

## **CORDÉE DE LA RÉUSSITE**

### **TUTO ELECTRICITÉ**

Dans ce tutoriel, nous allons réaliser un jeu de rapidité. Le but étant d'appuyer plus vite que son adversaire sur un bouton lorsqu'une lampe s'allume.

<b>I - Choisir les composants</b>	<b>2</b>
Histoire de l'Arduino :	2
Comment fonctionne "Arduino".	2
<b>II - La DEL RVB</b>	<b>3</b>
<b>III - Le bouton</b>	<b>7</b>
<b>IV - Le projet final</b>	<b>8</b>

## **I - Choisir les composants**

Comme pour tous les projets, il faut savoir ce que l'on veut faire (dans notre cas un jeu de rapidité) et ce que l'on va utiliser pour le faire. Cependant, comme vous n'êtes pas (encore) des ingénieur.e.s on va vous guider pour ce projet. Pour commencer vous aurez besoin de :

Dans ce projet on va donc utiliser :

- 1 Arduino et son câble de branchement
- 1 plaquette d'essai
- Des câbles pour plaquette
- Des résistances ( 3\*220 Ohms et 2\*15 kOhms)
- 3 DEL RVG (ou led rgb en anglais)
- 2 boutons poussoirs

Votre première mission va être de les trouver et de vérifier que vous avez bien tout.

### **Histoire de l'Arduino :**

L'Arduino est à l'origine un projet d'étudiants de l'école de Design d'Interaction<sup>1</sup> d'Ivrea en Italie. Au début des années 2000, les outils de conception de projets dans le domaine du design d'interaction étaient onéreux, proches d'une centaine d'euros. Ces outils étaient pour la plupart conçus pour le domaine de l'ingénierie et de la robotique. Maîtriser et utiliser ces composants demandait beaucoup de temps et d'apprentissage et ralentissait fortement le processus de création pour les jeunes étudiants.

Vient alors l'idée de créer une plateforme plus abordable et plus simple à utiliser à une équipe d'étudiants et de professeurs qui finirent par concevoir la toute première Arduino en 2005. Entièrement open source, l'Arduino présente l'avantage d'être multiplateforme et d'être en perpétuelle optimisation par la communauté d'utilisateurs.

### **Comment fonctionne "Arduino".**

Il faut tout d'abord savoir que "Arduino" désigne deux choses. D'une part un logiciel qui permet d'écrire du code et de vérifier qu'il n'y est pas d'erreurs et d'autre part une carte électronique qui va exécuter ce code et interagir avec l'extérieur.

Il est donc important de travailler avec la synergie des deux pour réussir ses projets.

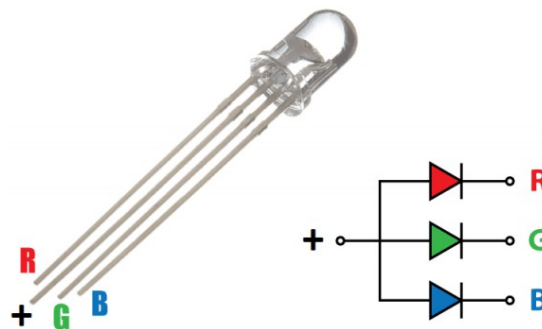
---

<sup>1</sup> Le design d'interaction définit la façon dont un objet ou un service va se comporter dans le dialogue avec l'utilisateur.

## II - La DEL RVB

Une DEL RVB (ou Diode ElectroLuminésante de couleur Rouge Verte et Blue, aussi applé LED RGB en anglais) est un assemblage de 3 lampes avec 1 verte, 1 rouge et 1 bleu.

Pour brancher ce type de DEL on doit suivre un schéma, voici celui de notre DEL. Arrivez vous à identifier la broche "+" sur vos DEL ?



