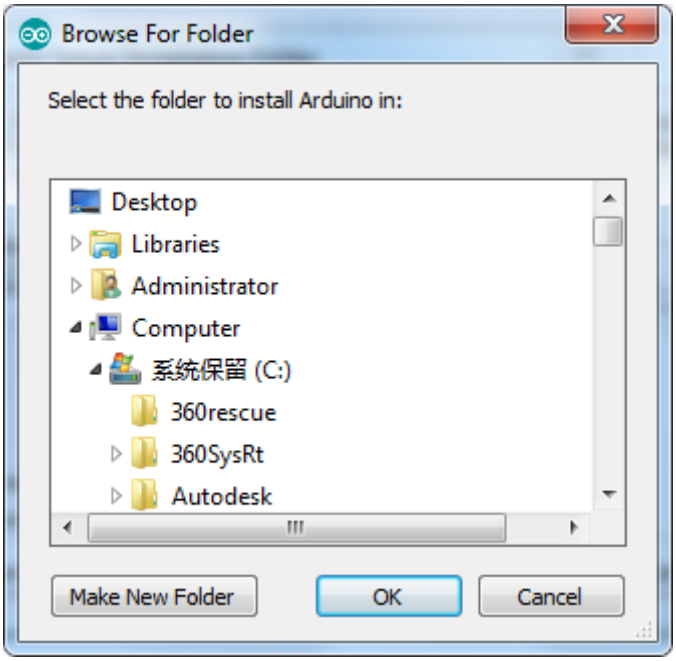
Cliquez sur *I Agree*



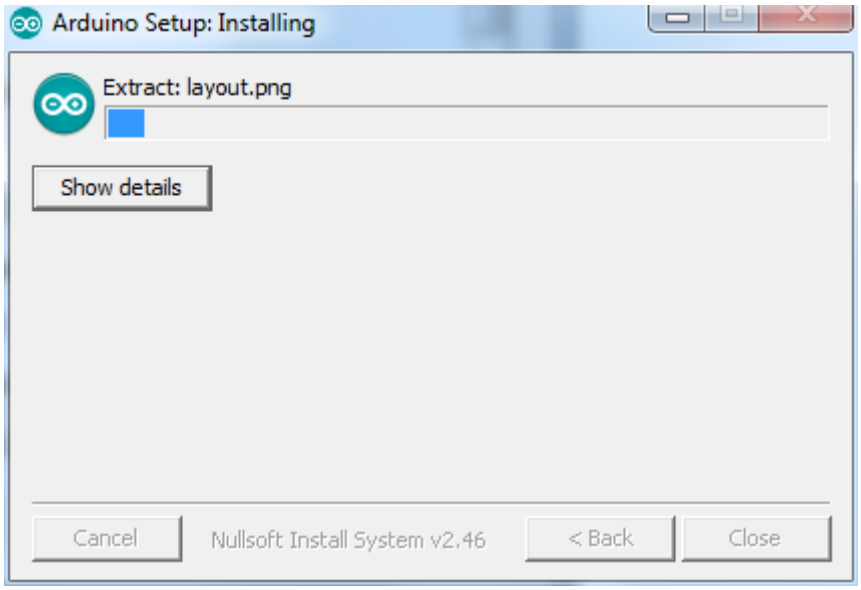
Cliquez *Next*



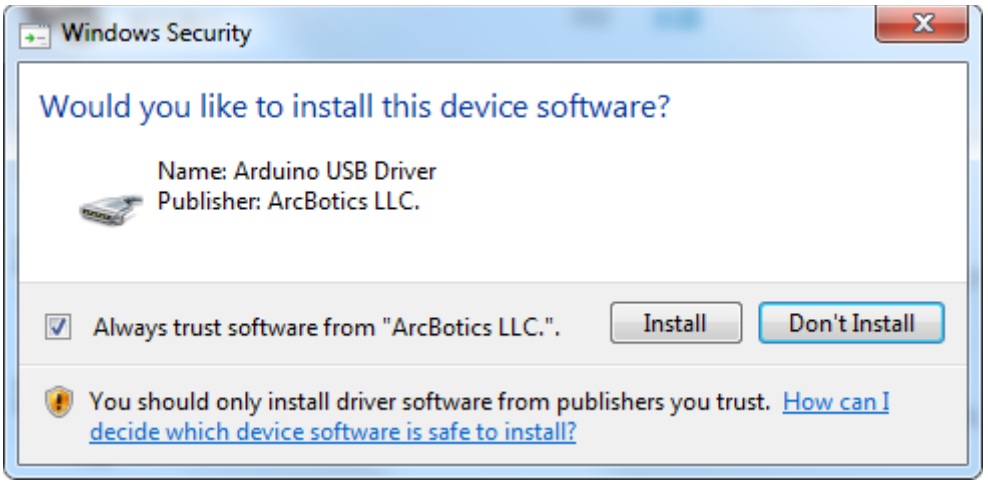
Cliquez sur *Browse* si vous souhaitez définir un autre répertoire.



Cliquez sur *Install*

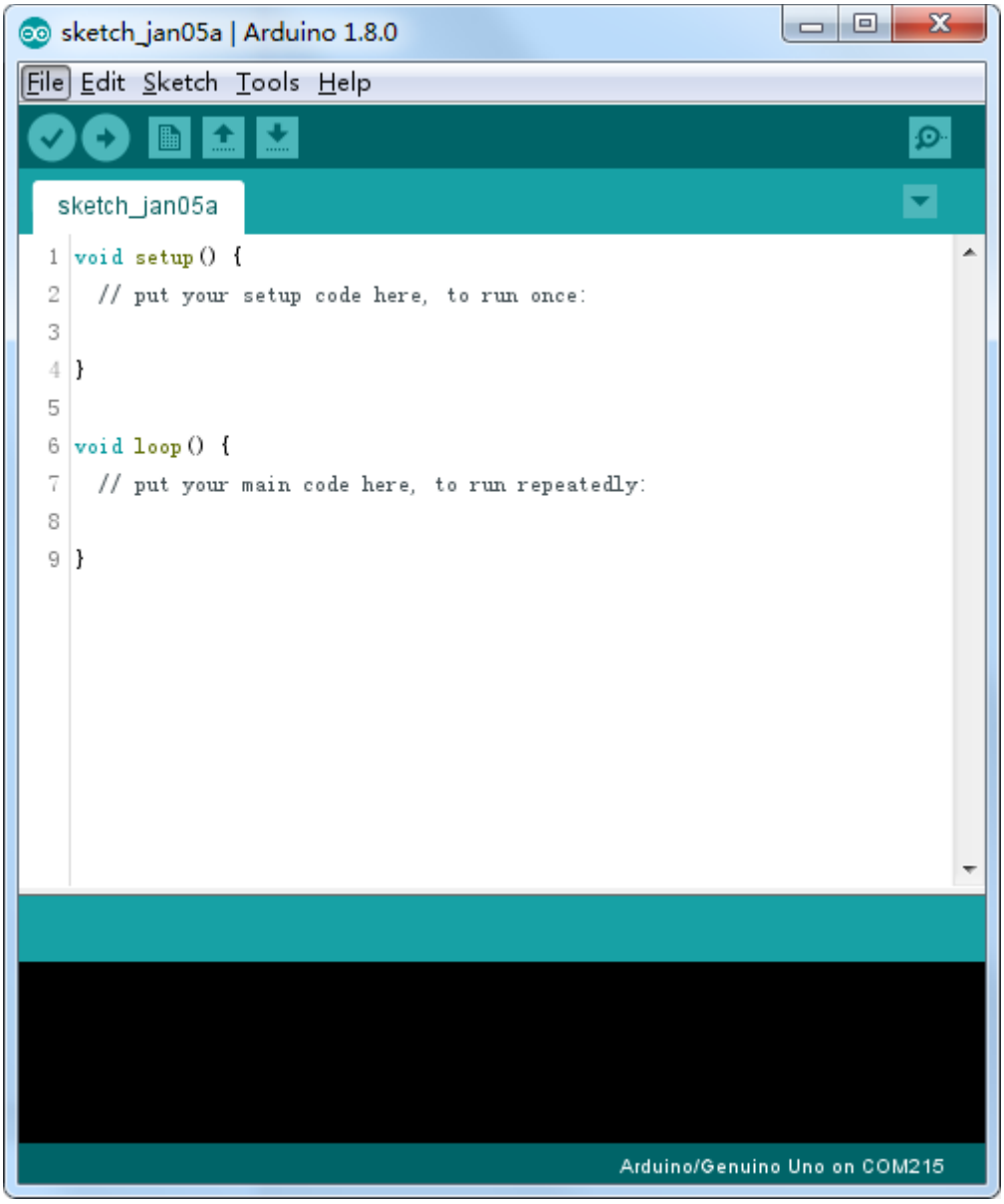


Enfin, cliquez de nouveau sur *Install*.



L'icône suivant apparait sur votre bureau

Double cliquez pour lancer l'IDE

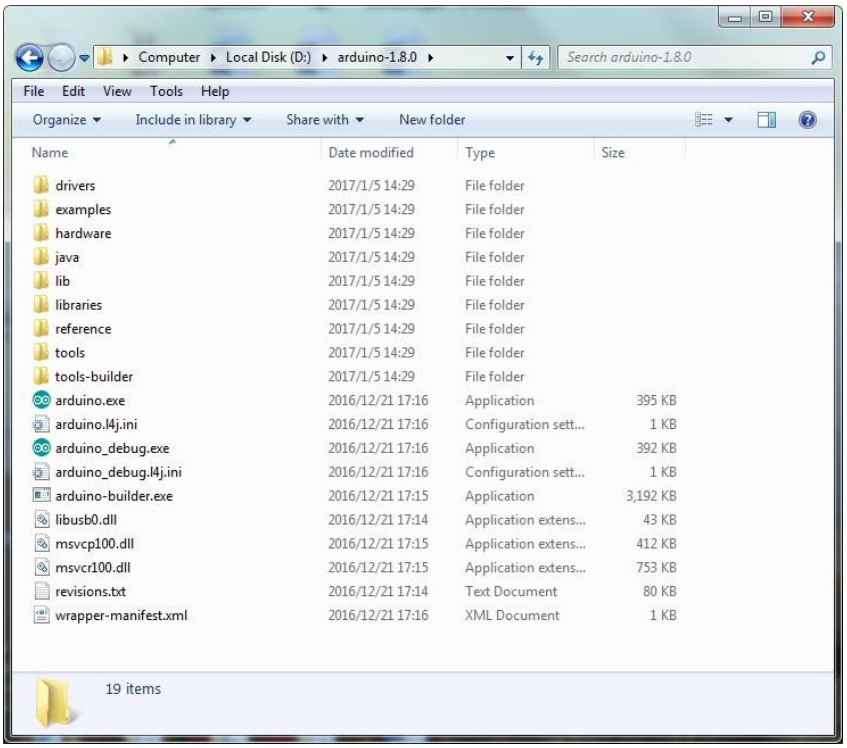


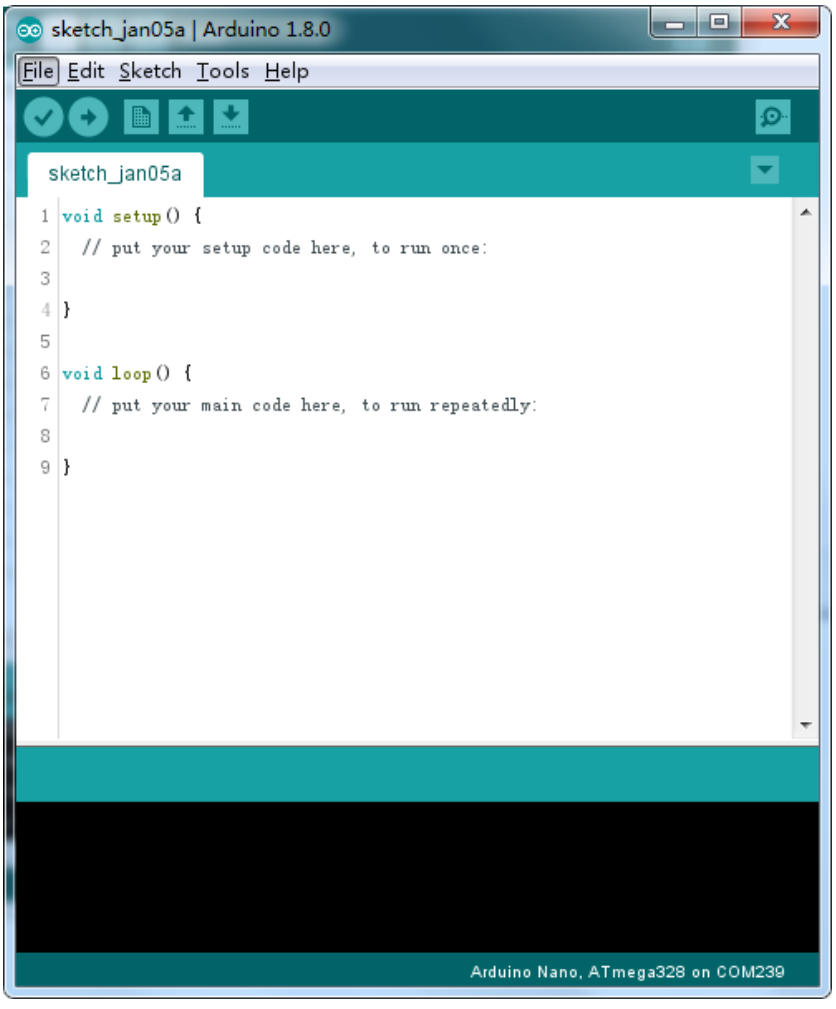
Ceci est la fenêtre de départ. Elle contient l'ossature de code commune et obligatoire.

Pour installer le logiciel depuis un fichier zip, veuillez suivre les instructions suivantes.

Téléchargez le fichier zip. Faites l'extraction des fichiers dans un dossier.

Lancer le fichier *arduino.exe*



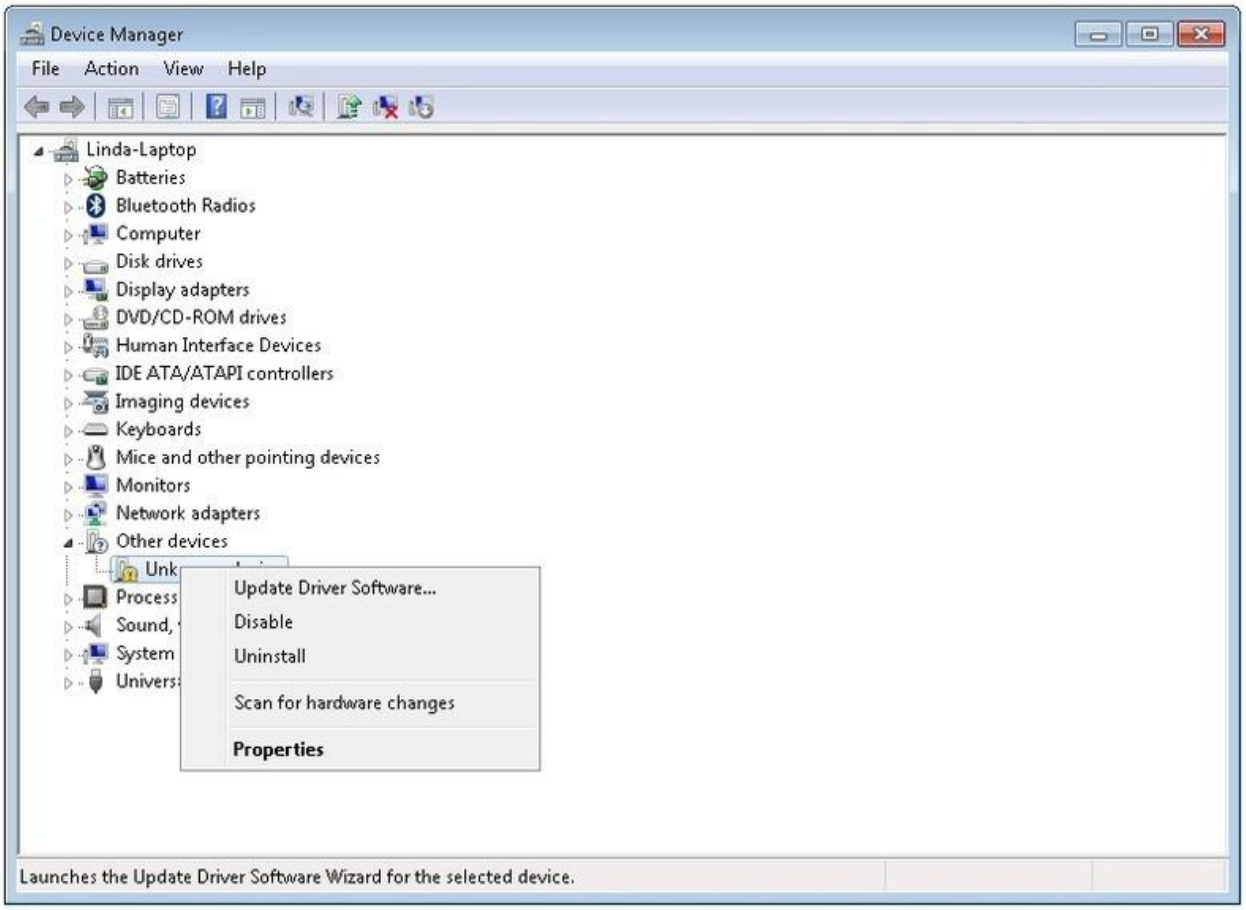


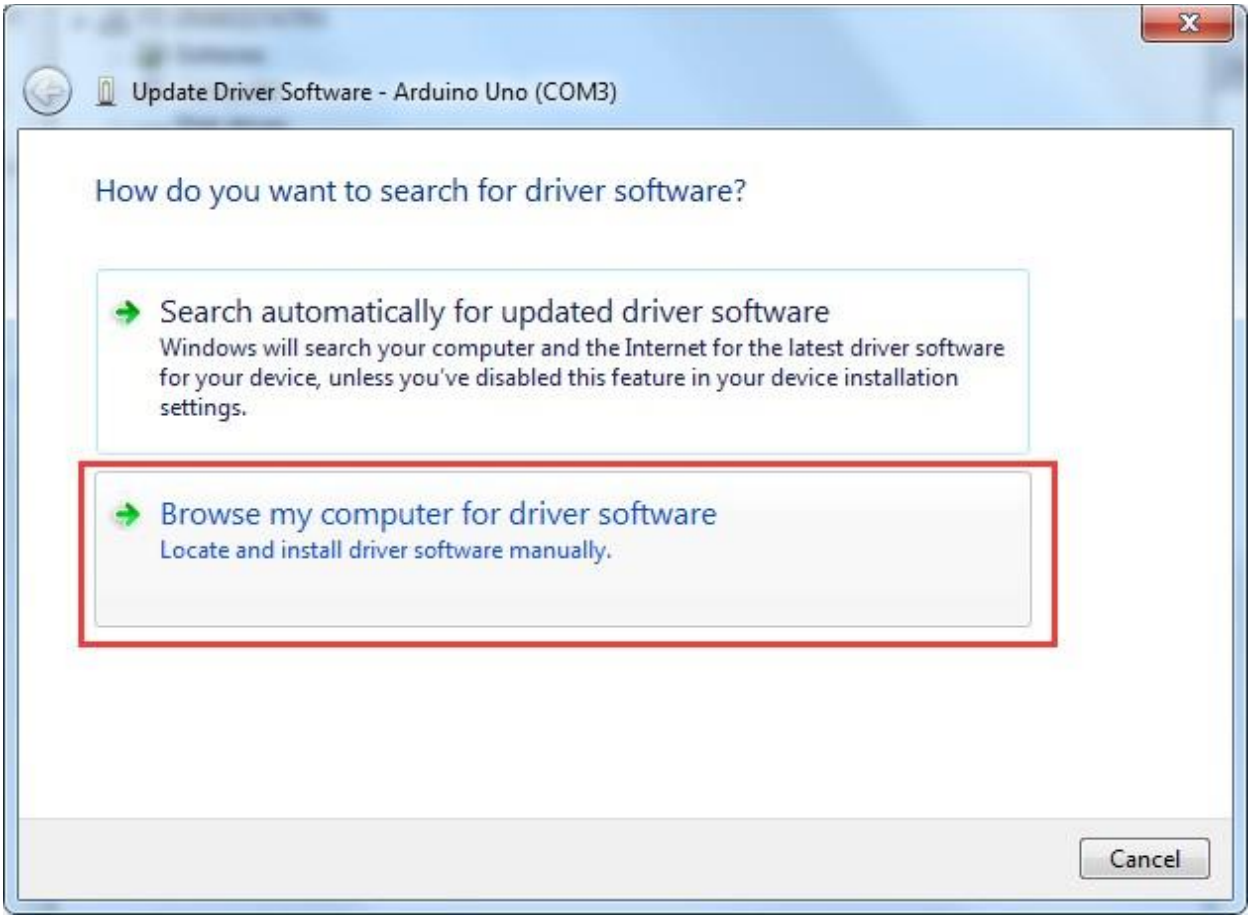
Si vous procédez de cette dernière manière, il faut installer les drivers de la carte.

Le répertoire dézippé contient à la fois les fichiers nécessaires au bon fonctionnement de l'IDE, mais aussi ceux nécessaires à l'installation des drivers USB de la carte UNO.

Branchez une carte UNO à un port USB de votre ordinateur. Vous allez certainement voir un message apparaitre mentionnant que Windows a découvert un nouveau matériel. **Ignorez ce message et fermez les tentatives de Windows de faire l'installation.**

La méthode la plus fiable est d'aller dans le gestionnaire de périphérique et de faire l'installation manuellement.





Faites un clic droit et sélectionnez le menu "*Mettre à jour le driver*". Sélectionnez ensuite *parcourir* et allez chercher le répertoire dans lequel vous avez dézippé le fichier Arduino.

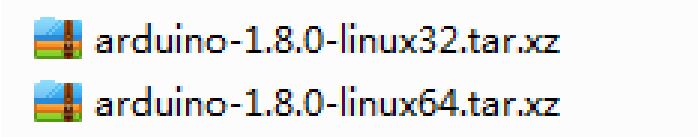
Cliquez '*Suivant*'. Après un message de sécurité, vous obtenez l'écran de confirmation de la bonne installation.

Installation sous Mac OS X

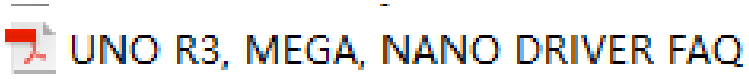
Téléchargez et dézippez le fichier suivant:

Installation Linux

Sélectionnez le fichier correspondant à votre environnement de travail



Conseil : vous trouverez des conseils et astuces, des réponses aux questions fréquentes :



Leçon 1 Ajouter une bibliothèque / utiliser le moniteur série

Installer des bibliothèques complémentaires

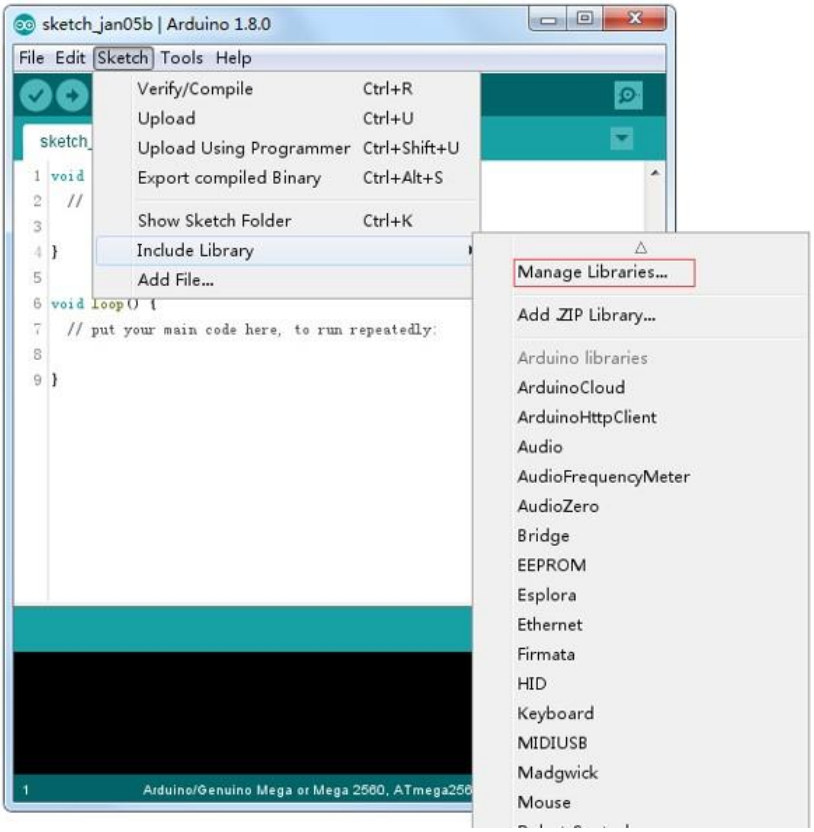
Lorsque vous aurez bien saisi les fonctions intégrées de l'environnement Arduino, vous aurez certainement le besoin ou l'envie d'aller encore plus loin, avec pourquoi pas des bibliothèques complémentaires.

Qu'est-ce qu'une bibliothèque?

