

MEETUP MAKER GIRLS

Introduction à l'Arduino

(Tutorial basé sur celui de Limor Fried qui est disponible à <http://www.ladyada.net/learn/arduino> et celui de Sparkfun disponible à <https://learn.sparkfun.com/tutorials/what-is-an-arduino>, ainsi que du matériel pris du site Arduino (<http://www.arduino.cc/>), tous les trois sous une licence [CC BY-NC-SA 3.0](#), ce qui est aussi la licence de ce tutorial)

Sommaire

- Installation
- L'électronique de base
- « Hello World! » électronique
- Programmation de base
- Utilisation des composants

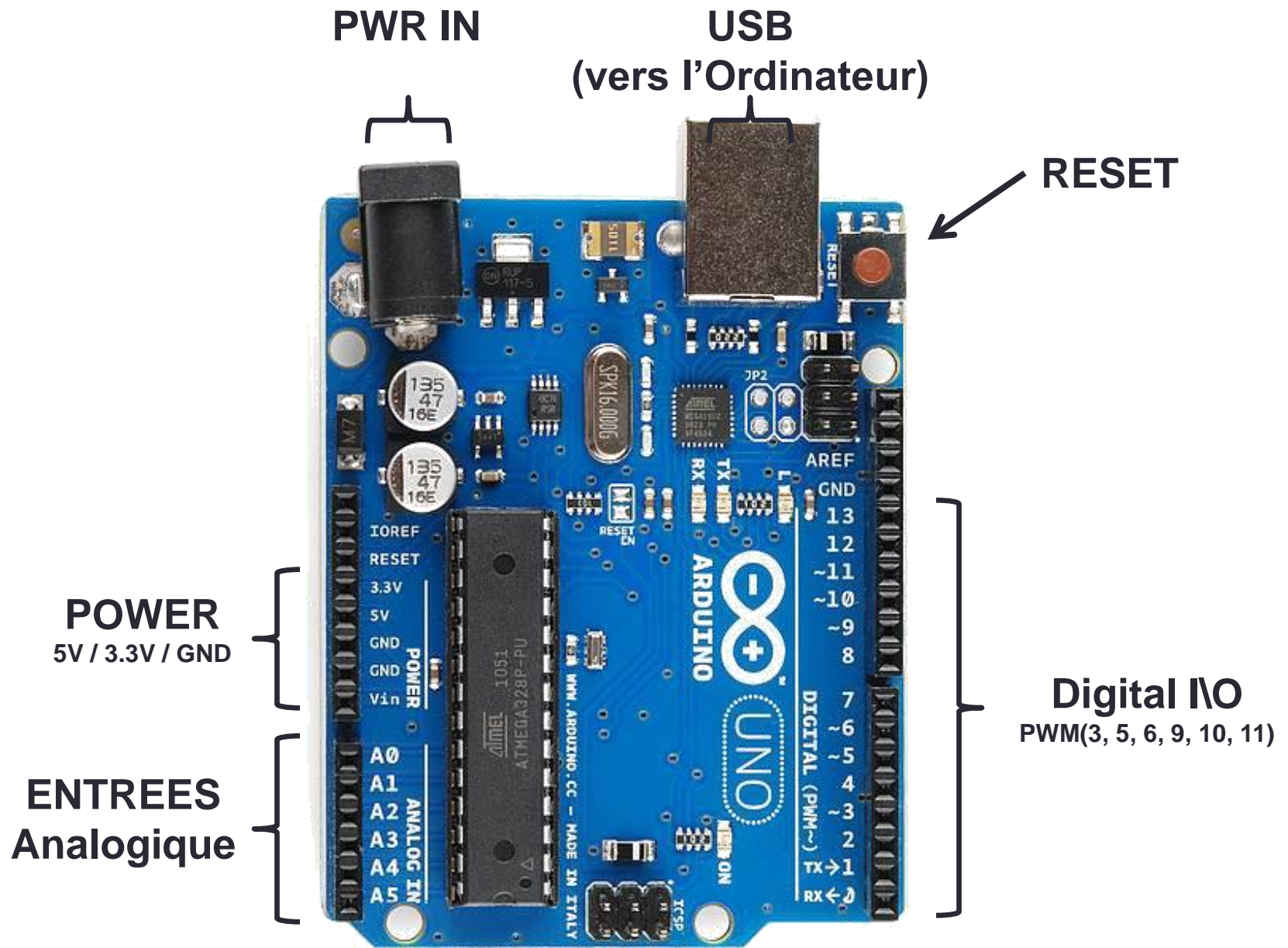
Sommaire

- **Installation**
- L'électronique de base
- « Hello World! » électronique
- Programmation de base
- Utilisation des composants

