DEVOXX™ 4KIDS

Session électronique avec Arduino

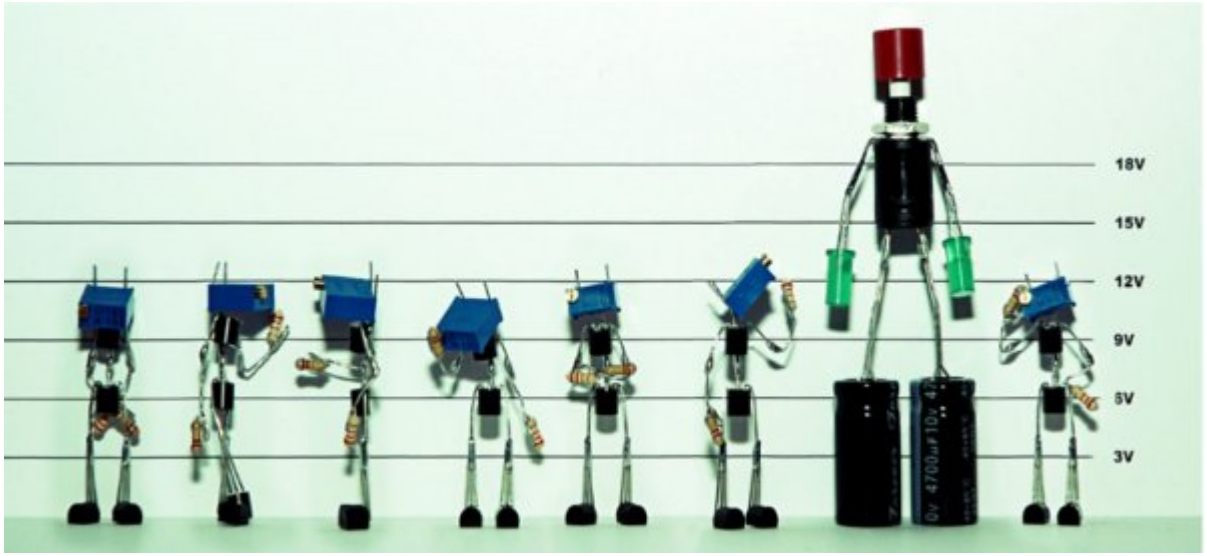
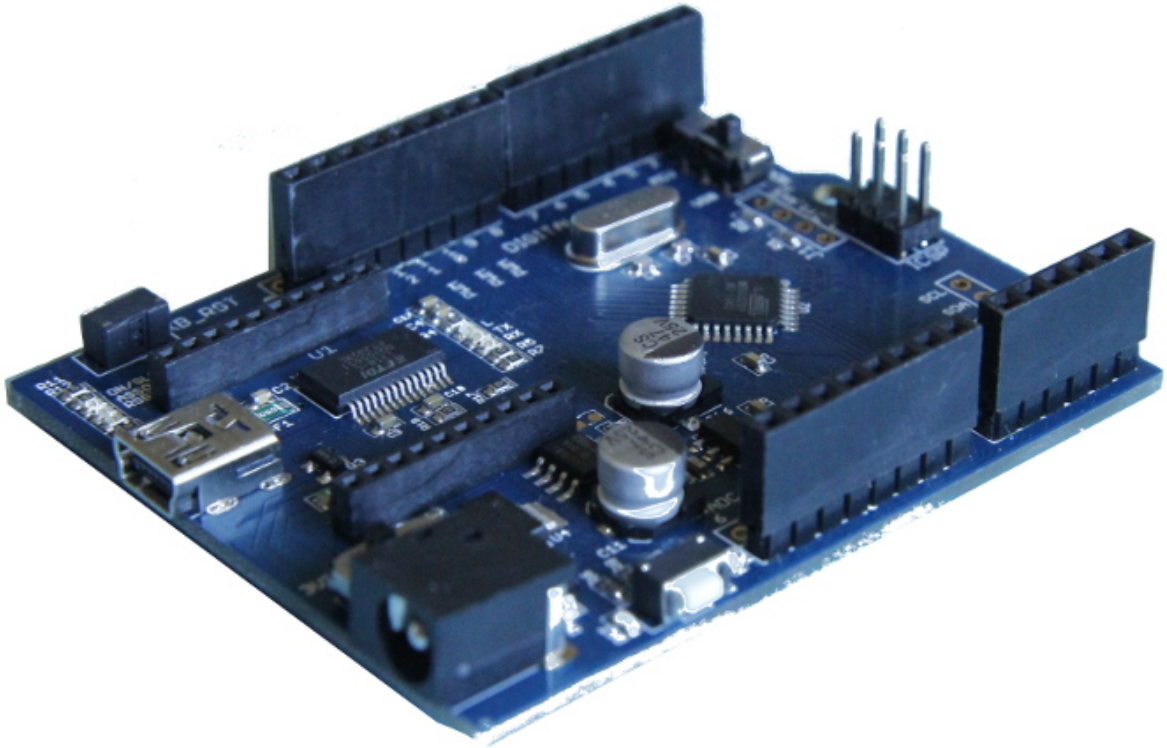