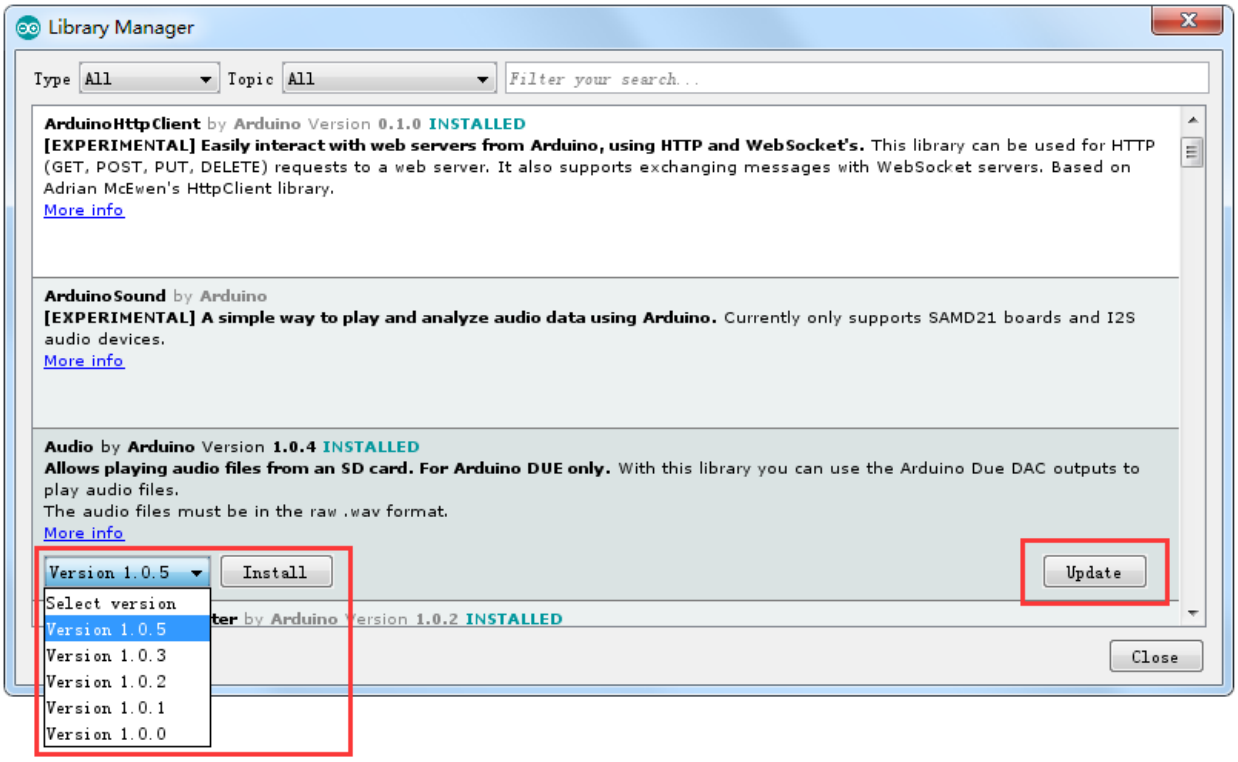
Une bibliothèque est une suite d'instructions qui rendent beaucoup plus facile l'utilisation de composants complexes. Cela peut être pour l'utilisation d'un écran à cristaux liquide, pour l'utilisation d'un servomoteur etc…

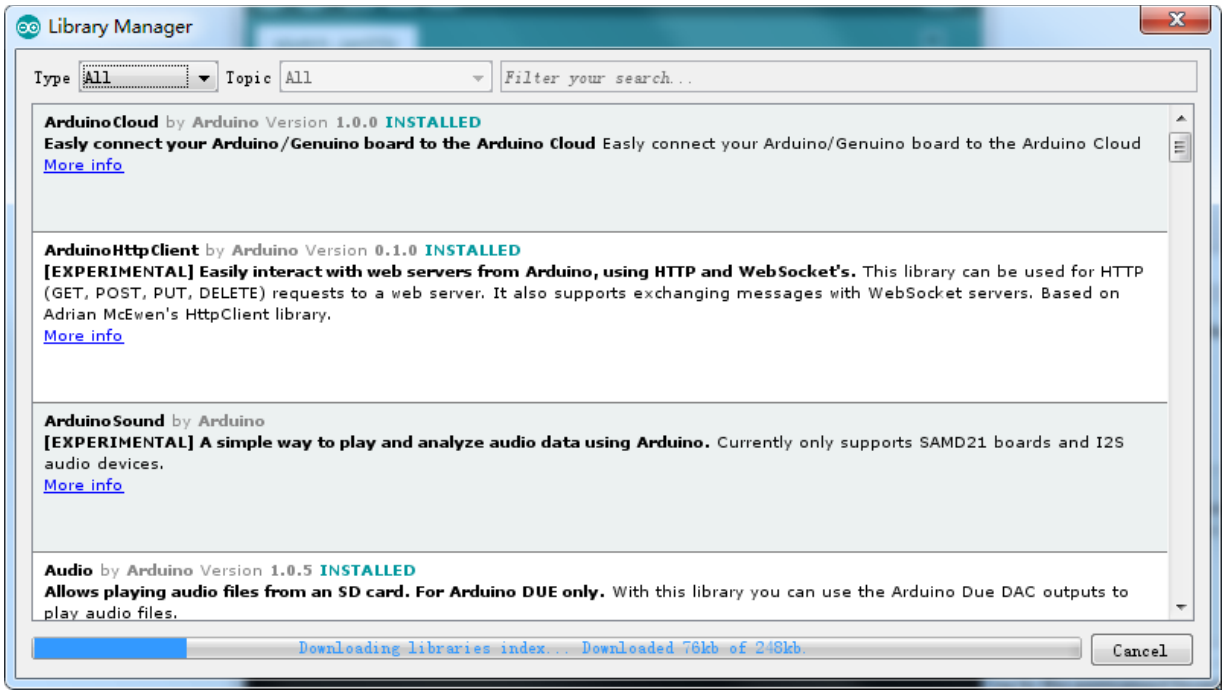
Comment installer

L'IDE contient un centre de management des bibliothèques qui permet de vérifier la bonne installation de telle ou telle bibliothèque ou d'en ajouter de nouvelles
Pour installer une nouvelle bibliothèque, cliquez sur le menu suivant :

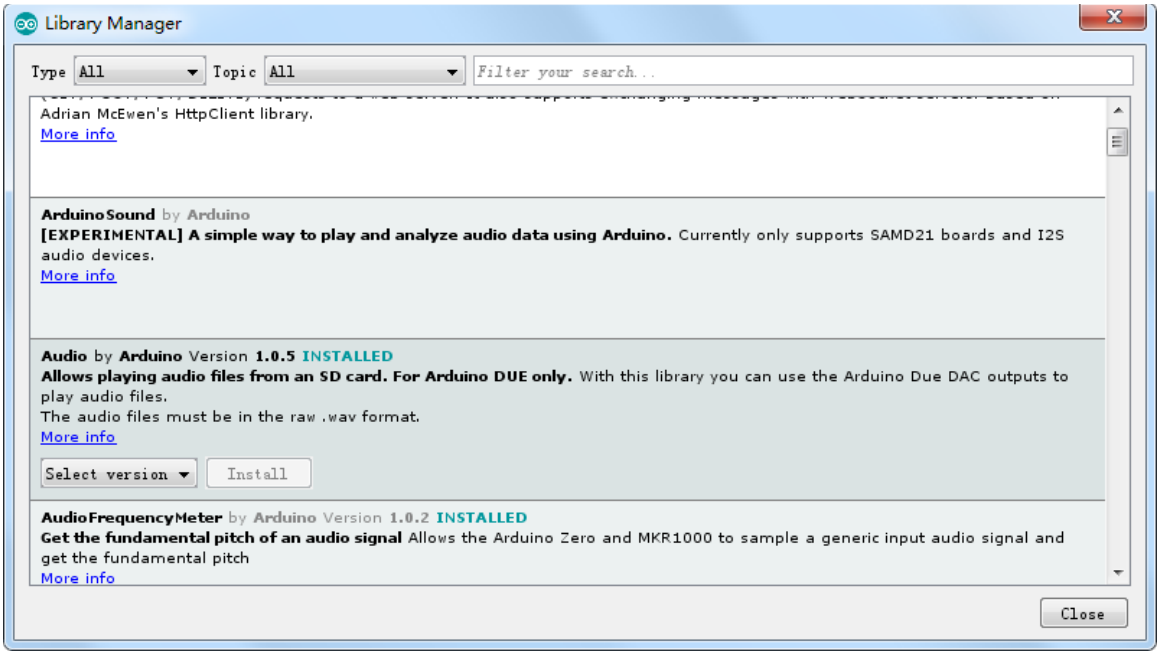


Vous allez pouvoir voir l'ensemble des bibliothèques déjà présentes ainsi que la version utilisée :





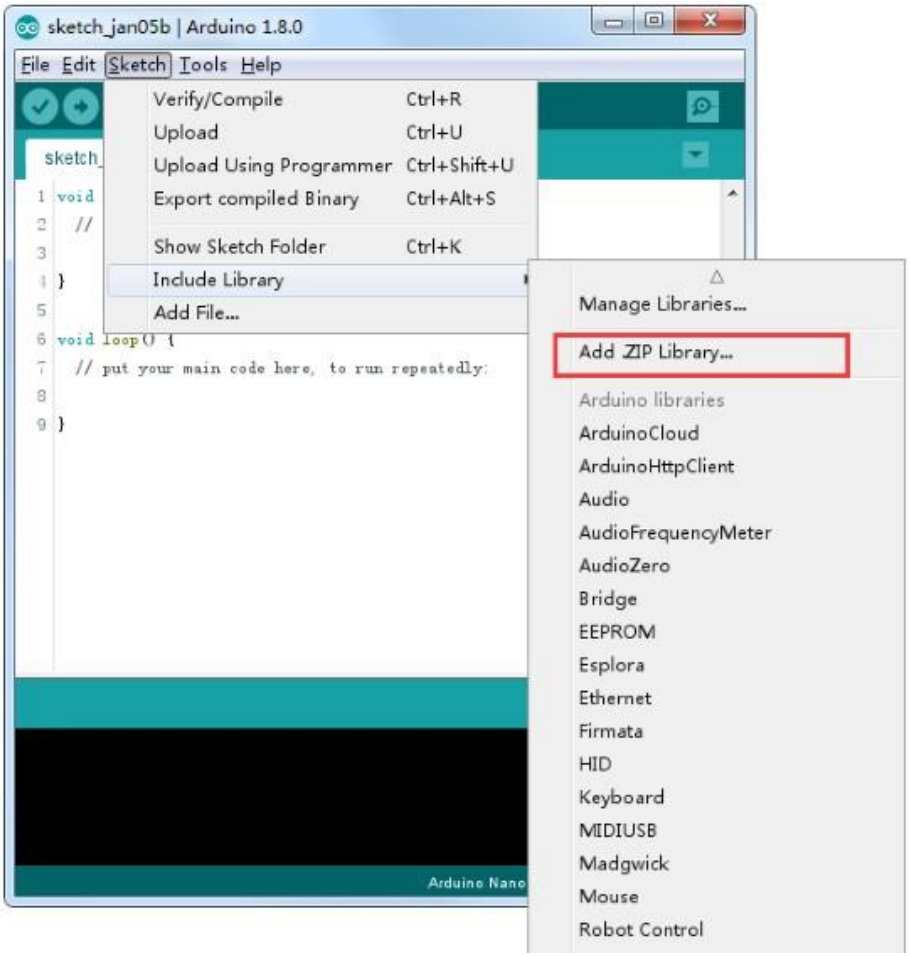
Si la dernière version d'une bibliothèques n'est pas installée, le bouton "Installer" se dégrise, vous pouvez procéder à son installation.



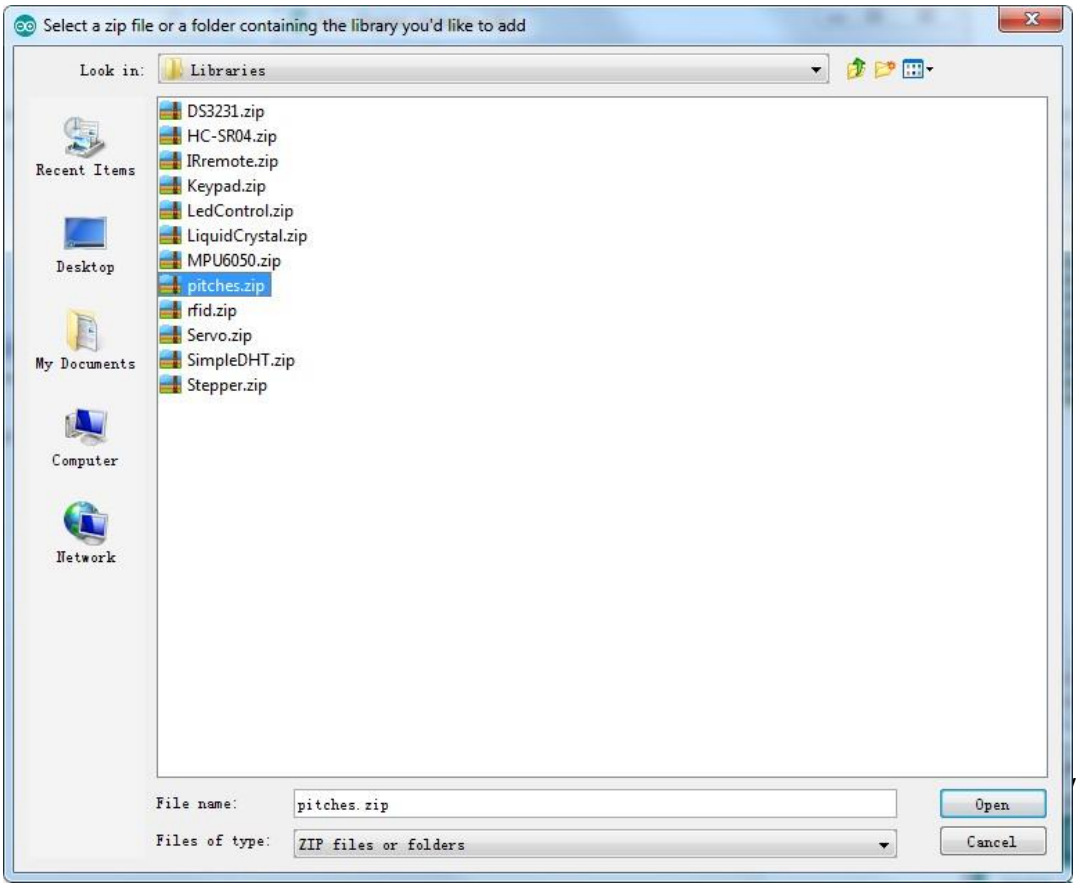
Si la bibliothèque recherchée n'apparait pas

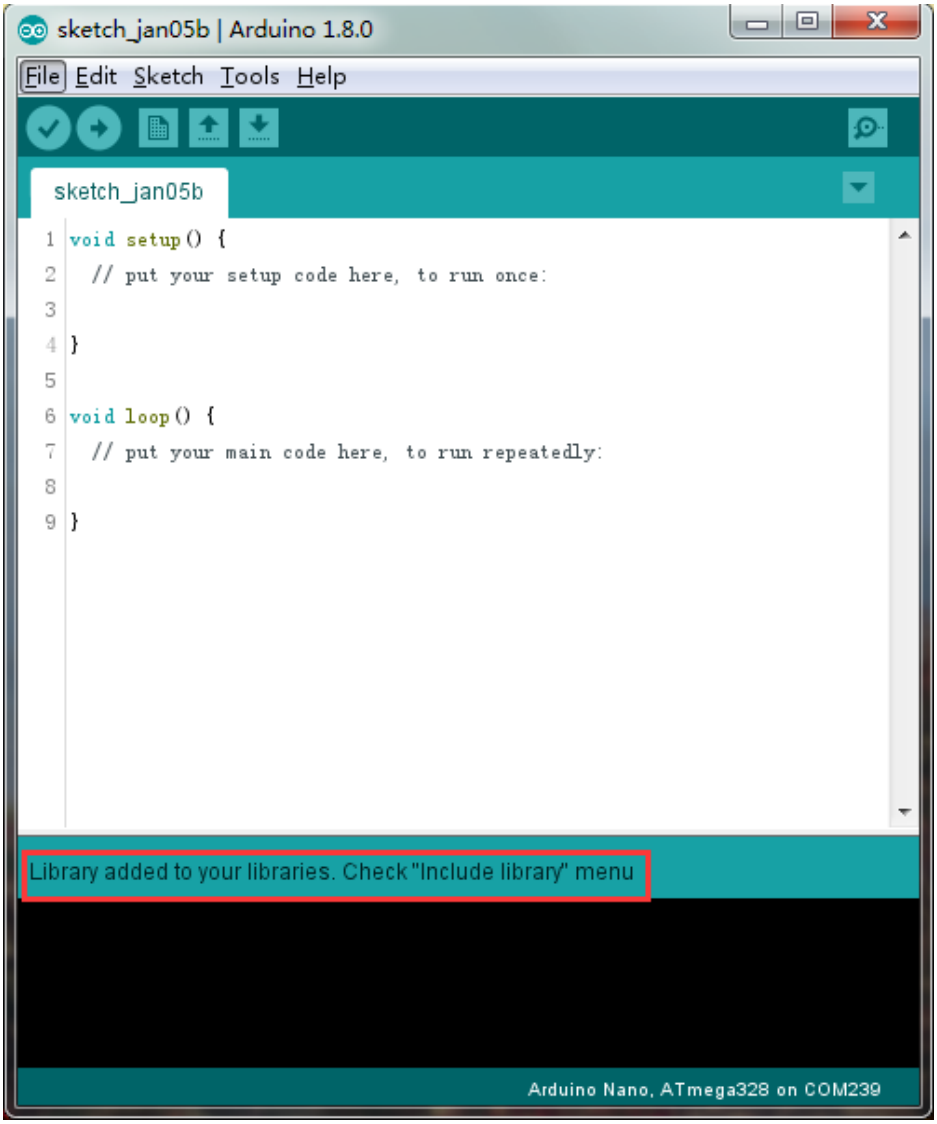
Procurez-vous sur internet (notamment sur GITHUB) le zip de la bibliothèque que vous souhaitez ajouter. Il n'est pas nécessaire de dézipper les fichiers, mais notez bien l'emplacement du fichier.

Dans le menu « Sketch », sélectionner « Inclure une bibliothèque ». Sélectionnez ensuite « Ajouter .ZIP Library ».



Il vous suffit ensuite de sélectionner le zip contenant le fichier





Revenez au menu Sketch> Import Library. Vous devriez maintenant voir la bibliothèque en bas du menu déroulant. Il est prêt à être utilisé dans votre croquis. Le fichier zip sera étendu dans le dossier des bibliothèques dans votre répertoire d'esquisser Arduino. NB: la bibliothèque sera disponible à utiliser dans des croquis, mais les exemples pour la bibliothèque ne seront pas exposés dans le fichier> Exemples jusqu'à la redémarrage de l'IDE.

Ces deux sont les approches les plus courantes. Les systèmes MAC et Linux peuvent être gérés de la même manière. L'installation manuelle à introduire ci-dessous comme alternative peut être rarement utilisée et les utilisateurs sans besoins peuvent l'ignorer.

Installation manuelle

Pour installer la bibliothèque, quittez d'abord l'application Arduino. Ensuite, décompressez le fichier ZIP contenant la bibliothèque. Par exemple, si vous installez une bibliothèque appelée

"ArduinoParty", décompressez ArduinoParty.zip. Il devrait contenir un dossier appeléArduinoParty, avec des fichiers comme ArduinoParty.cpp et ArduinoParty.h à l'intérieur. (Si les fichiers .cpp et .h ne sont pas dans un dossier, vous devrez en créer un. Dans ce cas, vous devez créer un dossier appelé "ArduinoParty" et y déposer tous les fichiers qui se trouvaient dans le ZIP Fichier, comme ArduinoParty.cpp et ArduinoParty.h.)
Faites glisser le dossier ArduinoParty dans ce dossier (votre dossier de bibliothèques). Sous Windows, il sera probablement appelé "Mes documents \ Arduino \ bibliothèques". Pour les utilisateurs de Mac, il sera probablement appelé «Documents / Arduino / bibliothèques». Sur Linux, ce sera le dossier "bibliothèques" dans votre carnet de croquis.

Votre dossier de bibliothèque Arduino devrait maintenant ressembler à ceci (sous Windows):

Mes documents \ Arduino \ bibliothèques \ ArduinoParty \
ArduinoParty.cpp Mes documents \ Arduino \ bibliotecas \ ArduinoParty \
ArduinoParty.h Mes documents \ Arduino \ bibliotecas \ ArduinoParty \
examples

Ou comme ceci (sur Mac et Linux):

Documents / Arduino / bibliothèques / ArduinoParty / ArduinoParty.cpp
Documents / Arduino / bibliothèques / ArduinoParty / ArduinoParty.h
Documents / Arduino / bibliothèques / ArduinoParty / examples.
...

Il peut y avoir plus de fichiers que les fichiers .cpp et .h, assurez-vous qu'ils sont tous là. (La bibliothèque ne fonctionnera pas si vous mettez les fichiers .cpp et .h directement dans le dossier des bibliothèques ou si elles sont imbriquées dans un dossier supplémentaire. Par exemple: Documents \ Arduino \ bibliotecas \ ArduinoParty.cpp et Documents \ Arduino \ Les bibliothèques \ ArduinoParty \ ArduinoParty \ ArduinoParty.cpp ne fonctionneront pas.)

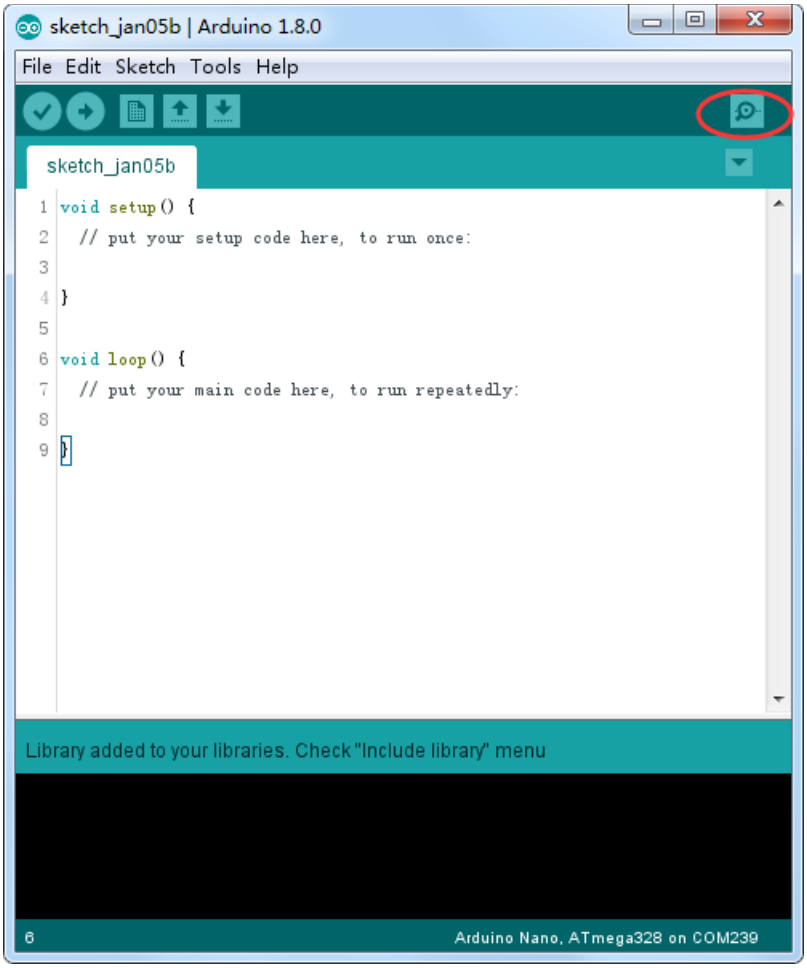
Redémarrez l'application Arduino. Assurez-vous que la nouvelle bibliothèque apparaît dans l'élément de menu Sketch-> Import Library du logiciel. C'est tout! Vous avez installé une bibliothèque!

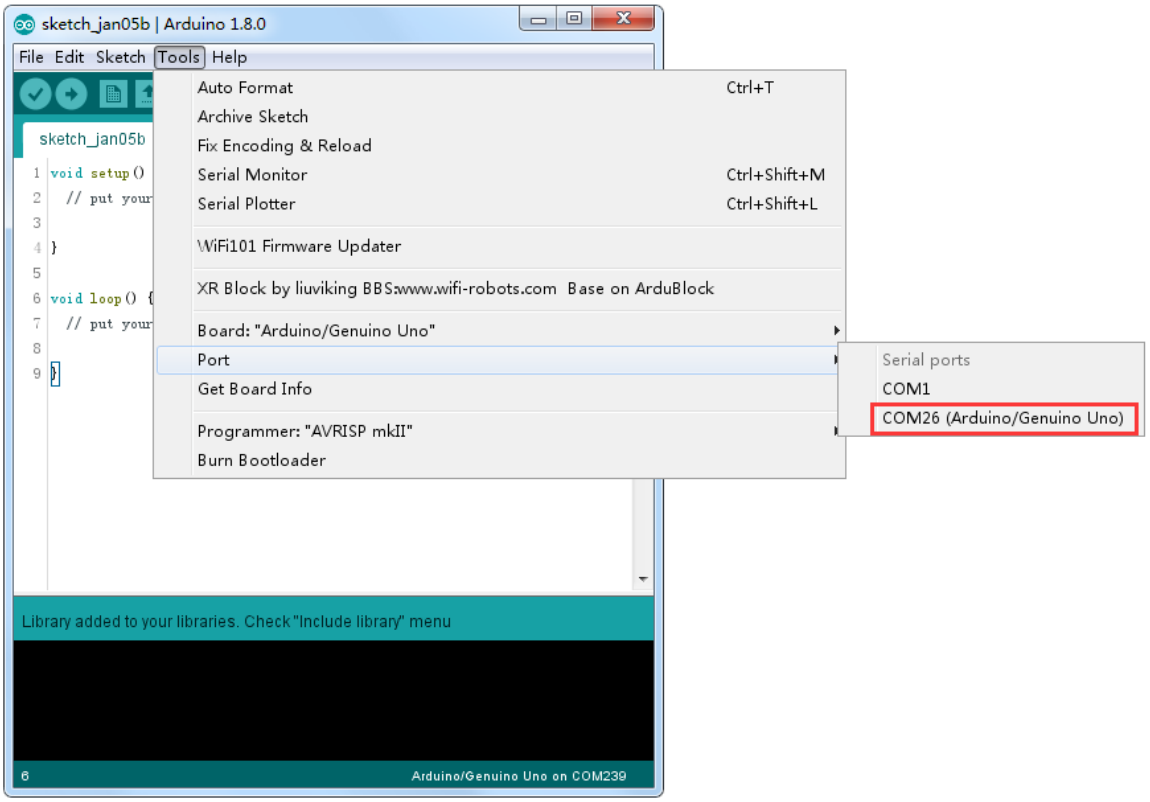
Utilisation du moniteur série (Windows, Mac, Linux)

Le moniteur série est une petite fenêtre très utile qui va vous permettre d'interagir en temps réel avec la carte UNO. Vous pouvez lui envoyer des informations et elle pourra à son tour faire de même. Vous verrez dans les différentes leçons que le recours au moniteur série est très fréquent.

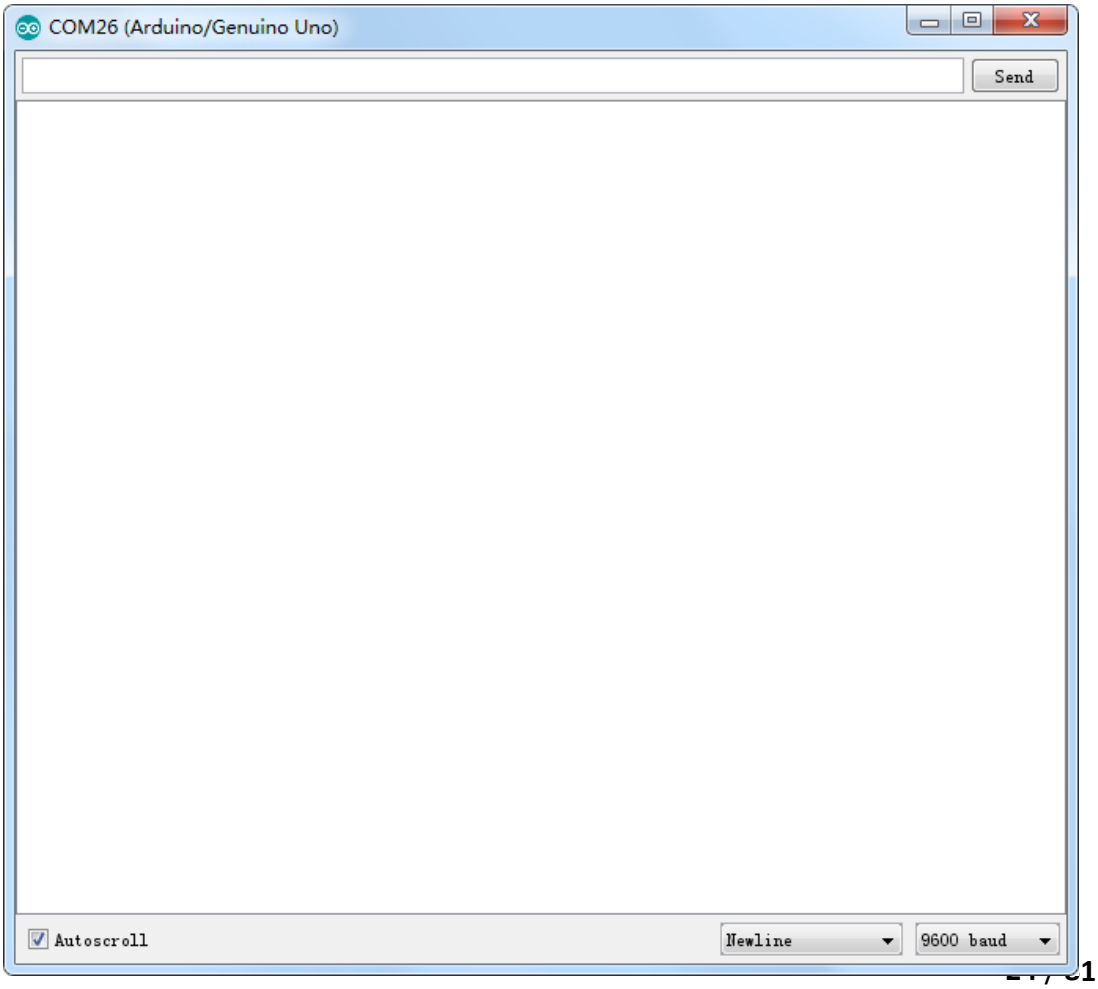
Se Connecter

Pour l'ouvrir, c'est très simple, il suffit de cliquer sur le petit icône entouré en rouge.



