

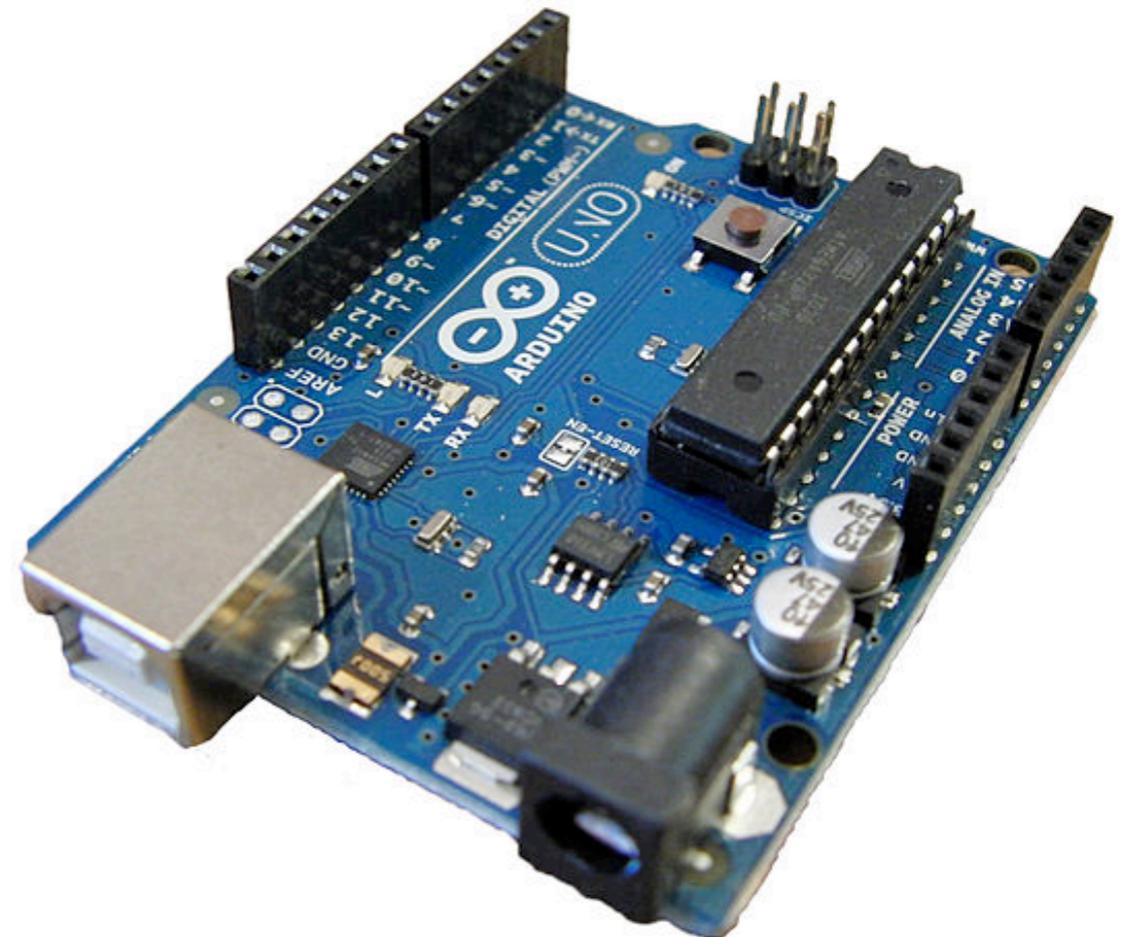


# Atelier électronique avec Arduino



On va jouer avec un **Arduino**.  
Trouve le dans la boîte et regarde le de plus près.

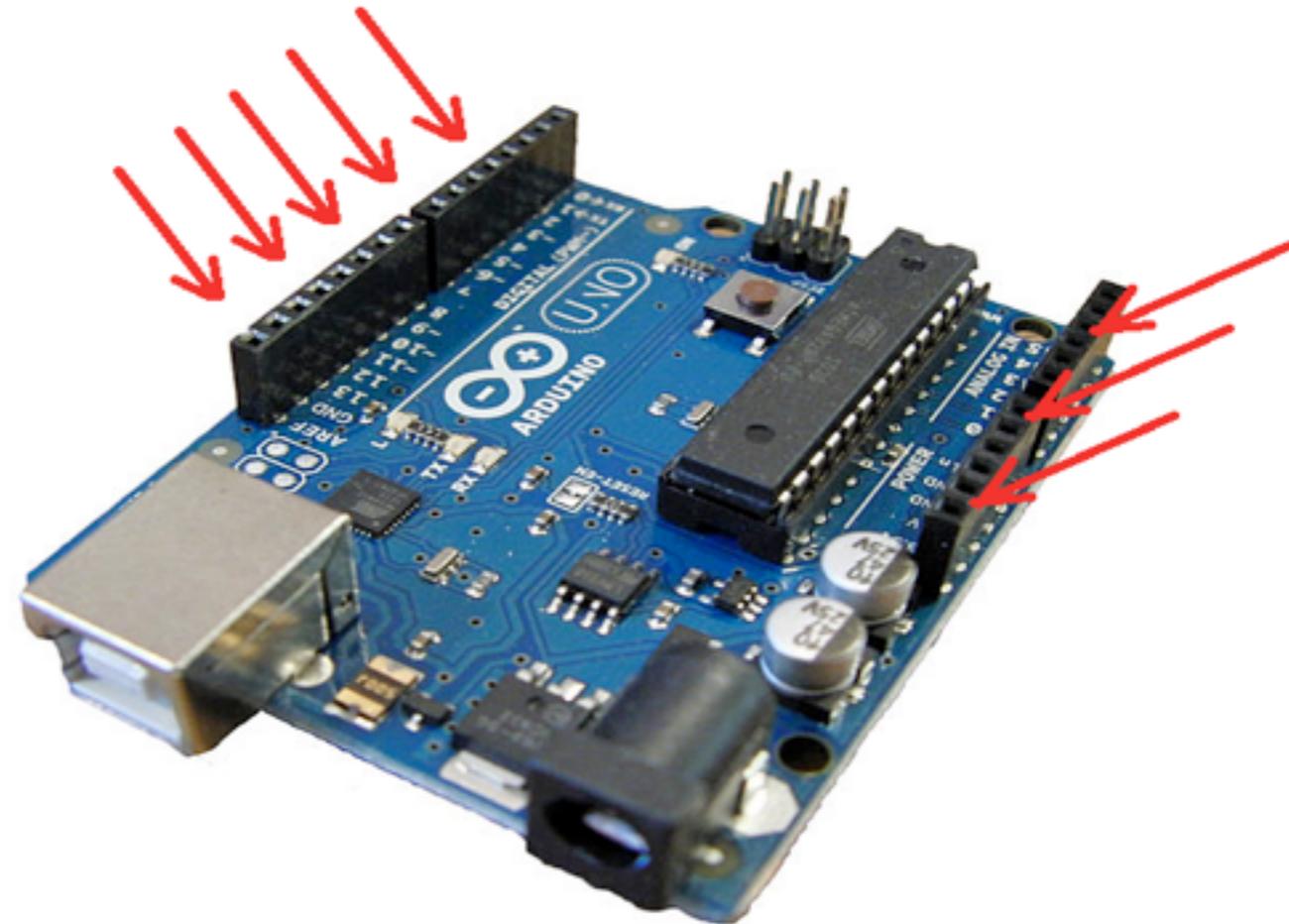
OK, il y a beaucoup de choses  
bizarres dessus, mais qu'est ce  
que c'est ?





C'est un **composant électronique** qu'on appelle une **carte**. C'est une sorte de **cerveau** pour les appareils.

Il sait faire des opérations simples comme compter et il peut envoyer et recevoir des informations.  
Toutes les choses que l'on voit sur la carte s'occupent de ça.





## **Mais à quoi ça sert ?**

C'est le cerveau des téléphones mobiles, des robots, de certains jouets, des machines à laver le linge ...

L'ordinateur a aussi une carte mais beaucoup plus puissante que l'Arduino.

On pourrait construire un robot, mais pour le moment il faut apprendre à travailler avec l'Arduino.

Nous allons d'abord faire clignoter une lampe.



Les composants électroniques ont besoin de **courant électrique** pour fonctionner.

Ils utilisent du courant **continu** (comme celui des piles). On ne peut pas les brancher directement sur la prise de courant qui fournit beaucoup trop de courant et pas sous la bonne forme. C'est du courant alternatif.

On va utiliser l'ordinateur qui sait faire cette conversion pour lui. On va l'utiliser pour fournir le courant continu.