Photo by Lenny et Meriel ©

www.devoxx4kids.com

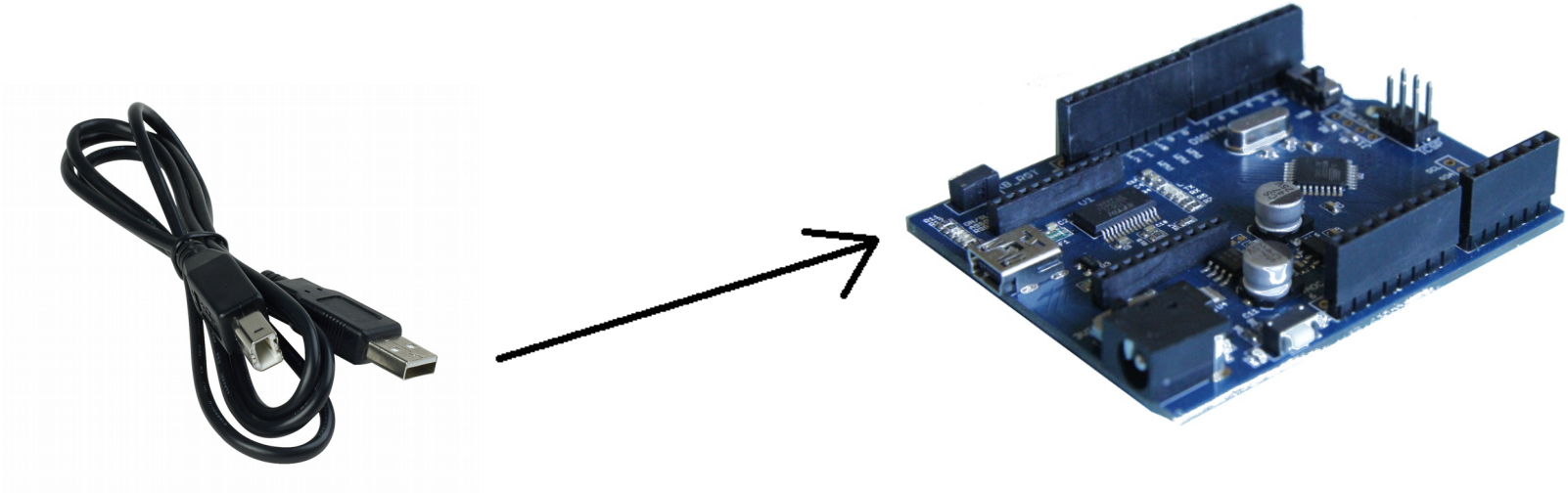
DEVOXX™ 4KIDS

Ouvrez la boîte... et cherchez ce truc:

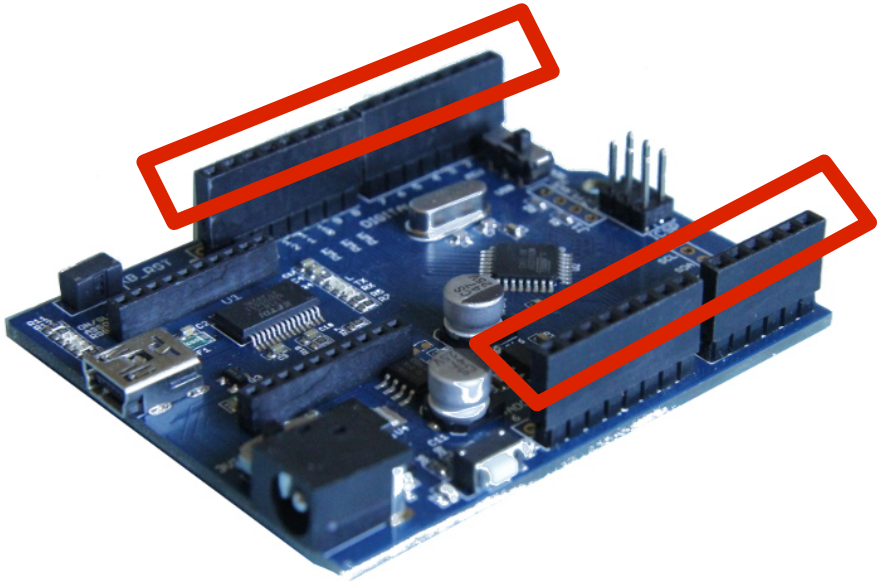


www.devoxx4kids.com

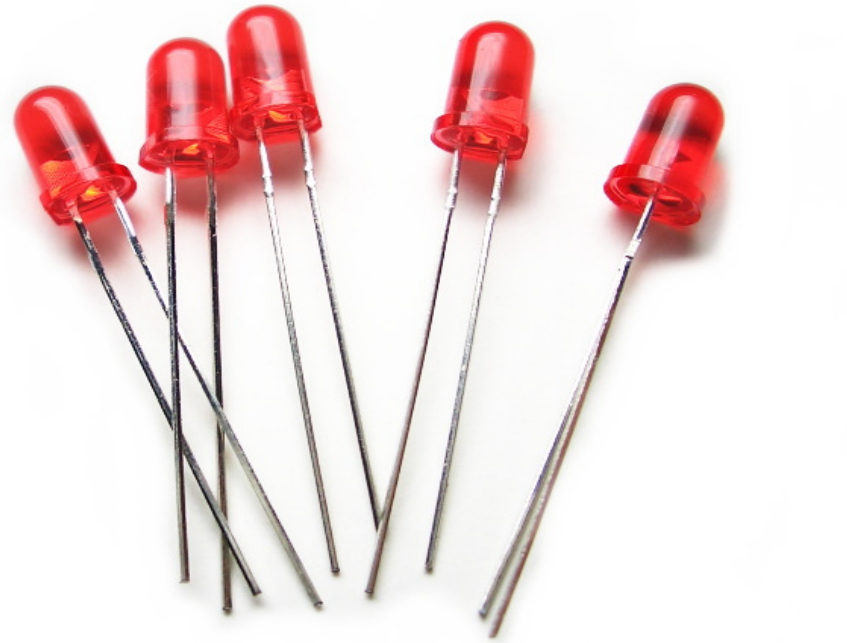
> On peut demander à Arduino d'exécuter une tâche (un **programme**) que l'on réalise, au préalable, sur un ordinateur.
> Pour lui envoyer le programme, il faut le connecter à l'ordinateur avec le câble **USB**.