Installation

- La Carte Arduino c'est:
 - Du matériel open-source
 - Créé en Italie en 2005
 - Avec un processeur Atmel
 - Programmation accessible
 - Pour des artistes, bricoleu(ses)(rs), étudiant-e-s





Installation

- Environnement de développement pour Arduino
 - <http://arduino.cc/en/Main/Software>

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.0.3". The menu bar includes "File", "Edit", "Tools", and "Help". The toolbar contains icons for opening files, saving, compiling, uploading, and a serial monitor. The main text area displays the "Blink" example code. The code includes a multi-line comment describing the function, a pin definition for pin 13, a setup function to initialize the pin as an output, and a loop function that toggles the LED on and off with 1000ms delays. The status bar at the bottom shows "24" on the left and "Arduino Mega (ATmega1280) on /dev/tty.usbserial-A600enbz" on the right.

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

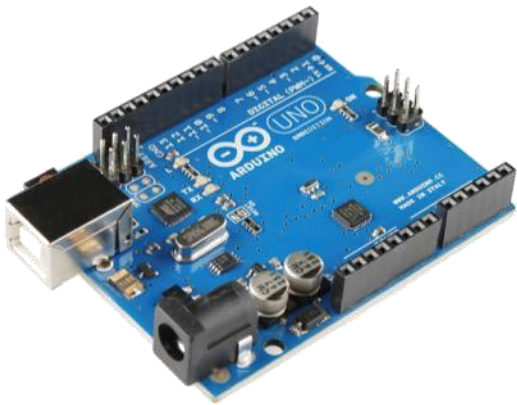
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

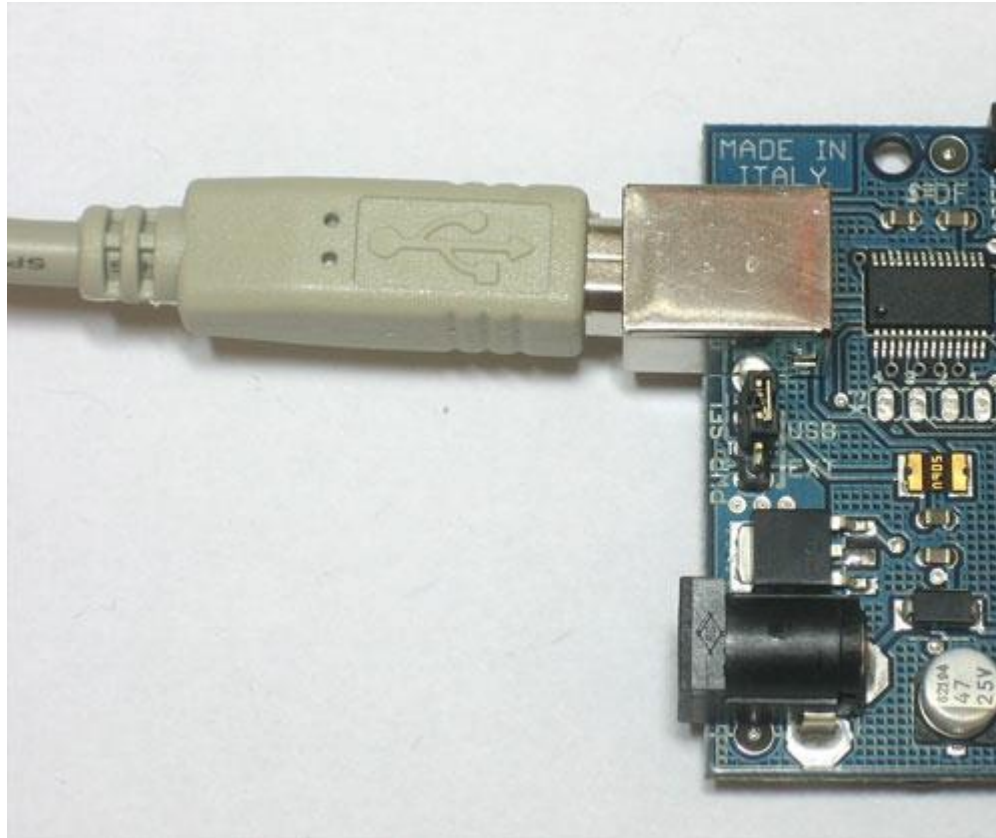
// the loop routine runs over and over again forever:
void loop(){
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