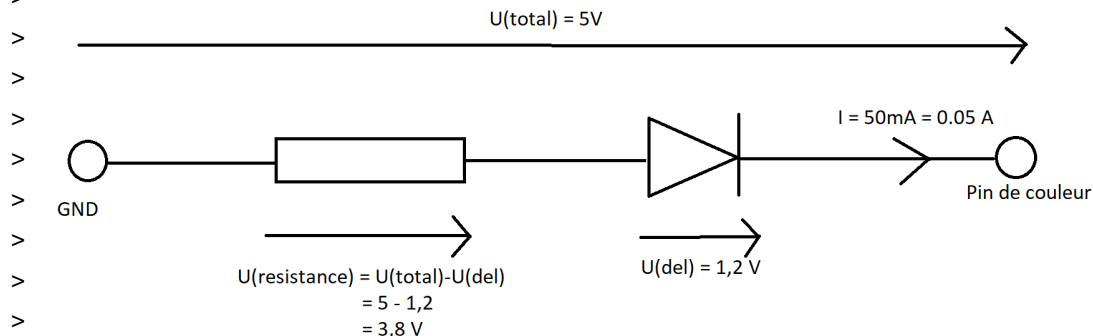
Attention, on ne peut pas laisser une DEL alimentée directement par l'Arduino, elle brûlerait instantanément, au sens propre du terme. Il faut donc une résistance. C'est là que l'ingénieur électricien arrive pour dimensionner les composants avec les fiches techniques (ou datasheets) et les outils à sa disposition, comme les formules mathématiques mais aussi les logiciels. Pour les plus motivés, voici une section "Pour aller plus loin" qui vous explique comment calculer la valeur de résistance, pour les autres, rendez-vous à la page suivante 😊.

### > Pour aller plus loin

>

> Dans ce schéma on retrouve de gauche à droite notre pin de masse (le 0 volt), notre  
> résistance (le rectangle), la del (le triangle avec une barre verticale), une flèche pour  
> donner le sens du courant (ici avec  $I = 50 \text{ mA} = 50 \text{ milliampère}$ , donnée par la  
> documentation de la del) et enfin le pin du choix de la couleur que l'on alimente ou non  
> en 5V (c'est la valeur du  $U(\text{total})$ ), c'est la sortie de l'arduino.

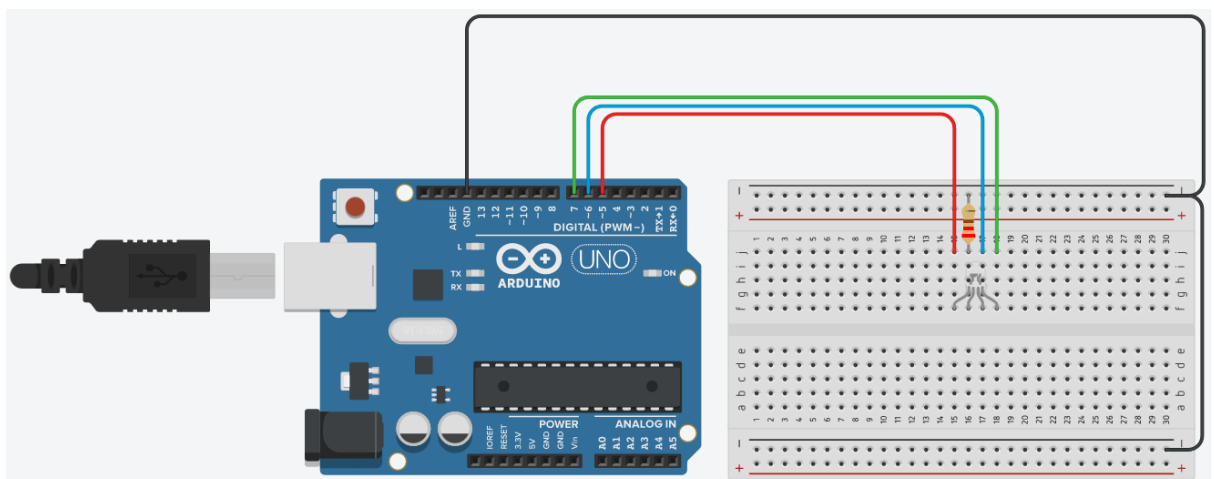
>



- > D'après la loi des mailles on sait que  $U_{Total} = U_{résistance} + U_{del}$  et donc que
- >  $U_{résistance} = U_{total} - U_{del}$ . En remplaçant par les valeurs on obtient  $U_{résistance} = 3,8V$ .
- > Ensuite on utilise la loi dans mailles avec  $U = R \times I$  donc  $R = \frac{U}{I} = \frac{3,8}{0,05} = 76\Omega$
- > Il nous faut donc une résistance de 76 Ohm pour avoir notre del au mieux de sa forme.
- > Dans notre cas on met 220 Ohm pour que la del brille moins fort (car la tension à ces bornes sera plus faible), et aussi par mesure de sécurité pour ne pas la faire brûler.