> Via le câble USB, Arduino reçoit du courant et est dit *sous tension*.
> Arduino a des **pins** (les trous) sur les côtés qui peuvent envoyer ou recevoir du courant!

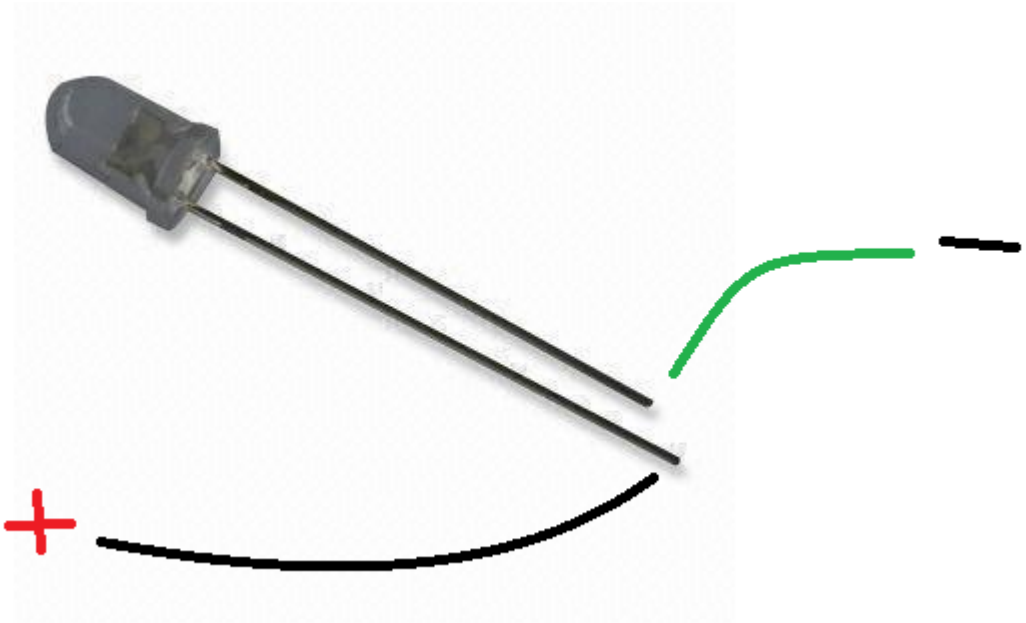


> Une **LED** est une très petite lumière, souvent de couleur (blanc/rouge/jaune/vert)
> Regardez celles que vous avez dans la boîte!



DEVOXX™ 4KIDS

> Chaque LED a deux **jambes**... une courte et une longue
> La longue représente le PLUS (**+**) (pôle positif)
> La courte... le MOINS (**–**) (pôle négatif)
> Le courant circule toujours du **+** vers le **-**

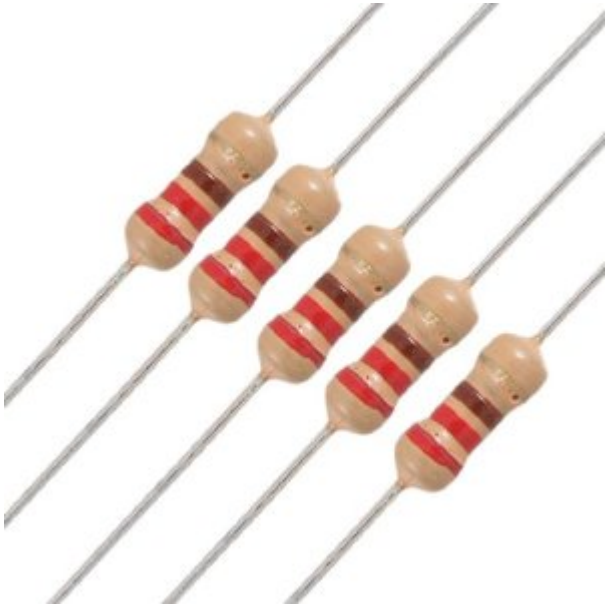


> Arduino reçoit donc son courant via l'**USB** (et donc du PC)
> On mesure la quantité de courant qui passe dans le fil en **AMPÈRE** (c'est le *litre* du courant électrique, ou le *mètre*...)
> La quantité de pression (tension) pour que le courant se fasse se mesure en **VOLT** (pas le chien!)

> Une LED a une tension de 5V (5 Volts)
> Une LED peut faire circuler jusqu'à 15mA (15 milli ampères, 15 millièmes de 1 ampère)

> Donc il faut faire attention au courant qu'on lui envoie !!!

Une **résistance** assure qu'il n'y a pas trop de courant qui passe.
C'est une espèce de goulot de bouteille... mais pour l'électricité...



> Tout ce beau monde se combine comme ça:
ampère (courant) * **ohm** (résistance) = **volt** (tension)

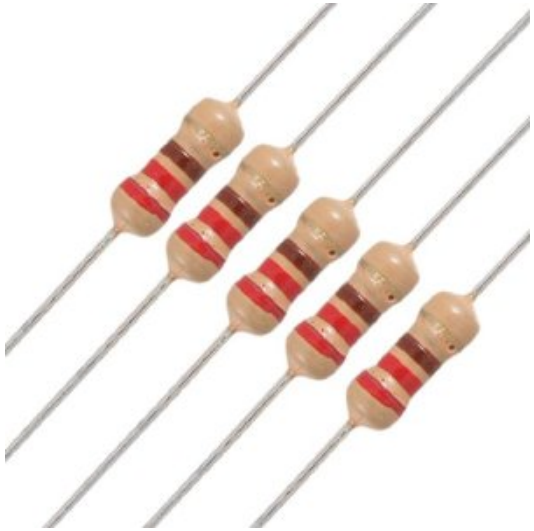
> Donc, on peut savoir la résistance dont on a besoin:
ohm = **volt** / **ampère**

> Pour nos LEDs:
(5 – 1.8) volt / 0,015 ampère = 213 ohm

Les résistances ont des **lignes colorées**... en plus de *faire joli*, ces lignes sont un **code** qui représente la quantité d'**ohms** qu'elles offrent.