Installation

- Connectez la carte Arduino à l'ordinateur avec un câble USB
- La DEL verte (PWR) devrait être allumée



Installation



Installation: Le pilote

- Cliquez sur « Démarrer » et ouvrez le Panneau de Configuration
- Choisissez « Système »
- Choisissez « Gestionnaire de périphériques »
- Sous « Ports (COM et LPT) » un port « Arduino UNO (COMxx) » devrait y être
- Cliquez droit sur le port « Arduino UNO (COMxx) » et choisissez « Mettre à jour le pilote »
- Choisissez « Recherchez un pilote sur mon ordinateur »
- Parcourez vers le répertoire où vous avez installé le logiciel Arduino et choisissez le fichier « arduino.inf » qui devrait être dans le répertoire « Drivers »
- Laissez Windows compléter l'installation
- Pour Windows XP: <http://arduino.cc/en/Guide/UnoDriversWindowsXP>

Sommaire

- Installation
- L'électronique de base
- « Hello World! » électronique
- Programmation de base
- Utilisation des composants

La Charge Electrique

- La **tension**
 - La différence de charge entre deux points
- Le **courant**
 - La vitesse à laquelle la charge s'écoule
- La **résistance**
 - La tendance d'un matériau à résister à l'écoulement de la charge (courant)
- Un **circuit** est une boucle fermée qui permet la charge de se déplacer d'un endroit à l'autre.
- Les **composants** dans le circuit nous permettent de contrôler cette charge et l'utiliser pour faire du travail

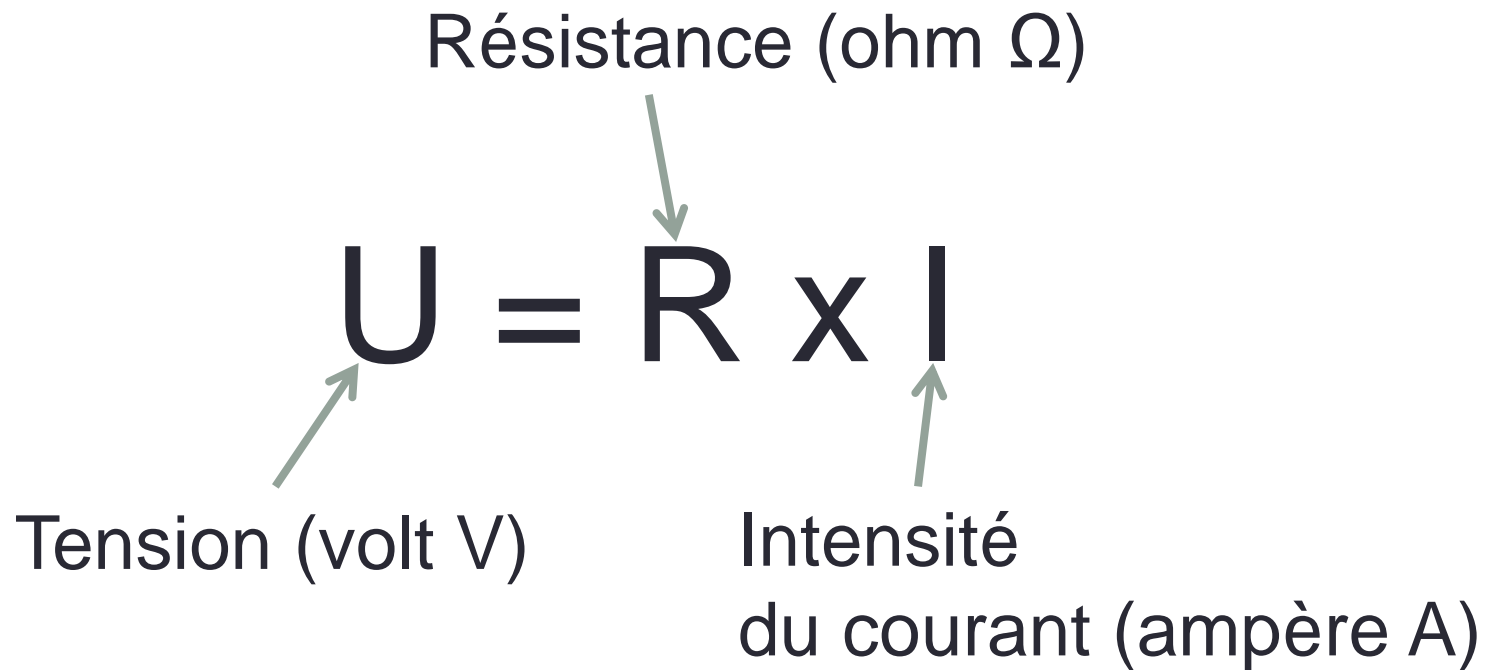
La Loi d'Ohm

Résistance (ohm Ω)

$$U = R \times I$$

Tension (volt V)

