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.

Principe

Matériel nécessaire: (1) x la carte UNO R3

ELEGOO

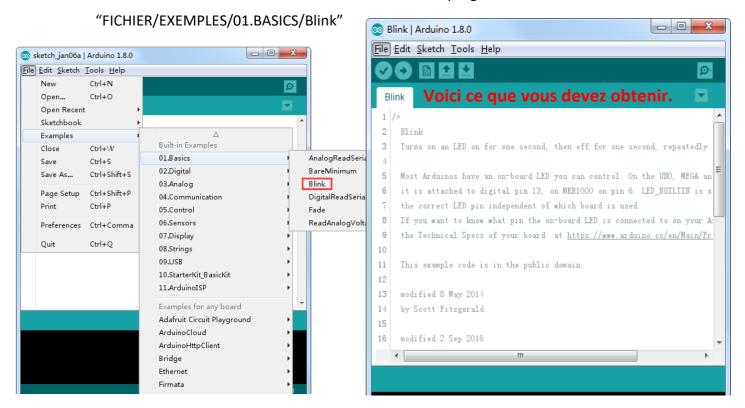
La carte UNO R3 possède deux rangées de connecteurs le long de ses extrémités qui sont utilisés pour brancher une vaste gamme de composants électroniques ou des cartes d'extensions (appelées shields) qui augmentent ses capacités.

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.

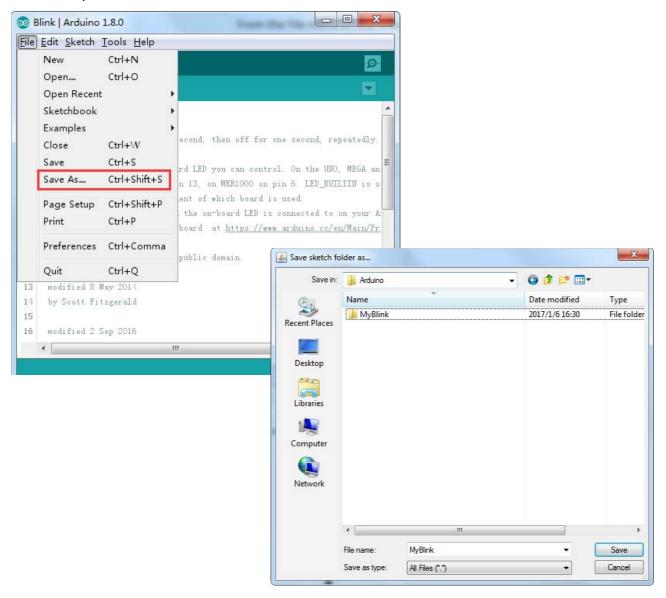
Dans cette leçon, nous allons reprogrammer la carte UNO R3 avec notre propre programme « BLINK », ce qui va nous permettre notamment de changer la fréquence de clignotement de la LED.

Ouvrez le sketch "BLINK" dans l'environnement de programmation via le menu

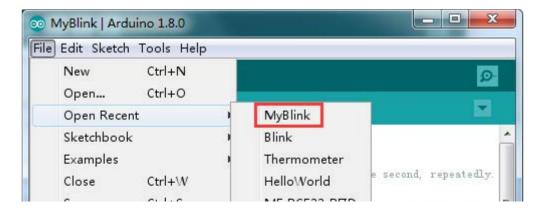


Comme vous allez apporter des modifications à ce sketch, la première étape consiste à l'enregistrer dans votre répertoire personnel, les sketchs exemples étant en lecture seule.

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

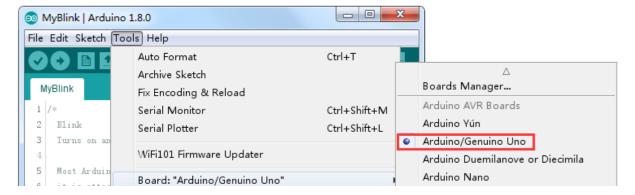


Vous venez d'enregistrer le fichier dans votre répertoire.

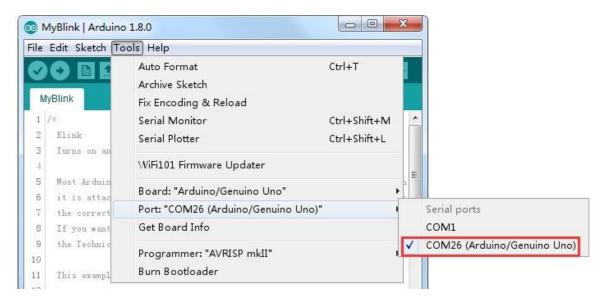


Vous pouvez accéder à vos fichiers récents de la même manière que la plupart de vos autres logiciels.