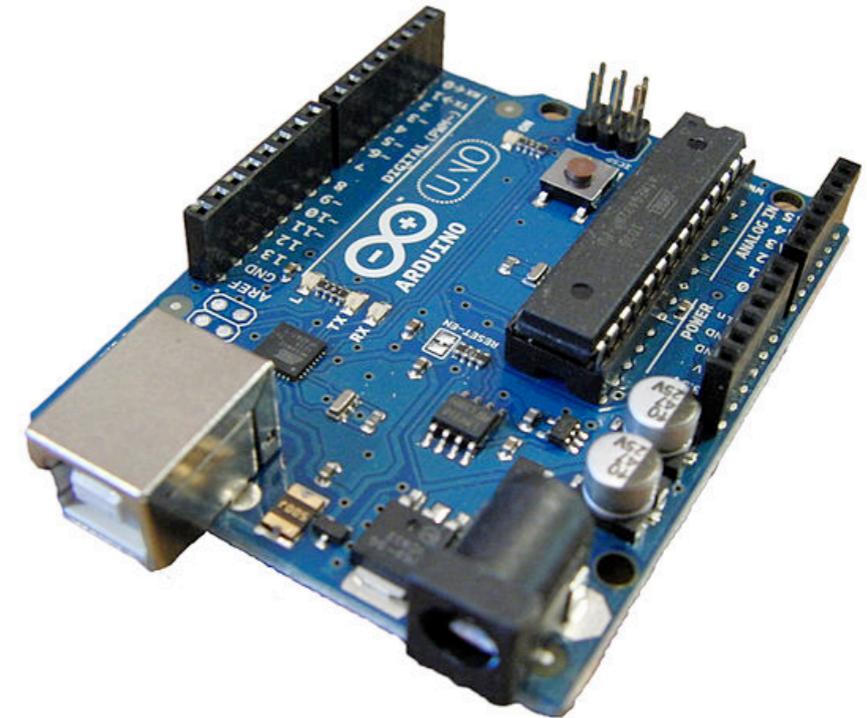


Cherche le cable et **connecte** l'ordinateur et l'Arduino.  
Attention, il faut que tu cherche les bons connecteurs.



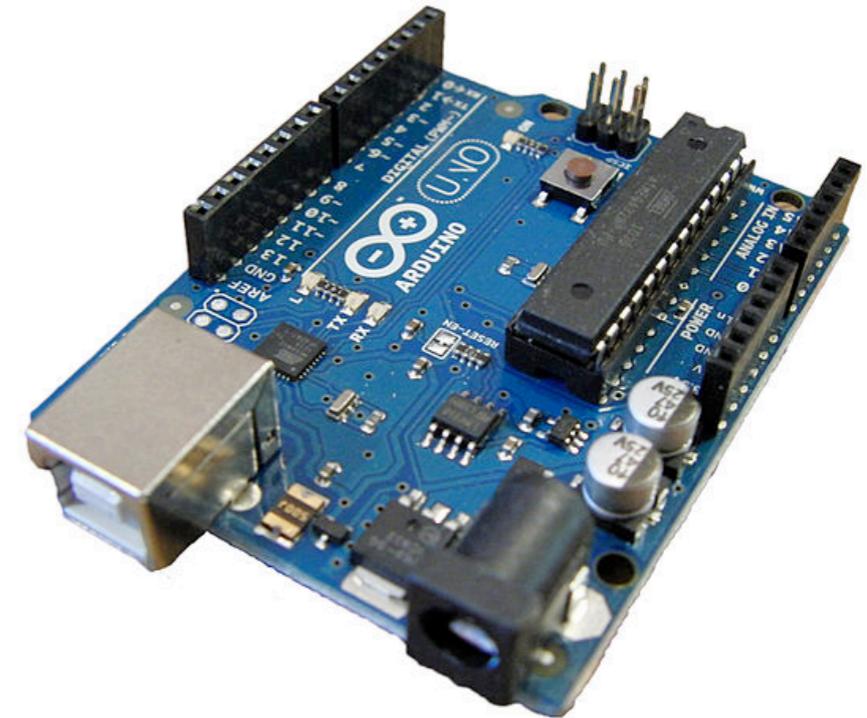


C'est fait ?  
Que se passe t'il sur l'Arduino ?





C'est fait ?  
Que se passe t'il sur l'Arduino ?



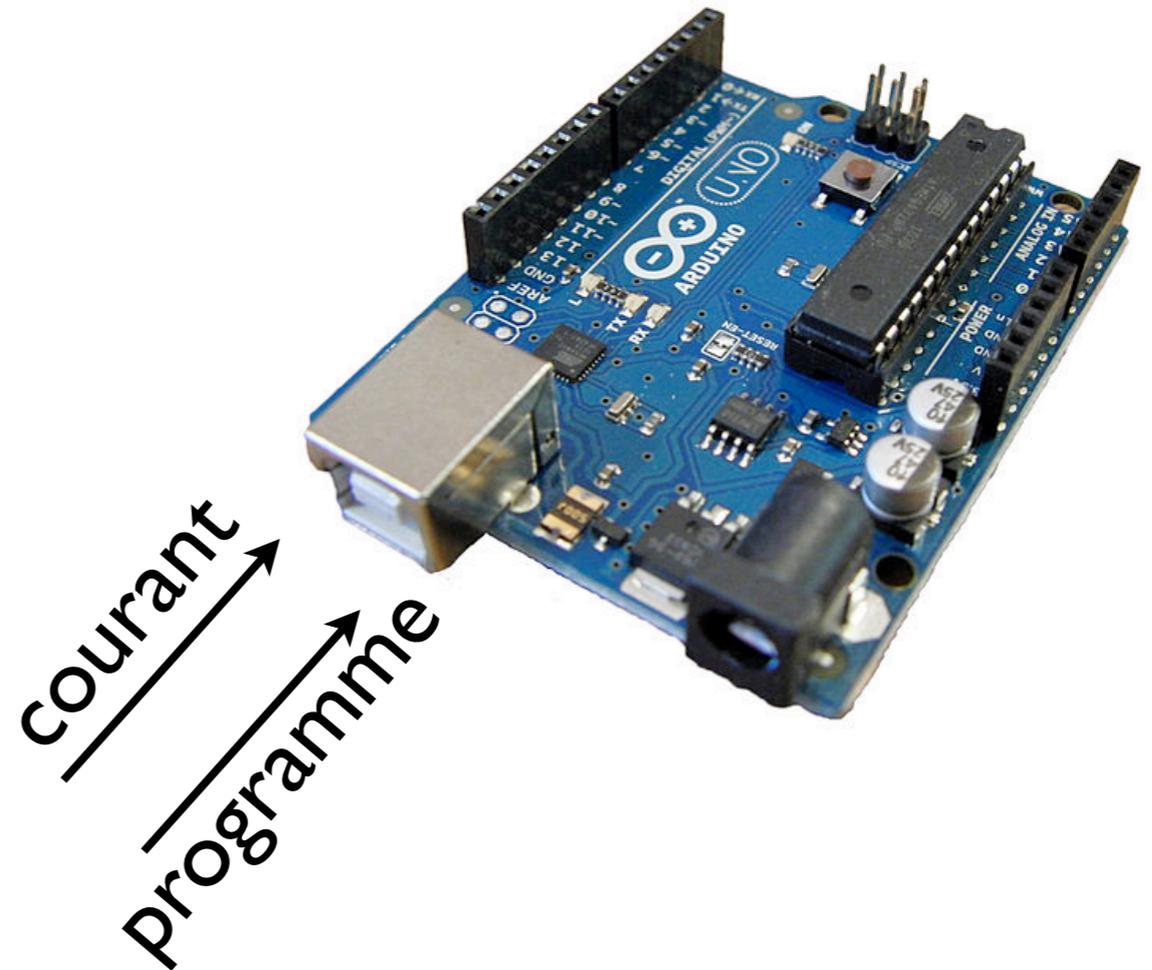
**Réponse :**  
La lampe verte s'allume et l'orange clignote



L'Arduino est un cerveau mais pour le moment il ne sait rien faire.

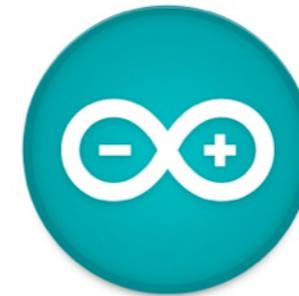
Il attend qu'on lui donne un programme à exécuter.

On va écrire ces programmes sur l'ordinateur, puis les transférer dans l'Arduino par le cable.





Lance le programme qui a ce logo



C'est l'IDE qui va nous permettre d'écrire des programmes.