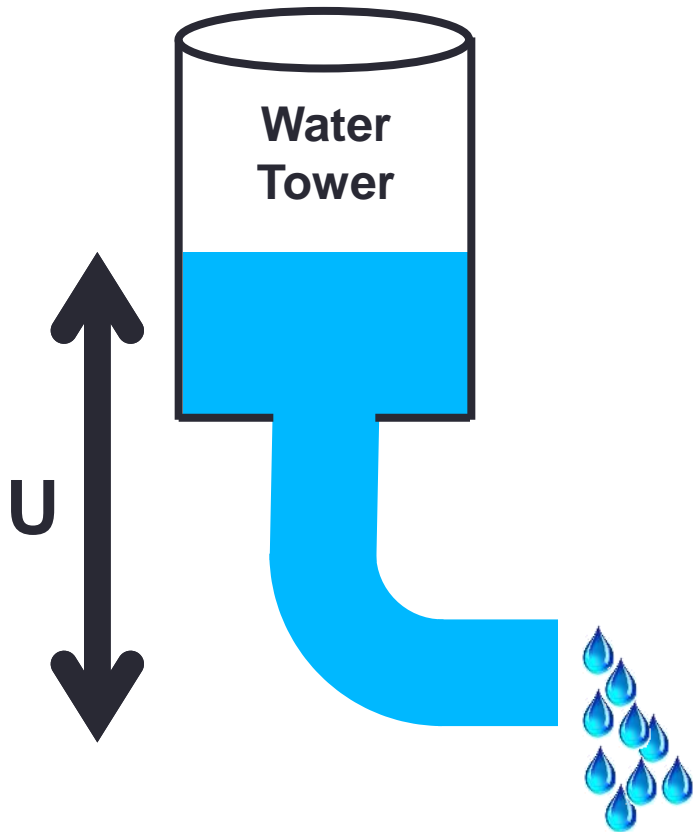
Intensité
du courant (ampère A)



The diagram illustrates Ohm's Law with the equation $U = R \times I$. Three labels with arrows point to the variables: 'Résistance (ohm Ω)' points to 'R', 'Tension (volt V)' points to 'U', and 'Intensité du courant (ampère A)' points to 'I'.