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

```
// the loop function runs over and over again forever

void loop() {

digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the volt

delay(500) // wait for a second

digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the vold

delay(500) // wait for a second

// wait for a second

// wait for a second

// wait for a second
```

Leçon 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 Uno R3

(1) x LED rouge de 5mm

(1) x résistance 220 ohm

(1) x résistance 1 kohm

(1) x résistance 10 kohm

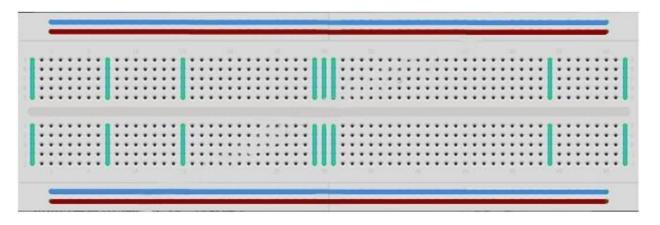
(2) x câbles mâle-mâle

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:

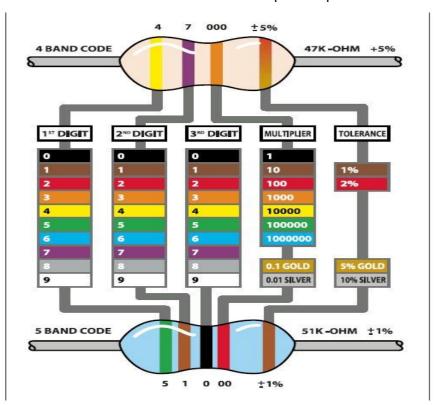
Comme leur nom le suggère, les résistances s'opposent au passage du flux d'électricité. Plus la valeur est grande plus la résistance l'est aussi. Vous allez pouvoir calibrer la brillance de la led en jouant sur la valeur de la résistance.



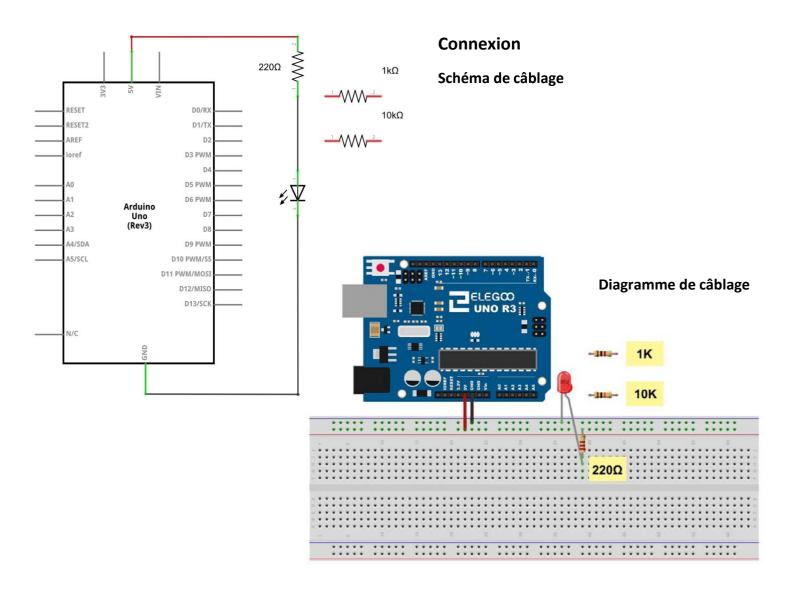
Mais tout d'abord, quelques informations supplémentaires...

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 : $k\Omega$ (1,000 Ω) et $M\Omega$ (1,000,000 Ω).

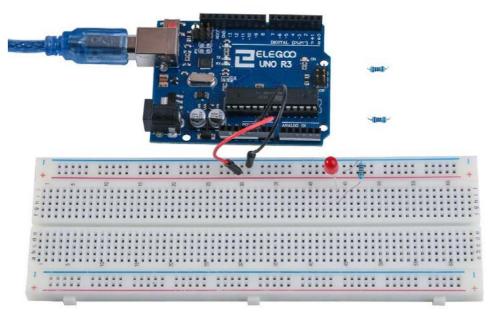
Dans cette leçon, vous allez utiliser trois valeurs de résistance : 220Ω , $1k\Omega$ and $10k\Omega$. C'est grâce aux anneaux de couleurs et au code associé que l'on peut connaître la valeur d'une résistance.



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.



Illustration



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 $1k\Omega$ et à peine brillante avec une résistance de $10 k\Omega$.

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

Leçon 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 Uno R3

(1) x Planche prototype

(4) x Câbles mâle-mâle

(1) x led RGB

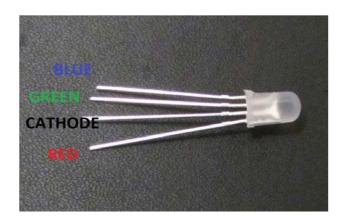
(3) x résistance 220 ohms

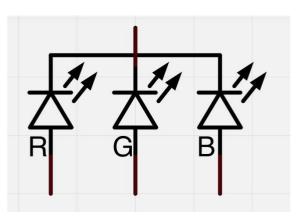
Présentation des composants

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.