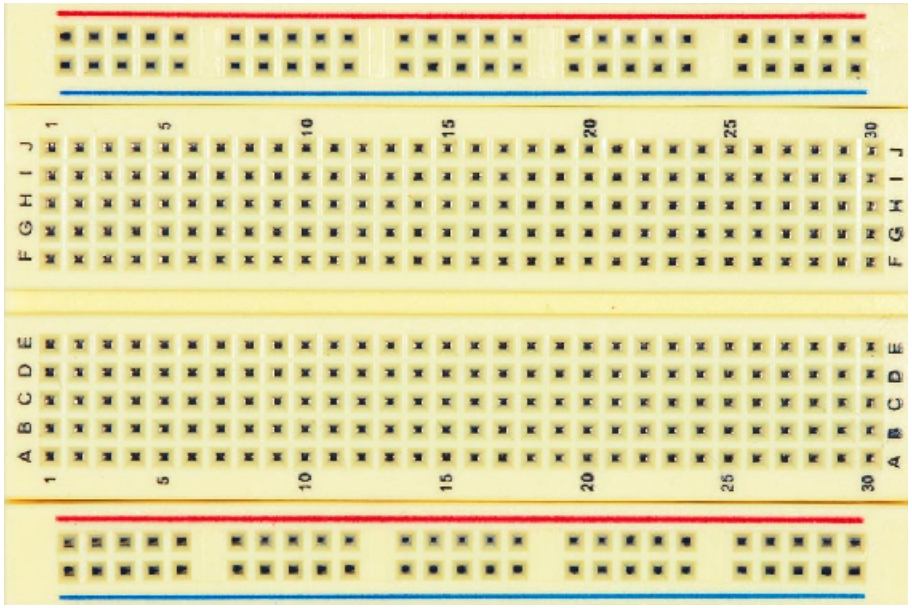
Pour nos LEDs on a besoin de 213 ohms et dans le langage secret des électriciens, le code **Rouge-Rouge-Brun** veut dire 220 ohms.

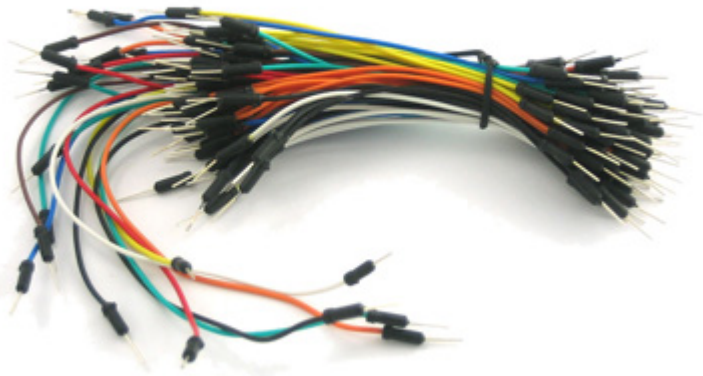
Vu qu'on n'est pas à 7 ohms près,
on se servira de celles-là **->**



DEVOXX™ 4KIDS

La planche à pain (breadboard) et ses fils...





DEVOXX™ 4KIDS

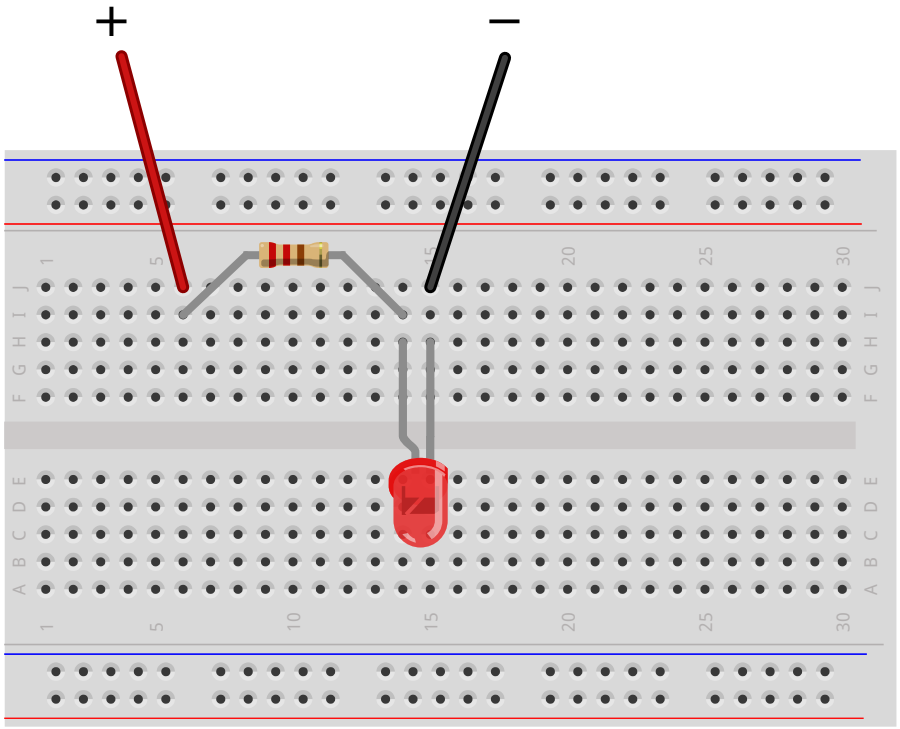
Mouais et? “Comment ça marche?”