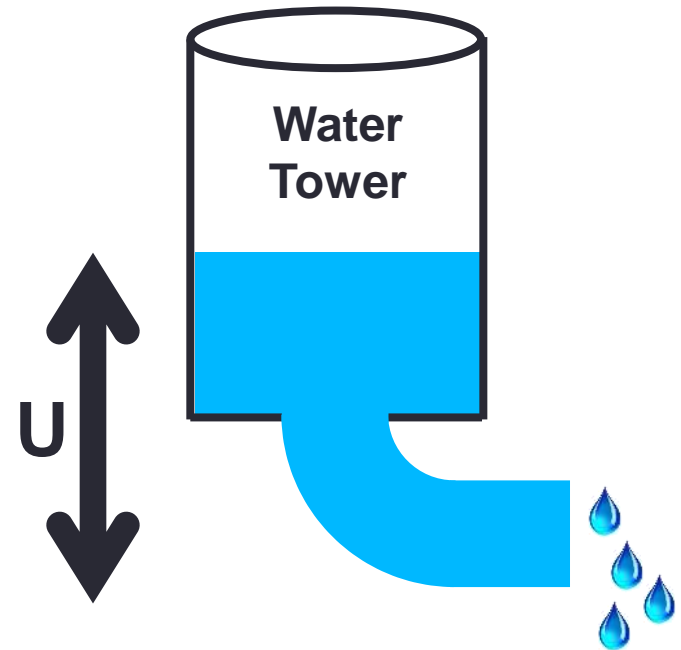
La Tension

$$U = R.I$$



Plus d'énergie == Plus de Tension

$$U = R.I$$



Moins d'énergie == Moins de tension

$$U = R.I$$

Le Courant

$$U = R.I$$



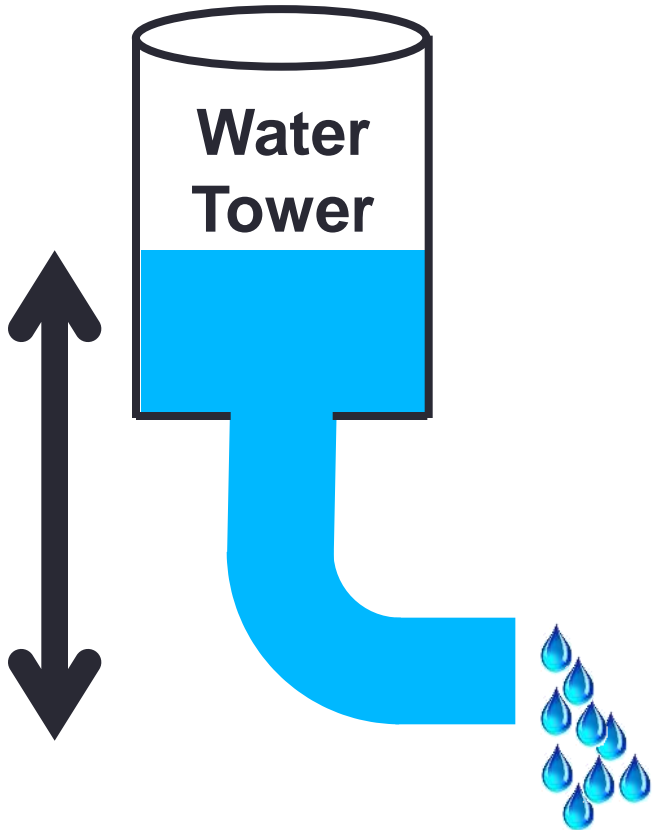
Courant élevé



Faible courant

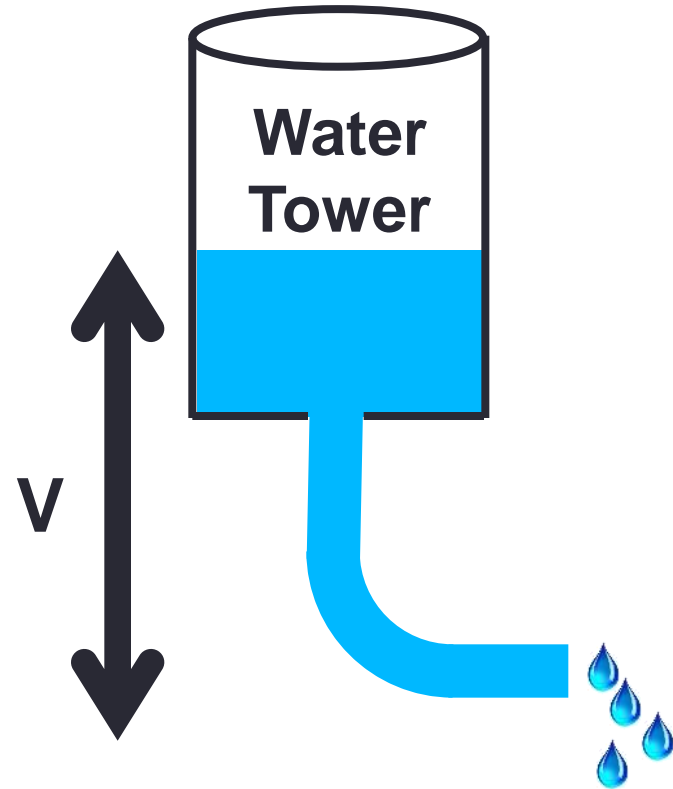
La Résistance

$$U = R \cdot I$$



Grand tuyau == Moins de Résistance

$$U = R \cdot I$$

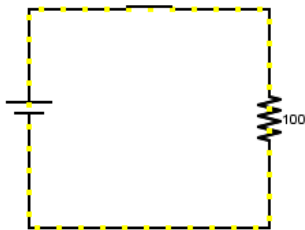


Petit tuyau == Plus de Résistance

$$U = R \cdot I$$

Le Circuit

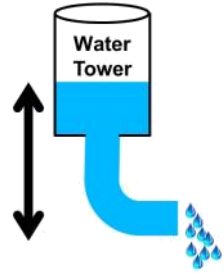
- Le circuit doit être connecté dans une boucle depuis l'alimentation (V_{cc}) en passant par des composants, à la terre (GND)
- Un multimètre peut analyser si le circuit a un problème de connexion (eg si la boucle est « cassée »)