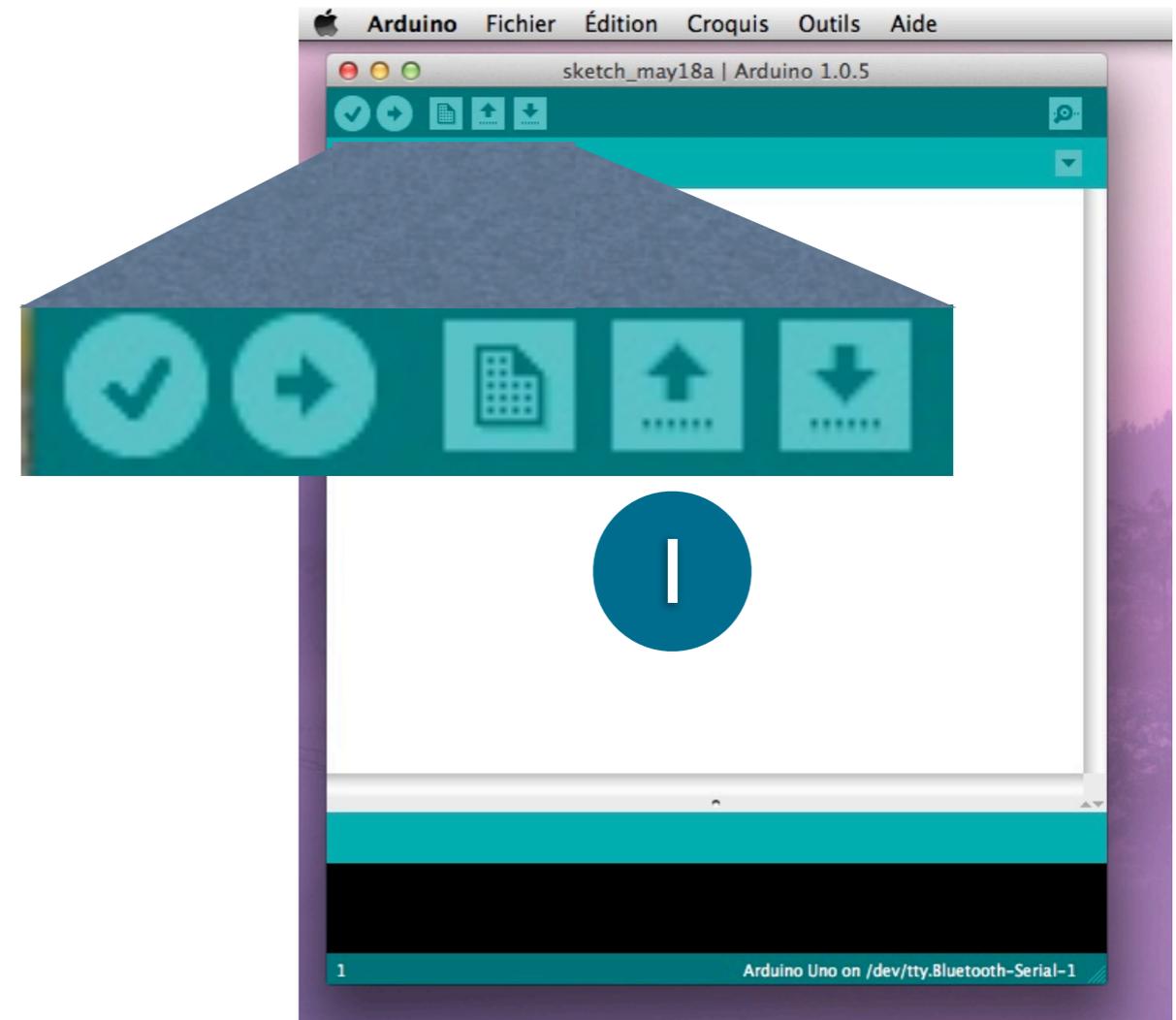
Un **programme** c'est une suite d'indications, un peu comme une route à suivre pour aller quelque part ou une recette de cuisine.



Nous allons utiliser un programme déjà fait :

**I** : charge le programme  
*Clignote* dans l'IDE

Lit le programme et décrit ce que l'Arduino devrait faire



# DEVOXX<sup>TM</sup> 4KIDS

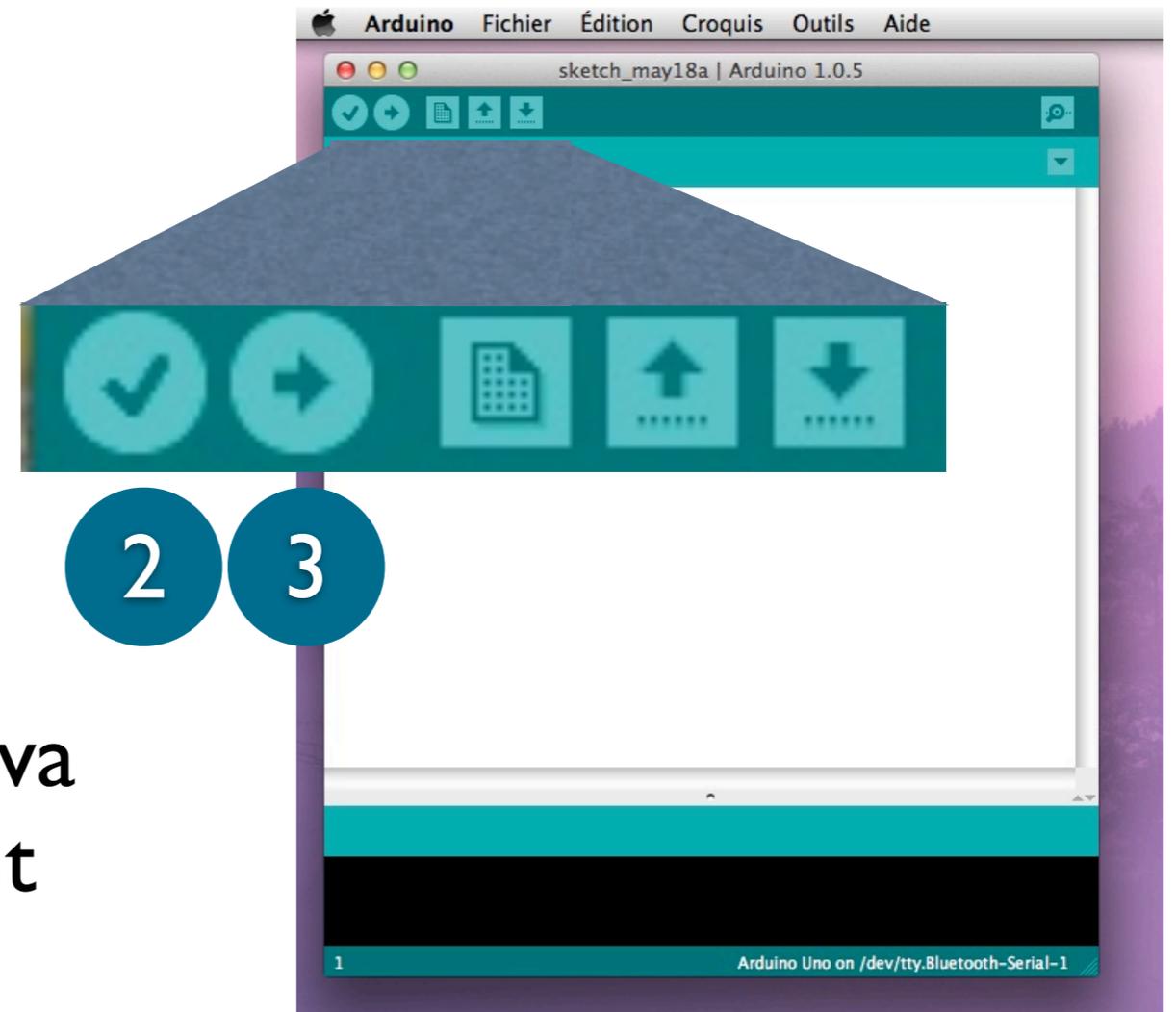


**2** : vérifie le programme avant de l'envoyer

**3** : envoie le programme à l'Arduino

Après un petit moment, l'IDE va indiquer que le «téléversement est terminée».

Que fait l'Arduino ?





Change des valeurs dans l'IDE (par exemple change les temps d'attente pour 10s), vérifie le programme, et envoi le à nouveau à l'Arduino

Que se passe t'il sur l'Arduino ?



Change des valeurs dans l'IDE (par exemple change les temps d'attente pour 10s), vérifie le programme, et envoi le à nouveau à l'Arduino

Que se passe t'il sur l'Arduino ?

**Réponse :**

La lampe orange clignote toutes les 10 secondes

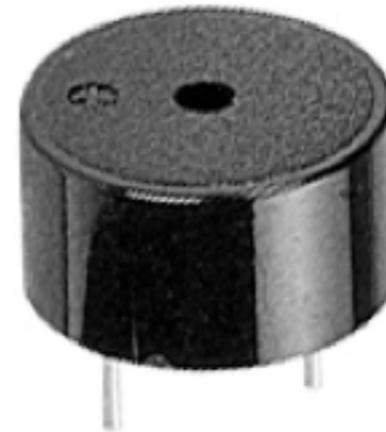


Maintenant il faut ajouter **les composants** que l'Arduino va piloter. Ce sont eux qui vont faire de la lumière, bouger, faire du bruit.