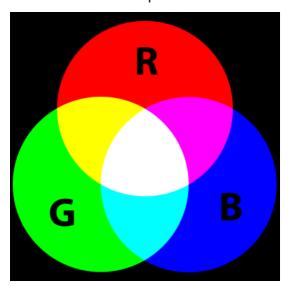


Il faut connecter une sortie analogique de la carte UNO R3 sur les pattes ROUGE/VERT/BLEU et une masse sur la patte (GND)

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.

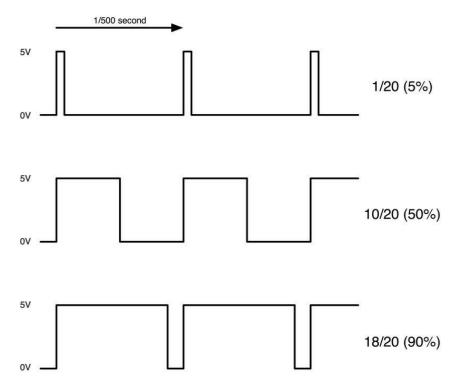


Théorie (PWM)

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

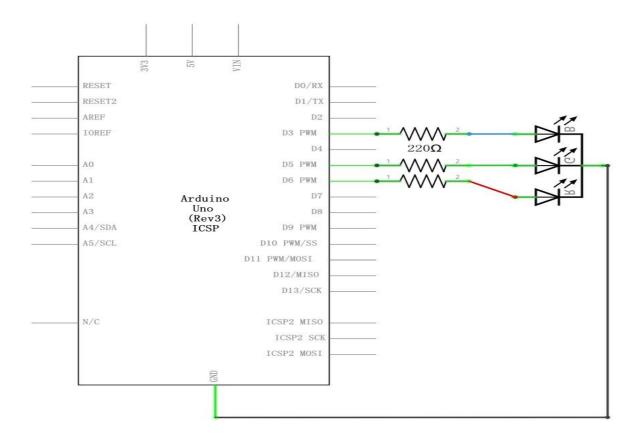
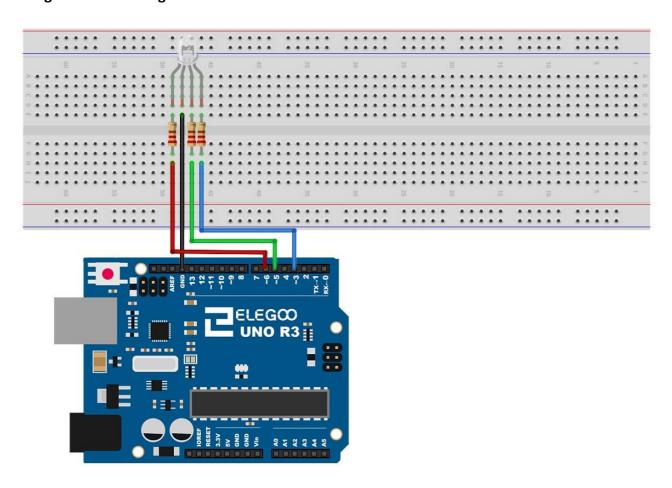
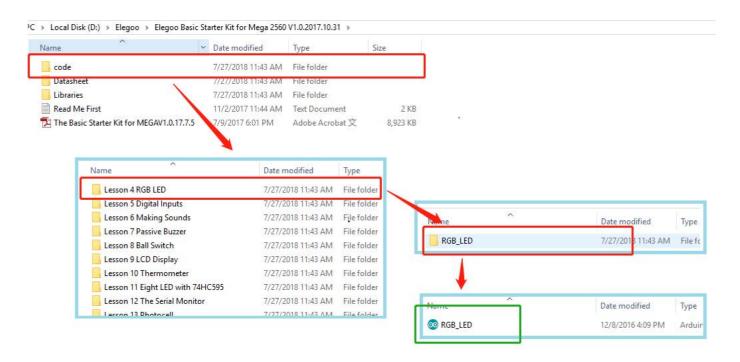


Diagramme de câblage



Code

After wiring, please open the Sketch in folder path: Tutorial > English > code > Lesson RGB LED > RGB LED, and click UPLOAD to upload the program.



See Lesson 2 for details about program uploading if there are any errors.

The sketch starts by specifying which pins are going to be used for each of the colors:

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

The next step is to write the 'setup' function. As we have learnt in earlier lessons, the setup function runs just once after the Arduino has reset. In this case, all it has to do is define the three pins we are using as being outputs.