Made with Fritzing.org

www.devoxx4kids.com

DEVOXX™ 4KIDS

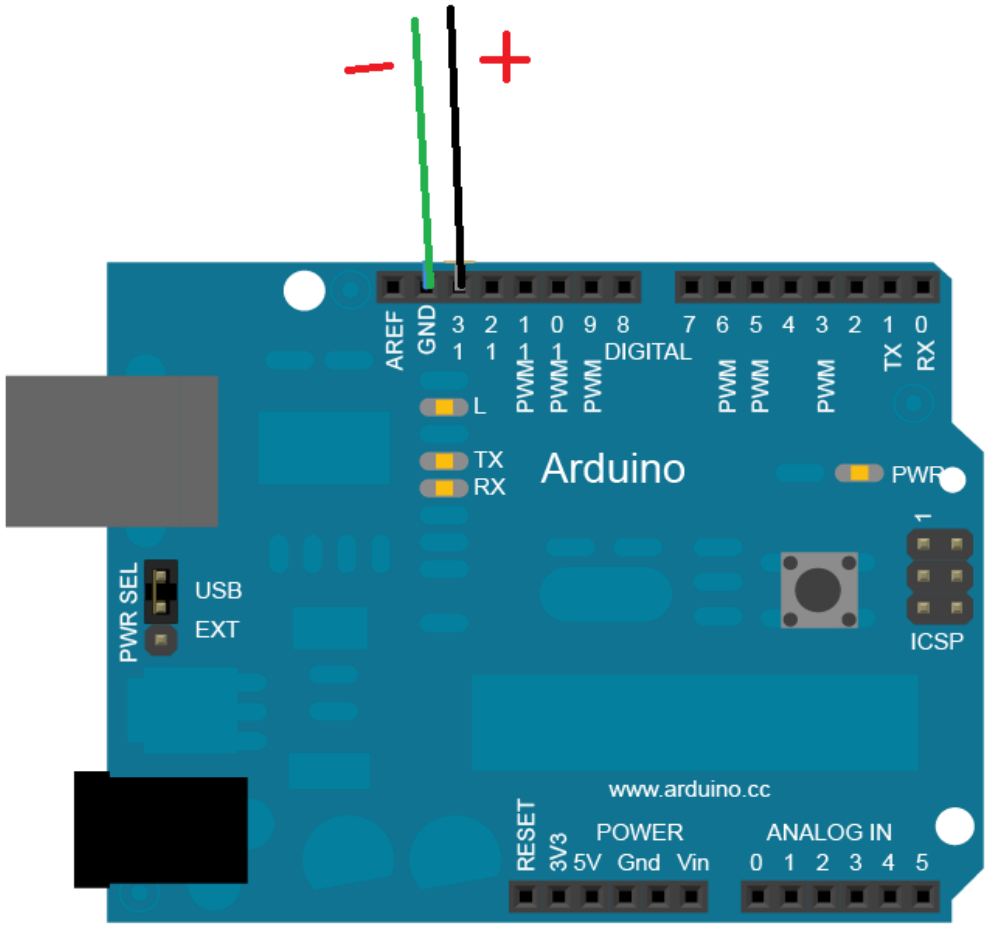
Allez, zou on met tout ça ensemble!



Made with Fritzing.org

www.devoxx4kids.com

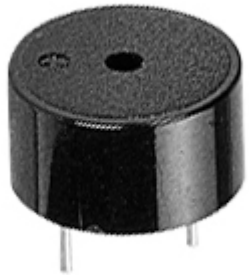
DEVOXX™ 4KIDS



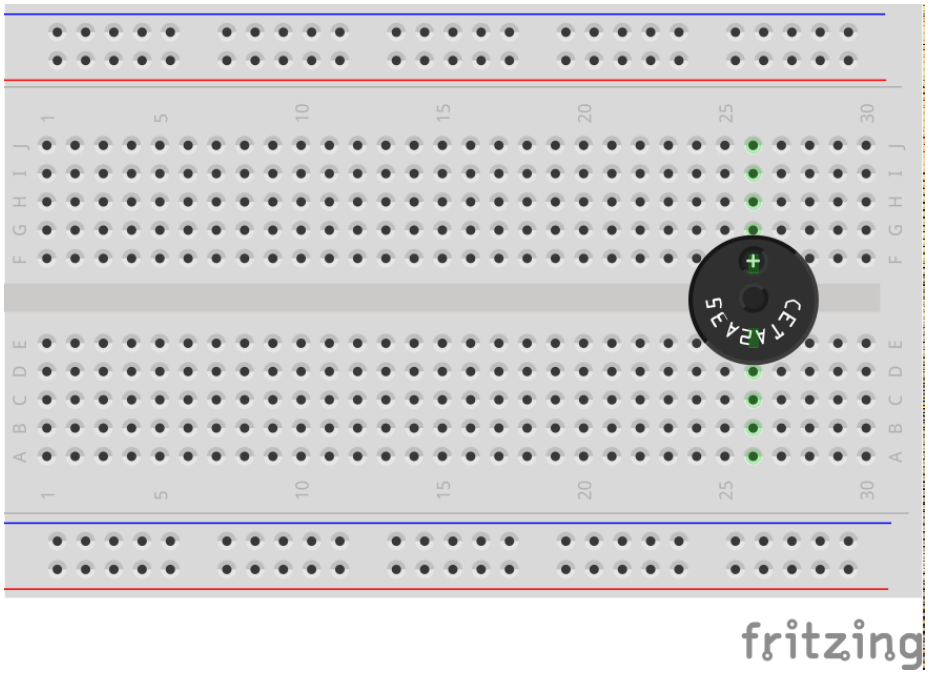
www.devoxx4kids.com


```
long dernierTemps = 0;
long periode = 1000;
int etatLED = LOW;

void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  unsigned long maintenant = millis();

  if(maintenant - dernierTemps > periode) {
    dernierTemps = maintenant;
    if (etatLED == LOW)
      etatLED = HIGH;
    else
      etatLED = LOW;
    digitalWrite(13, etatLED);
  }
}
```