Les **LED** sont des petites lampes.



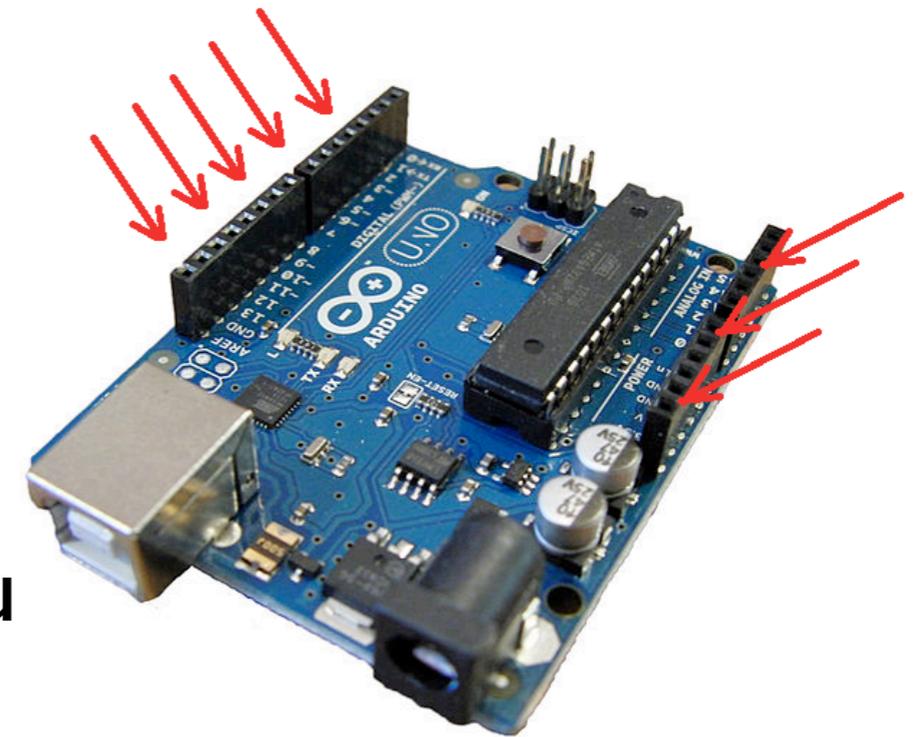
Celui ci est un **buzzer**. Il joue des sons.





On va créer un **circuit** qui relie l'Arduino et les composants pour que le courant puisse passer.

Les **broches** sur les côtés vont permettre au programme de contrôler quand le courant passe.



La lampe s'allume quand elle reçoit du courant. Elle s'éteint quand elle n'en a plus.

# DEVOXX™ 4KIDS



Retourne dans l'IDE et regarde le programme.

Il est en 2 parties

- Dans Clignote, il y a les **fonctions** où tu indique ce que tu veux faire (setup et loop)

- Dans LED, il y a les fonctions qui traduisent tes demandes dans le langage de l'Arduino

```
Clignote § LED §  
  
// Sur la plupart des Arduino il y a une LED sur  
// On va donner un nom au pin, ça sera plus par  
int led = 13;  
  
void prepareLED () {  
  // dit à la broche 13 qu'elle est un OUTPUT,  
  pinMode (led, OUTPUT);  
}  
  
void allumeLED () {  
  digitalWrite (led, HIGH); // allume la LED  
}  
  
void eteintLED () {  
  digitalWrite (led, LOW); // eteinf la LED (  
}  
  
void attend (int n ) { // n est en secondes  
  delay (n * 1000);  
}
```

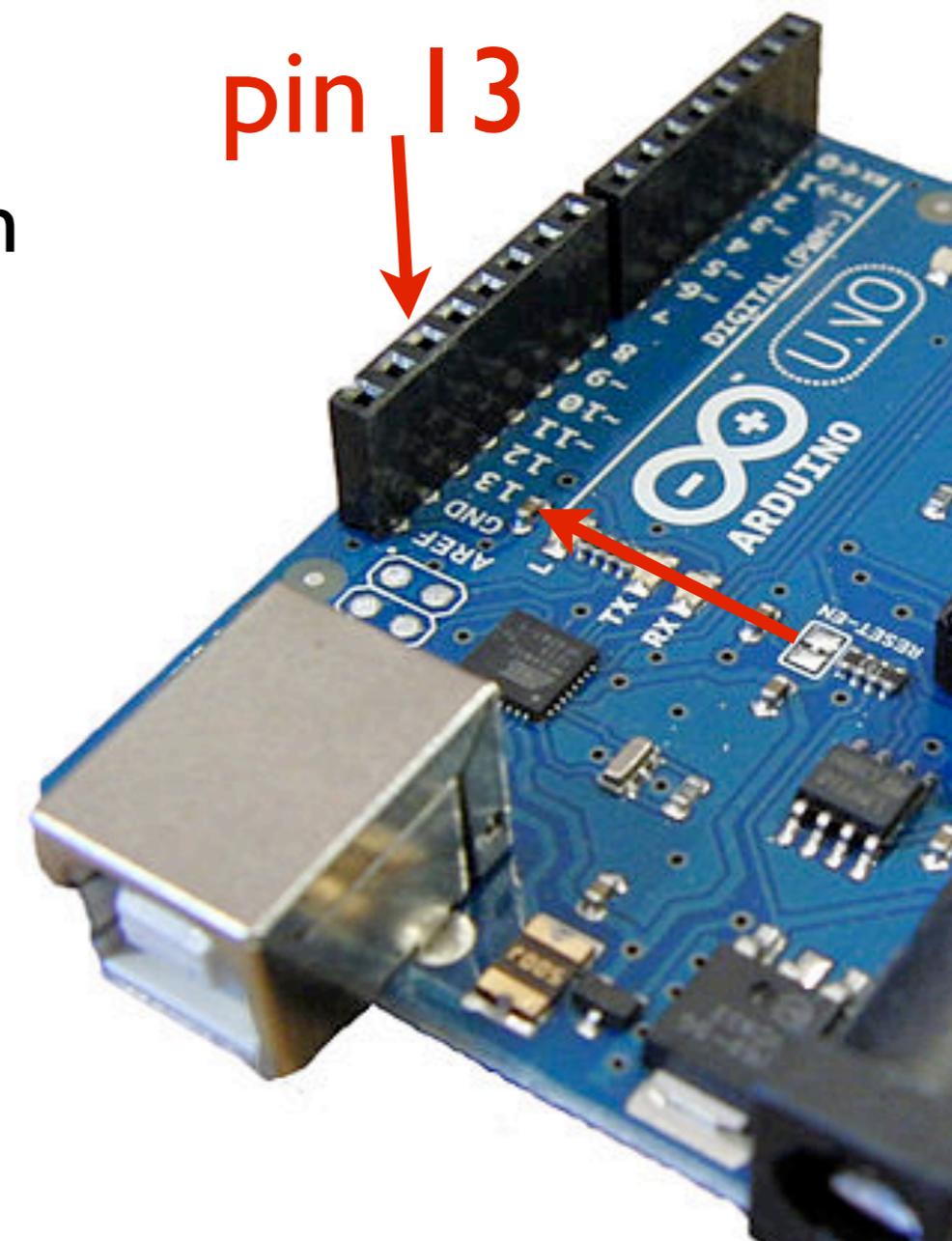


Que fait le programme ?

Le programme indique qu'il va travailler sur le pin 13 (*pinMode*). pin est le nom en anglais pour broche. C'est celui qui est à côté de la LED orange de la carte.

Ensuite, il écrit soit HIGH soit LOW. C'est de l'anglais, et ça signifie HAUT et BAS.