Après ce long raisonnement, on arrive à une résistance de 220 Ohm, exactement celle que l'on vous a donnée.

Il ne reste maintenant plus qu'à faire le câblage et pour ça vous pouvez suivre le schéma qui suit :



Pensez bien à tout relier, et une fois que vous êtes sûr de vous, ouvrez le logiciel "Arduino" pour commencer à coder. Ne branchez l'arduino qu'APRES avoir été validé par l'un de nous.

Pour commencer, on va te donner le code déjà fait et on va essayer de le comprendre après.

```
//Définir le numéro pour chaque entrée/sortie
#define delr 5
#define delv 7
#define delb 6

void setup()
{
  //Définir le mode de chaque entrée/sortie
  pinMode(delv, OUTPUT);
  pinMode(delr, OUTPUT);
  pinMode(delb, OUTPUT);
}

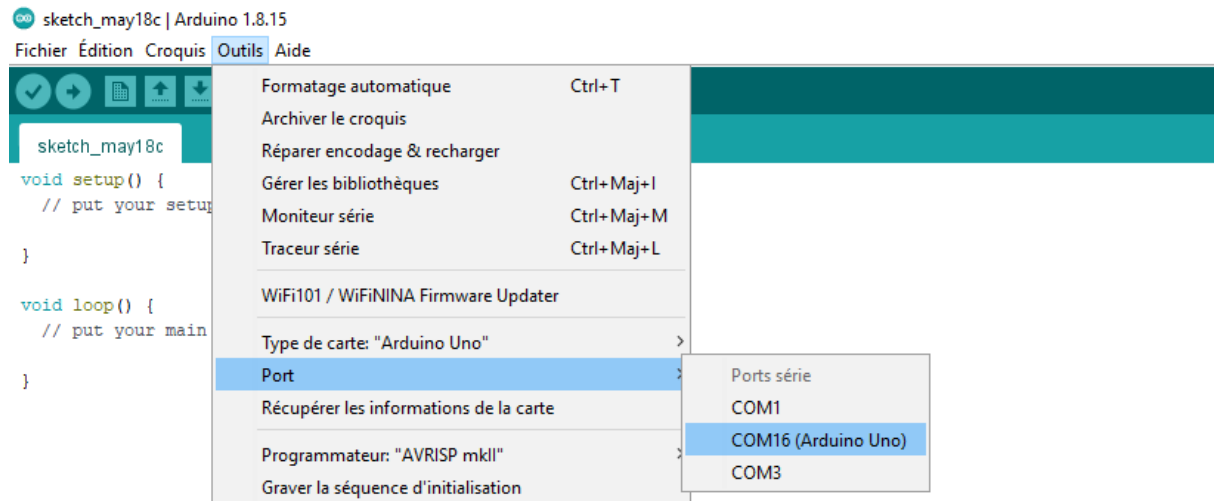
void loop()
{
  // Allumer le vert de la del centrale
  digitalWrite(delv, HIGH);
  digitalWrite(delr, LOW);
  digitalWrite(delb, LOW);

  delay(2000); // Attendre 2 secondes

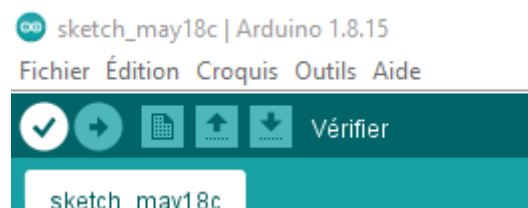
  // Allumer le rouge de la del centrale
  digitalWrite(delv, LOW);
  digitalWrite(delr, HIGH);
  digitalWrite(delb, LOW);

  delay(2000); // Attendre 2 secondes
}
```