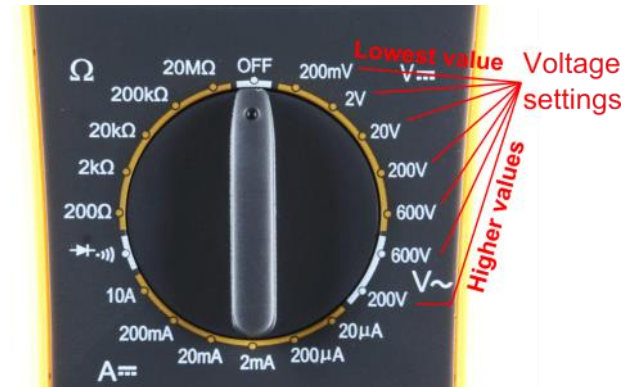
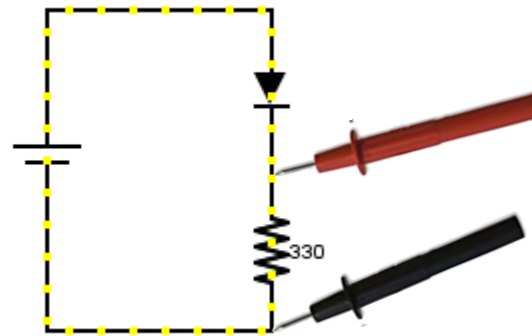
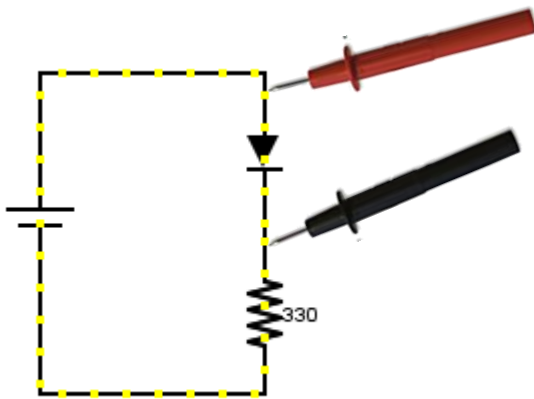
Continuity
setting



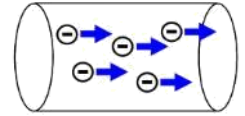
Comment Mesurer l'Electricité



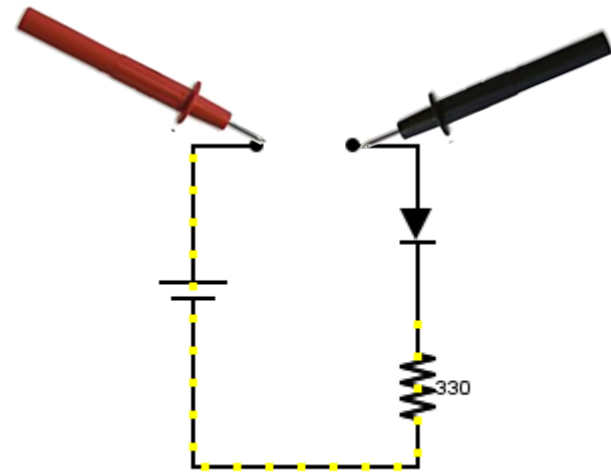
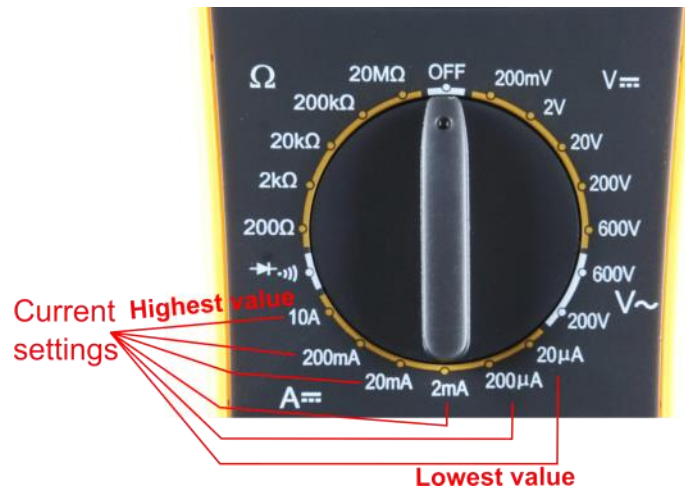
- La tension mesure la différence d'énergie entre deux points dans un circuit