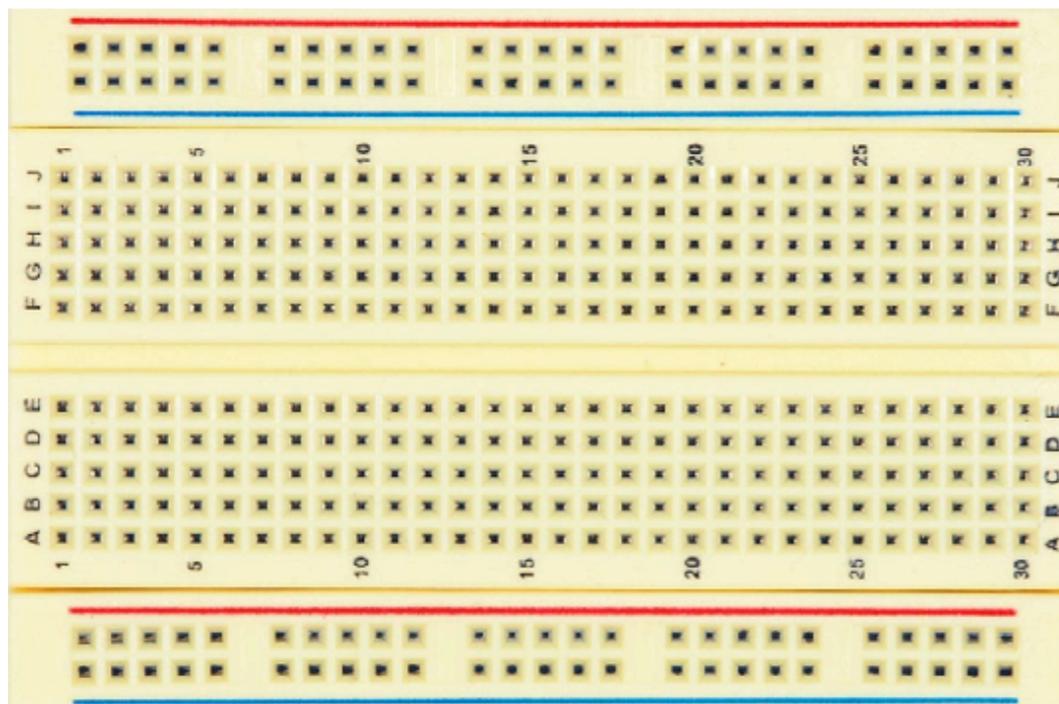
Que se passe t'il pour la lampe ?



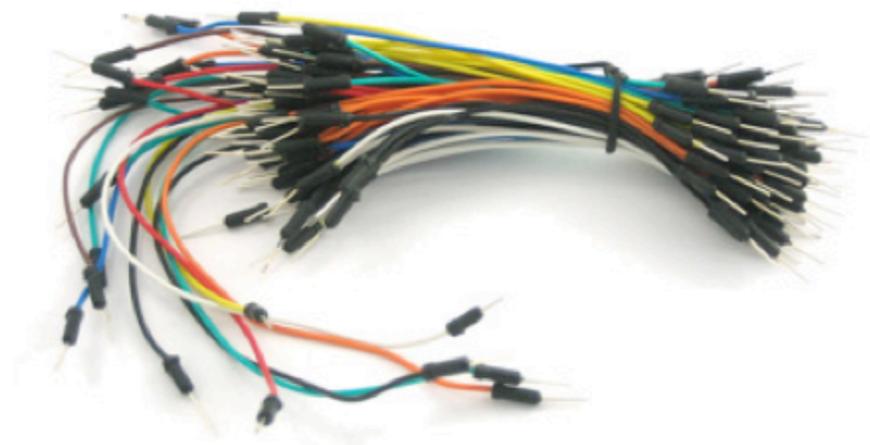


Dans un appareil, les circuits sont fabriqués en usine et soudés pour qu'ils soient solides.

Nous allons utiliser une **plaque de cablage** qui permet de monter et démonter les circuits.



Et des  **fils**  ..., beaucoup de fils



On va aussi avoir besoin de **résistances**. Elles limitent le courant qui passe pour protéger les composants et l'Arduino

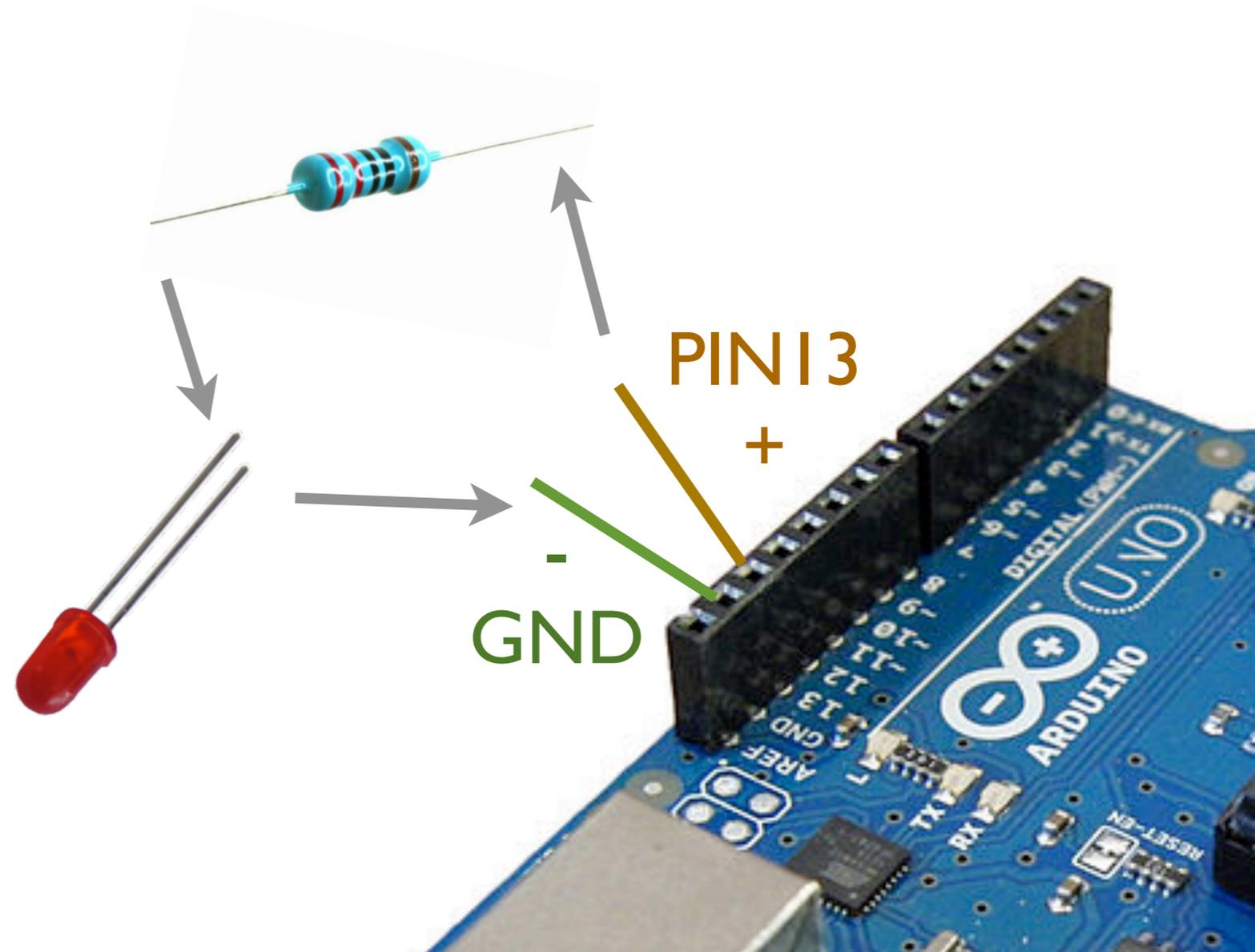
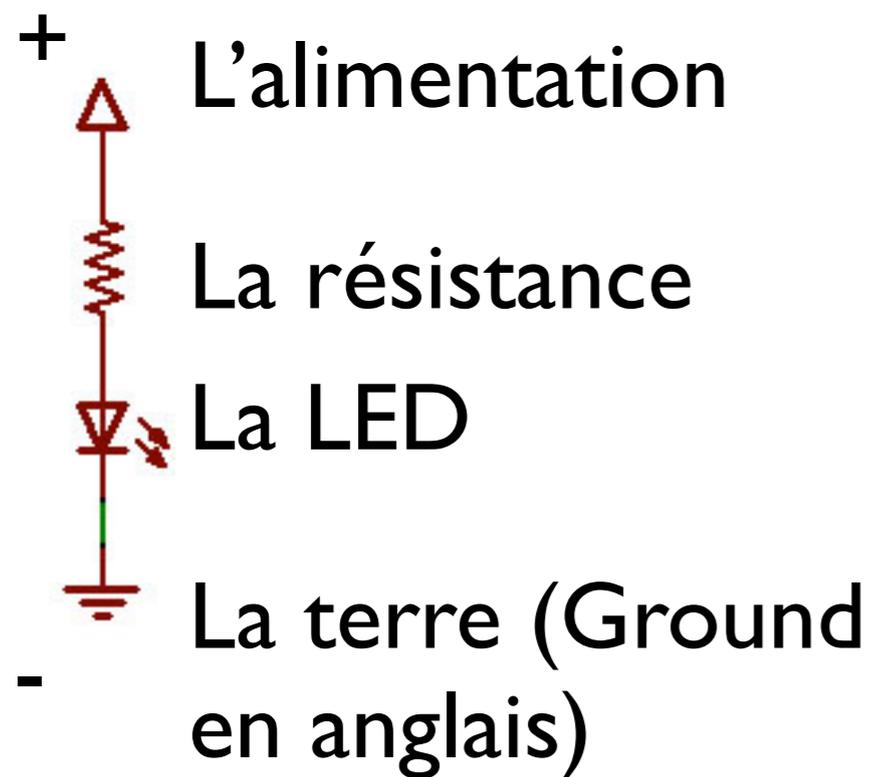


Regarde dans la boîte, il y en a beaucoup de différentes.