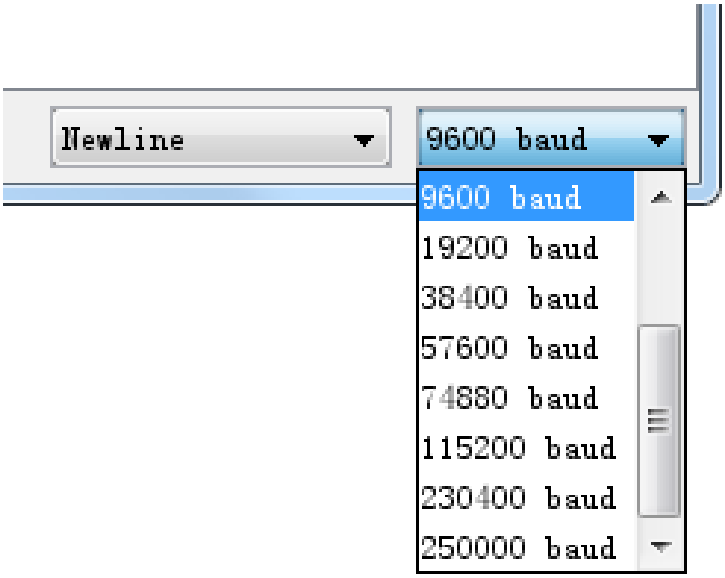


A l'ouverture, vous devez voir cette fenêtre:

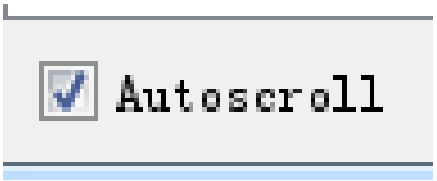


Settings

Vous pouvez définir le taux de transfert de la connexion de la manière suivante



Et vous pouvez avoir un défilement automatique de l'écran comme suit



Avantages

Le Serial Monitor est un excellent moyen d'établir rapidement une connexion série avec votre Arduino. Si vous travaillez déjà dans l'IDE Arduino, il n'est vraiment pas nécessaire d'ouvrir un terminal distinct pour afficher les données.

Les inconvénients

Le manque de paramètres laisse beaucoup à désirer dans le Serial Monitor, et, pour les communications en série avancées, il se peut que ce ne soit pas le tour.

Leçon 2 Blink

But de la leçon

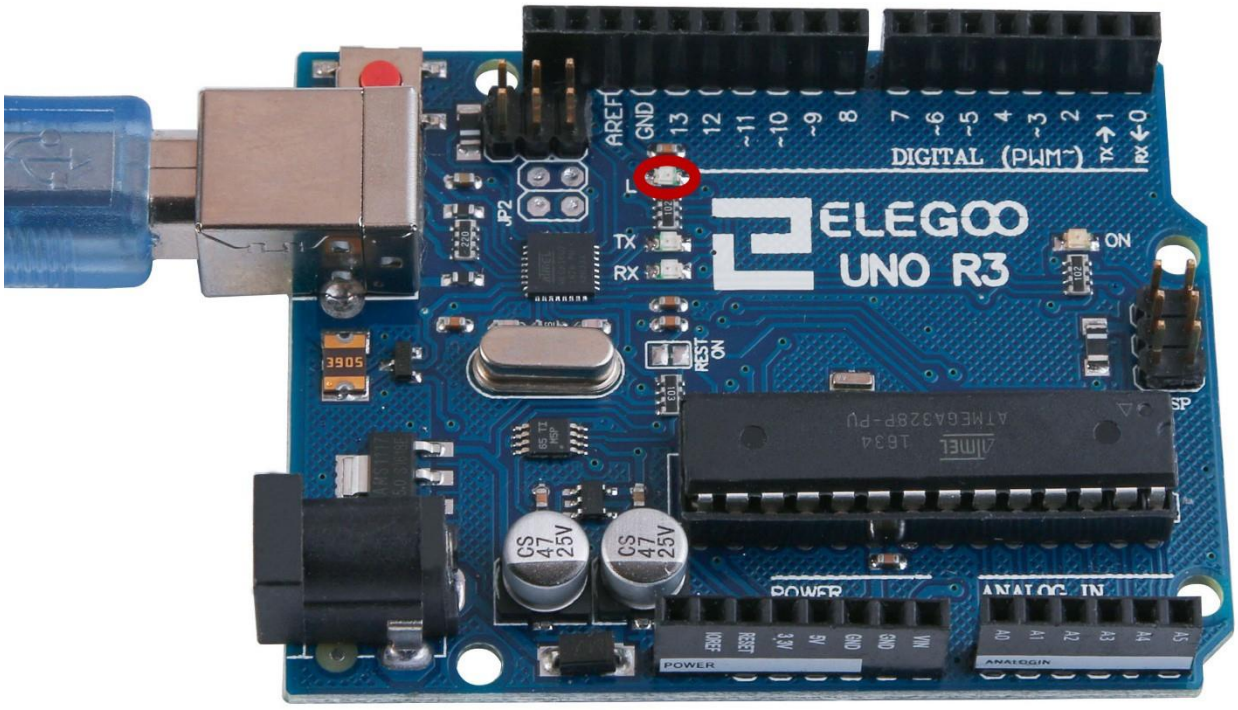
Dans cette première leçon, vous allez prendre en main l'IDE et apprendre à faire clignoter la LED intégrée à la carte UNO R3.

Matériel nécessaire:

(1) x Elegoo Uno R3

Principe

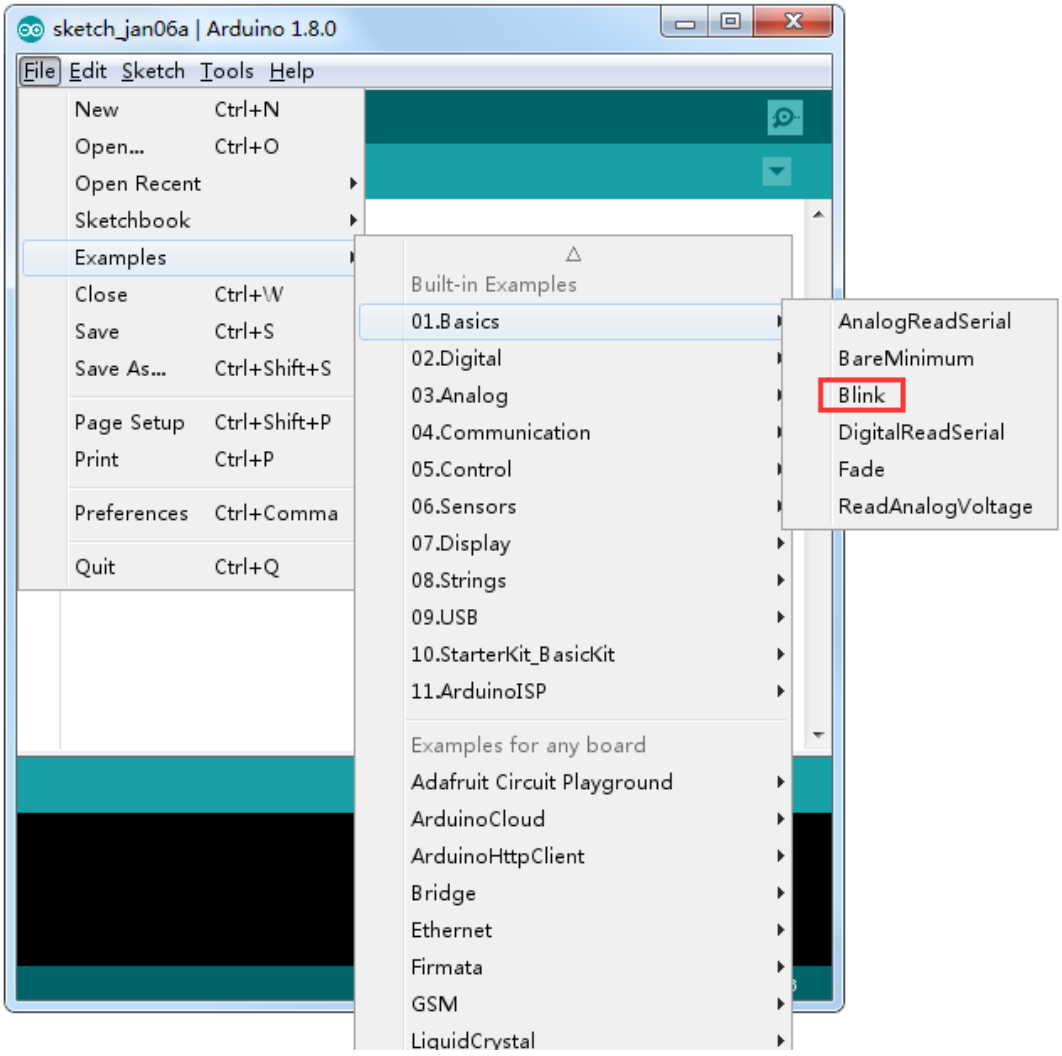
La carte UNO R3 possède deux rangées de connecteurs le long de ses extrémités qui sont utilisés pour brancher une vaste gamme de composants électroniques ou des cartes d'extensions (appelées shields) qui augmentent ses capacités.
Elle est aussi équipée d'une LED intégrée qu'il est possible de commander au travers de vos programmes. Vous pouvez apercevoir cette LED sur l'image ci-dessous, elle est repérable grâce au « L » visible sur la carte.



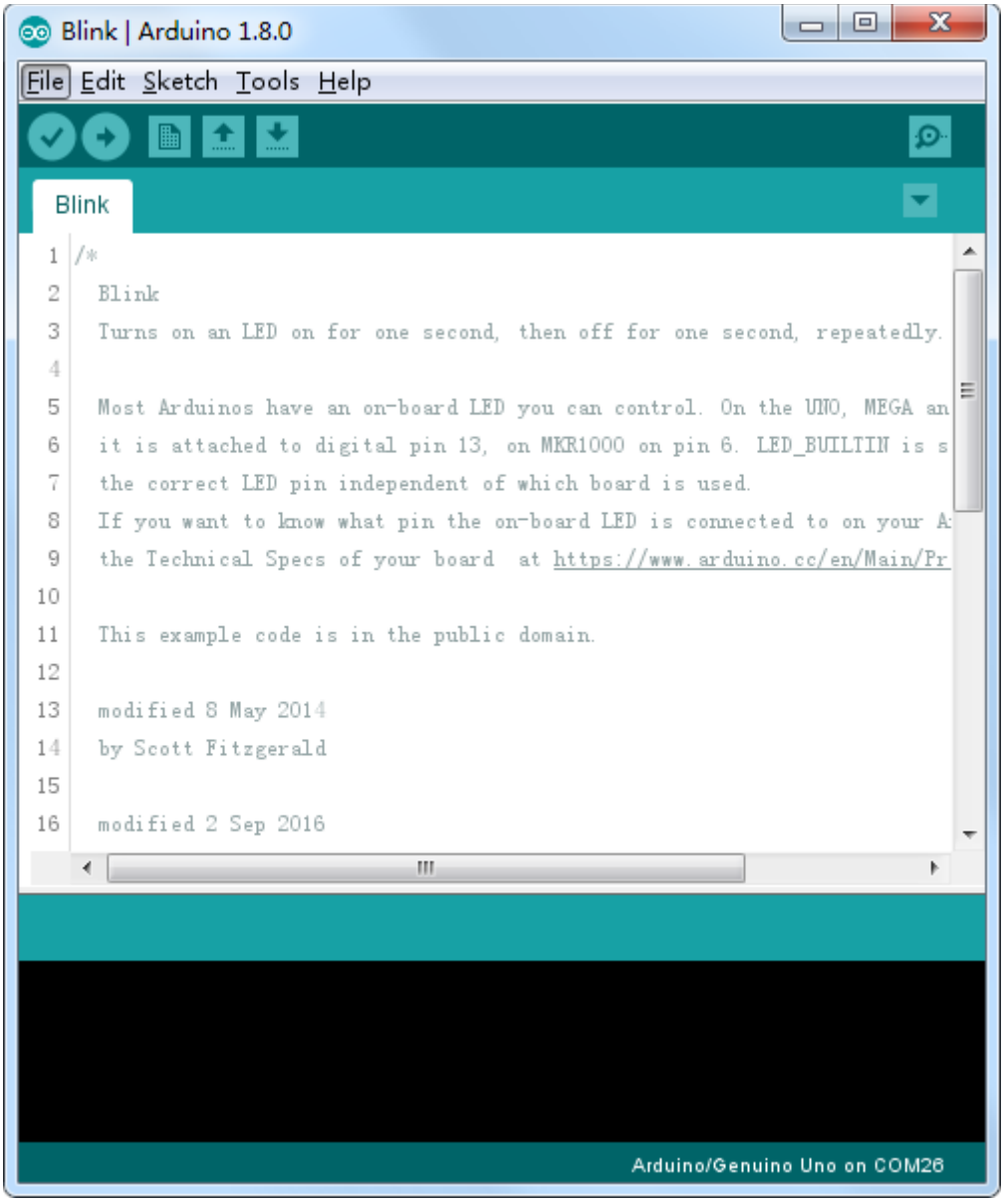
Lorsque vous connectez votre carte pour la première fois, il est possible que la LED se mette à clignoter. C'est parce que les cartes sont fréquemment expédiées avec le programme « BLINK » préinstallé.

Dans cette leçon, nous allons reprogrammer la carte UNO R3 avec notre propre programme « BLINK », ce qui va nous permettre notamment de changer la fréquence de clignotement de la LED.

Ouvrez le sketch "BLINK" dans l'environnement de programmation via le menu "FICHIER/EXEMPLES/01.BASICS/Blink"

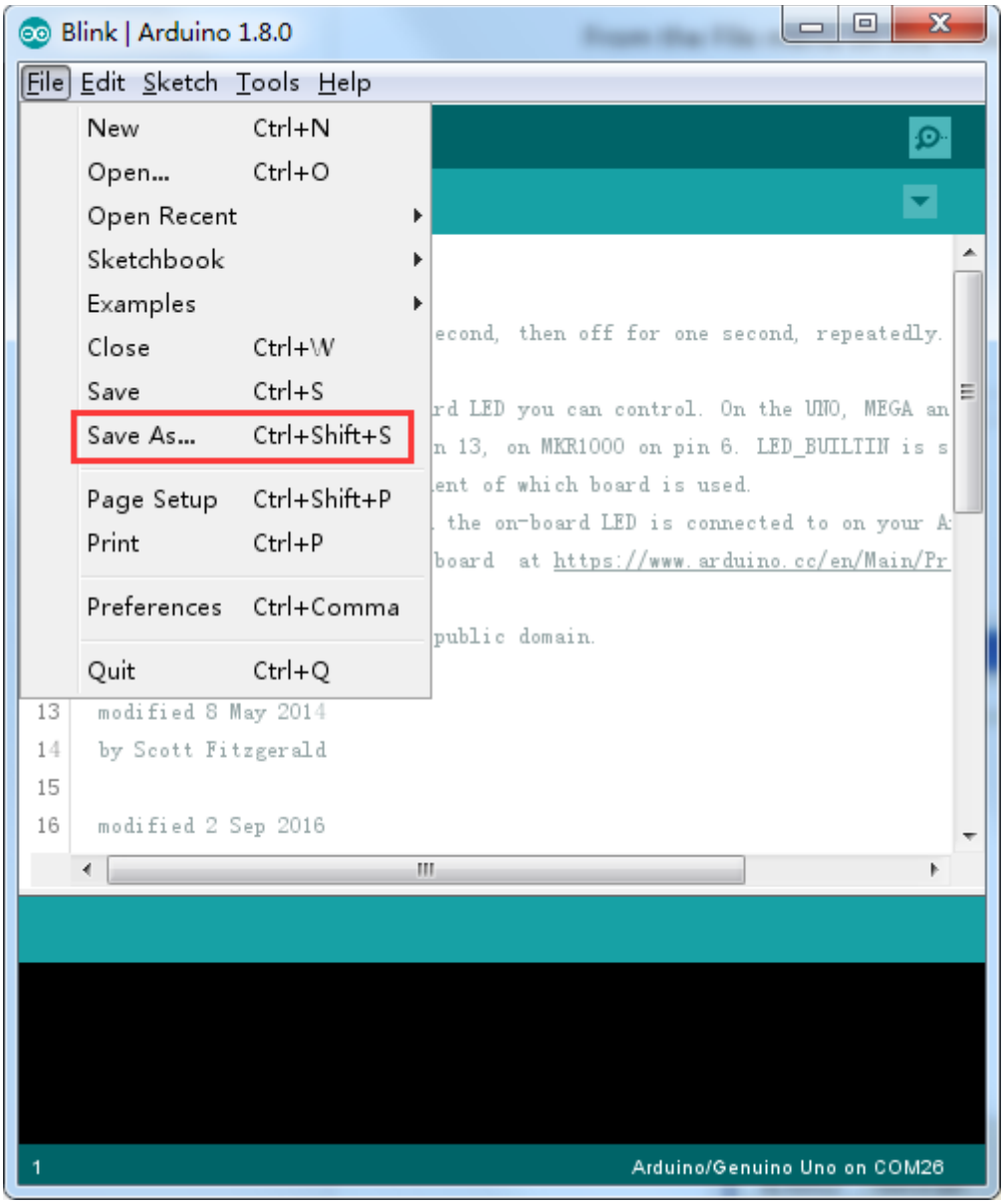


Voici ce que vous devez obtenir.

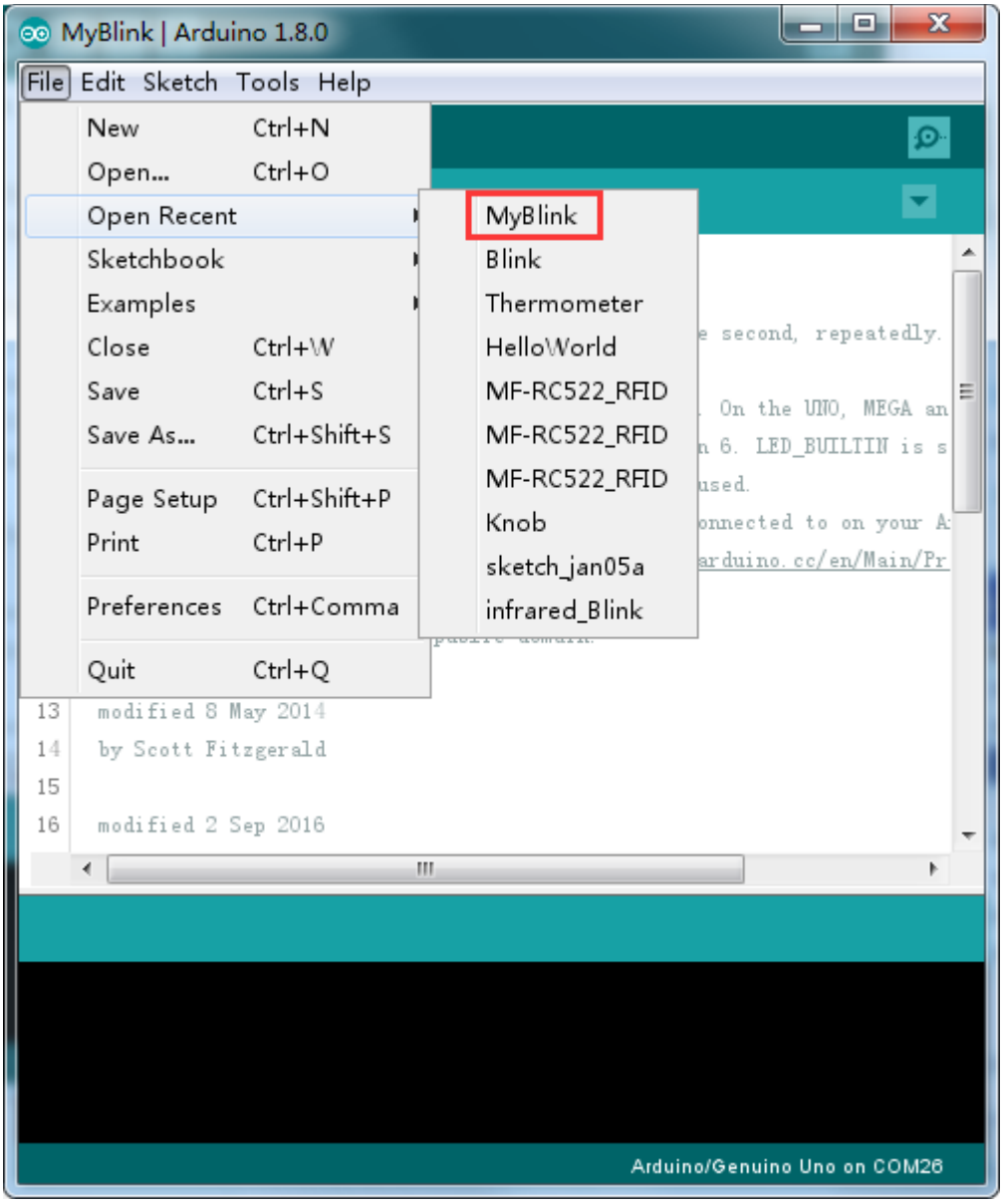


Comme vous allez apporter des modifications à ce sketch, la première étape consiste à l'enregistrer dans votre répertoire personnel, les sketchs exemples étant en lecture seule.

Utilisez le menu « enregistrer sous… » et définissez le répertoire et nom de fichier que vous souhaitez.

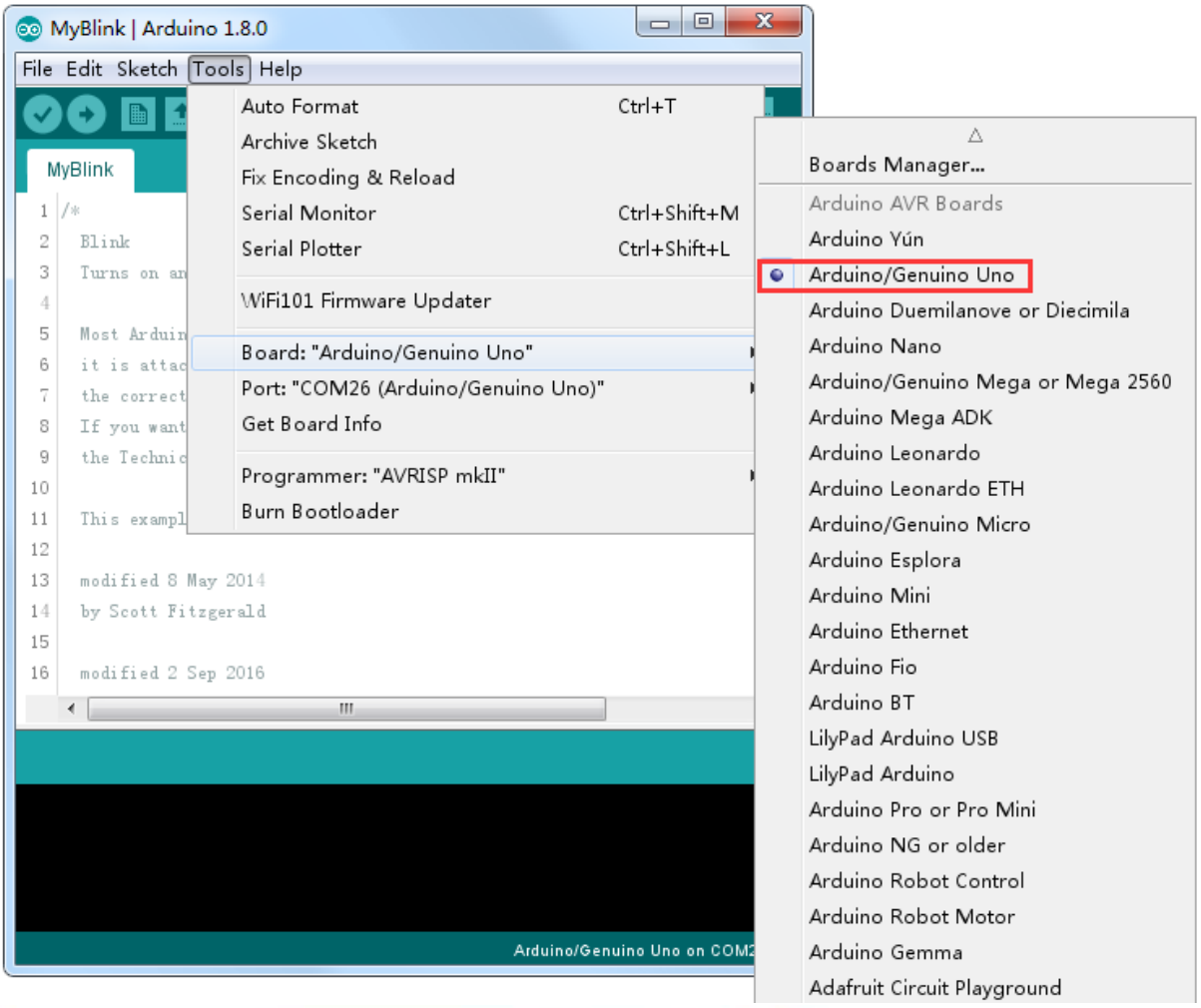


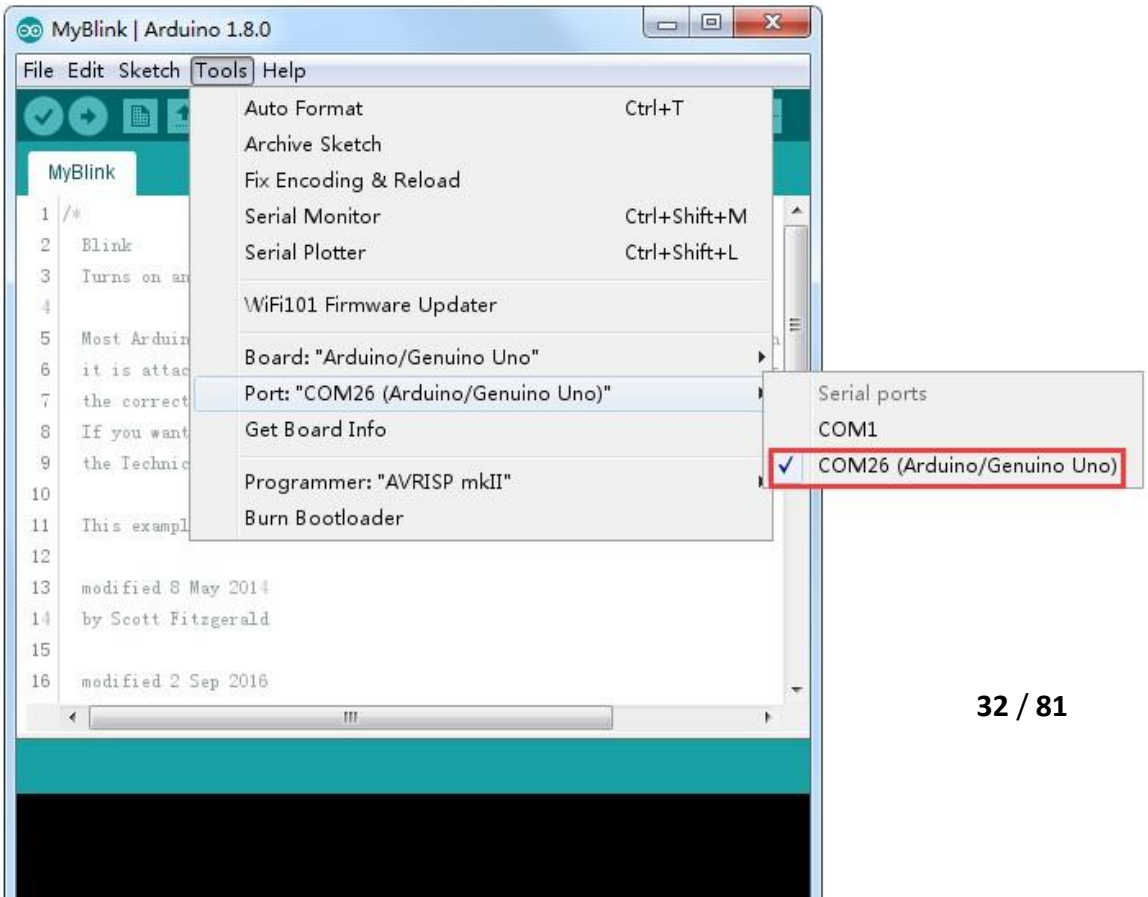
Vous venez d'enregistrer le fichier dans votre répertoire.