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

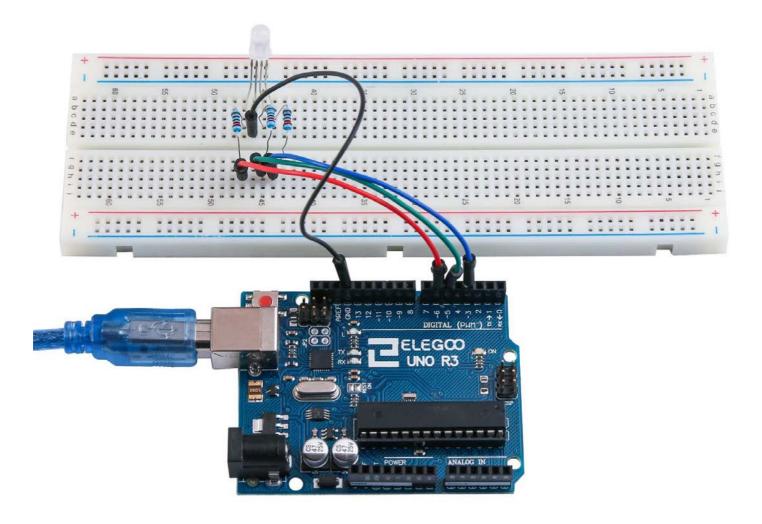
varie entre 0 et 255. La fonction appelle ensuite analogWrite pour définir la brillance de chaque led.

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 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.

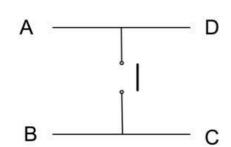
Matériel nécessaire:

Présentation du composant

- (1) x 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
- **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.

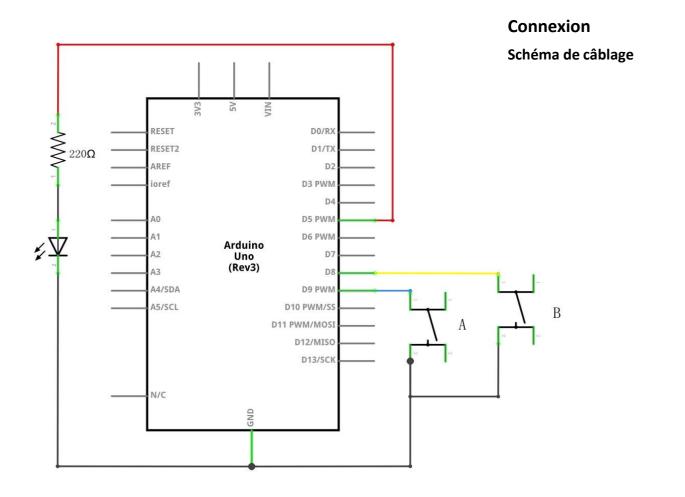
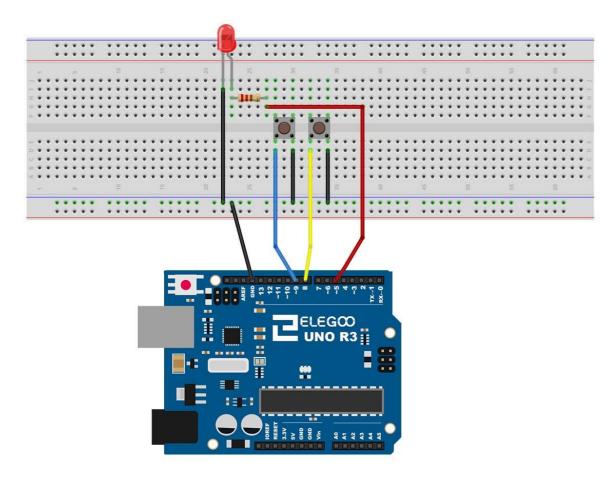


Diagramme de câblage



Code

Après avoir réalisé le câblage, ouvrez le sketch "Leçon Digital Inputs" puis téléversez le code sur la carte UNO R3 comme présenté lors de la leçon "BLINK"