Les barres de couleur sont un code pour sa résistance . Nous avons besoin de 220 Ohms.

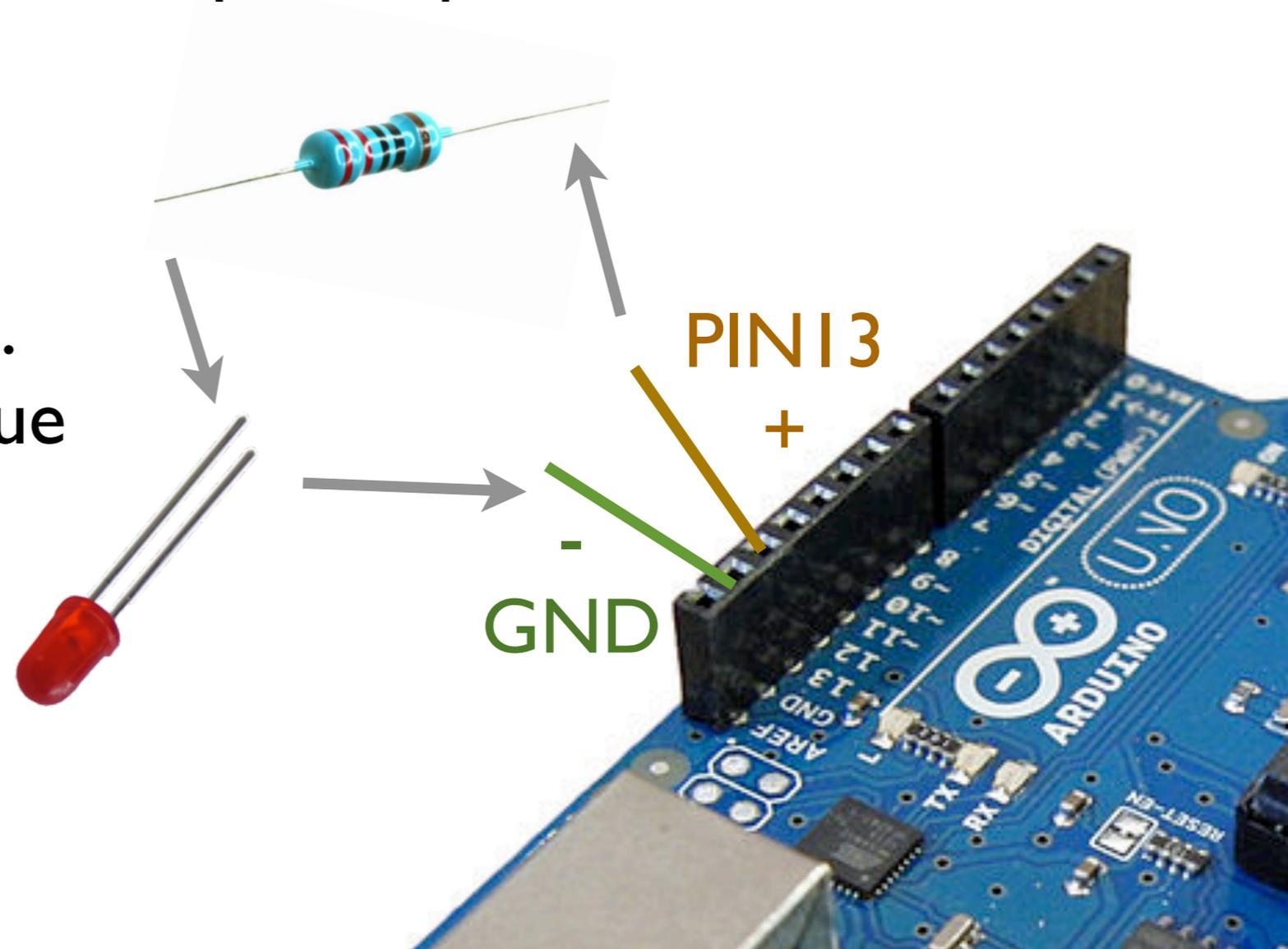
Ce sont les plus nombreuses. Bleu avec des barres Rouge-Rouge-Noir-Noir.

On dessine le circuit de cette manière.  
Le courant circule toujours du + vers le -

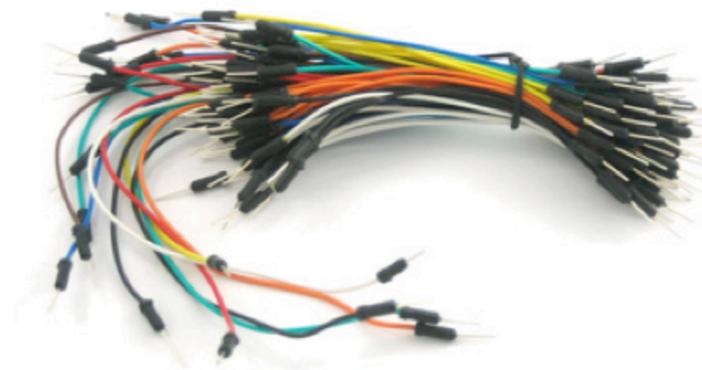
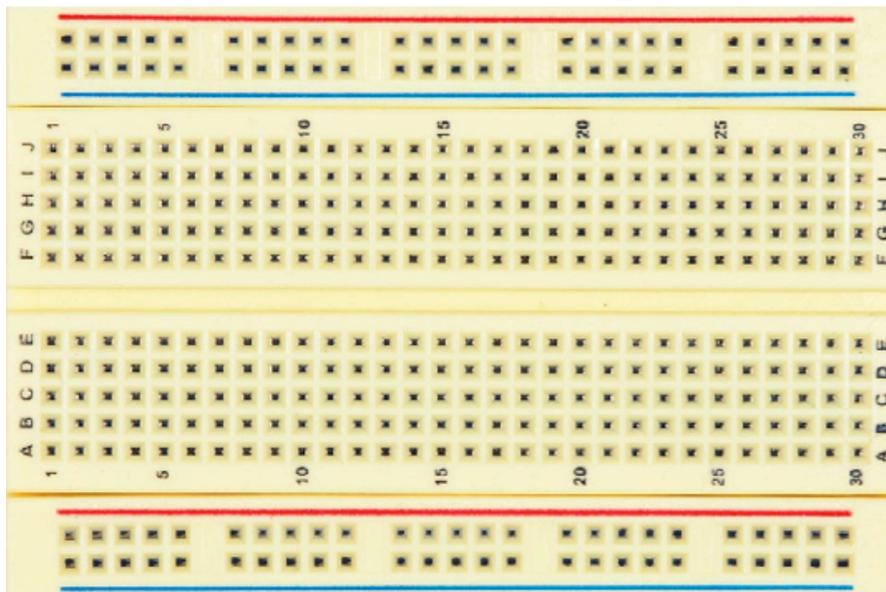
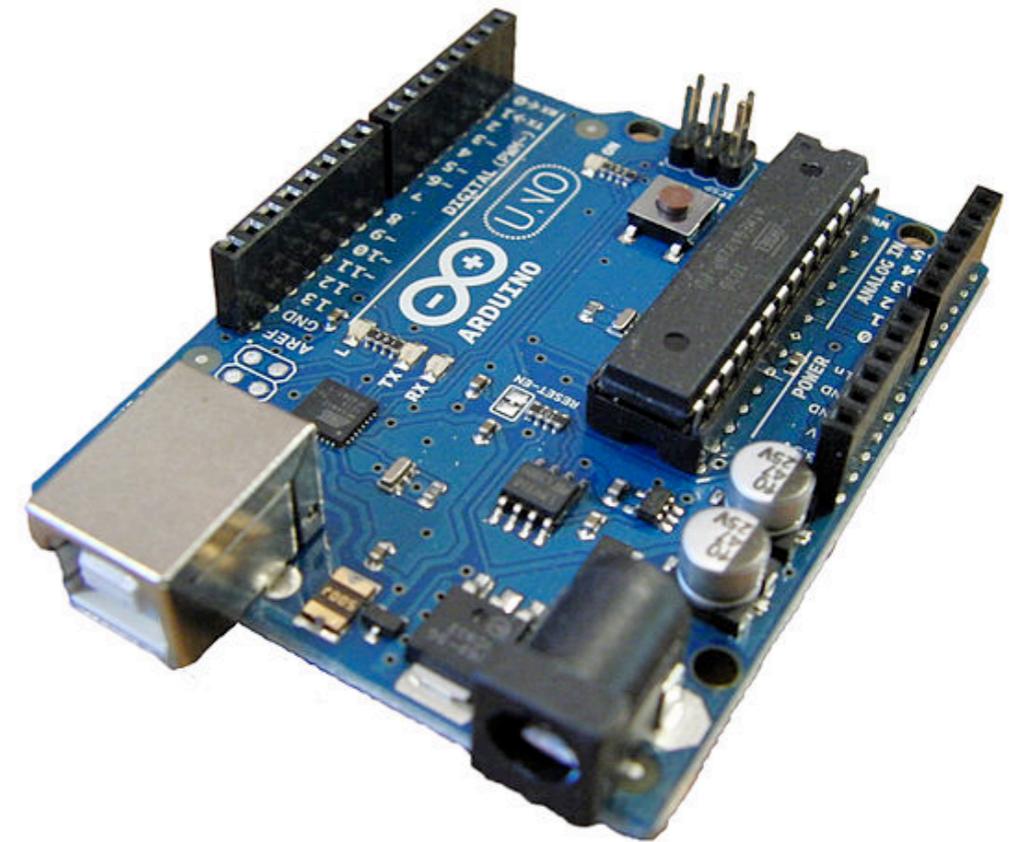


La résistance peut être utilisée dans n'importe quel sens.