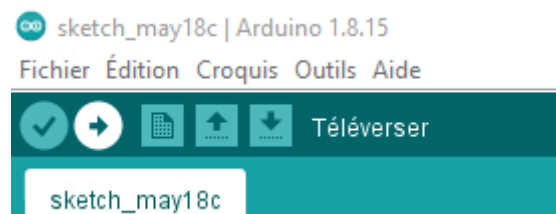
Après avoir copié/collé le texte dans arduino; et l'avoir branché; après vérification de l'un de nous, vous pouvez ouvrir "Outils -> Port -> COM (Arduino Uno)" (PS : Ce n'est pas forcément COM16, le numéro peut changer, ce n'est pas grave)



Puis cliquer sur la coche en haut à gauche pour vérifier si le code n'a pas d'erreurs



Et enfin sur la flèche juste à côté pour téléverser le code, c'est-à-dire le mettre sur la carte Arduino.



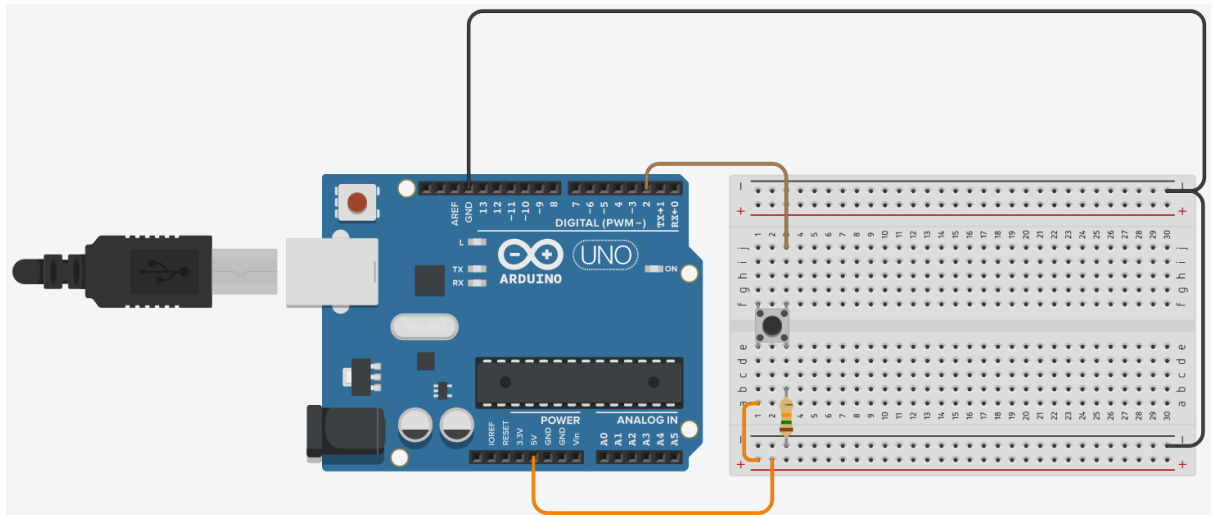
Si tout s'est bien passé votre DEL devrait se mettre à clignoter en rouge et vert.

> **Pour aller plus loin**

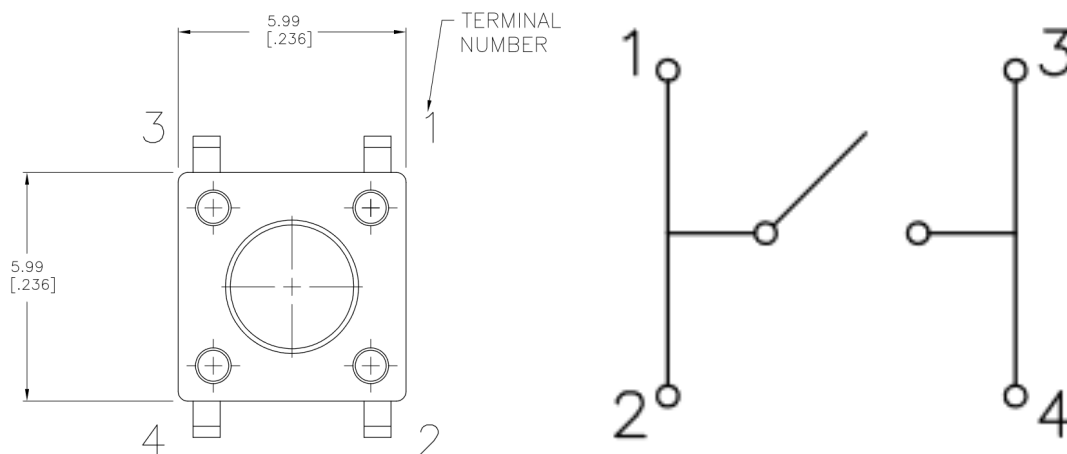
- > Pour vérifier que vous avez compris le code, essayez de faire clignoter en rouge, vert et
- > bleu, bon courage :)