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);
```

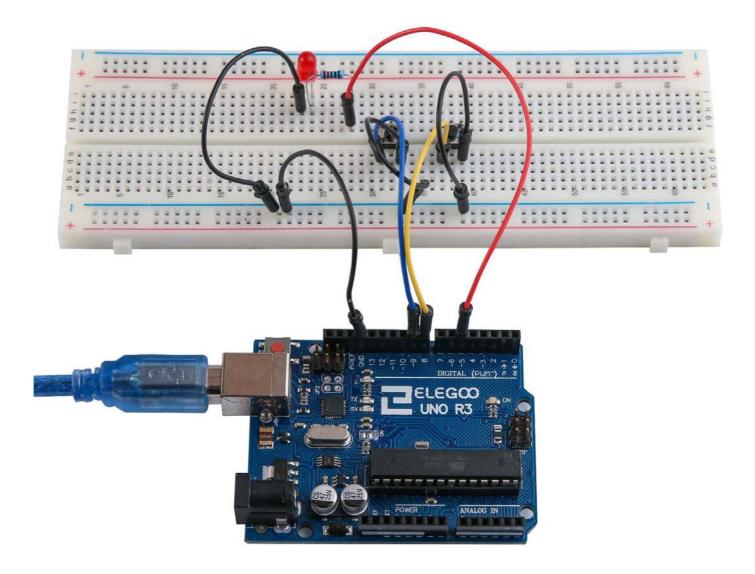
Petite particularité, les entrées sont définies en tant qu'entrées avec résistance de PULLUP. Pourquoi cette subtilité ? Comme vous pouvez le constater sur le schéma de câblage, le fait de presser un bouton met la pin associée à la masse. Mais lorsque le bouton n'est pas pressé, il peut subsister des 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 Active buzzer

But de la leçon

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

Matériel nécessaire:

- (1) x Uno R3
- (1) x Active buzzer
- (2) x Câbles Mâle-Femelle





Présentation du composant

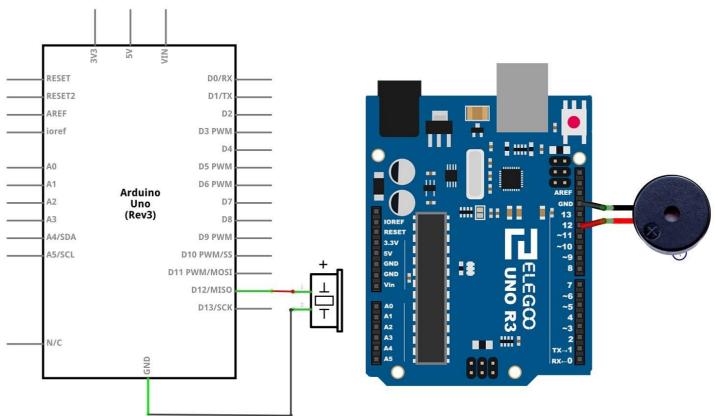
BUZZER:

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 Making Sounds" et Téléversez le code sur la carte UNO R3 comme expliqué plus haut.

Leçon Passive Buzzer

But de la leçon

Dans cette leçon, vous allez apprendre comment utiliser un buzzer passif.

Le but est de générer des notes durant chacune 0.5 seconde Do (523Hz), Ré (587Hz), Mi (659Hz), Fa (698Hz), So (784Hz), La (880Hz), Si (988Hz).

Matériel nécessaire:

- (1) x Uno R3
- (1) x Buzzer passif
- (2) x Câbles Mâle-Femelle

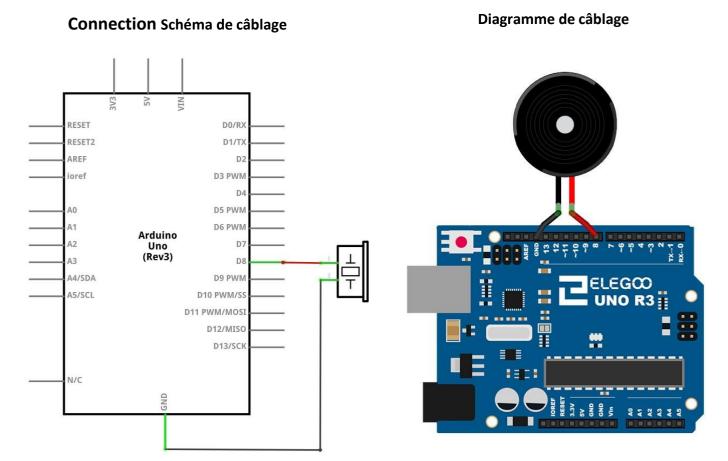


Présentation du composant

Passive Buzzer:

Le principe de fonctionnement d'un buzzer passif est d'utiliser un signal PWM. Modulé à la bonne fréquence, la vibration génère différents sons. Exemple, envoyer une pulsation de 523Hz donne un Do, 587Hzdonne un Ré...

Attention: il n'est pas possible d'utiliser la fonction analogWrite() de la carte UNO car le signal est fixe et vaut 500Hz. Il va falloir faire autrement, c'est ce que nous allons voir dans le code qui suit.