Vous pouvez accéder à vos fichiers récents de la même manière que la plupart de vos autres logiciels.

Il est temps maintenant de connecter la carte UNO R3 via le port USB de votre ordinateur et de vérifier la bonne reconnaissance de celle-ci.

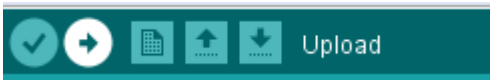




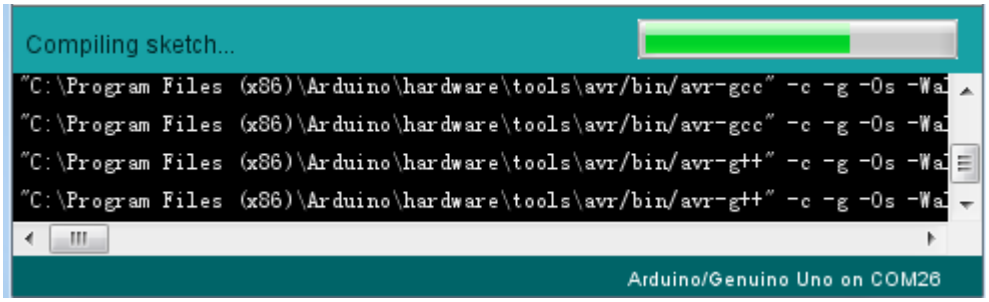
Note: le numéro de port COM ne sera pas nécessairement celui que vous pouvez apercevoir sur les captures d'écran. En effet, c'est votre ordinateur qui va affecter une référence au moment de la connexion et si vous possédez plusieurs cartes, chacune aura potentiellement un numéro différent.

Une fois connecté, vous pouvez apercevoir ce petit texte en bas à droite de votre écran.

Cliquez sur le bouton avec la flèche pour déclencher le téléversement.



Vous pouvez constater que la première étape est la compilation (le logiciel vérifie l'intégrité du code).



Ensuite, le téléversement à proprement parler commence.

Enfin, le statut "terminé" s'affiche.

Un message vous renseigne sur la taille du programme et la quantité de mémoire utilisée sur la carte. Si votre carte n'est pas correctement connectée ou reconnue, vous obtiendrez l'écran suivant. Reportez-vous aux explications données dans les premières leçons pour veillez à avoir correctement installé les drivers.

Vous pouvez constater qu'une part importante du code est consacrée aux commentaires. Les commentaires sont importants ils permettent de bien comprendre le code, surtout quand celui-ci est complexe. Cela permet d'y revenir plus tard lorsqu'il est nécessaire de le faire évoluer.

Il existe deux façons de mettre du texte en commentaire. La première est de commencer la ligne par « // ». La deuxième manière permet de placer une portion de code/texte en commentaire (idéal lorsque l'on veut tester plusieurs alternatives par exemple). Débuter la zone commentaire par « /* » et terminez celle-ci par « */ »

Dans le code « blink », la vraie première ligne d'instruction est la suivante:

```
int led = 13;
```

Cette instruction permet de créer une constant nommée "led" et de lui affecter la valeur numérique "13".

Le bloc suivant représente le code qui sera exécuté une unique fois lors de la mise sous tension de la carte.

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

Tous les sketchs doivent obligatoirement avoir une bloc “setup” dans lequel vous pouvez placez autant d’instructions que nécessaires. Les instructions s’écrivent entre {}.

Dans « BLINK » la seule instruction consiste à affecter le pin 13 en tant que sortie (OUTPUT).

Il est aussi obligatoire dans un sketch d’avoir un bloc “loop”. Contrairement au premier bloc celui-ci s’exécutera en boucle tant que la carte sera sous tension.

```
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
}
```

Le bloc « loop » est constitué de 4 instructions. La première déclenche l’allumage de la led, la deuxième une attente de 1000ms, la troisième l’extinction de la led. La dernière instruction une attente de 1000ms.

Vous allez maintenant augmenter la fréquence de clignotement de la LED. Vous avez certainement deviné que le paramètre à changer est celui entre () sur les instructions « delay ». Téléversez le programme et observez.

```
30 // the loop function runs over and over again forever
31 void loop() {
32   digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the volt
33   delay(500)                         // wait for a second
34   digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the vo
35   delay(500)                         // wait for a second
36 }
```

Leçon 3 LED

But de la leçon

Dans celle leçon, vous allez apprendre à faire varier la luminosité d'une LED en utilisant plusieurs valeurs de résistances.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x LED rouge de 5mm

(1) x résistance 220 ohm

(1) x résistance 1 kohm

(1) x résistance 10 kohm

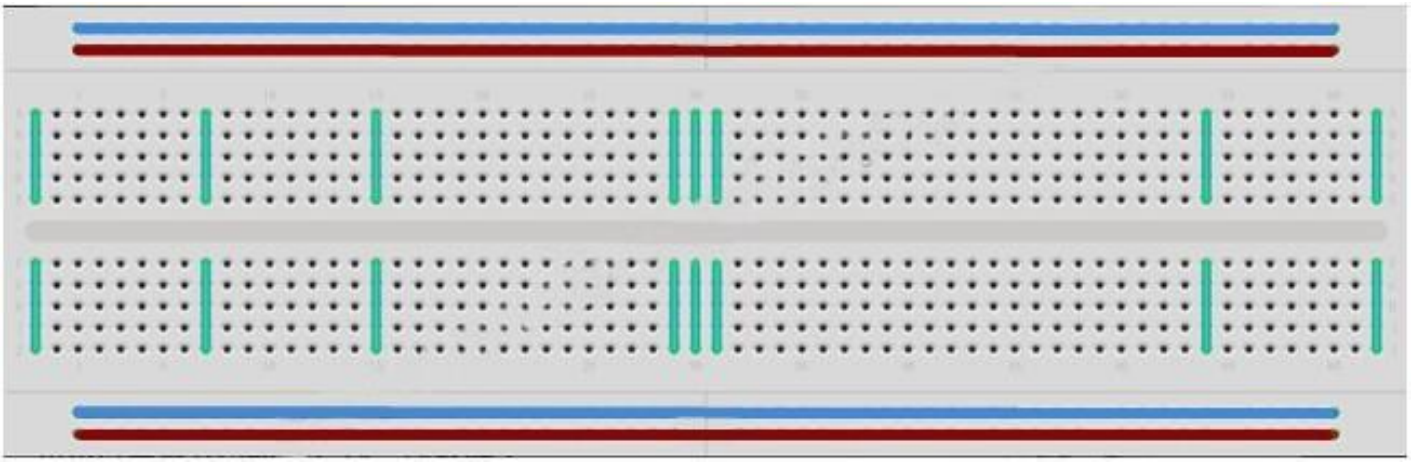
(2) x câbles mâle-mâle

Présentation des composants

PLANCHE PROTOTYPE MB-102：

Une planche prototype vous permet de réaliser des circuits très rapidement, sans avoir besoin de réaliser de soudures.

Exemple :

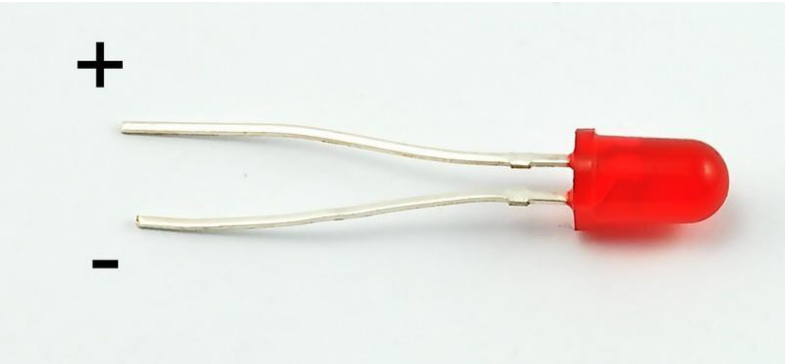


Il existe une grande variété de planches prototypes. La plus simple est une grille de trous dans un bloc de plastique. A l'intérieur se trouvent des lames métal permettant la connexion électrique entre les différents trous d'une même ligne. Mettre les branches de deux composants sur une même ligne les "assemble" électriquement. La ligne creusée au centre de la plaque symbolise une rupture électrique entre la partie haute et la partie basse. Cela signifie aussi que vous pouvez positionner une puce à cheval entre haut et bas. Certaines planches prototypes ont aussi deux lignes horizontales en haut et en bas. On les utilise généralement pour créer une ligne d'alimentation +5V et masse.
Attention tout de même, les planches prototypes ont comme limite d'utilisation la qualité des connexions qui ne valent pas une soudure et peuvent entraîner parfois des dysfonctionnements.

LED:

Les leds font de parfaits indicateurs lumineux. Elles consomment peut de courant et ont une très bonne durée de vie.
Dans cette leçon, vous allez utiliser certainement le type de leds le plus commun, la led 5mm. Il en existe d'autres de 3 à 10mm.

Attention: il n'est pas possible de brancher directement une led sur une batterie car l'intensité de courant est trop forte.



Attention aussi au sens de branchement de la led. La patte la plus longue doit être connectée à la borne + et la patte courte à la borne -.

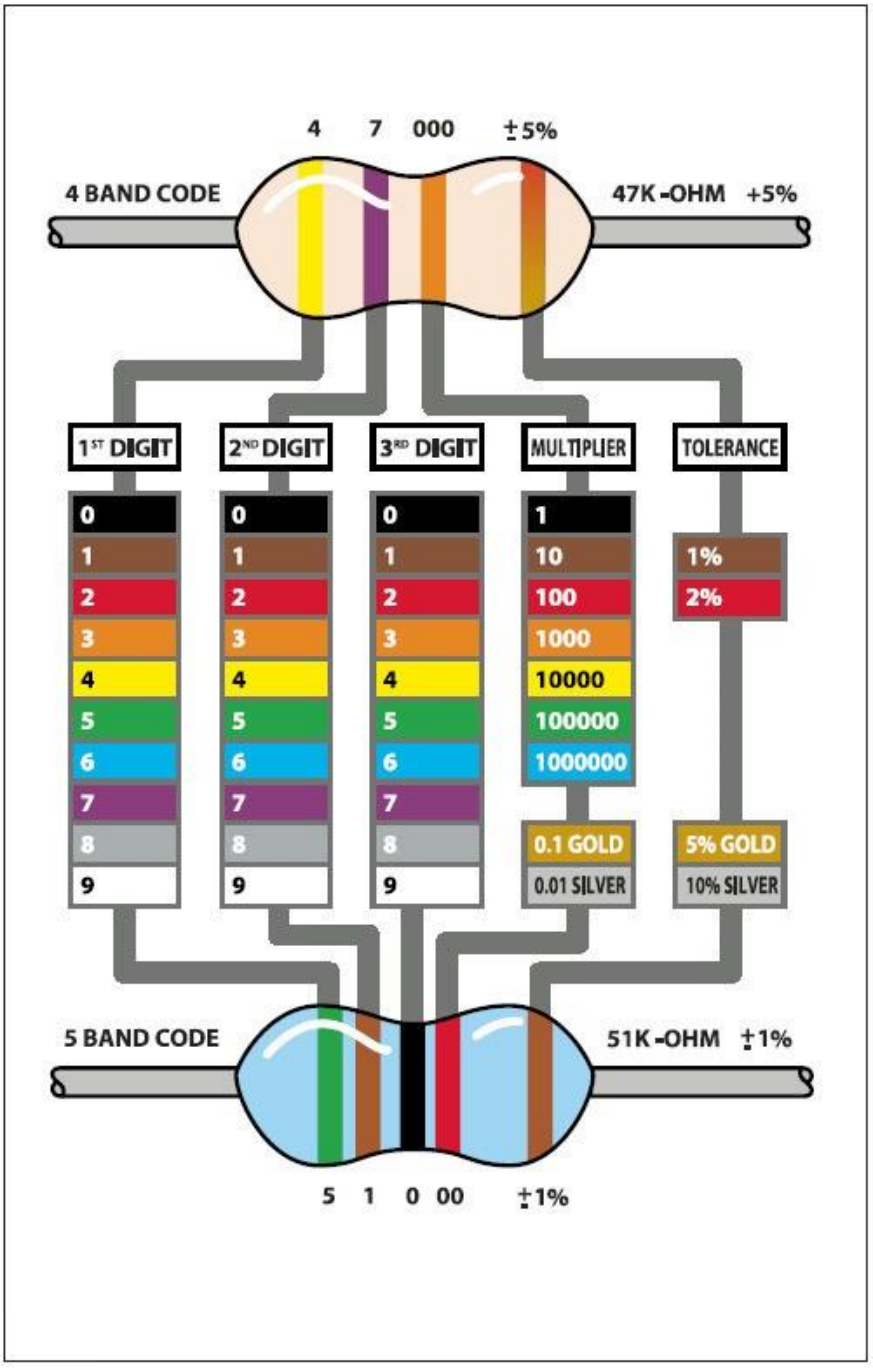
RESISTANCES:

Comme leur nom le suggère, les résistances s'opposent au passage du flux d'électricité. Plus la valeur est grande plus la résistance l'est aussi. Vous allez pouvoir calibrer la brillance de la led en jouant sur la valeur de la résistance.

Mais tout d'abord, quelques informations supplémentaires…

L'unité des résistances est l'Ohm, que l'on représente généralement avec la lettre grecque Ω. Il faut beaucoup d'ohms pour avoir un minimum de résistance, c'est pour cela que l'on retrouve généralement des kilos, voir mégas ohms : kΩ (1,000 Ω) et MΩ (1,000,000 Ω).

Dans cette leçon, vous allez utiliser trois valeurs de résistance : 220Ω, 1kΩ and 10kΩ. C'est grâce aux anneaux de couleurs et au code associé que l'on peut connaitre la valeur d'une résistance.



Notez aussi que contrairement aux leds, les résistances n'ont pas de sens.

Pour simplifier la tâche, il est aussi possible d'utiliser un appareil de mesure afin de connaître la valeur de la résistance.

Connexion

Schéma de câblage

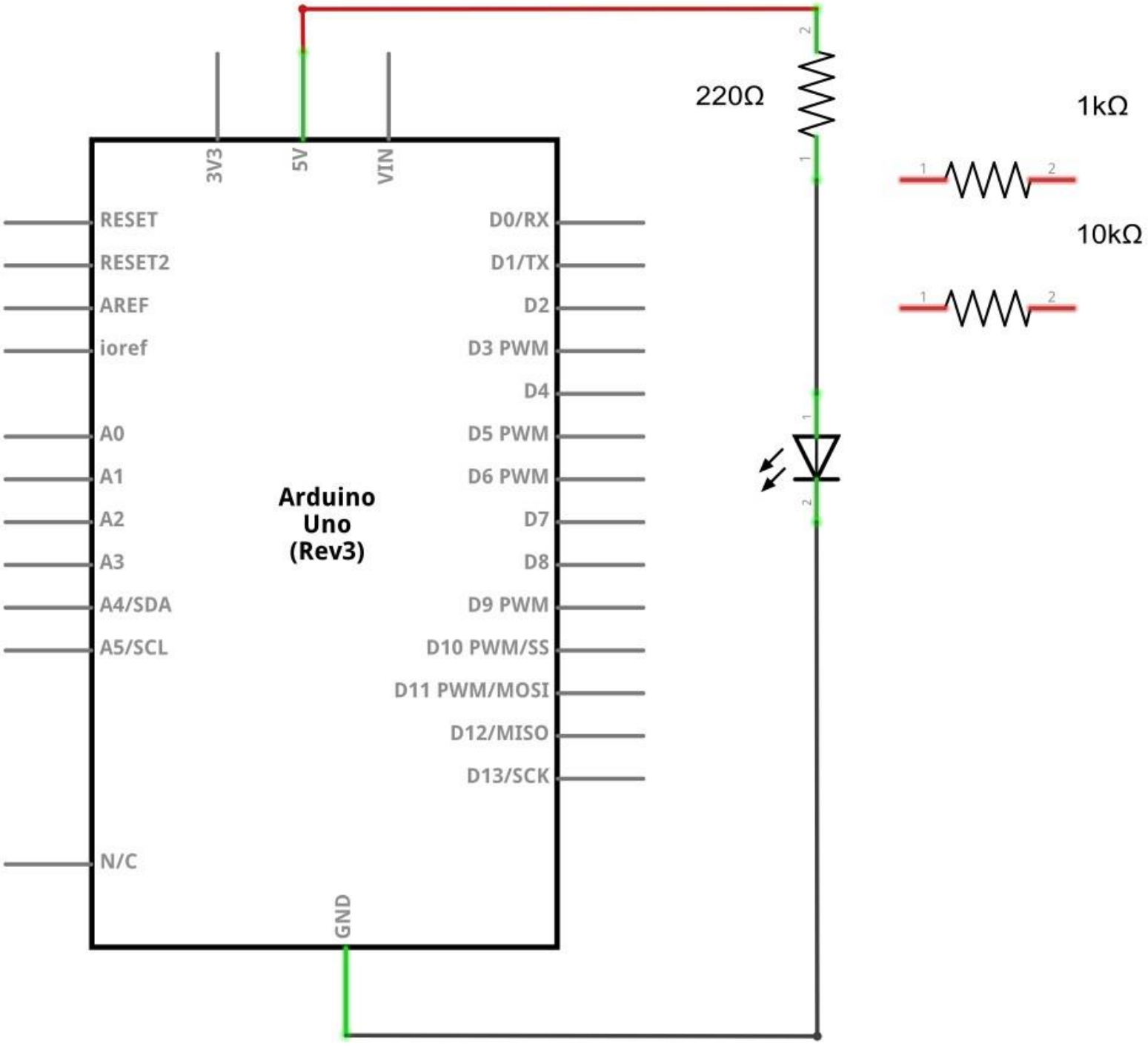
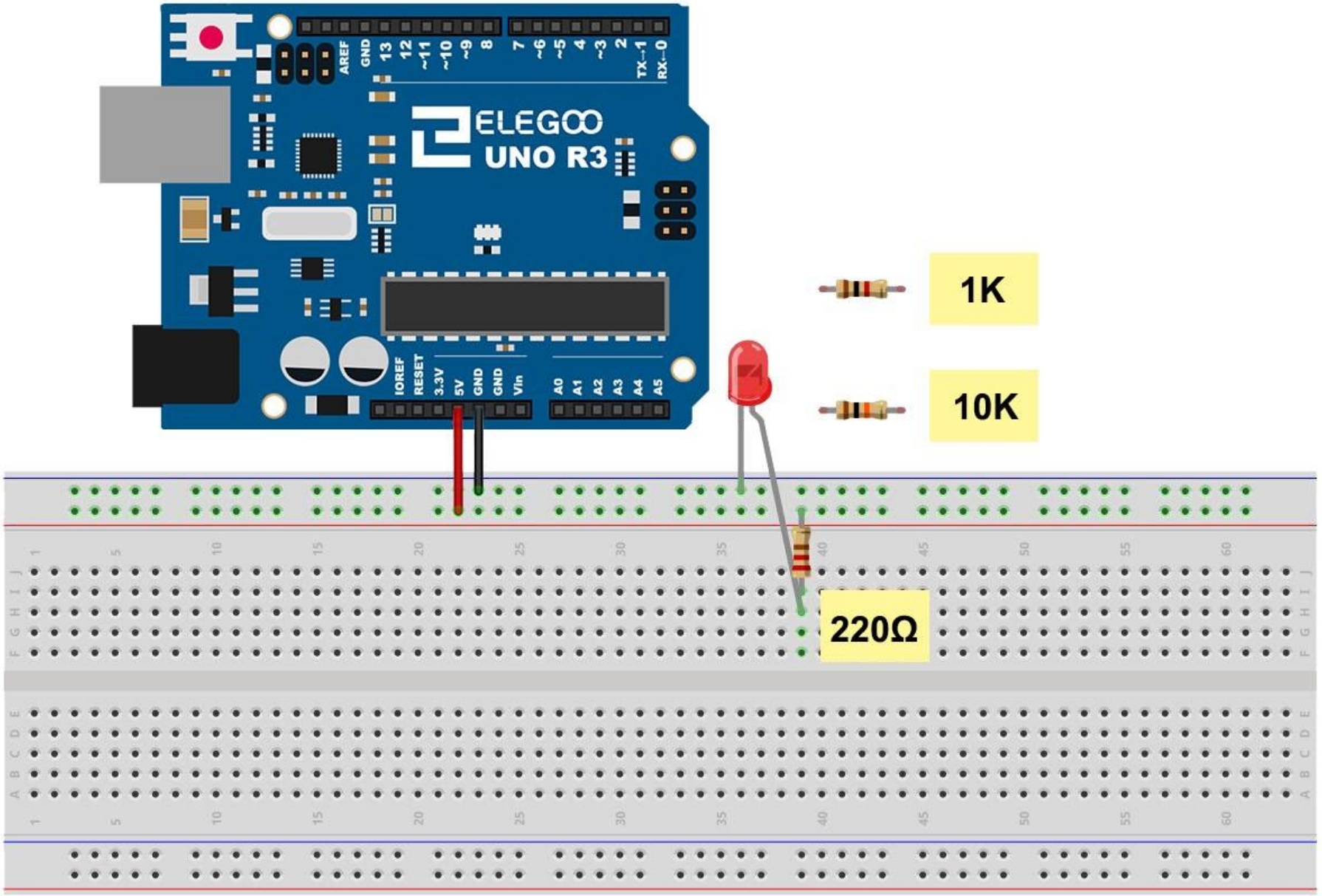


Diagramme de câblage

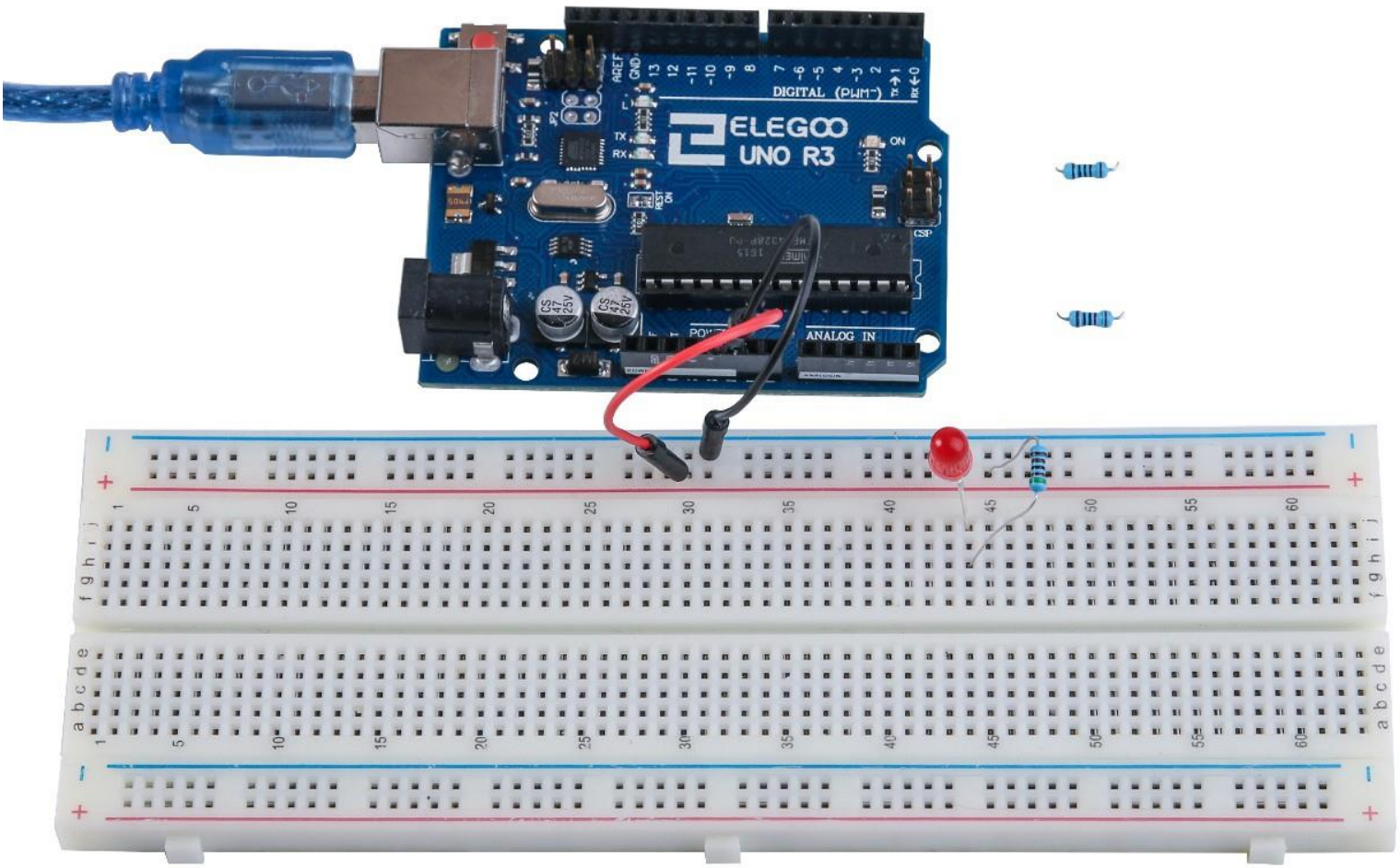


L’UNO est une bonne source de +5V. Vous allez donc pouvoir l’utiliser pour fournir le bon voltage à la LED et la résistance. Il n’est pas nécessaire de téléverser un code pour cet exemple, simplement connecter la carte à un ordinateur pour celle-ci soit sous tension.