Comment Mesurer le Courant



- Le courant est la mesure de la vitesse de la charge (le mouvement des électrons)
- Pour le mesurer, il faut casser le circuit



Comment Mesurer la Résistance



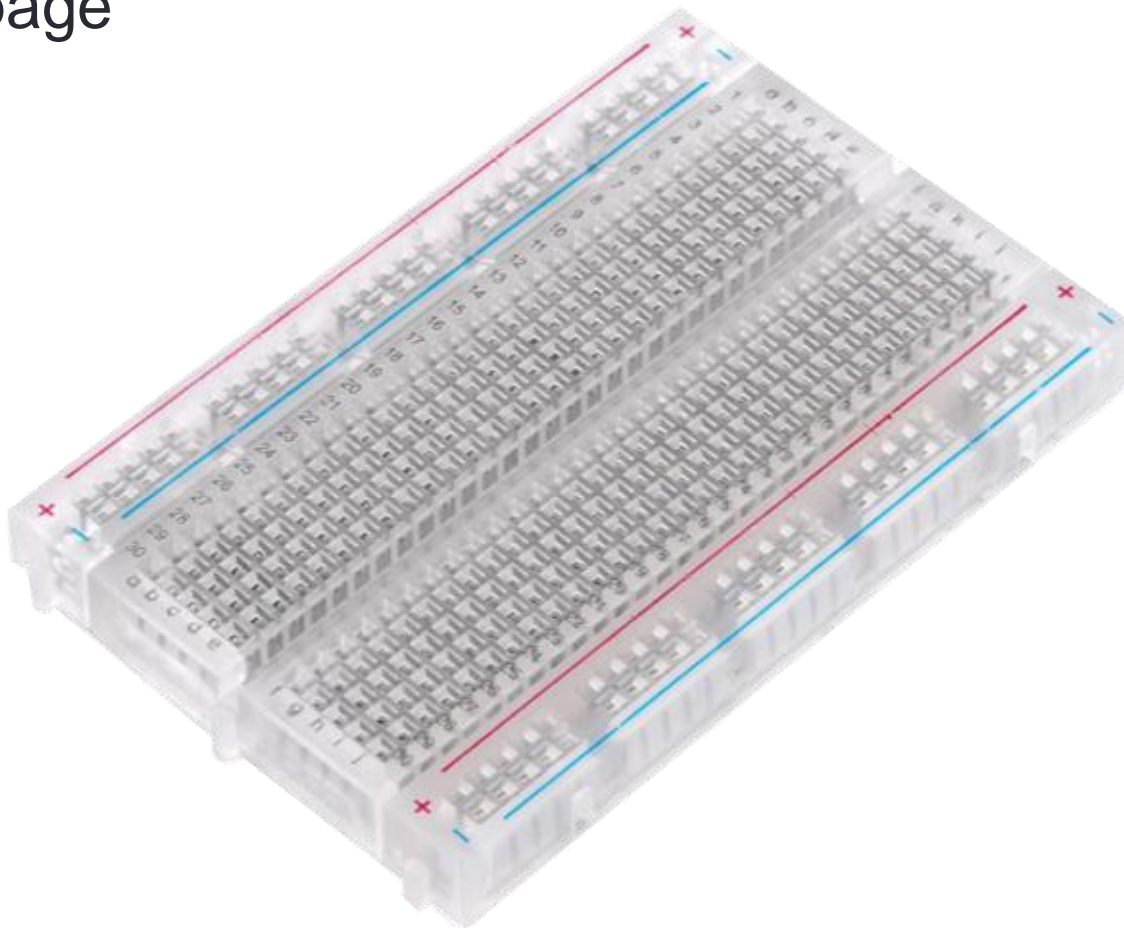
- La résistance est la mesure de l'opposition au flux du courant dans le circuit
- Les composants doivent être tous retirés du circuit pour mesurer la résistance