Code

Après avoir réalisé le câblage, ouvrez le sketch "Leçon Passive Buzzer" et Téléversez- le sur la carte UNO R3. Attention : veillez d'abord à bien avoir installé la bibliothèque <PITCHES>

Leçon 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 Uno R3
- (1) x capteur Tilt Ball switch
- (2) x Câbles Mâle-Femelle



Présentation du composant

Capteur « Tilt »:

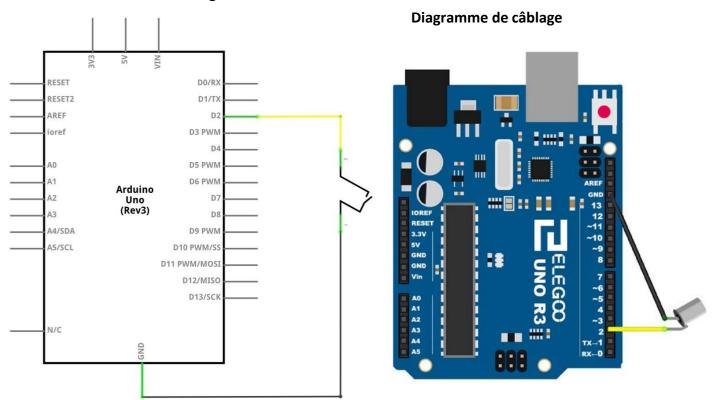
Ce capteur vous permet de détecter une orientation ou inclinaison. Il est petit, pas cher est consomme très peu d'énergie.

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



Code

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

Leçon Servomoteur

But de la leçon

Le servomoteur est un type de moteurs qui peut seulement tourner de 180 degrés. Il est contrôlé par l'émission de pulsations électriques depuis la carte UNO R3. La pulsation donne au moteur la position qu'il doit prendre.

Le moteur a trois connexions. Le fil brun (masse), le fil rouge (positif), le fil orange (signal – à connecter sur la pin 9 de la carte UNO R3)

Matériel nécessaire:

- (1) x Uno R3
- (1) x Servo (SG90)
- (3) x Câbles mâle-mâle



Le composant SG90

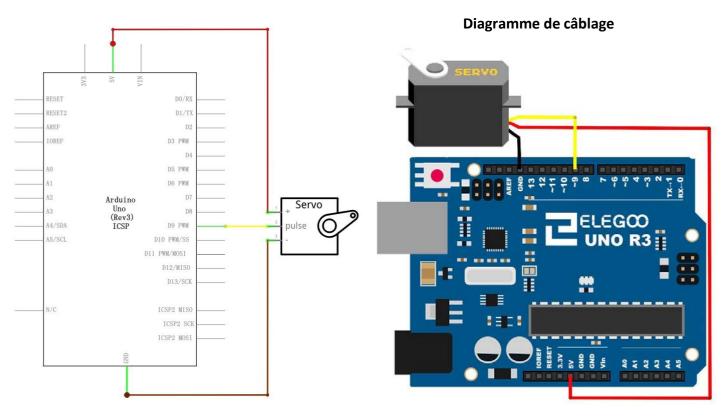
Longueur de câble : 25cm Couple (à 4.8V): 1.6kg/cm Temperature : -30~60'C

Voltage: 3.5~6V

Dimensions: 3.2cm x 3cm x 1.2cm

Poids : 134 g

Connection Schéma de câblage



Code

Après avoir réalisé le câblage, ouvrez le sketch « Leçon Servo » et téléversez le code sur la carte comme expliqué à la leçon 2.

Attention de bien avoir installé la bibliothèque <SERVO>,

Leçon Ultrasonic Sensor Module

But de la leçon