Avec une résistance de 220 Ω, la LED est très brillante. Elle l’est un peu moins avec une résistance de 1kΩ et à peine brillante avec une résistance de 10 kΩ.

Vous pouvez positionner la résistance avant ou après la led et constater que cela n’a pas d’importance.

Illustration



Leçon 4 RGB LED

But de la leçon

Les LED RGB sont une façon simple et amusante d'ajouter de la couleur à vos projets.
Il en existe deux types, à anode commune et à cathode commune.
A anode commune utilisent le +5V en entrée commune, à cathode commune la masse en sortie commune.
Il ne faut pas oublier d'utiliser des résistances pour limiter le courant et protéger les leds de la destruction.
Dans le projet, vous allez commencer avec une couleur Verte, passer par le Bleu pour enfin terminer avec le Rouge.

Matériel nécessaire:

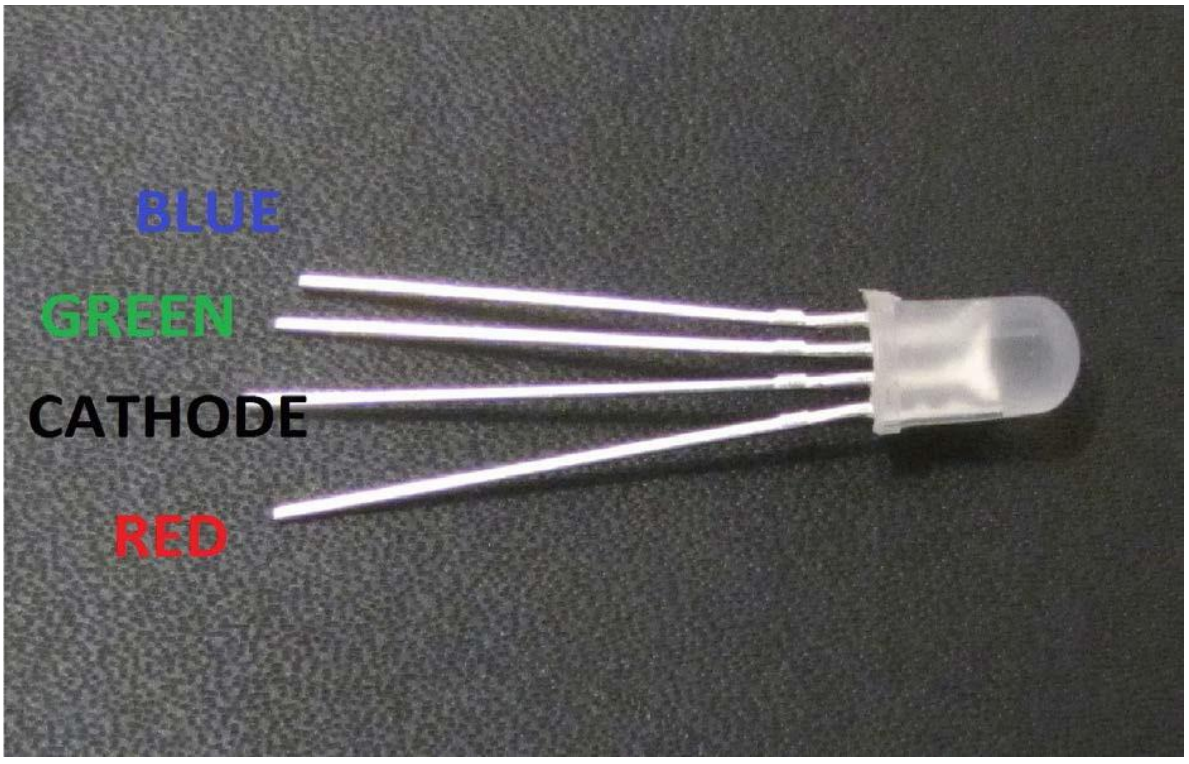
(1) x Elegoo Uno R3
(1) x Planche prototype
(4) x Câbles mâle-mâle
(1) x led RGB
(3) x résistance 220 ohms

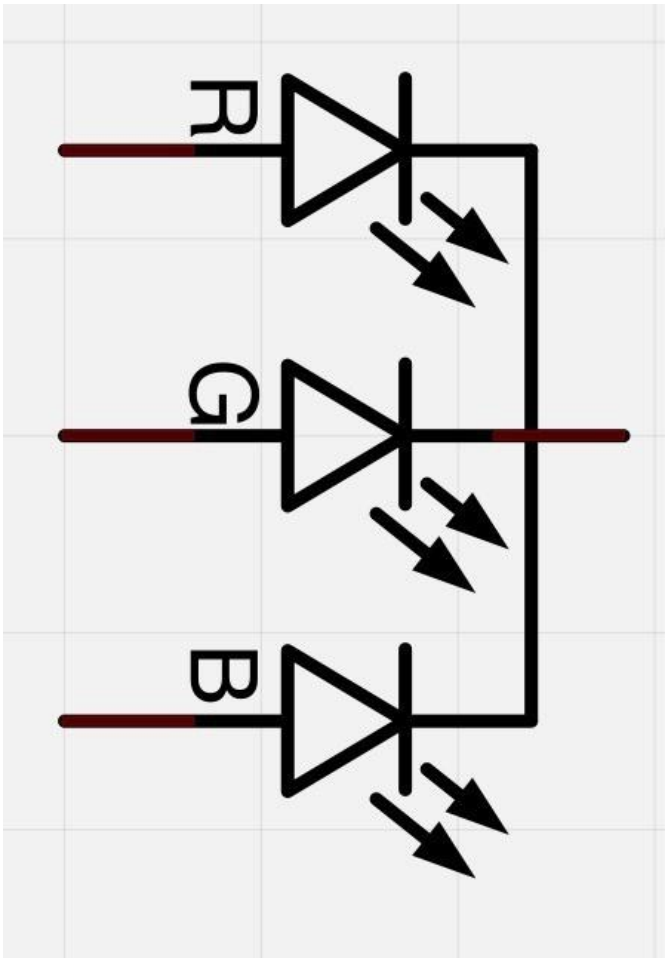
Présentation des composants

RGB:

Au premier abord, la led ressemble à une led standard monochrome. En fait, à l'intérieur, ce sont 3 leds qui sont installées : une bleue, une rouge et une verte. Il est possible en mixant les couleurs de créer une large palette de coloris. Pour cela, il suffit d'ajuster la brillance de chaque led. On pourrait pour cela utiliser des valeurs de résistances différentes, un travail bien compliqué pour obtenir la couleur souhaitée. Heureusement, la carte UNO R3 va être très utile. Grâce à ses sorties analogiques, l'ajustement d'intensité sera un jeu d'enfant.

L'illustration suivante montre comment se répartissent les différentes pattes de la led.



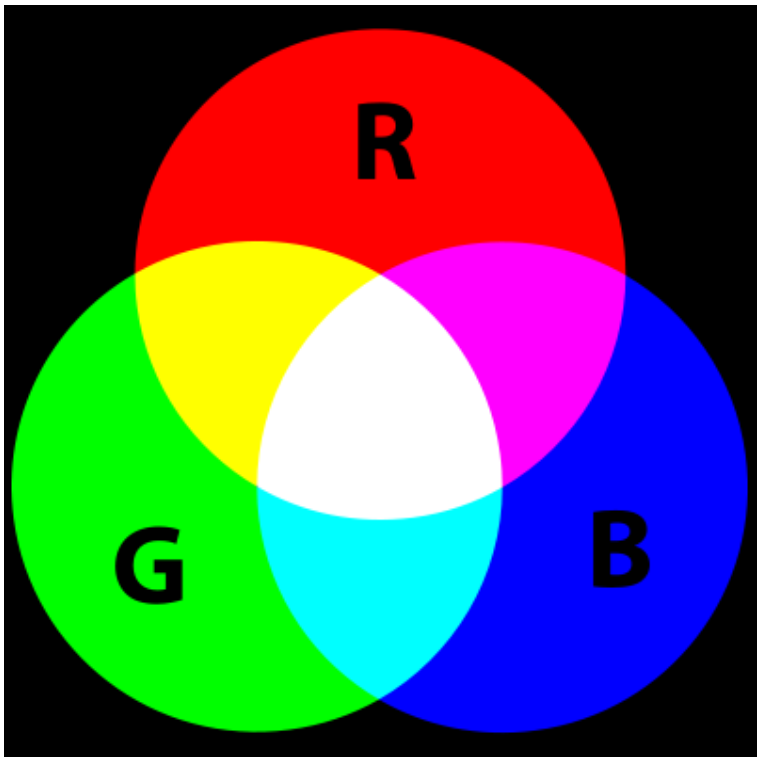


Il faut connecter une sortie analogique de la carte UNO R3 sur les pattes ROUGE/VERT/BLEU et une masse sur la patte (GND)

N’oubliez pas encore une fois de positionner une résistance 220 ohms avant chaque patte d’entrée.

COLEURS:

L'illustration suivante montre comment se créent les différentes couleurs en fonction de l'association des couleurs de bases. Servez-vous en pour créer vos couleurs personnalisées.

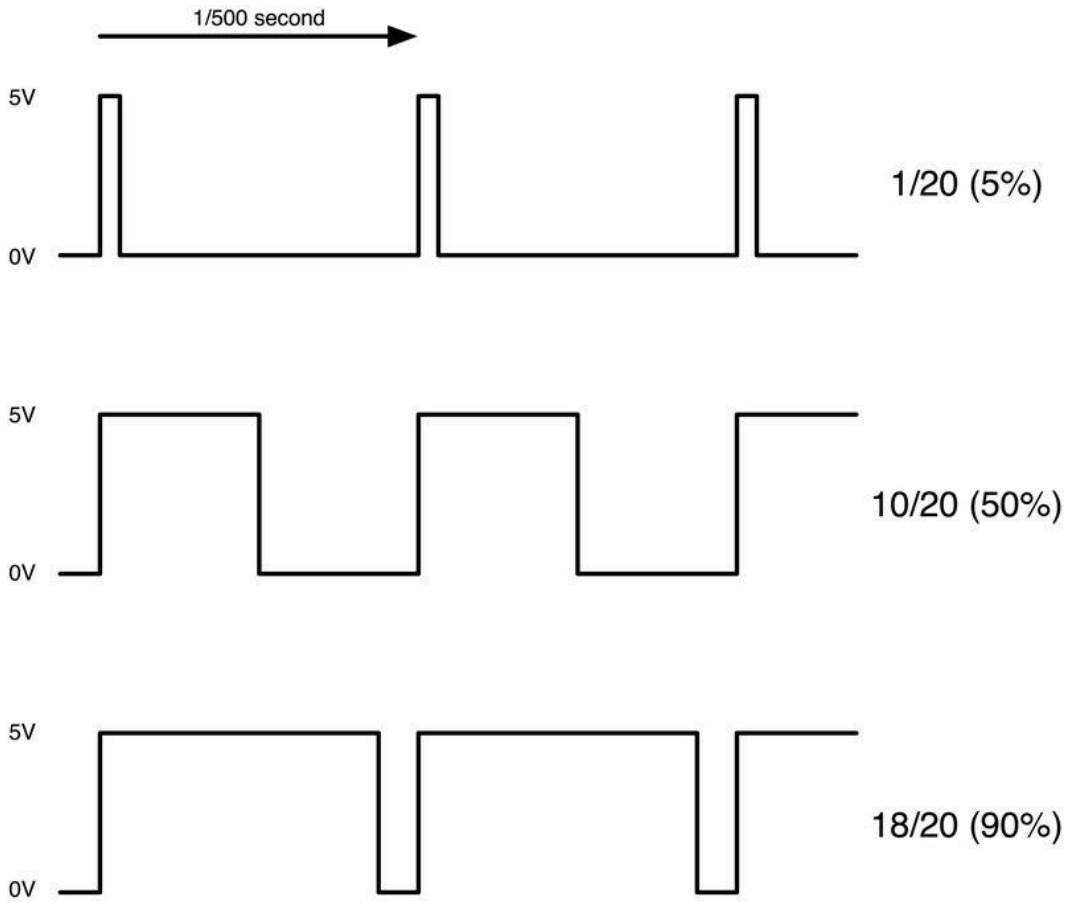


Théorie (PWM)

Pulse Width Modulation (PWM) ou modulation de largeur de pulsation est une technique pour contrôler la puissance d'une sortie.

Vous pouvez l'utiliser pour contrôler la luminosité de chaque led.

Les diagrammes suivants illustrent le fonctionnement des sorties PWM de la carte UNO R3:



Toutes les 1/500 de seconde, les sorties PWM vont produire une impulsion. La longueur de ces impulsions peut être contrôlée avec la fonction 'analogWrite'. Ainsi 'analogWrite(0)' ne produira pas d'impulsion et 'analogWrite(255)' produira une impulsion qui durera jusqu'au prochain déclenchement. Il est donc possible en ajustant la valeur entre 0 et 255 de moduler la puissance en sortie en 0% et 100%

Si la sortie est de 5V sur 100% du temps, en modulant à 90%, nous obtiendrons 90% de 5V.

Connexion

Schéma de câblage

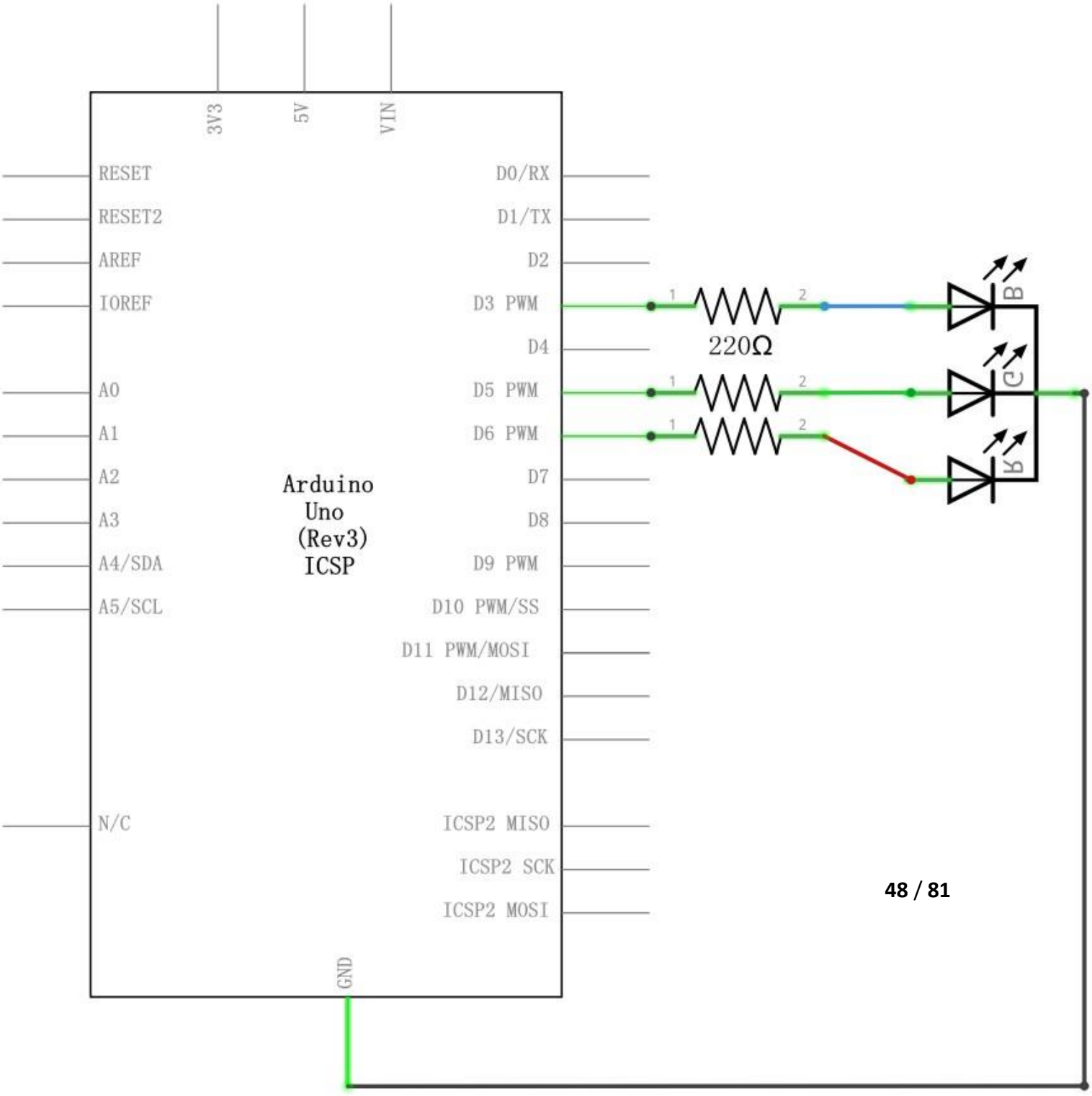
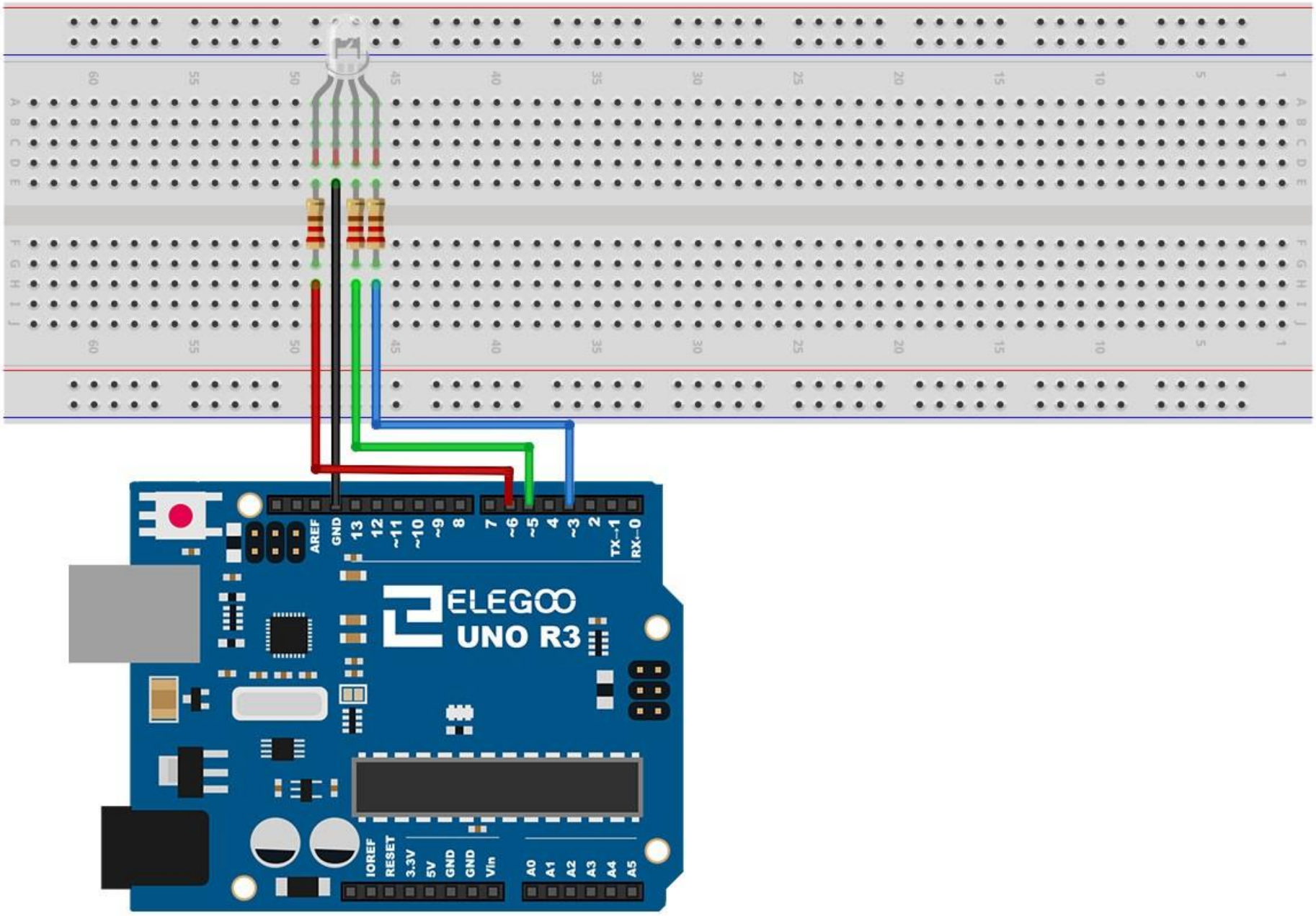


Diagramme de câblage



Code

Après avoir réalisé le câblage, ouvrir le code « Leçon 4 RGB LED », téléversez le code de la même manière que présenté en leçon 2.

Notre code va utiliser une boucle « FOR » pour boucler au travers des couleurs.
Première boucle : rouge vers vert.
Deuxième boucle : vert vers bleu.
Dernière boucle : bleu vers rouge.

Le sketch commence en spécifiant les pins utilisées pour chaque couleur:

```
// Define Pins
#define BLUE 3
#define GREEN 5
#define RED 6
```

Le bloc “setup” associe les pins à leur couleur et initialise les pins en tant que sorties analogiques :

```
void setup()
{
pinMode(RED, OUTPUT);
pinMode(GREEN, OUTPUT);
pinMode(BLUE, OUTPUT);
digitalWrite(RED, HIGH);
digitalWrite(GREEN, LOW);
digitalWrite(BLUE, LOW);
}
```

Avant de regarder le bloc “loop” en détail, regardons la dernière fonction du sketch

```
redValue = 255; // choose a value between 1 and 255 to change the color.
greenValue = 0;
blueValue = 0;
```

Cette fonction prend en compte 3 arguments. Un pour la brillance du rouge, un pour celle du vert et enfin un pour celle du bleu. Dans chaque cas, la valeur à assigner

varie entre 0 et 255. La fonction appelle ensuite analogWrite pour définir la brillance de chaque led.

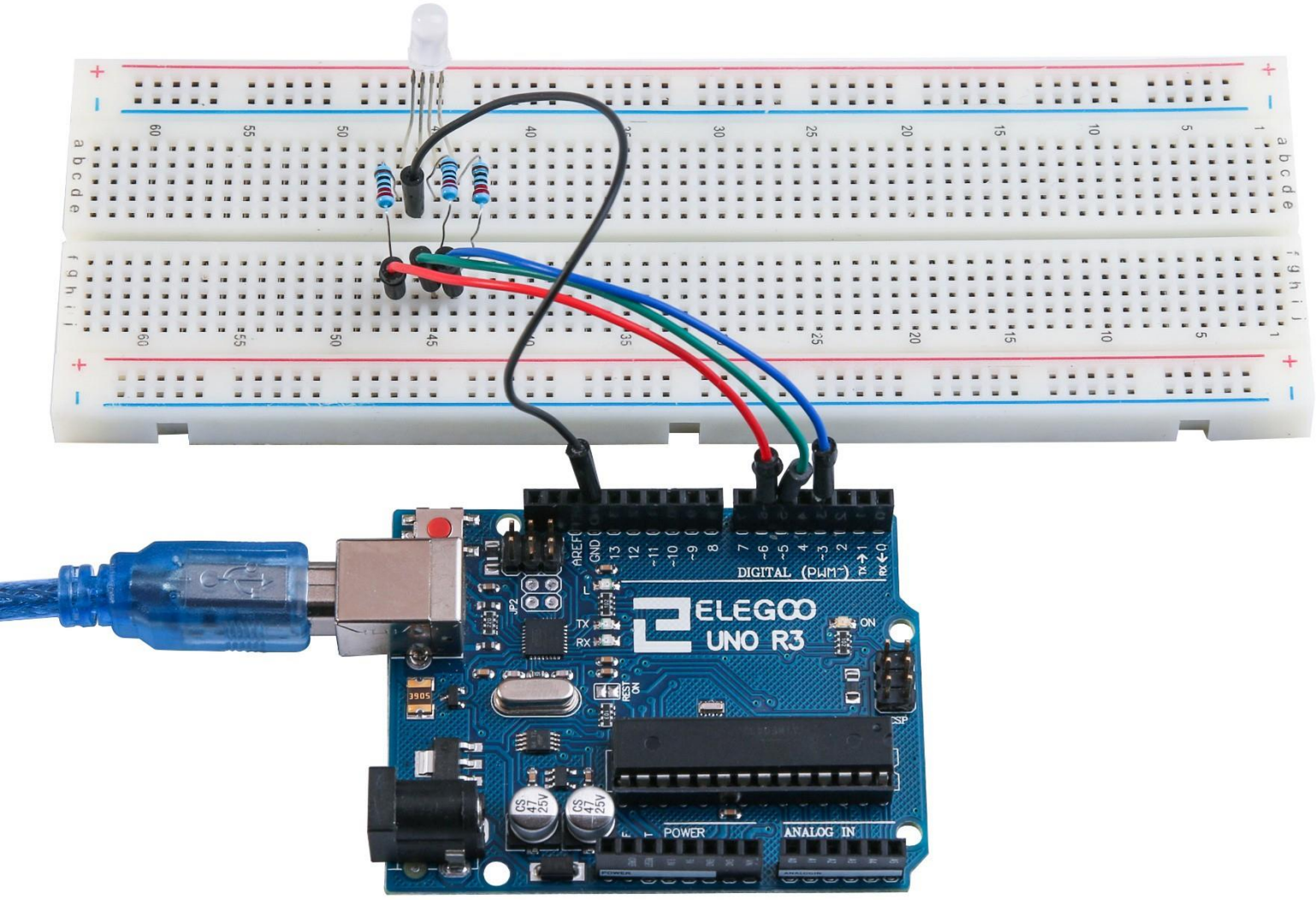
Si vous regarder le bloc “loop” vous constatez que l’on définit la quantité de rouge, de vert, de bleu et que l’on fait une pause avant de passer à la suivante.

#define delayTime 10 // fading time between colors

Delay(delayTime);

Essayez d'ajouter quelques couleurs de votre choix à l'esquisse et regardez l'effet sur votre LED.

Illustration



Leçon 5 Digital Inputs

But de la leçon

Dans cette leçon, vous allez apprendre à utiliser les entrées digitales pour allumer/éteindre une led.

Presser un premier bouton allumera la led, presser un autre bouton l'éteindra.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Planche prototype

(1) x LED rouge 5mm

(1) x résistance 220 ohms

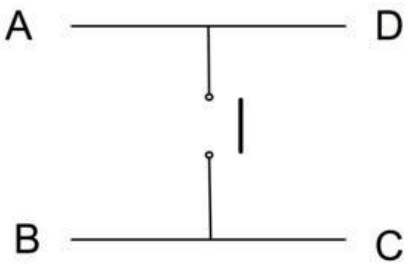
(2) x boutons poussoirs

(7) x Câbles mâle-mâle

Présentation du composant

Boutons poussoirs:

Les boutons poussoirs sont des composants très simples. Lorsque vous pressez le bouton, un contact électrique se fait et laisse passer le courant. Les boutons poussoirs utilisés ont 4 pattes, ce qui peut créer une certaine confusion.



En fait, il n'y a bien que 2 connexions électrique. A et D sont connectées ensemble et B et C aussi. Presser le bouton permet au courant de lier électriquement A-D avec B-C.