La LED est directionnelle. Elle a une patte plus longue que l'autre, c'est celle qui doit être côté +.

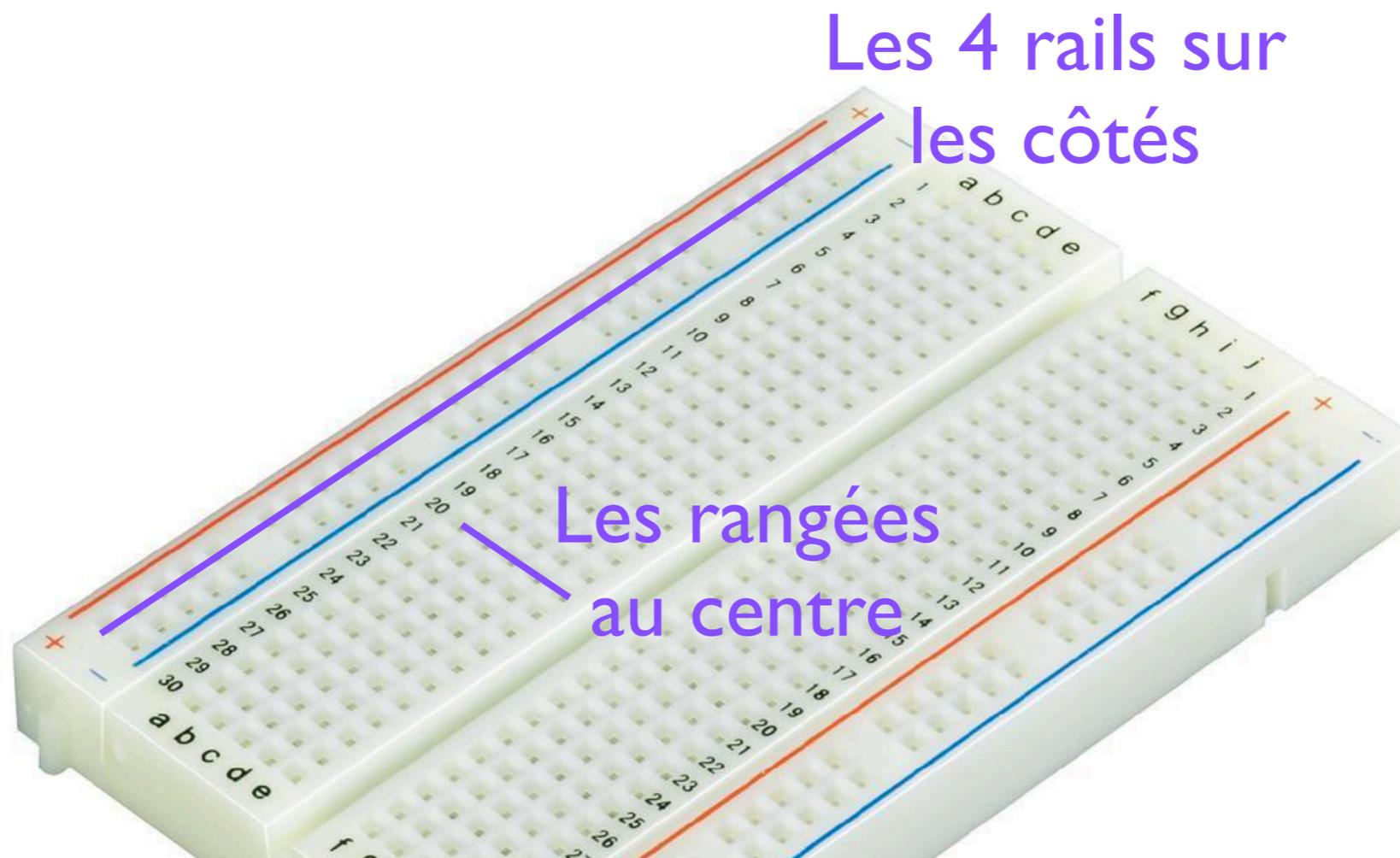


Et ma plaque d'essai la dedans ?



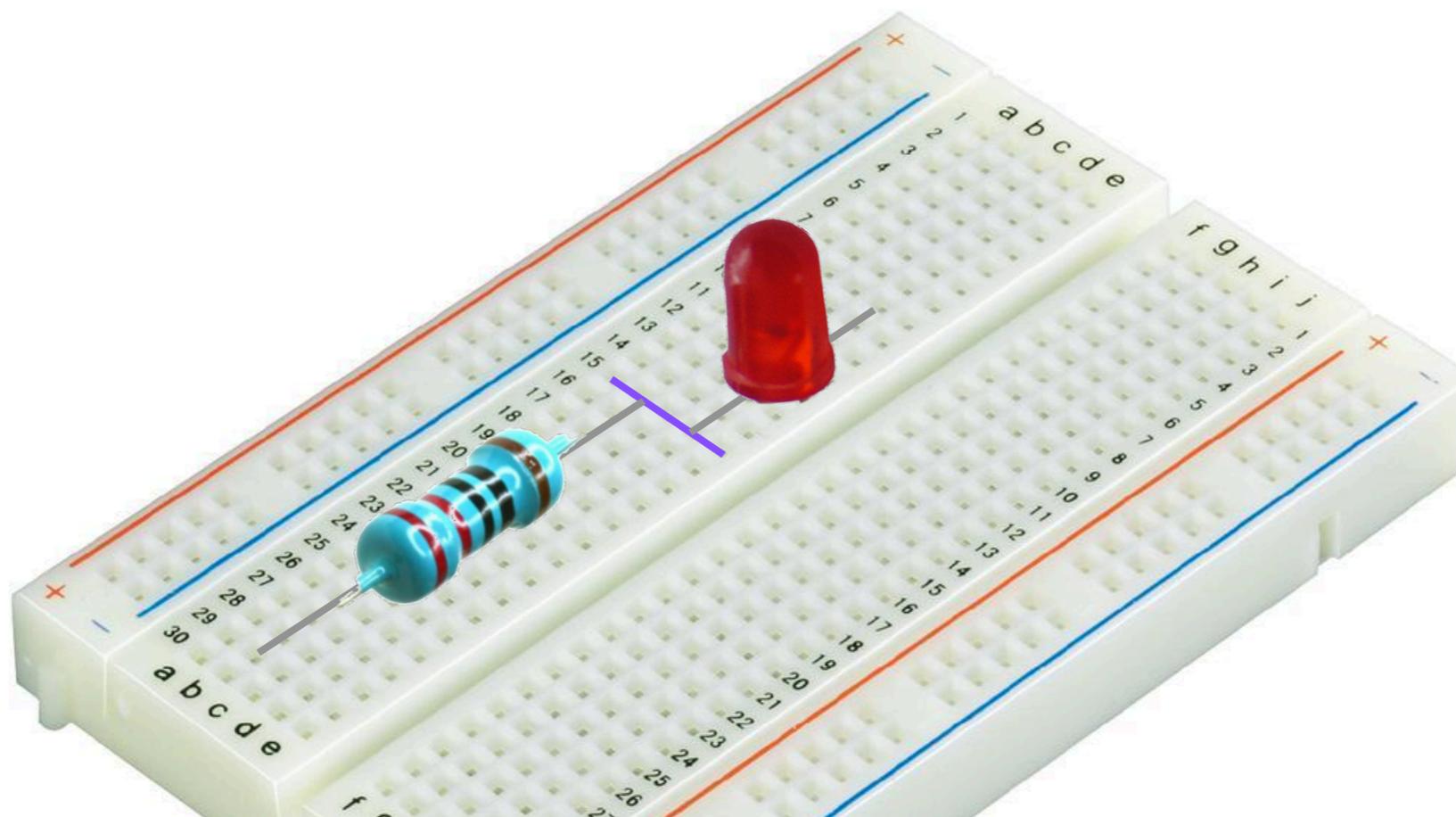


On va piquer les composants dans la plaque.  
A l'intérieur de la plaque certains trous sont reliés entre eux.