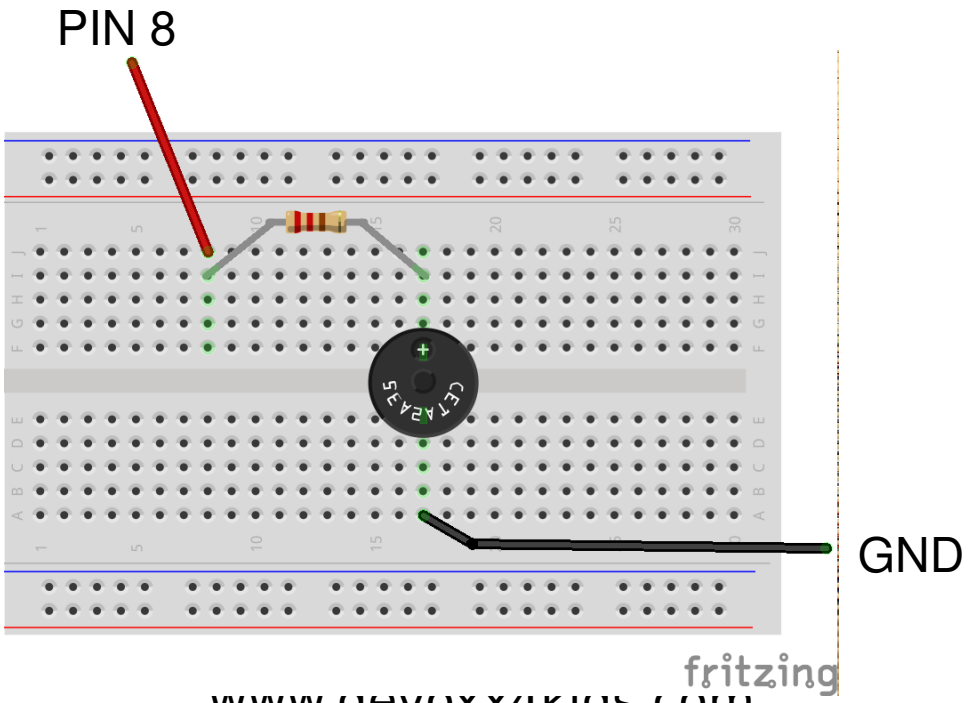
> Et si on buzzait... en utilisant le **buzzer** (composant qui fait pouet-pouet)
> On va s'en servir comme **haut-parleur**!!



> vidons d'abord la ~~planche à pain~~... heu la breadboard.
> ajoutons le buzzer et connectons à cheval sur les deux côtés



> Puis les derniers ingrédients, pour qu'on puisse faire quelque chose avec... sans tout griller (oui oui griller)
> Attention les fils sont maintenant sur la **Terre** et le pin **8**



Un peu de zik'mu?

```
int zikmu[] = {262,196,196,220,196,0,247,262};
int pouetDuree[] = {4,8,8,4,4,4,4,4 };

void setup() {
  for (int pouet = 0; pouet < 8; pouet++) {
    long duree = 1000/pouetDuree[pouet];
    tone(8, zikmu[pouet],duree);
    delay(duree * 1.4);
    noTone(8);
  }
}

void loop() {
}
```

> Et si on jouait avec... de la lumière?
> Jusque là, la résistance laissait passer un courant fixe!

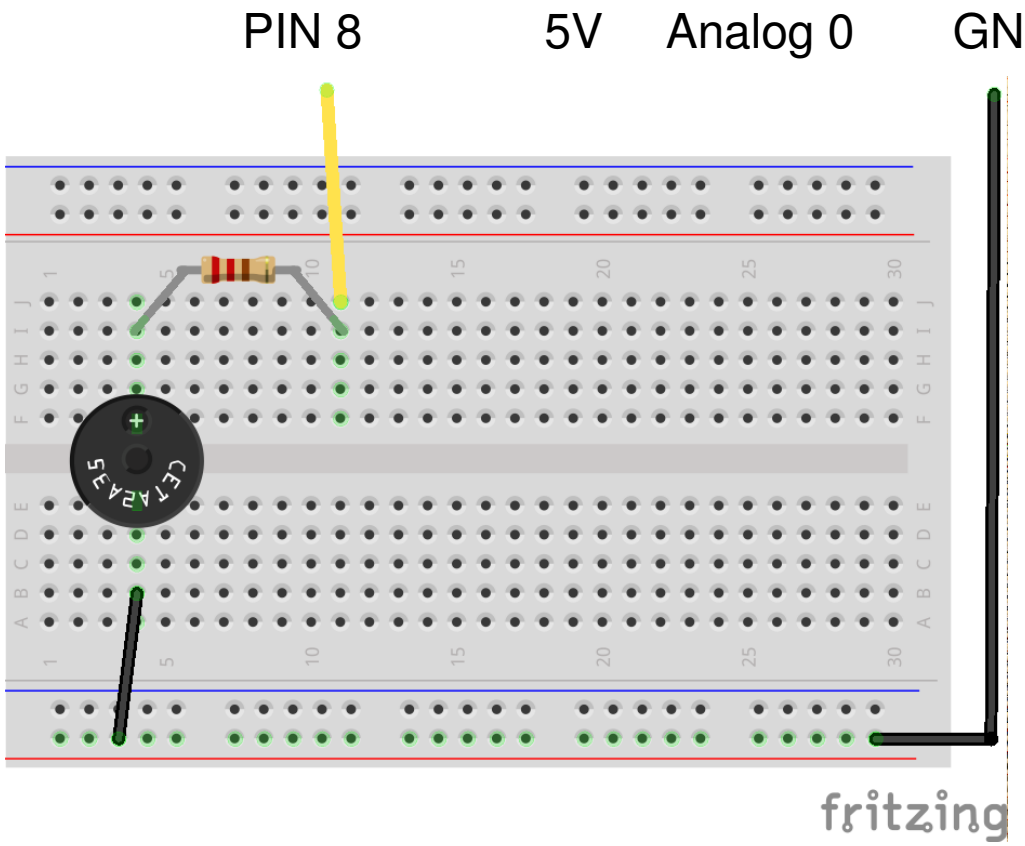
> Mais il existe aussi des Photorésistances, elles varient avec la lumière!
> Ca ressemble à ça (et c'est dans votre boîte!)