Le capteur ultrasons est parfait pour tous les projets nécessitant de faire de la mesure de distances, en évitement d'obstacles par exemple.

Le HC-SR04 est un produit peu onéreux et facile à utiliser.

Matériel nécessaire:

- (1) x Uno R3
- (1) x Capteur Ultrasons
- (4) x Câbles Mâle-Femelle



Présentation du composant

Module Ultrasons

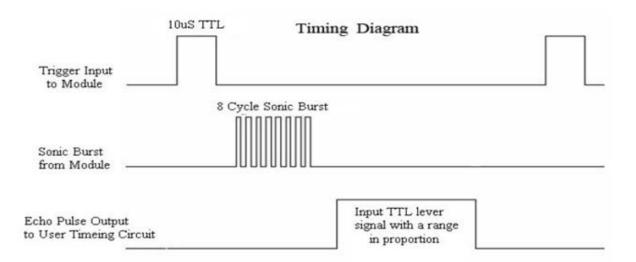
Le capteur permet une mesure d'un objet situé à une distance allant de 2cm à 4m et fourni une mesure avec une précision de 3mm.

Principe de fonctionnement:

- (1)Un signal de déclenchement (trigger) est émis pendant 10µs
- (1)Le module émet 8 signaux (minimum) à 40kHz.
- (2)Le module se met à l'écoute d'un signal de retour
- (3)Le temps entre l'émission et la réception est le temps nécessaire au signal pour faire l'aller et le retour vers l'objet qui a réfléchi celui-ci.

On a donc:

Distance = (Temps_mesuré * Vitesse_du_son)/2 (340m/s pour la vitesse du son)



Connection

Schéma de câblage

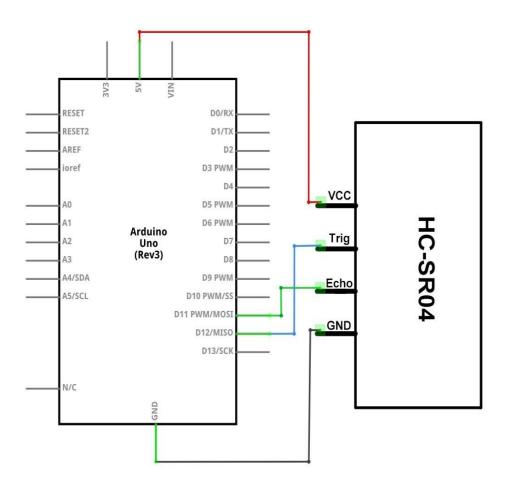
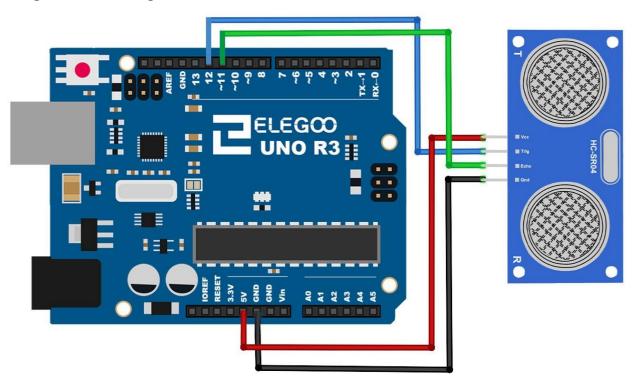


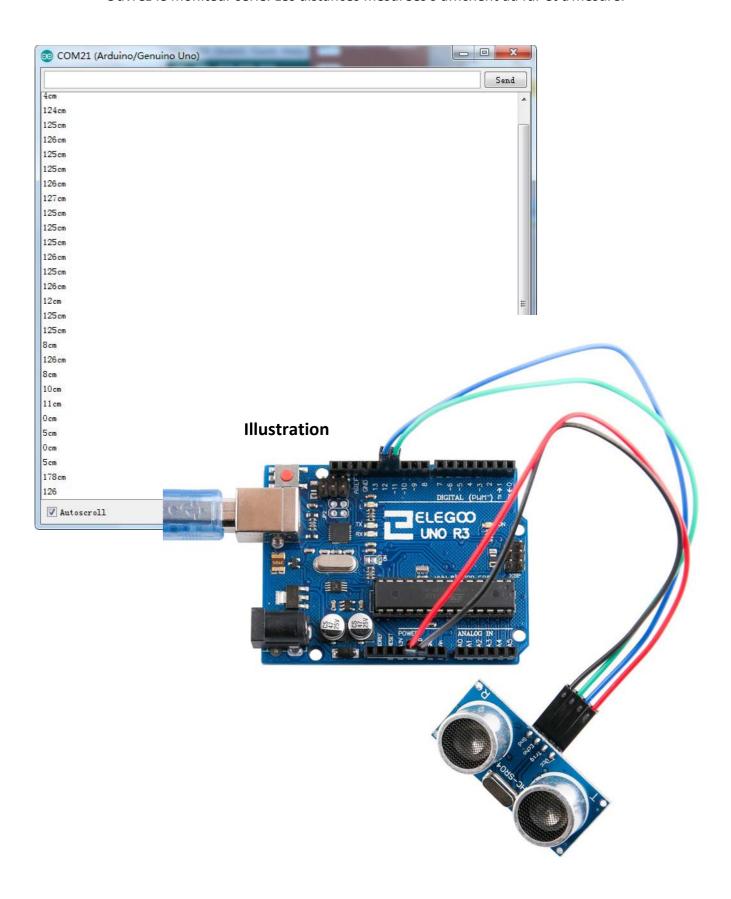
Diagramme de câblage



Code

Ouvrez le sketch "Leçon Ultrasonic Sensor Module" et Téléversez-le sur la carte UNO R3.

Ouvrez le moniteur série. Les distances mesurées s'affichent au fur et à mesure.



Leçon DHT11 Temperature and Humidity Sensor

But de la leçon

Dans cette leçon, vous allez apprendre à utiliser le capteur DHT11 qui permet une mesure de température et d'humidité.

C'est un capteur fiable qui permettra de répondre à la plupart des projets nécessitant un suivi de température et/ou d'humidité.