### III - Le bouton

Pour cette partie on va devoir accepter quelques petites choses, comme le schéma électrique que l'on ne cherchera pas à prouver. (Pour ceux et celles qui veulent aller plus loin vous pouvez vous renseigner sur les résistances de pull-down et de pull-up) Pour la résistance vous allez prendre la bleue, elle vaut 15 000 Ohm soit 15 kOhm



Pour câbler le bouton il faut suivre son schéma sur la datasheet, voici les parties importantes. Mais attention, pour bien poser le bouton sur la plaquette d'essai il faut que le numéro 3 et 1 soit du même côté et que les numéros 4 et 2 soit de l'autre côté. Sinon les broches ne vont pas bien rentrer, il ne faut pas les plier, ça doit rentrer tout seul.



Pour notre schéma nous relierons le "5V" au numéro 2 du bouton, le "GND" ou "0V" au numéro 4 et enfin le pin de l'arduino que l'on souhaite être celui de réception du signal au numéro 3. Et oui, si vous vous posez la question, on n'utilise pas le numéro 1 du bouton dans ce projet, pas d'inquiétude si tout n'est pas relié du coup.

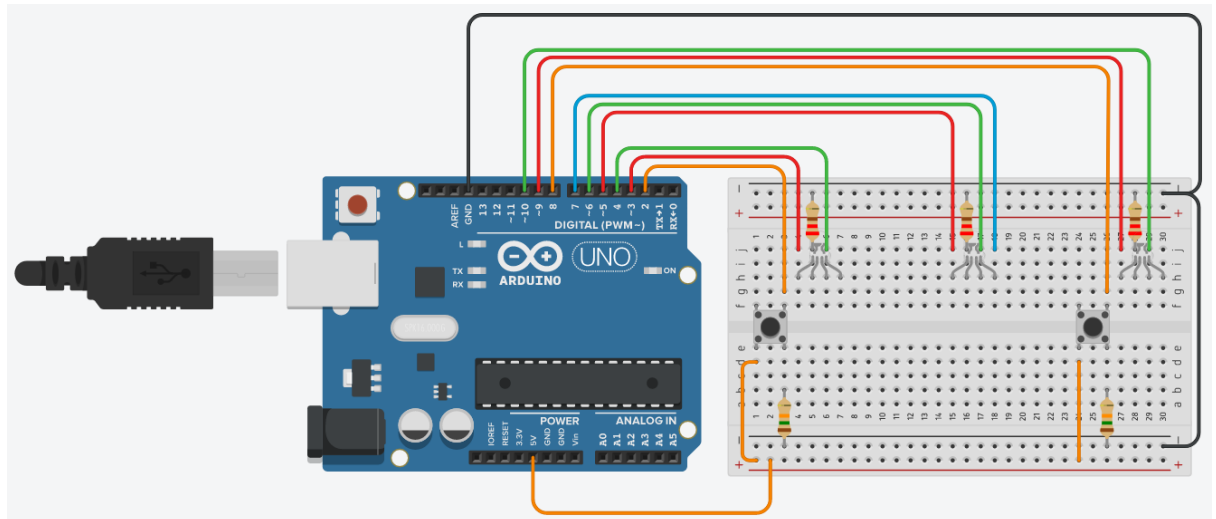
Cependant même avec tout ça il se peut que tu n'y arrives pas. Dans ce cas n'hésite pas à demander de l'aide aux intervenants :)

[Il faudrait rajouter un code où l'on teste le bouton !!!]



## IV - Le projet final

Comme on n'a pas beaucoup de temps pour créer notre propre jeu nous allons directement vous donner la solution... Enfin, seulement une partie de la solution. Voici donc le schéma électrique à reproduire.