> On va l'utiliser en mettant:
> le **+** sur du 5V
> le **–** sur un pin **analogique**

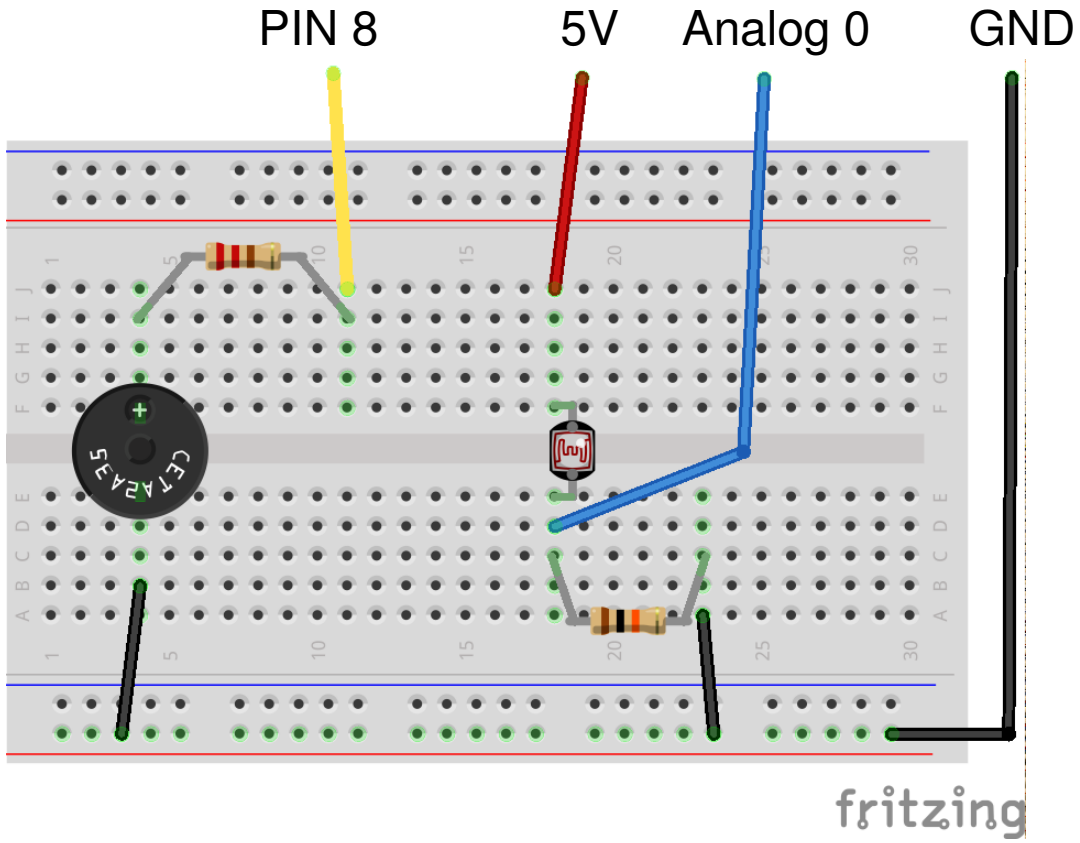
> un pin analogique peut lire/écrire plein de **nombres**, pas seulement 0 ou 1

> Ca va nous servir pour savoir **combien** de lumière est présente!

Et maintenant on retrousse ses manches... et on y va pour l'étape 1 !



Étape 2: le circuit de mesure



> De la zik'mu... et c'est nous qu'on joue!