L'emploi d'une bibliothèque permet en plus une implémentation simple du code dans vos sketchs.

Matériel nécessaire:

- (2) x Uno R3
- (1) x DHT11 Temperature and Humidity module
- (3) x Câbles Mâle-Femelle

Le DHT11 (Digitale Température Humidité 11) est un capteur qui génère un signal

digital en sortie codant une valeur de température et d'humidité mesurée en temps réel. Cette technologie est employée car elle permet de produire une mesure fiable sur le long terme. Il contient en plus de ces capteurs un microcontrôleur 8-bit. Encore une fois, il est utilisé dans un grand nombre d'applications qui nécessitent de moduler le fonctionnement d'un appareil en fonction de conditions de température ou d'humidité.

Paramètres: Humidité

relative: Résolution:

16Bit Répétabilité:

±1% RH

Précision: At 25 °C ±5% RH

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

1m / s air 6s

Hystérésis: <± 0.3% RH Stabilité: <± 0.5% RH / an Température: Résolution:

16Bit

Répétabilité: ±0.2℃

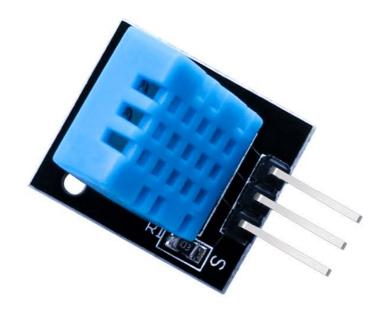
Gamme de températures: At 25° C $\pm 2^{\circ}$ C

Temps de réponse: 1 / e (63%) 10S

Alimentation: DC $3.5\sim5.5V$ Courant : 0.3mA standby 60μ A

Etalonnage: 2 secondes

Présentation du composant Capteur DHT11:



Connection Schéma de câblage

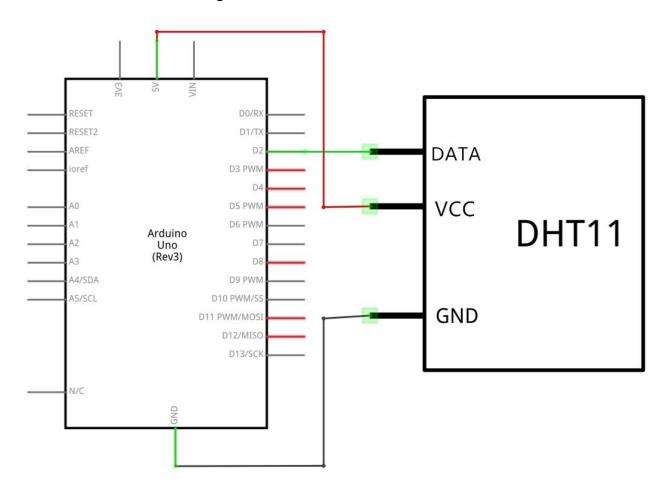
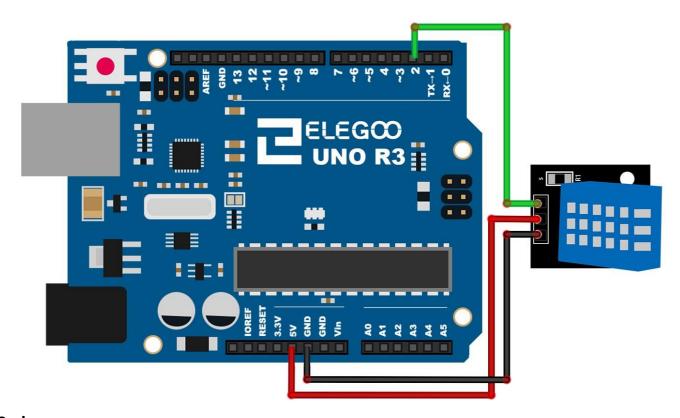


Diagramme de câblage

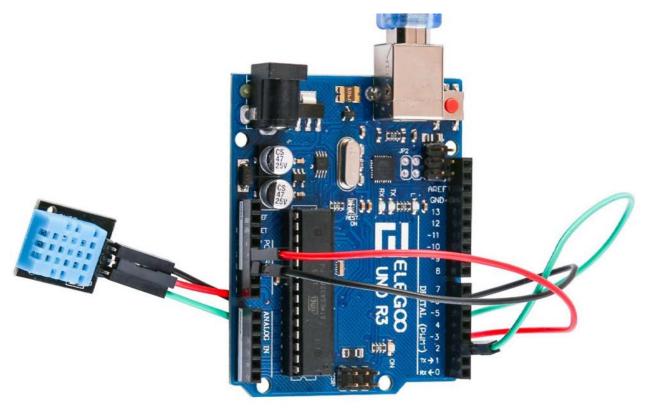


Code

Ouvrez le sketch "Leçon 12 DHT11 Temperature and Humidity Sensor" et Téléversez- le sur la carte UNO R3.

Avant toute chose, veillez à avoir installé la bibliothèque <SimpleDHT>

Illustration



Ouvrez le moniteur série sur votre ordinateur, les mesures s'affichent.