Connexion

Schéma de câblage

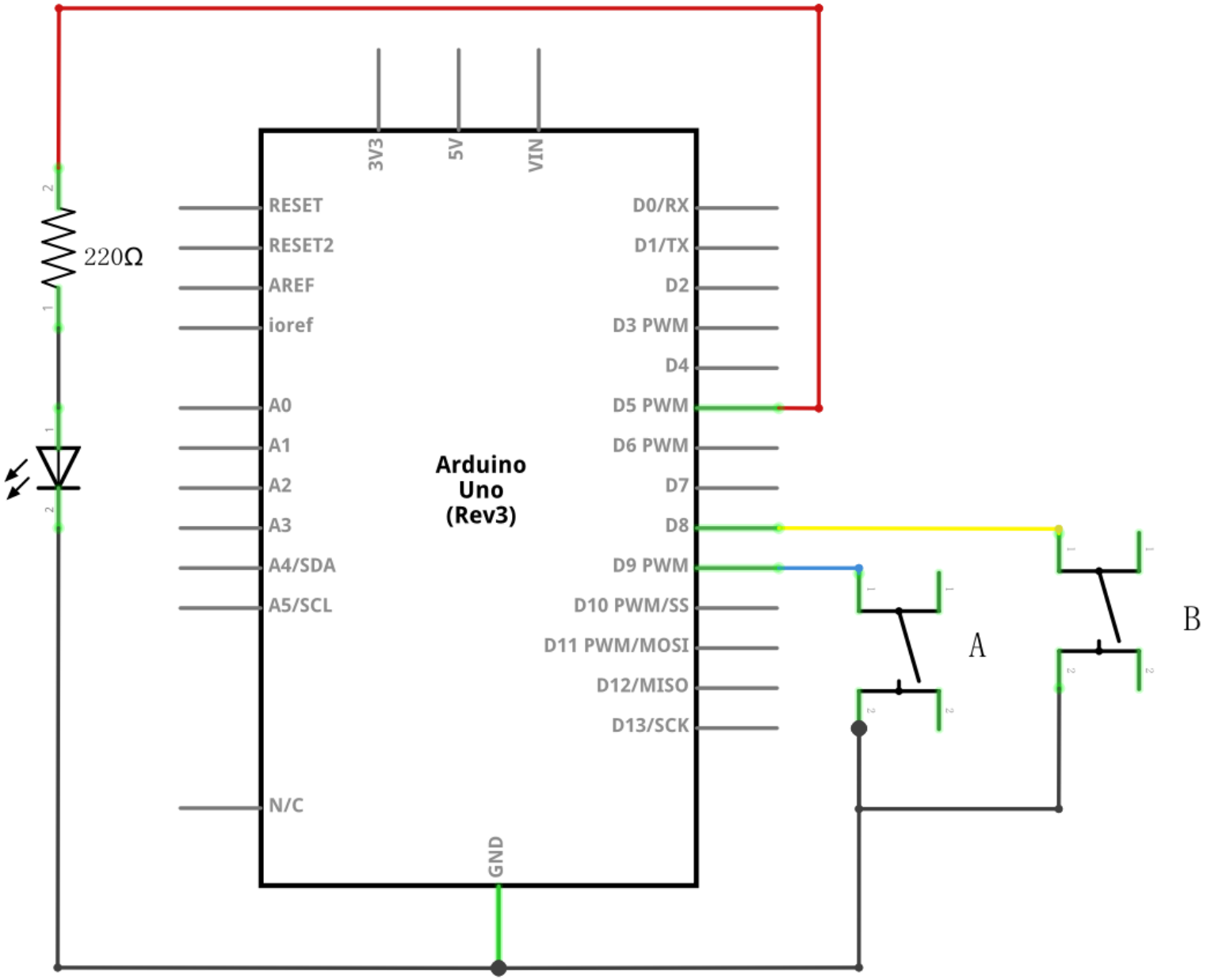
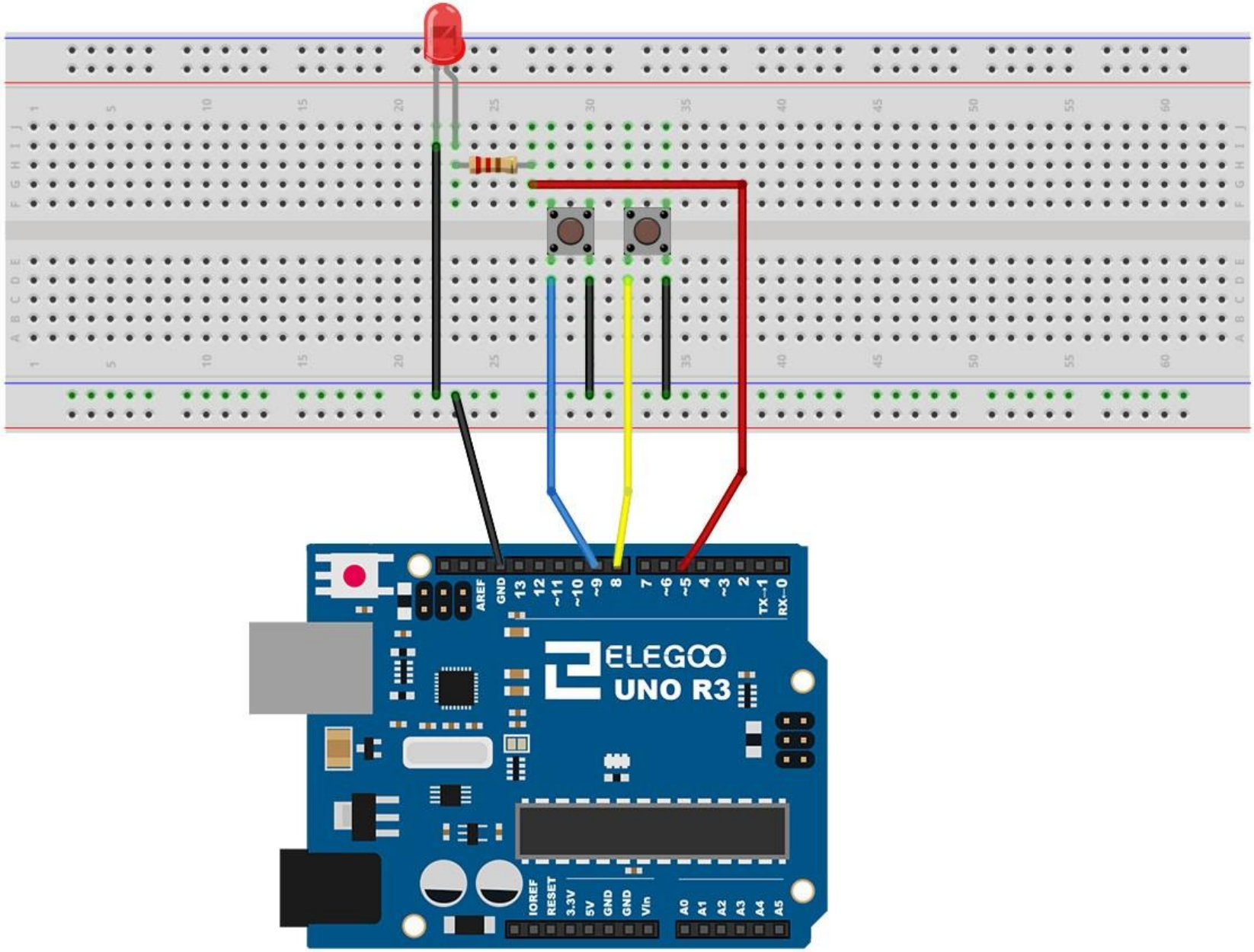


Diagramme de câblage



Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 5 Digital Inputs” puis téléversez le code sur la carte UNO R3 comme présenté lors de la leçon 2.

Presser le bouton de gauche allumera la led, presser le bouton de droite l’éteindra.

La première partie du sketch est consacrée à la définition des 3 variables pour les 3 pins qui seront nécessaires au fonctionnement du montage.
Dans le bloc « setup », les pins sont affectées en tant qu’entrées ou sorties.
‘ledPin' est défini en tant que sortie (OUTPUT), 'buttonApin' et 'buttonBpin' en tant qu’entrées (INPUT).

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

Petite particularité, les entrées sont définies en tant qu’entrées avec résistance de PULLUP. Pourquoi cette subtilité ? Comme vous pouvez le constater sur le schéma de câblage, le fait de presser un bouton met la pin associée à la masse. Mais lorsque le bouton n’est pas pressé, il peut subsister des signaux parasites qui peuvent être interprétés par la carte comme une mise à la masse alors que le bouton n’est pas pressé. Pour éviter ces parasites, on utilise la technique dite de « la résistance de « pullup » qui permet de forcer un état haut via une résistance connectée à une borne +. La carte UNO est capable de simuler ce branchement, pour cela, il suffit de déclarer la pin en INPUT_PULLUP au lieu de faire une déclaration INPUT simple.

Le bloc loop prend ensuite la mesure de la position de chaque bouton:

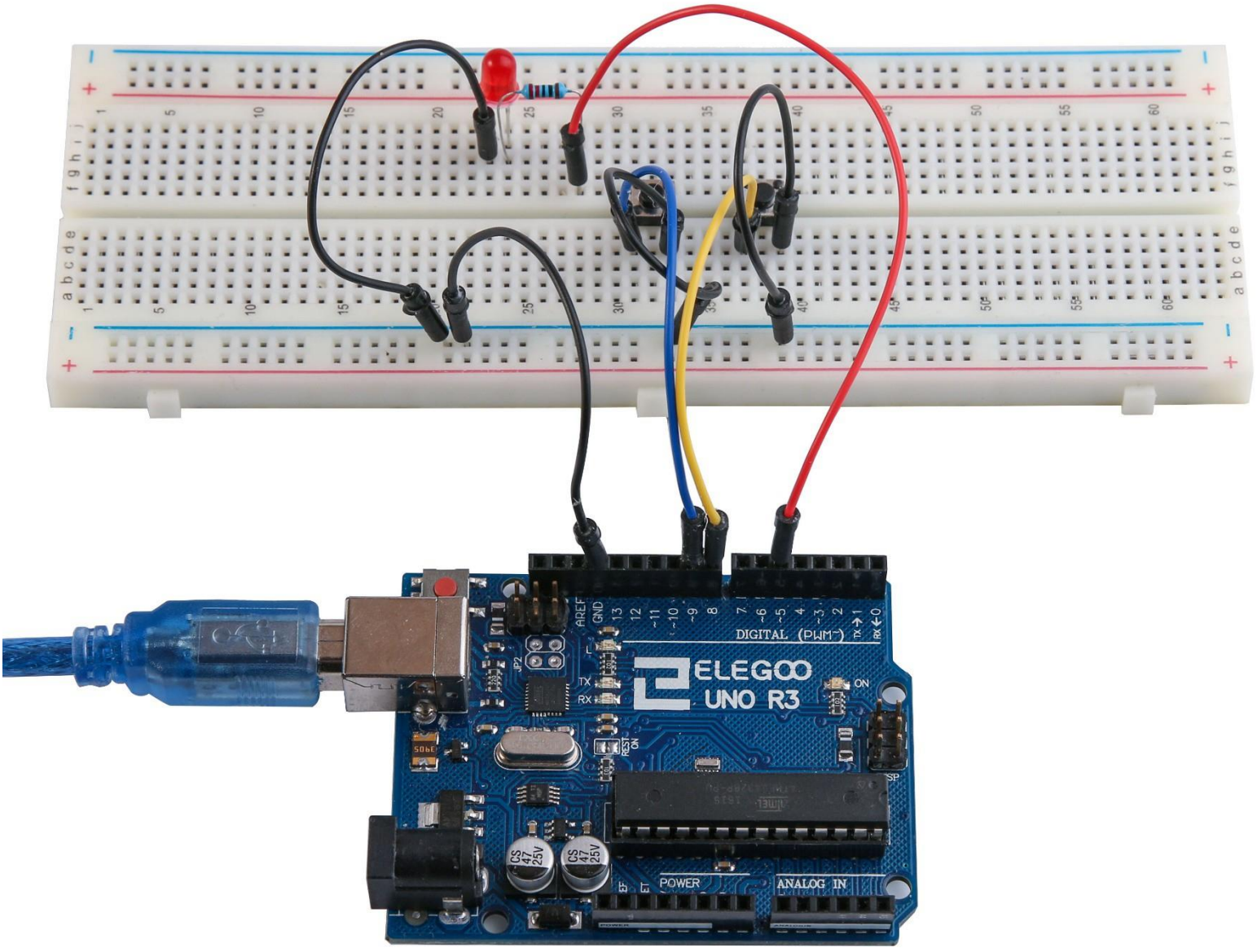
```
void loop()
{
  if (digitalRead(buttonApin) == LOW)
  {
    digitalWrite(ledPin, HIGH);
  }
  if (digitalRead(buttonBpin) == LOW)
  {
    digitalWrite(ledPin, LOW);
```

```
    }
}
```

On regarde si le bouton A est pressé (LOW). Si oui, on allume la led, si non, on ne fait rien.

On regarde ensuite si le bouton B est pressé. Si oui on éteint la led, si non, on ne fait rien.

Illustration



Leçon 6 Active buzzer

But de la leçon

Dans cette leçon, vous allez apprendre à générer des sons avec un « active buzzer ».

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x Active buzzer

(2) x Câbles Mâle-Femelle

Présentation du composant

BUZZER:

Les buzzers électroniques sont alimentés en courant continu et équipés de circuits intégrés. Ils sont largement utilisés dans les ordinateurs, imprimantes, alarmes, jouets etc…. Il en existe deux types : les actifs et les passifs. Placez le buzzer avec les pattes vers le haut. Celui où vous pouvez distinguer un petit circuit généralement vert est un buzzer passif.

La différence entre les deux réside dans le fait qu'un buzzer actif possède un oscillateur intégré, il va donc générer un son lorsque le courant passe. Un buzzer passif utilise un signal en entrée pour générer le son (généralement signal carré en 2kHz et 5kHz).

Connection

Schéma de câblage

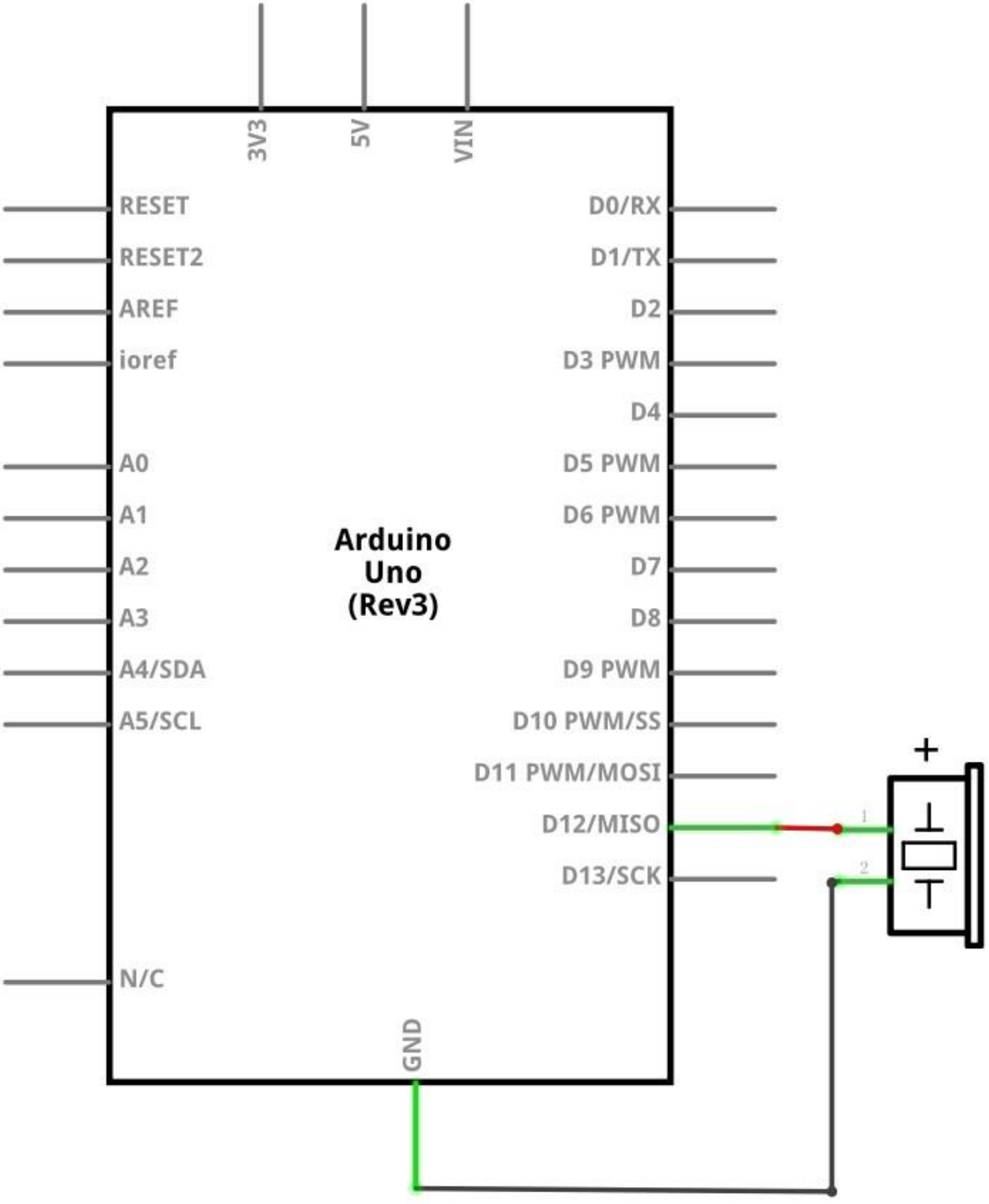
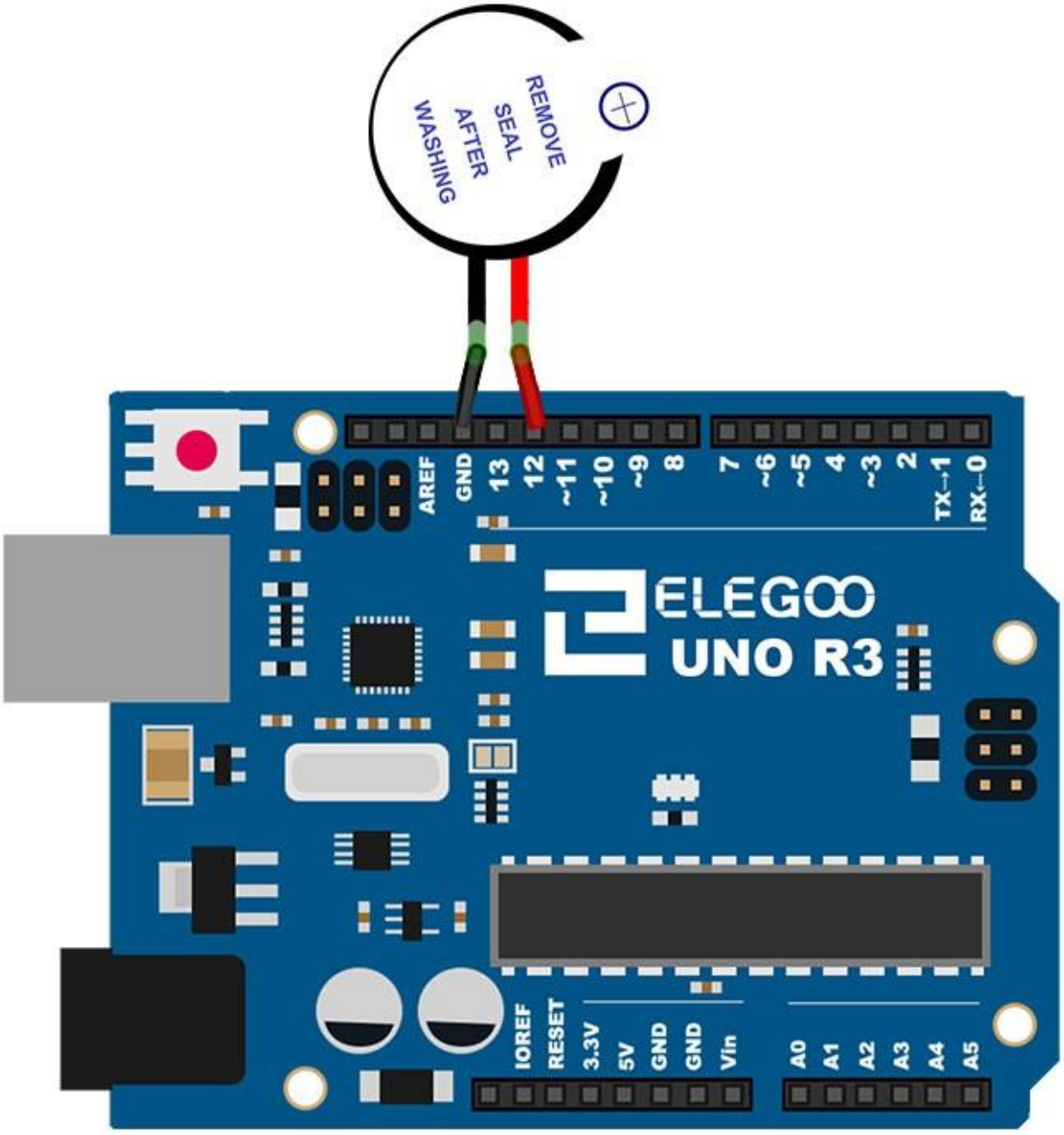


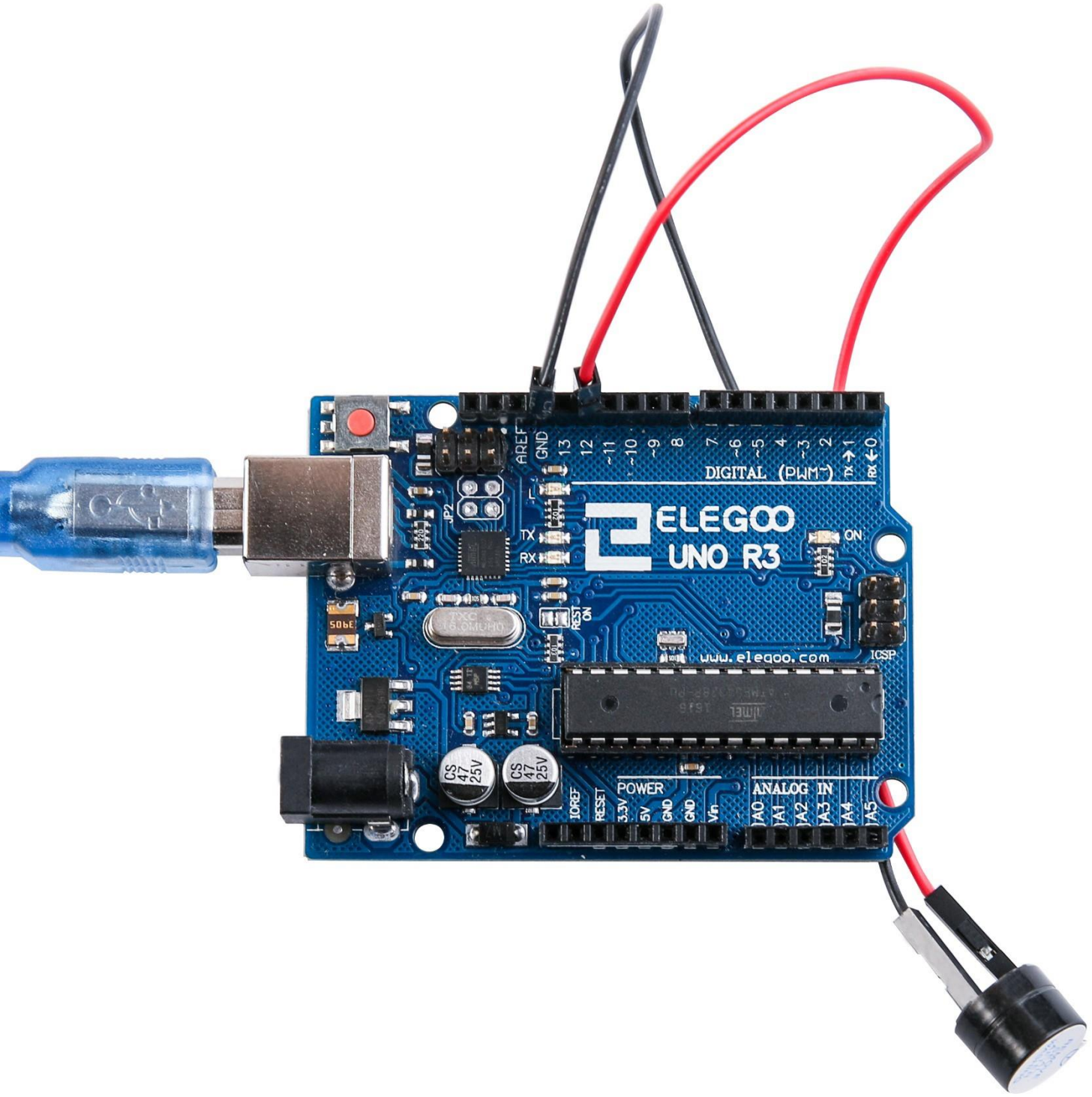
Diagramme de câblage



Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 6 Making Sounds” et Téléversez le code sur la carte UNO R3 comme expliqué avec la leçon 2.

Illustration



Leçon 7 Tilt Ball Switch

But de la leçon

Dans cette leçon, vous allez apprendre comment il est possible de détecter une inclinaison avec le capteur “tilt ball switch”.

Matériel nécessaire:

(1) x Elegoo Uno R3

(1) x capteur Tilt Ball switch

(2) x Câbles Mâle-Femelle

Présentation du composant

Capteur « Tilt » :

Ce capteur vous permet de détecter une orientation ou inclinaison. Il est petit, pas cher est consomme très peu d’énergie.

Sa simplicité fait qu’il est très répandu pour les jouets et toutes sortes de gadgets. Vous le trouverez sous différents types et noms : "mercury switches", "tilt switches" ou "rolling ball sensors".

Ces capteurs sont généralement faits d’une cavité cylindrique dans laquelle se trouve une masse libre de mercure par exemple. Au bout de la cavité, deux éléments conducteurs non reliés. Lorsque la cavité change d’orientation ou se retourne, la masse libre vient sur en contact avec les deux éléments conducteur et crée un contact électrique qui laisse passer le courant.

Alors qu’ils ne sont pas aussi précis qu’un accéléromètre, ils sont parfaits pour faire de la détection de mouvement ou d’orientation.