Et pour le code nous te proposons un petit jeu de puzzle, voici toutes les briques de code dont vous avez besoin pour réaliser le petit jeu (à la page suivante). A vous de les ranger dans l'ordre en les imbriquant dans le code là où il faut. Bon courage ;)

1	//Quand le temps est passé alors allumer la del digitalWrite(delv, HIGH); //allumer chaque couleur de la del centrale digitalWrite(delr, HIGH); digitalWrite(delb, HIGH);
2	// Attendre tant que les deux boutons sont désactivé while(digitalRead(bt1) == LOW && digitalRead(bt2) == LOW){}
3	// Allumer la del centrale toutes les 1 à 3 sec SANS empêcher la triche delay(random(1000,3000)); //attendre entre 1 et 3 sec
4	// C'est gagné si l'un des deux joueurs a appuyé sur son bouton en premier if(digitalRead(bt1) == HIGH){ //Si le joueur 1 gagne alors digitalWrite(del1v, HIGH); //Allumer le vert pour le 1 digitalWrite(del2r, HIGH); // Et allumer le rouge pour le 2 }else{ // Sinon le joueur 2 à gagner digitalWrite(del2v, HIGH); // Allumer le vert pour le 2 digitalWrite(del1r, HIGH); // Et allumer le rouge pour le 1 } delay(1000); // Attendre 1 sec le temps que les joueurs remarquent le vainqueur
5	// Éteindre toutes les lumières et recommencer digitalWrite(delv, LOW); digitalWrite(delr, LOW); digitalWrite(delb, LOW); digitalWrite(del2v, LOW); digitalWrite(del1r, LOW); digitalWrite(del1v, LOW); digitalWrite(del2r, LOW);
6	//Définir le numéro pour chaque entrée/sortie #define bt1 2 #define bt2 8 #define del1r 3 #define del1v 4 #define del2r 9 #define del2v 10 #define delr 5 #define delv 7 #define delb 6
7	//Définir le mode de chaque entrée/sortie pinMode(bt1, INPUT); pinMode(bt2, INPUT); pinMode(del1v, OUTPUT); pinMode(del1r, OUTPUT); pinMode(del2v, OUTPUT); pinMode(del2r, OUTPUT); pinMode(delv, OUTPUT); pinMode(delr, OUTPUT); pinMode(delb, OUTPUT);

### > Pour aller plus loin

> Petit bonus, nous avons posé deux fois le numéro 3 (le 3 et 3 bis). Ils font tous les deux la même chose ou presque. Le 3 va attendre sans rien faire entre 1 et 3 secondes alors que le 3 bis va attendre 1 à 3 secondes mais en vérifiant si on appuie pas sur les boutons durant ce laps de temps. Ainsi on ne peut pas gagner en gardant son doigt appuyé dessus. C'est un peu plus complet mais aussi beaucoup plus compliqué comme tu peux le remarquer et faire ce type de code c'est aussi dans les métiers de l'ingénierie. Ici nous n'irons pas beaucoup plus loin car nous ne nous spécialisons pas là dedans, mais pour être un.e bon.ne ingénieur.e il faut aussi connaître ce qui nous entoure. Il y a donc des d'enseignements communs ce qui nous ouvre aussi la porte à d'autres domaines > auxquelles tu ne t'attendais pas forcément.

3 bis	<pre>// Allumer la del centrale toutes les 1 à 3 sec EN empêchant la triche unsigned long x = millis()+random(1000,3000); // On pose le moment où l'on veut passer à la suite while(x&gt;millis()){ // tant que l'on a pas atteint ce chiffre on continue     if(digitalRead(bt1) == HIGH){ // si le joueur 1 a appuyer avant         digitalWrite(del1r, HIGH); //alors on le fait perdre         delay(1000);         digitalWrite(del1r, LOW);         x=millis()+random(1000,3000); // Et on attends de nouveau         pendant 1 à 3 sec     } else if(digitalRead(bt2) == HIGH){ // Si le joueur 2 appuie avant         on le fait aussi perdre         digitalWrite(del2r, HIGH);         delay(1000);         digitalWrite(del2r, LOW);         x=millis()+random(1000,3000);     } else {         //sinon on ne fait rien     } }</pre>
-------	--

Bravo, tu as atteint la fin de ce projet :)