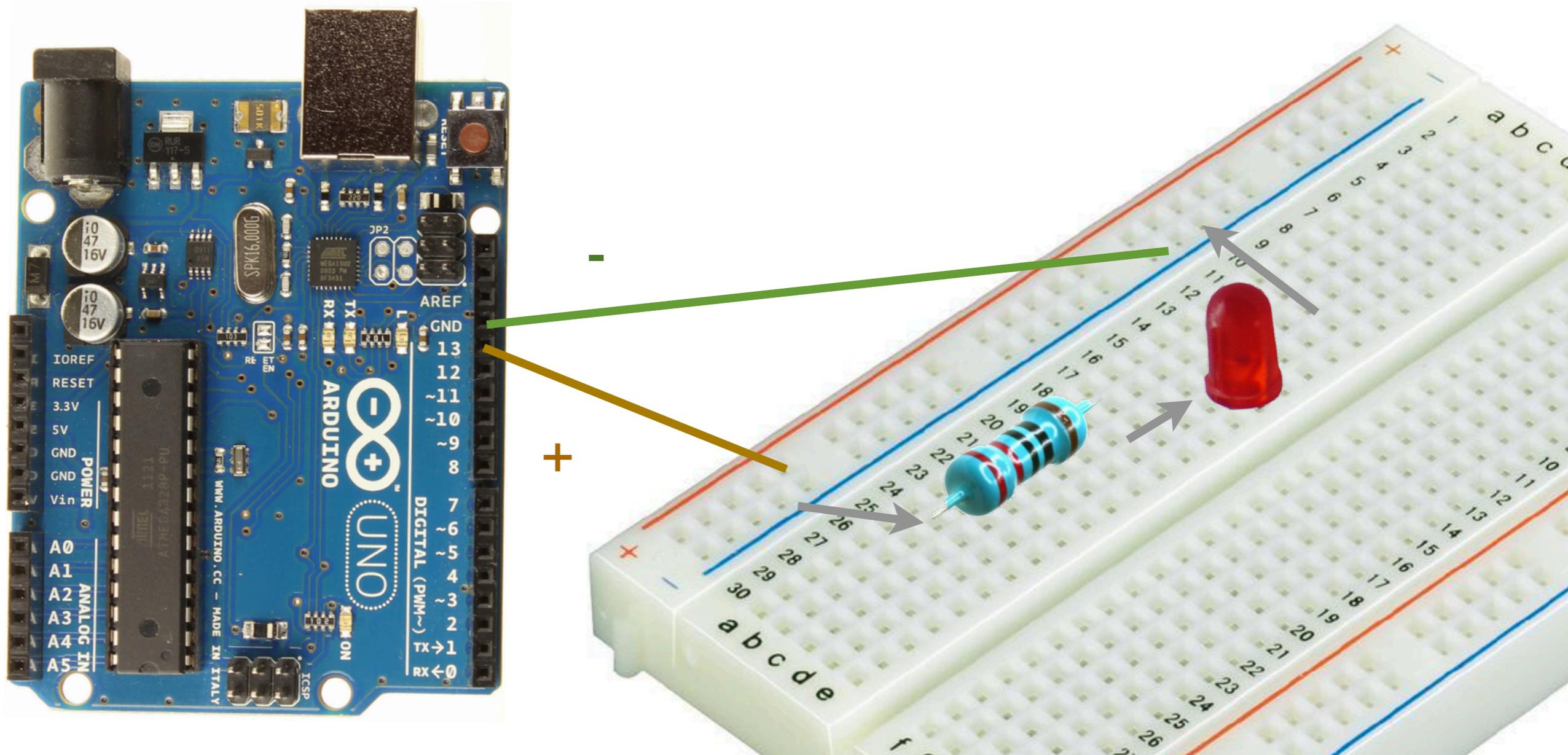




On peut relier les composants par les rangées ou les rails



Comment peux tu relier les composants pour construire le circuit en suivant les rails et les rangées ...





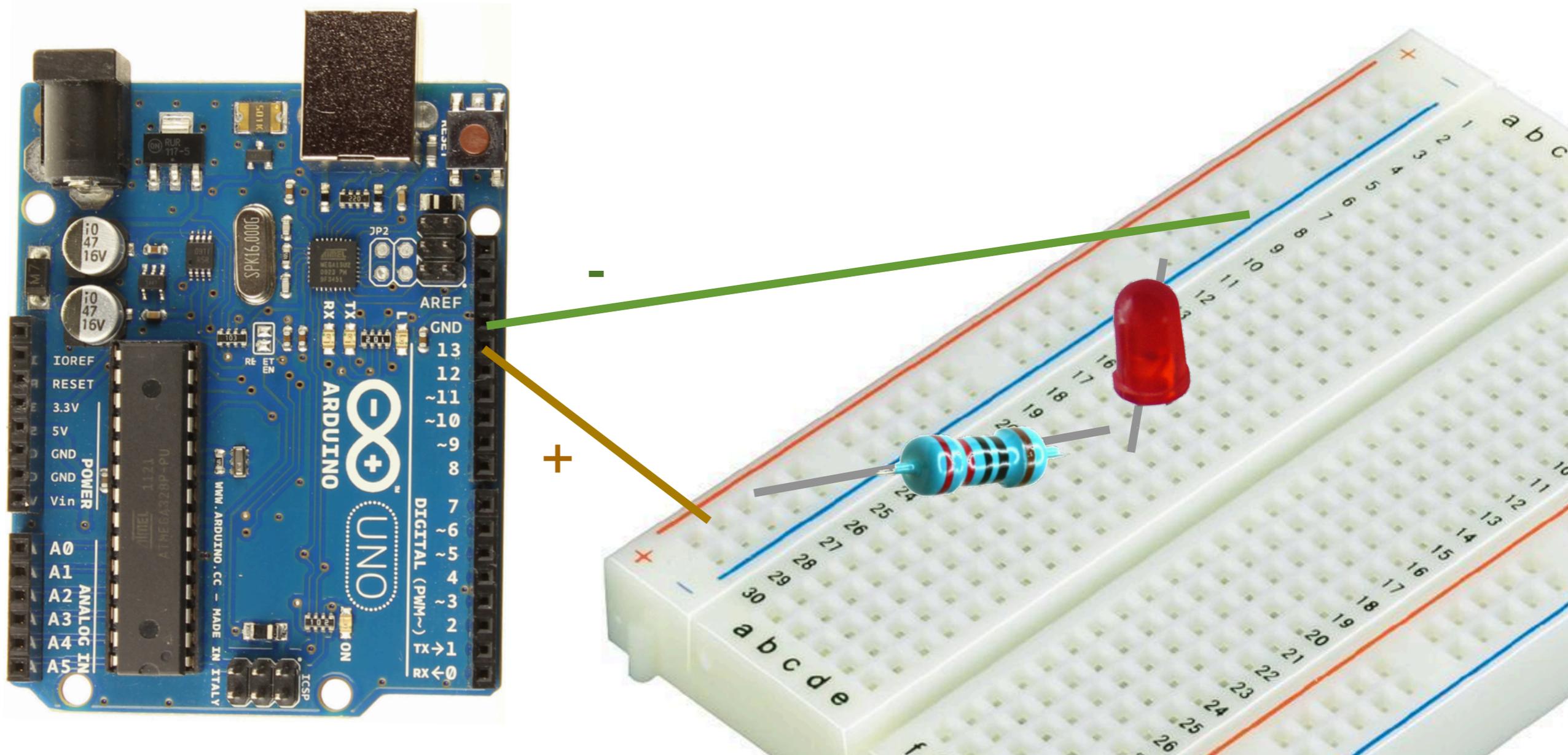
Tu peux maintenant faire du MORSE.  
Par exemple OK se dit --- -.-

Si le point . fait 1 seconde et se fait en allumant la lampe 1s puis en l'éteignant 1s, comment peux tu faire le trait - qui est 3 fois plus long et l'espace long ?

**Info 1** : tu peux créer tes propres fonctions «point» et «trait» par exemple `void point () { ... }`.

**Info 2** : tu n'as pas besoin de la loop car le message n'est envoyé qu'au démarrage ou au RESET (bouton sur la carte).

Une solution possible, mais pas la seule. Choisi la disposition qui t'arrange pour placer les pattes.





Beau travail.

Tu es un expert maintenant !

On va faire de la musique dans l'atelier suivant.