Connection

Schéma de câblage

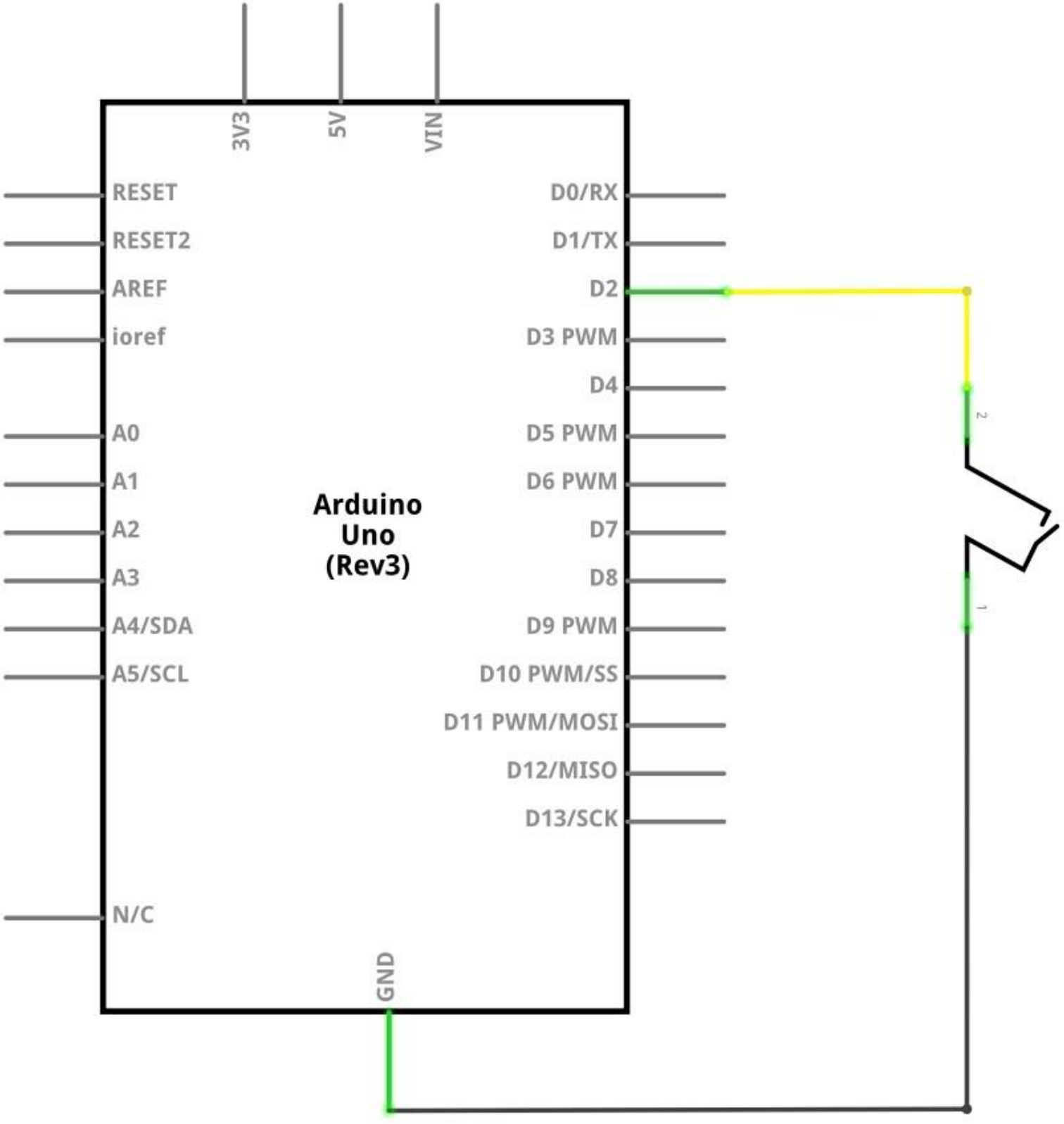
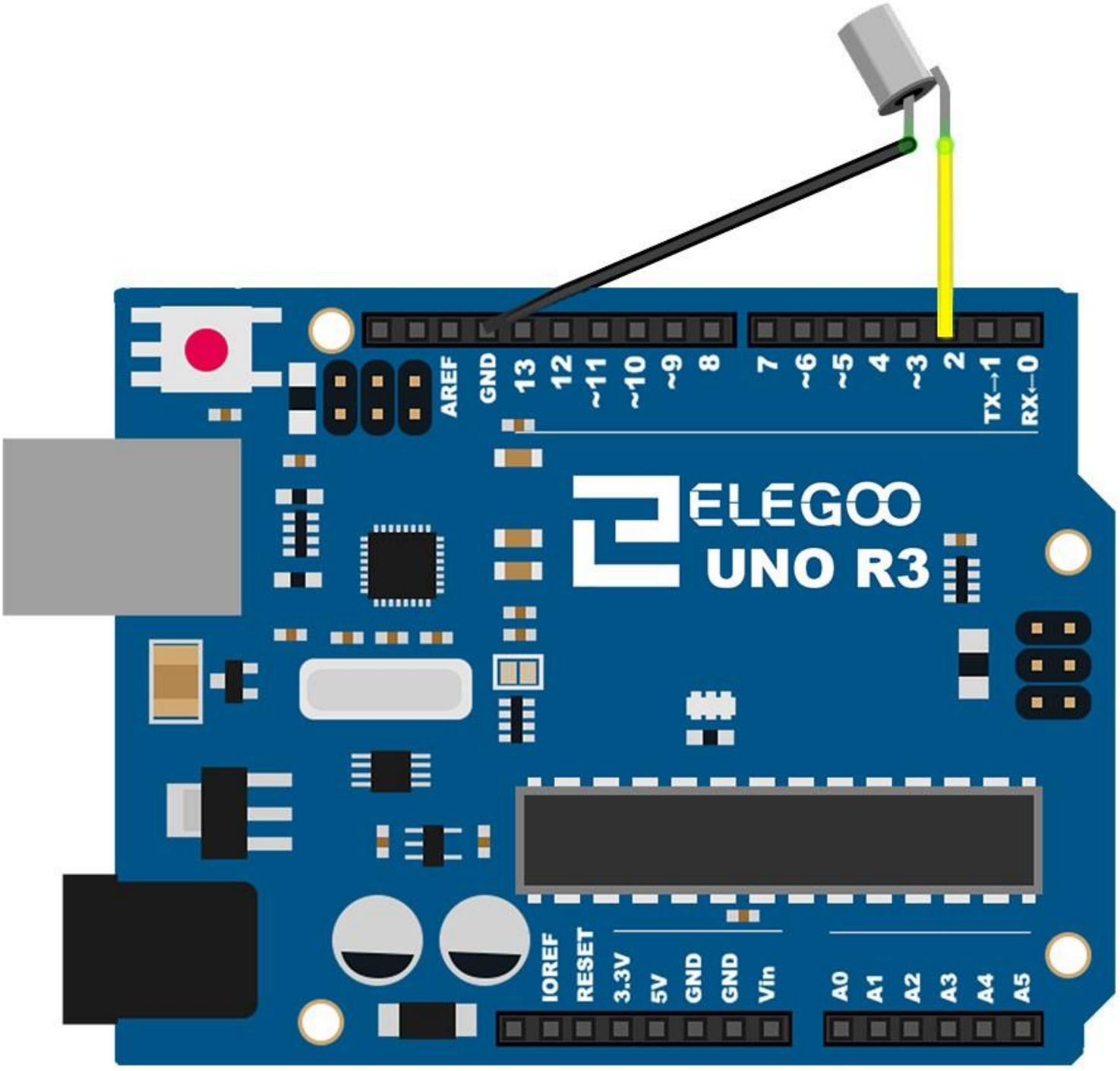


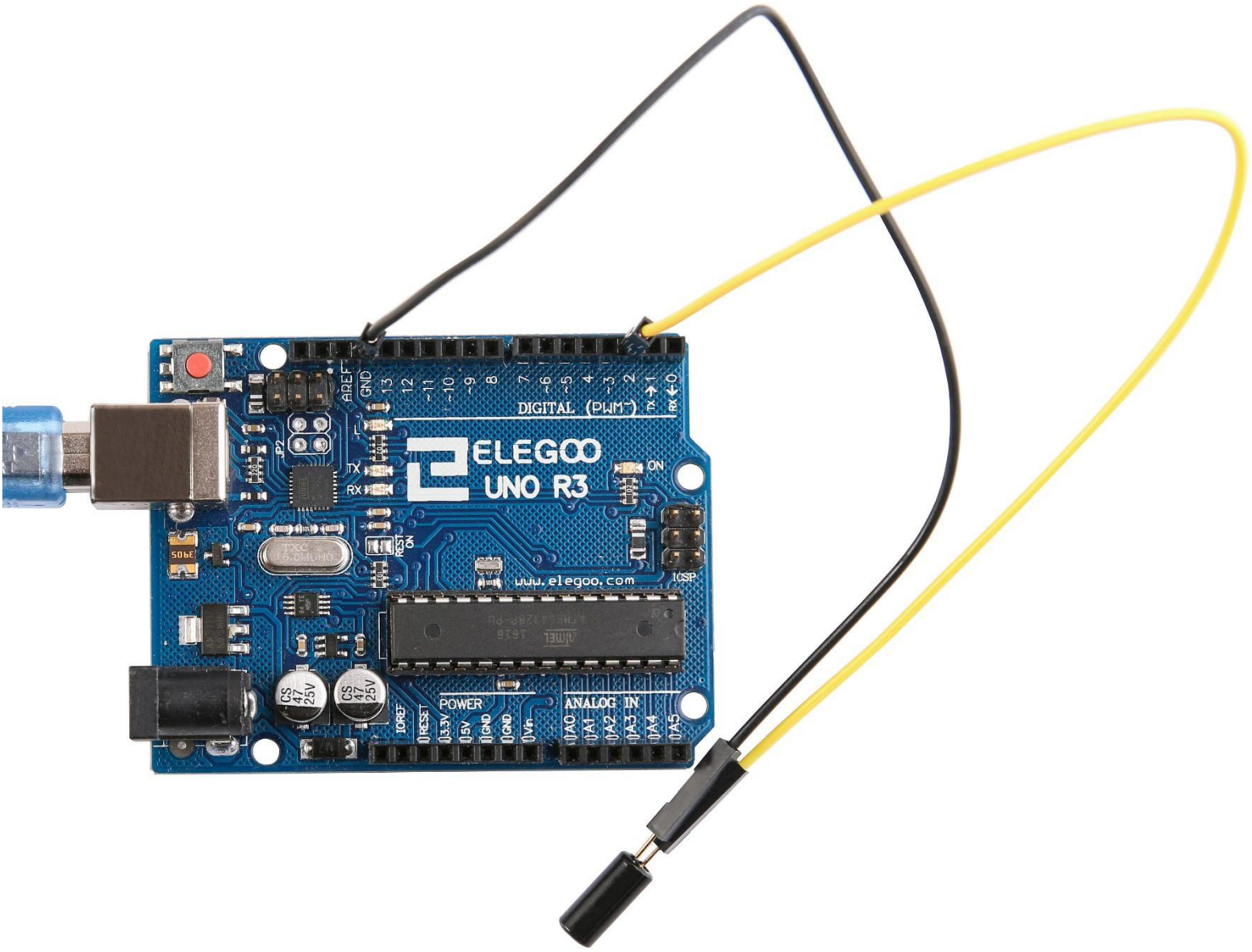
Diagramme de câblage



Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 8 Ball Switch” et Téléversez-le sur la carte UNO RO3.

Illustration



Leçon 8 Eight LED with 74HC595

But de la leçon

Dans cette leçon, vous allez apprendre comment il est possible de commander indépendamment 8 leds sans avoir besoin de mobiliser 8 pins sur la carte UNO R3. Il est bien évidemment possible de connecter 8 leds sur 8 pins de votre carte UNO R3 si vous n'avez pas besoin de beaucoup d'autres connexions pour votre projet, cela fonctionnera parfaitement, mais il est vrai que la plupart du temps, il est nécessaire de brancher tout un tas de boutons, capteurs et donc les pins deviennent rapidement pas assez nombreuses. C'est pour cela que nous allons voir comment interfacer les leds à la carte via une puce 74HC595 qui est un convertisseur Série / Parallèle. Cette puce dispose de 3 entrée et 8 sorties.

Le recours à cette puce ralentira un peu les possibilités de switch on-off (de 800000 à 500000 par seconde), mais cela reste une fréquence plus que raisonnable.

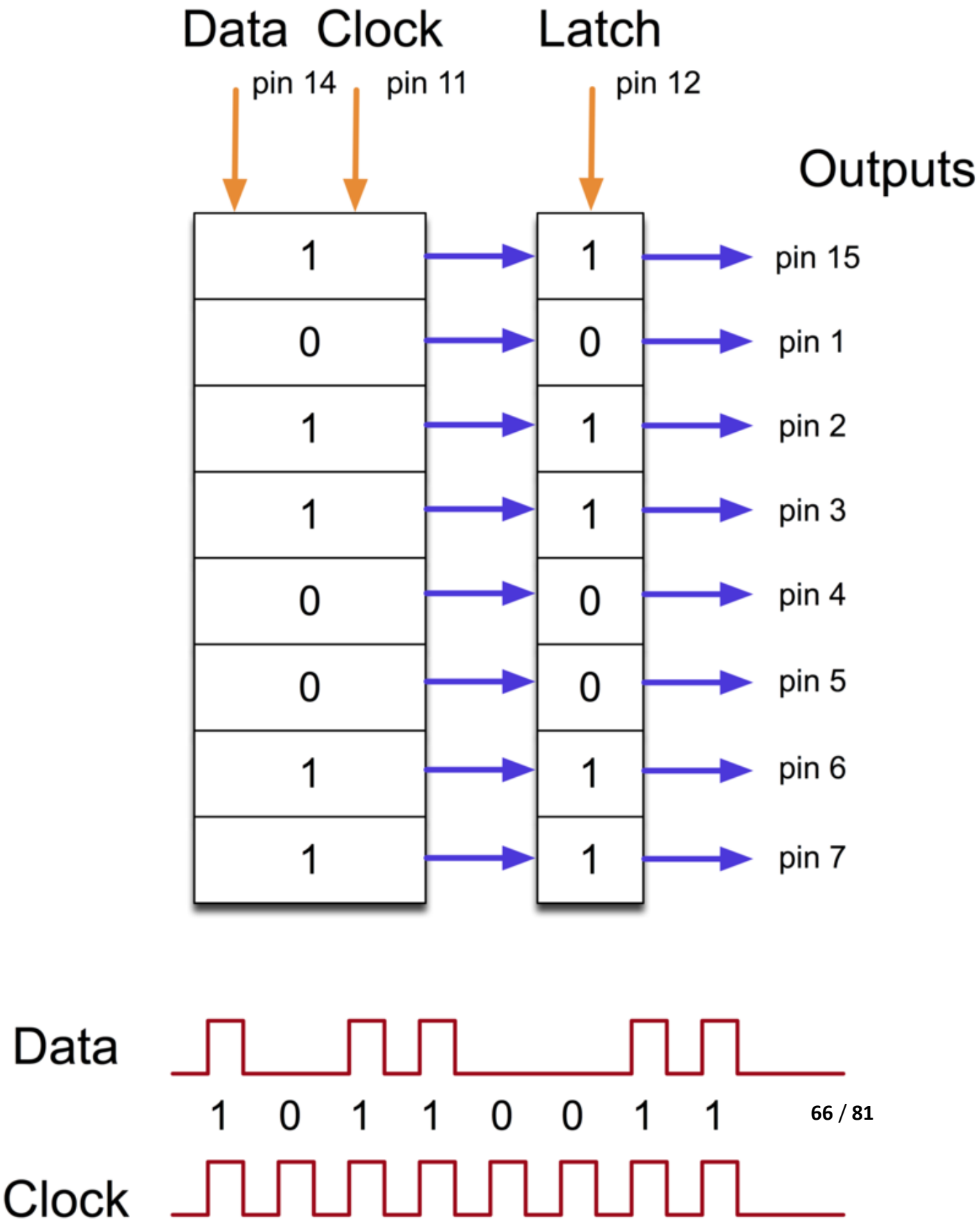
Matériel nécessaire:

(1) x Elegoo Uno R3
(1) x planche prototype
(8) x leds
(8) x résistances 220 ohm
(1) x puce 74hc595
(14) x Câbles mâle-mâle

Présentation du composant

74HC595 Shift Register:

Le "shift register" est un type de puce qui contient un nombre de registres mémoire qui peuvent pendre soit la valeur 0, soit la valeur 1. Il est possible de changer chaque valeur indépendamment.


Data
Clock
Latch
pin 14
pin 11
pin 12
Outputs
1
0
1
1
0
0
1
1
1
0
1
1
0
0
1
1
pin 15
pin 1
pin 2
pin 3
pin 4
pin 5
pin 6
pin 7
Data
1 0 1 1 0 0 1 1
Clock

Pour cette puce, l'horloge fonctionne avec 8 pulsations. A chaque pulsation correspond une adresse registre. Si la pin est à l'état haut, le registre prendra la valeur 1, s'il est à l'état bas, il prendra la valeur 0.

Connection

Schéma de câblage

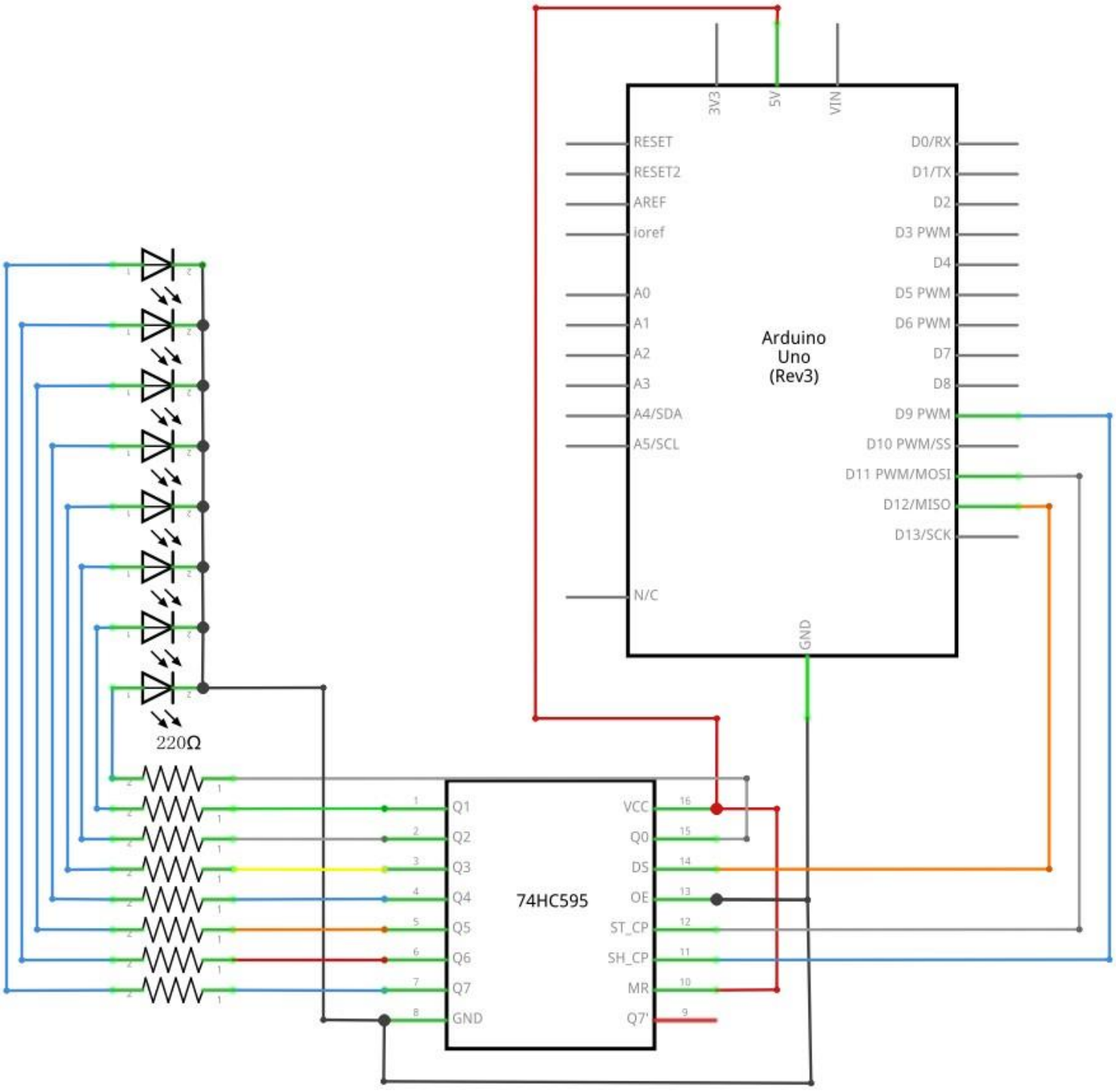
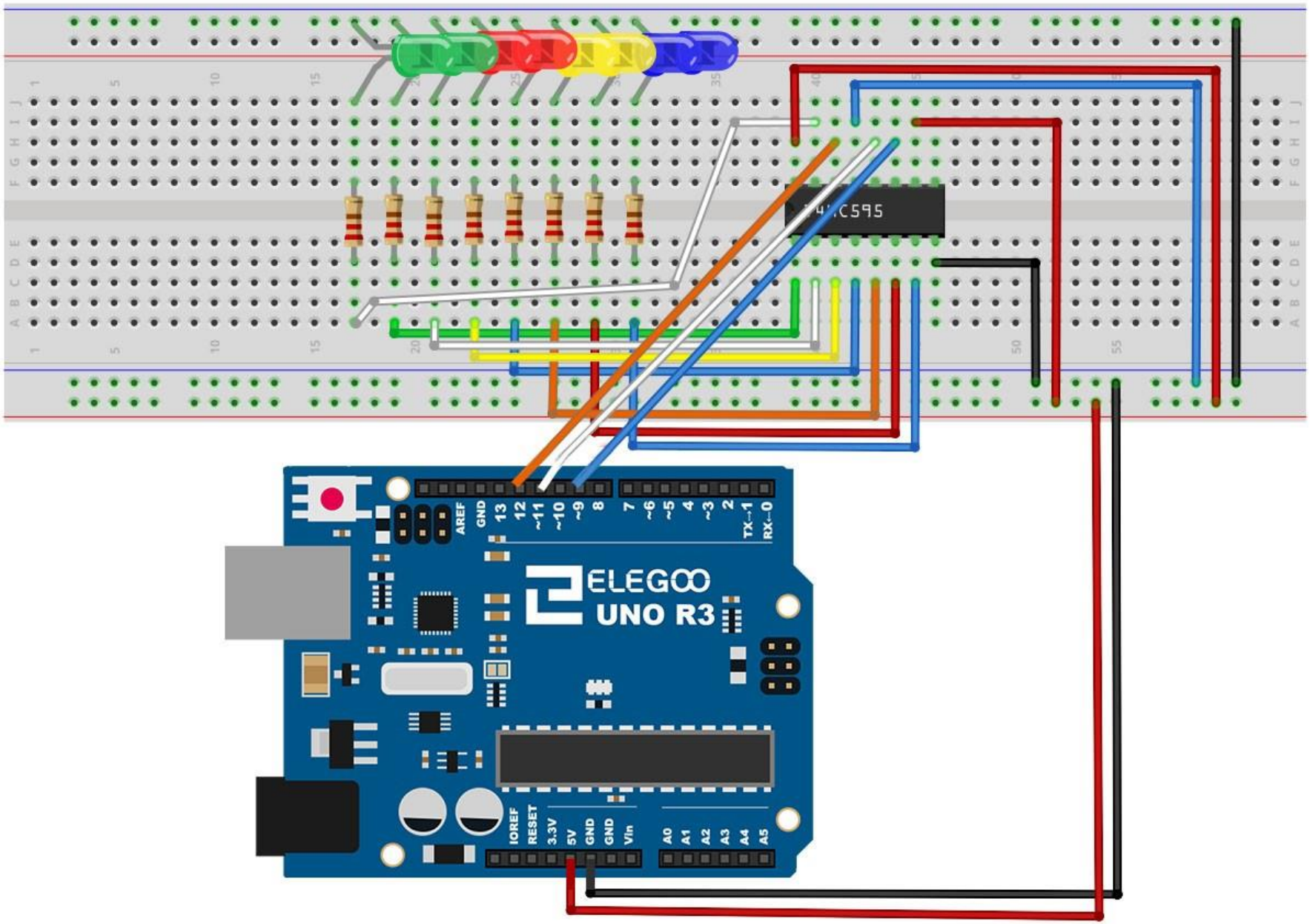


Diagramme de câblage



Ce schéma est assez complexe et demande d'être fait avec soin.
Le mieux étant de commencer par la mise en place de la puce 74HC595.
Toutes les leds sauf une sont sur le même côté de la puce.
Après avoir mis la puce en place installez les résistances.
Placez ensuite les différentes LEDS. Faites bien attention au sens des leds, la patte la plus longue devant être du côté de la borne positive du montage.

Code

Après avoir réalisé le câblage, ouvrez le sketch "Leçon 24 Eight LED with 74HC595" et Téléversez-le sur la carte UNO R3.

Les premières lignes du code définissent les pins de la carte UNO R3

```
int latchPin = 11;
int clockPin = 9;
int dataPin = 12;
```

Ensuite, une variable de type 'byte' est définie pour la définition de quelle led est à l'état allumée et quelle est à l'état éteinte.

```
byte leds = 0;
```

La function 'setup' réalise l'affectation des pins sur la carte

```
void setup()
{
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
}
```

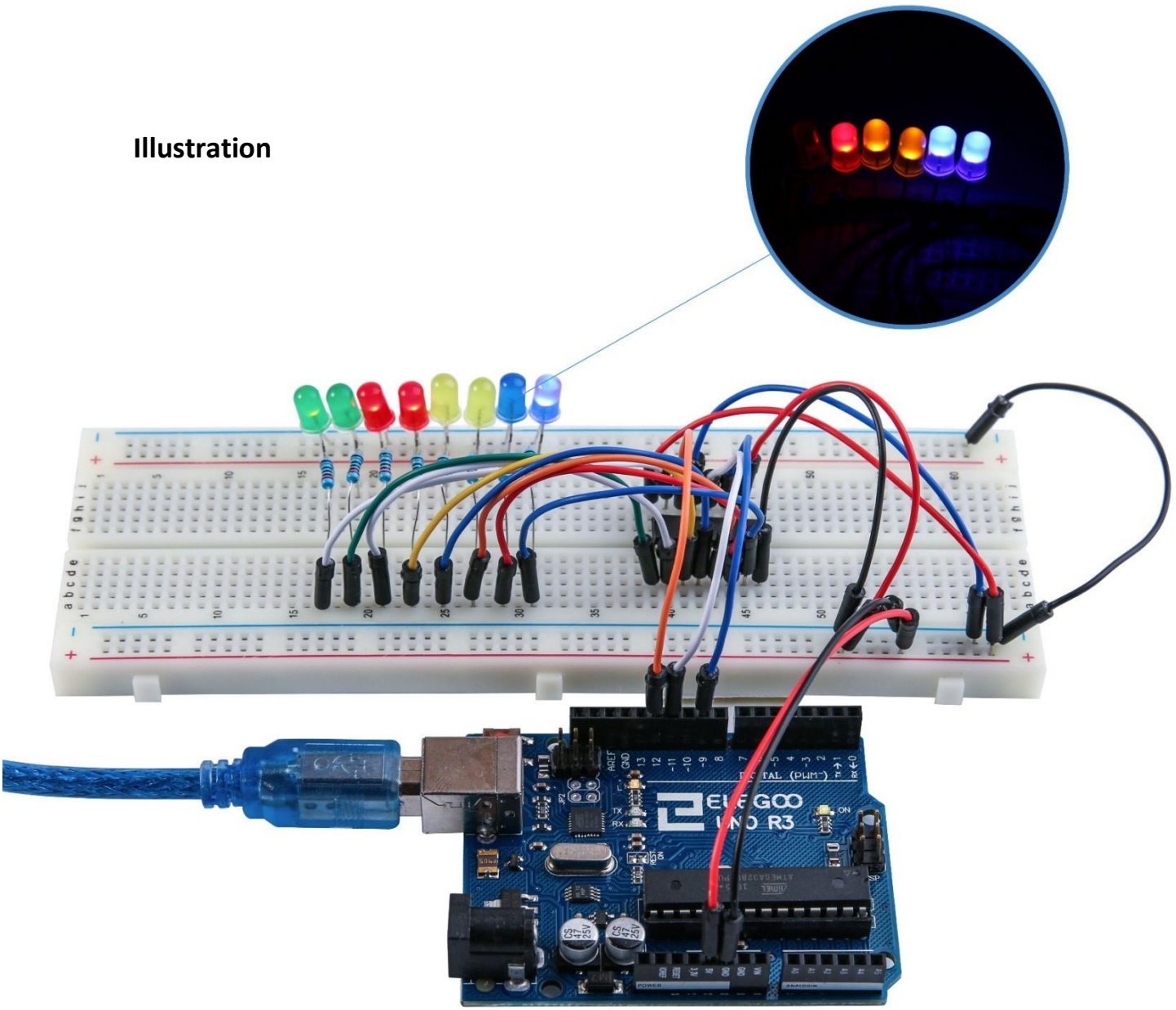
La fonction loop commence par définir toutes les leds à 'éteinte' en donnant la valeur 0 à la variable 'led'. Ensuite, elle appelle ensuite 'updateShiftRegister' qui va permettre l'allumage de certaines leds et l'extinction d'autres en définissant les valeurs 0 ou 1.
La fonction loop fait ensuite une pause de 500ms et procède ensuite à l'allumage/extinction des leds.