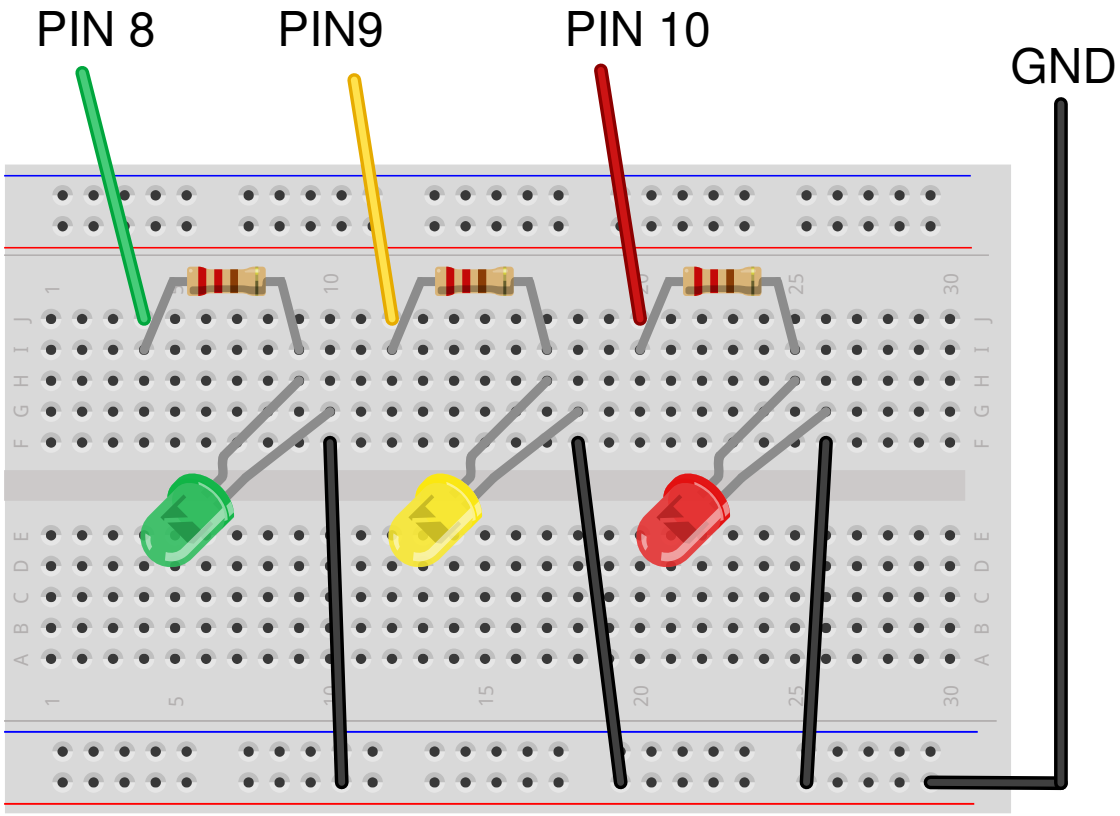
```
void setup() {
  // Configuration de la communication
  Serial.begin(9600);
}

void loop() {
  // Lire l'entrée analogique 0
  int lumiere = analogRead(A0);
  // S'il se bloque sur l'ordinateur, on peut lire la valeur
  Serial.println(lumiere);
  // Maintenant, nous prenons l'entrée (de 400 à 1000)
  // Et nous la traduisons à la sortie (entre 120 et 1500Hz)
  // Ces nombres peuvent changer avec la lumière dans la pièce
  int tonaliteCourante = map(lumiere, 400, 1000, 120, 1500);
  // Jouer le ton
  tone(8, tonaliteCourante, 10);
  delay(1);        // Courte pause, donne un meilleur résultat
}
```

> Maintenant, nous allons créer un feu de circulation avec 3 LEDs (vert / orange / rouge)

> On raccorde ces LEDs aux pins 8, 9 et 10 de la même manière qu'au début!

> On va les faire clignoter, et dans l'ordre s'il-vous-plaît!

DEVOXX™ 4KIDS



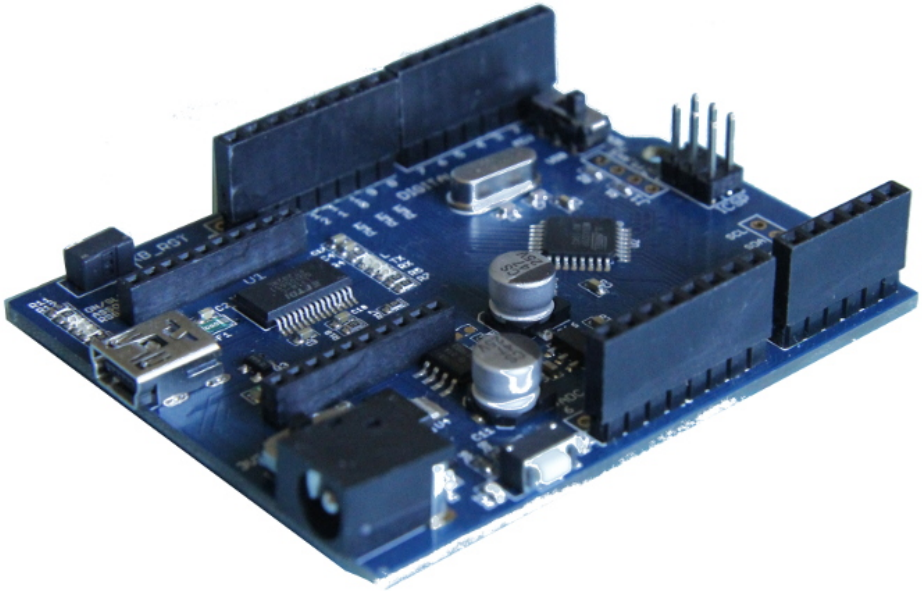
Made with Fritzing.org

www.devoxx4kids.com


```
long dernierTemps = 0;
long periode = 1000;
int ledOn = 0;

void setup() {
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
}

void loop() {
  unsigned long maintenant = millis();
  if(maintenant - dernierTemps > periode) {
    dernierTemps = maintenant;
    ledOn = (ledOn+1) % 3;
    digitalWrite(8,  0==ledOn);
    digitalWrite(9,  1==ledOn);
    digitalWrite(10, 2==ledOn);
  }
}
```

DEVOXX™ 4KIDS

Super les gars!



www.devoxx4kids.com