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

La fonction 'updateShiftRegister', commence par définir latchPin à bas, puis appelle la fonction UNO 'shiftOut' avant de remettre 'latchPin' à haut.

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

Illustration



Leçon 9 The Serial Monitor

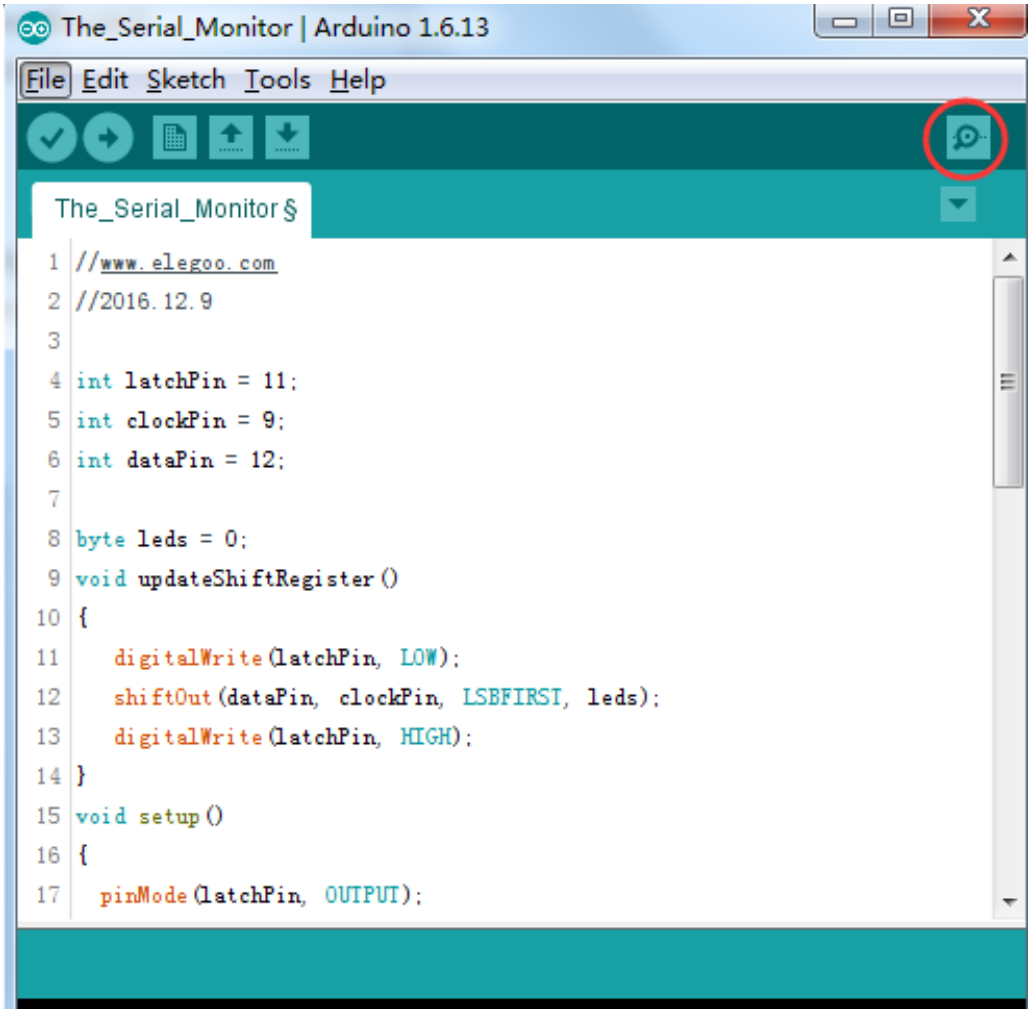
But de la leçon

Dans cette leçon, vous allez apprendre comment il est possible de contrôler l'allumage/extinction des leds avec le moniteur série. Cette leçon repose sur la leçon 8 qu'il est impératif d'avoir accompli avant.

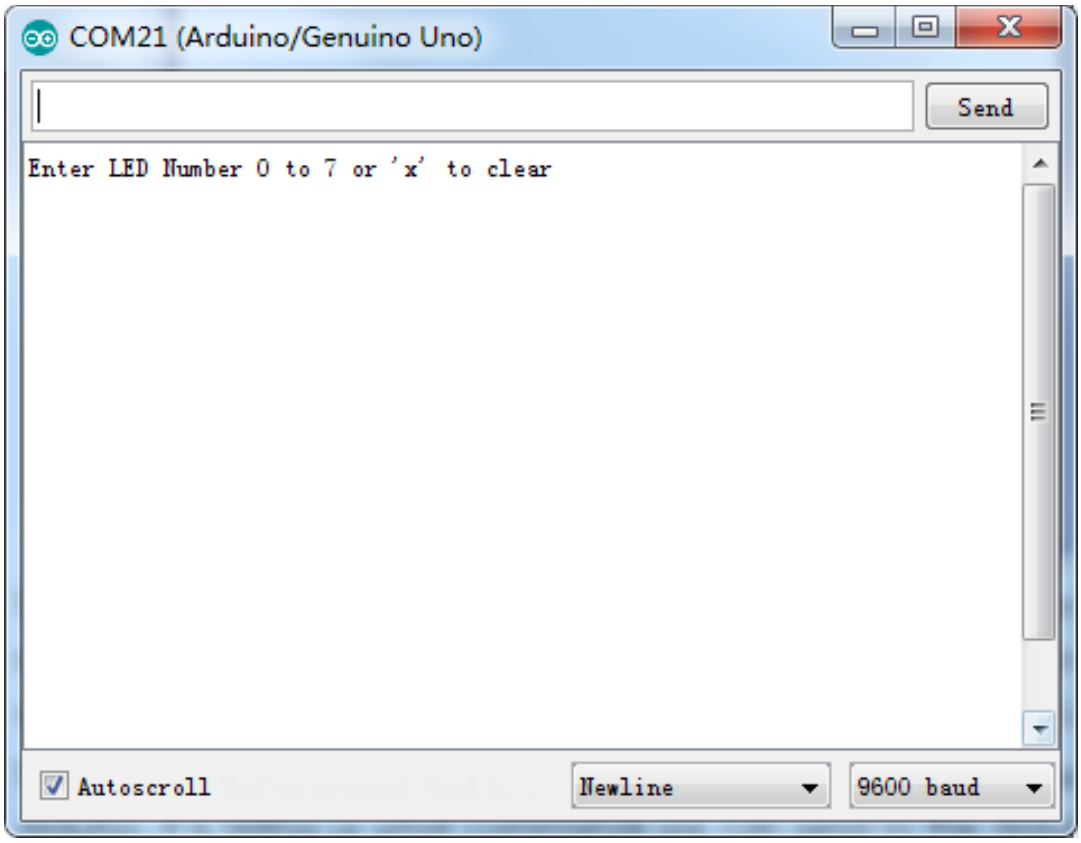
Dans cette leçon, le montage électronique est strictement identique à la leçon 8.

Etapes

Après avoir ouvert et téléversé le sketch, ouvrez le moniteur série.

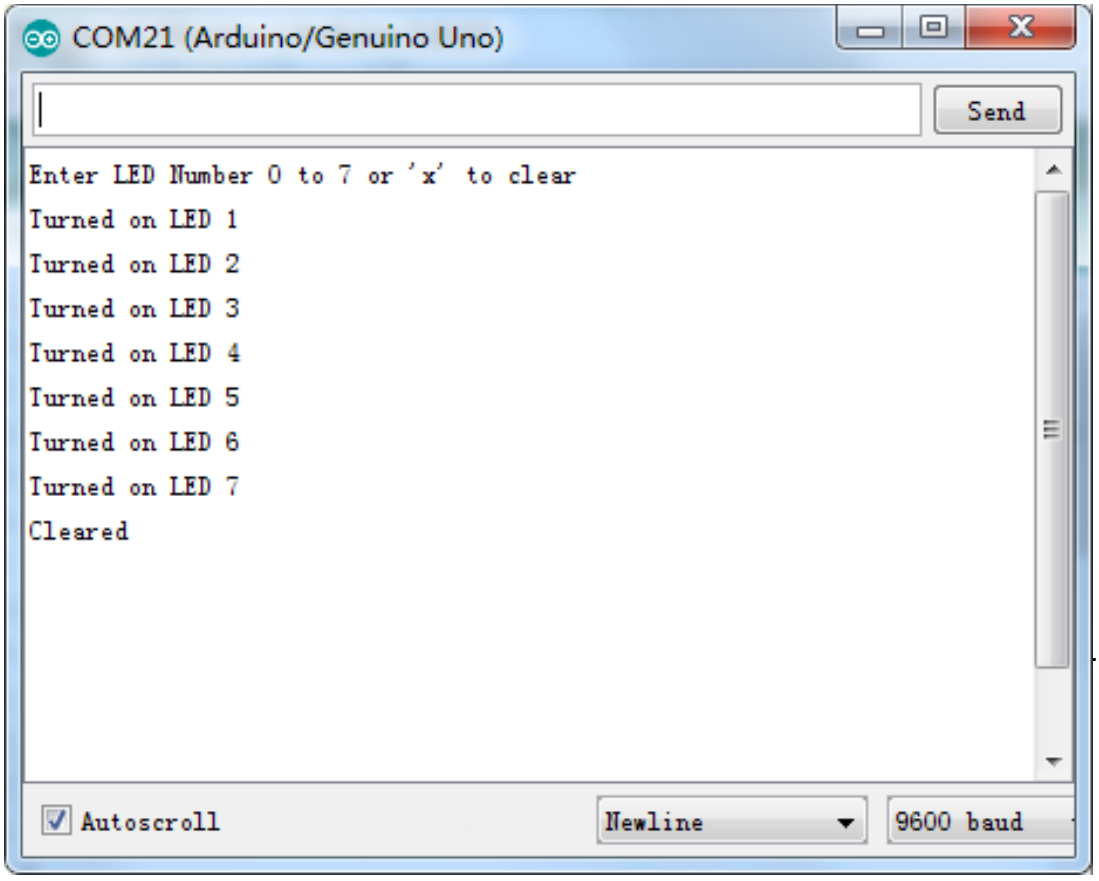


Voici ce que vous devez observer:



Vous êtes invités à entrer un chiffre ou la lettre *x* puis valider.

Envoyer un chiffre allumera la led correspondante, envoyer x les éteindra toutes.



Code

Ouvrez le sketch “Leçon 25 The Serial Monitor” et Téléversez-le sur la carte UNO R3. Comme vous pouvez vous y attendre, le code est très proche du précédent. Nous allons simplement nous attarder sur les instructions supplémentaires.

```
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 initialise la communication série, de sorte que la carte UNO R3 peut envoyer des commandes via la connexion USB. La valeur 9600 est appelée «débit en bauds» de la connexion. Cela correspond à la vitesse d’échange entre l’ordinateur et la carte.

La ligne commençant par 'while' garantit qu'il y a quelque chose à l'autre extrémité de la connexion USB pour l'Arduino avant de commencer à envoyer des messages. Sinon, le message peut être envoyé, mais pas affiché. Cette ligne n'est réellement nécessaire que si vous utilisez une carte Leonardo car la carte UNO R3 réinitialise automatiquement la carte lorsque vous ouvrez le moniteur série, alors que cela ne se produit pas avec la Leonardo.

La dernière des nouvelles lignes dans 'setup' envoie le message que nous voyons en haut du moniteur série.

```
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 dans la boucle se trouve dans une déclaration «if». Donc, à moins que l'appel à la fonction Arduino intégrée 'Serial.available ()' soit 'vrai', rien d'autre ne se produira.

Serial.available () renverra 'true' si les données ont été envoyées à l'UNO et sont prêtes à être traitées. Les messages entrants sont contenus dans ce qu'on appelle un tampon et Serial.available () renvoie true si ce buffer n'est pas vide.

Si un message a été reçu, il est sur la ligne de code suivante:

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

Cette instruction lit le caractère suivant du tampon et le supprime du tampon. Il l'affecte également à la variable 'ch'. La variable 'ch' est du type 'char' qui signifie 'caractère' et, comme son nom l'indique, détient un caractère unique.

Si vous avez suivi les instructions dans l'invite située en haut du Serial Monitor, ce caractère sera soit un chiffre d'un seul chiffre compris entre 0 et 7, soit la lettre «x». L'instruction 'if' sur la ligne suivante vérifie si c'est un seul chiffre en voyant si 'ch' est supérieur ou égal au caractère '0' et inférieur ou égal au caractère '7'. Il semble étrange de comparer des variables de cette façon, mais c'est parfaitement acceptable.

Chaque caractère est représentée par un nombre unique, appelé sa valeur ASCII.

Cela signifie que lorsque nous comparons des caractères en utilisant <= et> = c'est en fait les valeurs ASCII qui ont été comparées.

Si le test passe, nous arrivons à la ligne suivante:

```
int led = ch – '0';
```

Maintenant, nous interprétons l'arithmétique sur les charactères! Nous soustrayons le chiffre '0' de n'importe quel chiffre saisi. Donc, si vous avez tapé '0', alors '0' - '0' sera égal à 0. Si vous avez tapé '7', alors '7' - '0' sera égal au nombre 7 car il s'agit en fait des valeurs ASCII utilisées dans la soustraction.

Puisque nous connaissons le nombre de LED que nous voulons allumer, il suffit de définir ce bit dans la variable 'leds' et de mettre à jour le registre à décalage.

```
bitSet(leds, led);
updateShiftRegister();
Serial.print("Turned on LED ");
Serial.println(led);
```

La première ligne utilise Serial.print plutôt que Serial.println. La différence entre les deux est que Serial.print ne démarre pas une nouvelle ligne après l'affichage, ce qui est dans son paramètre. Nous utilisons cela en première ligne, car nous affichons le message en deux parties. Tout d'abord, le libellé général: «LED allumée», puis le nombre de LED.

Le nombre de LED est maintenu dans une variable 'int' plutôt que d'être une chaîne de texte. Serial.print peut prendre soit une chaîne de texte entre guillemets doubles, soit un «int» ou, en fait, pratiquement n'importe quel type de variable.

Après la déclaration «if» qui gère le cas, lorsqu'un seul chiffre a été géré, il y a une deuxième instruction «if» qui vérifie si «ch» est la lettre «x».

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

Leçon 10 Photocell

But de la leçon

Dans cette leçon, vous allez apprendre à mesurer l'intensité de la lumière en utilisant une entrée analogique. Vous allez utiliser le montage réalisé à la leçon précédente et utiliser vos nouvelles connaissances pour contrôler le nombre de leds allumées.

Matériel nécessaire:

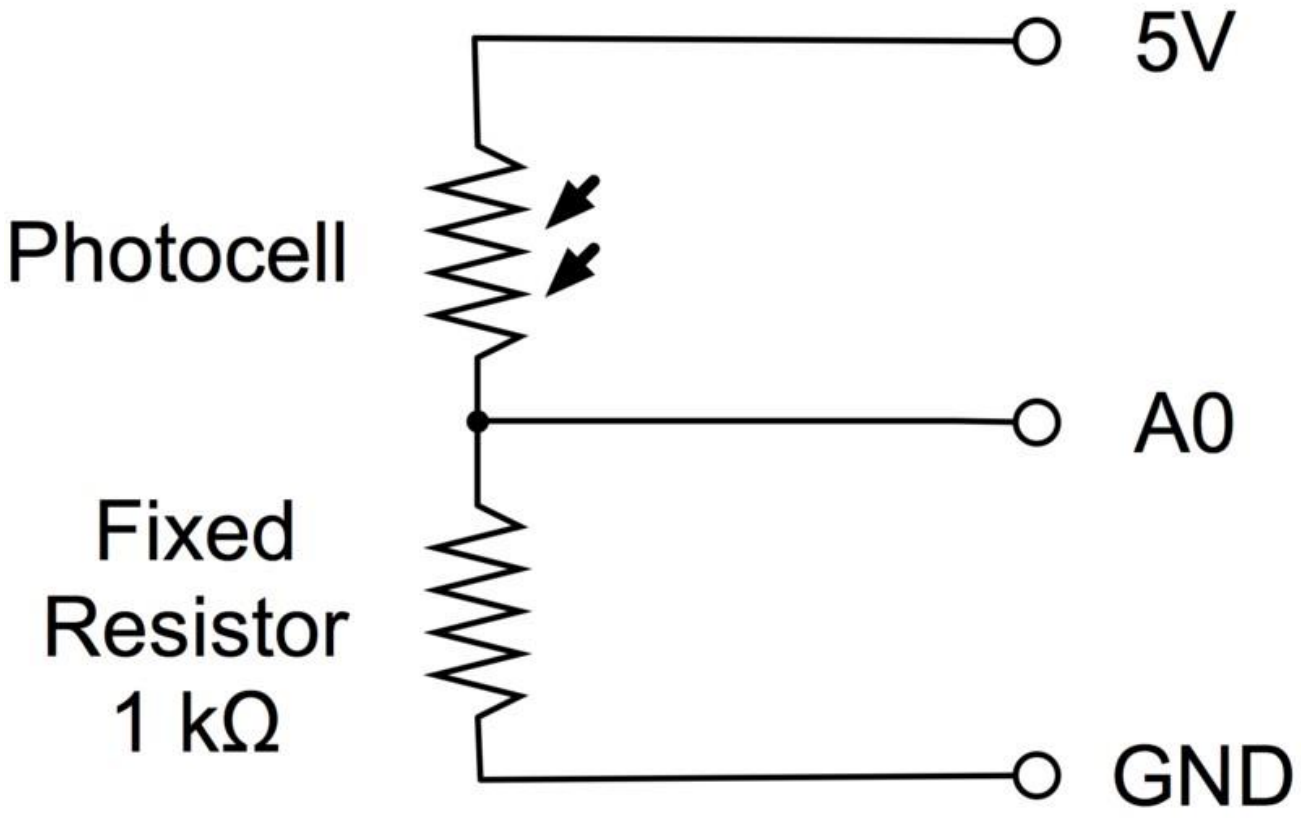
(1) x Elegoo Uno R3
(1) x Planche prototype
(8) x Leds
(8) x Résis220 ohm resistors
(1) x 1k ohm resistor
(1) x 74hc595 IC
(1) x Photorésistance (Photocell)
(16) x Câbles mâle-mâle

Présentation du composant

PHOTOCELL:

La photocell utilisée est appelée "résistance dépendante de la lumière" ou light dependent resistor (LDR). Comme son nom l'indique la valeur de la résistance dépend de la quantité de lumière reçue par le composant.

Cette résistance prend une valeur de 50 kΩ en obscurité et varie jusqu'à 500 Ω en pleine lumière. Pour convertir cette variation de valeur en variation de courant, il faut bien entendu alimenter la résistance.


5V
Photocell
A0
Fixed
Resistor
1 kΩ
GND

Connection

Schéma de câblage

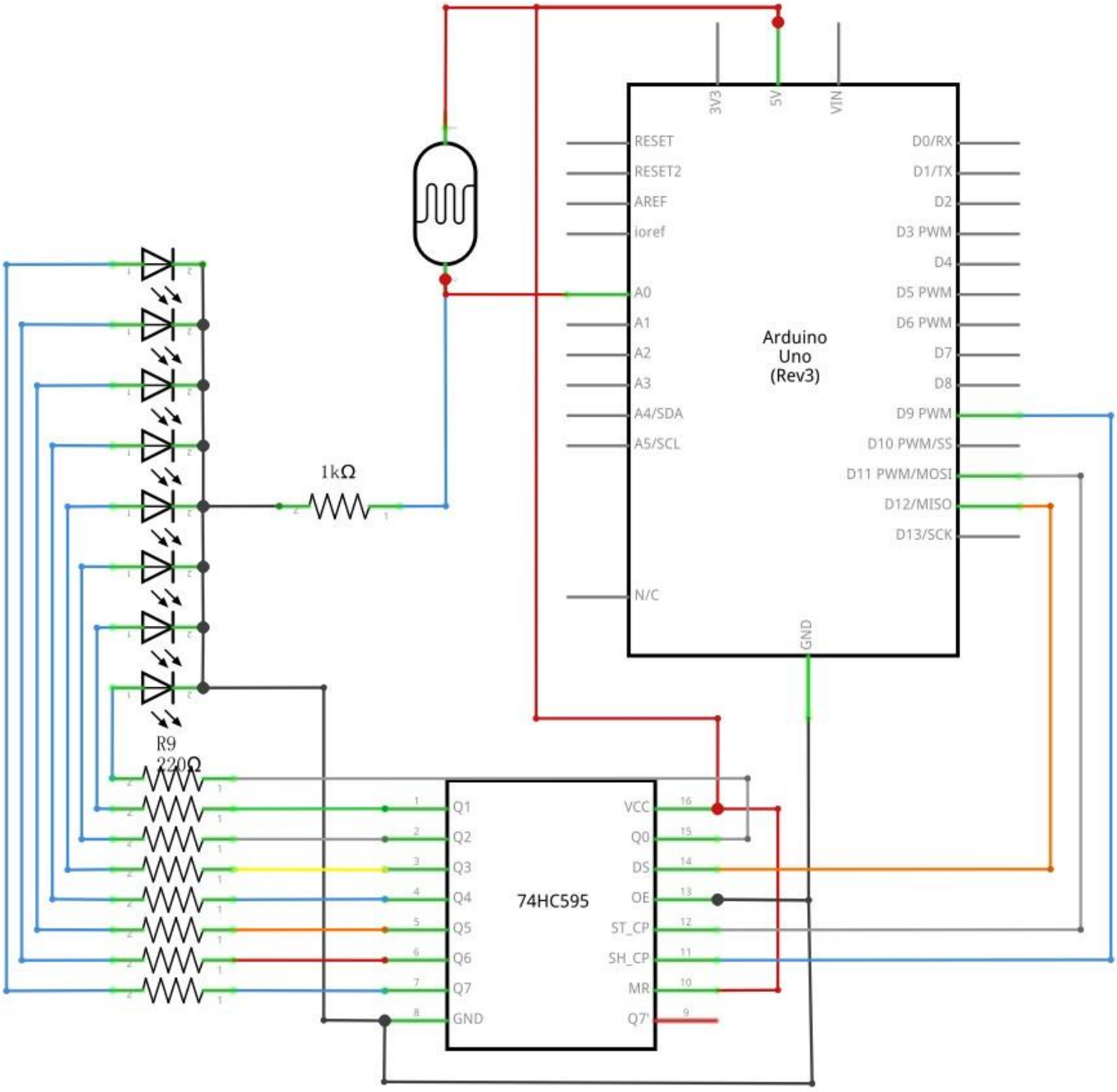
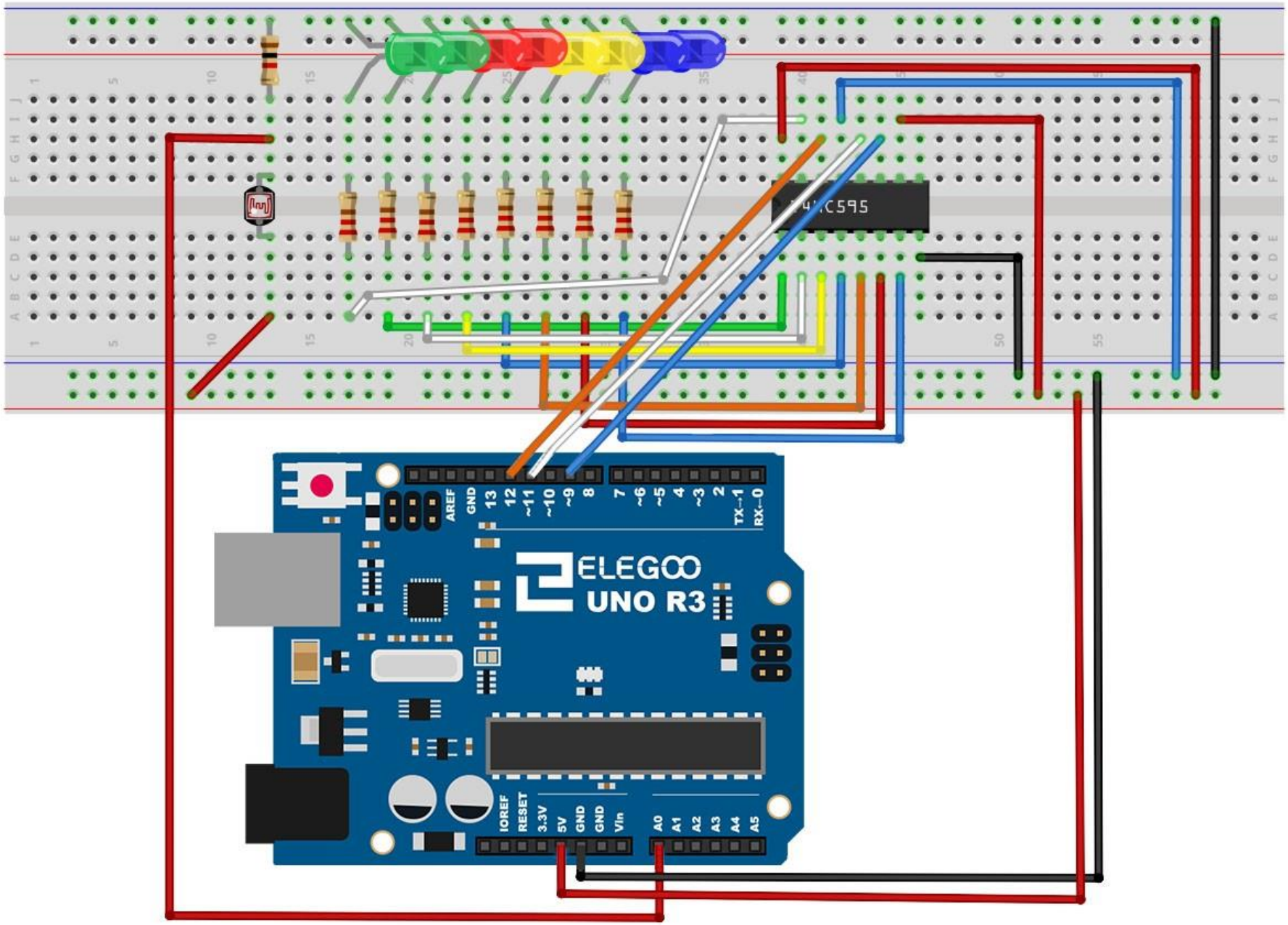


Diagramme de câblage



Code

Après avoir réalisé le schéma de câblage, ouvrez le sketch “Leçon 26 Photocell” et Téléversez-le sur la carte UNO.

La première chose à remarquer sur le code est qu’il y a la déclaration de la pin qui va permettre de mesure la valeur de la photorésistance : 'lightPin'.

L’autre changement est l’ajout de l’instruction qui calcule le nombre de leds allumées:

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

Cette fois, nous divisons la valeur brute mesurée par 57 plutôt que 114. Cela permet de créer 9 tranches : de 0 led à 8 leds allumées. Ce facteur est donnée par la valeur de la résistance fixe de 1kΩ, la valeur brute lue sera 1023 / 2 = 511.

Illustration