Resistance
settings

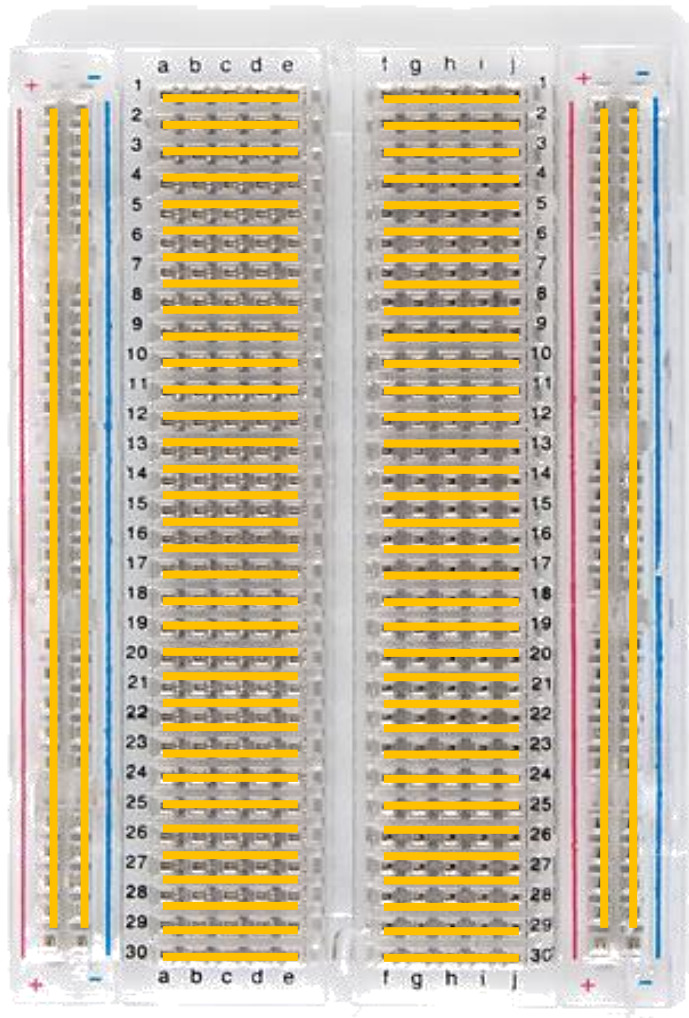


Le Breadboard

- Plus facile à utiliser (au lieu de souder le circuit) pour le prototypage



Le Breadboard



- Chaque ligne horizontale (de 5 trous) est connectée
- Les colonnes verticales à gauche et à droite sont aussi connectées et servent de connexion à l'alimentation et à la terre

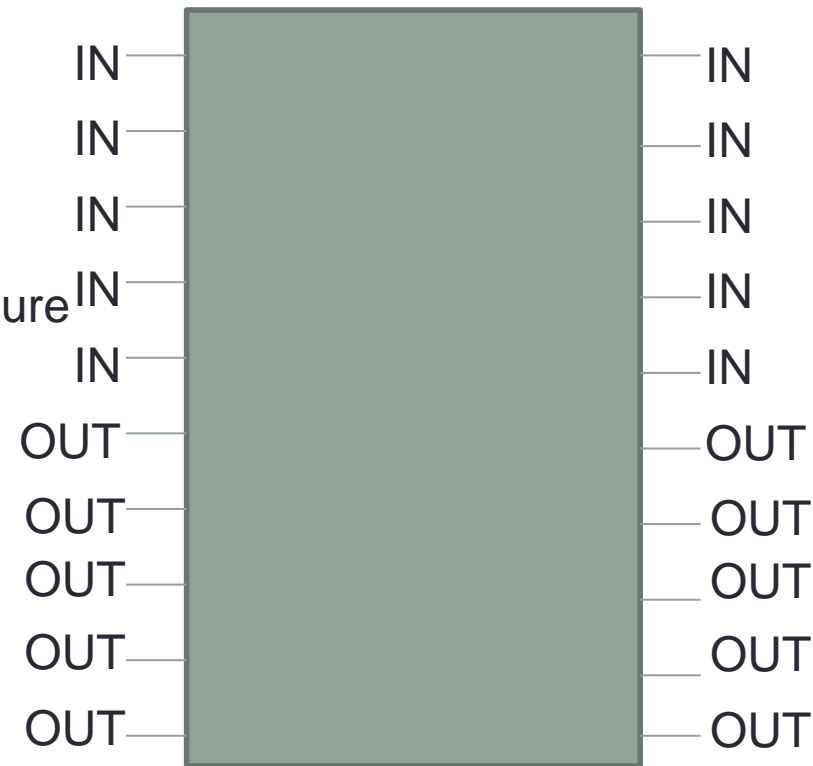
Les Périphériques

- Les Entrées

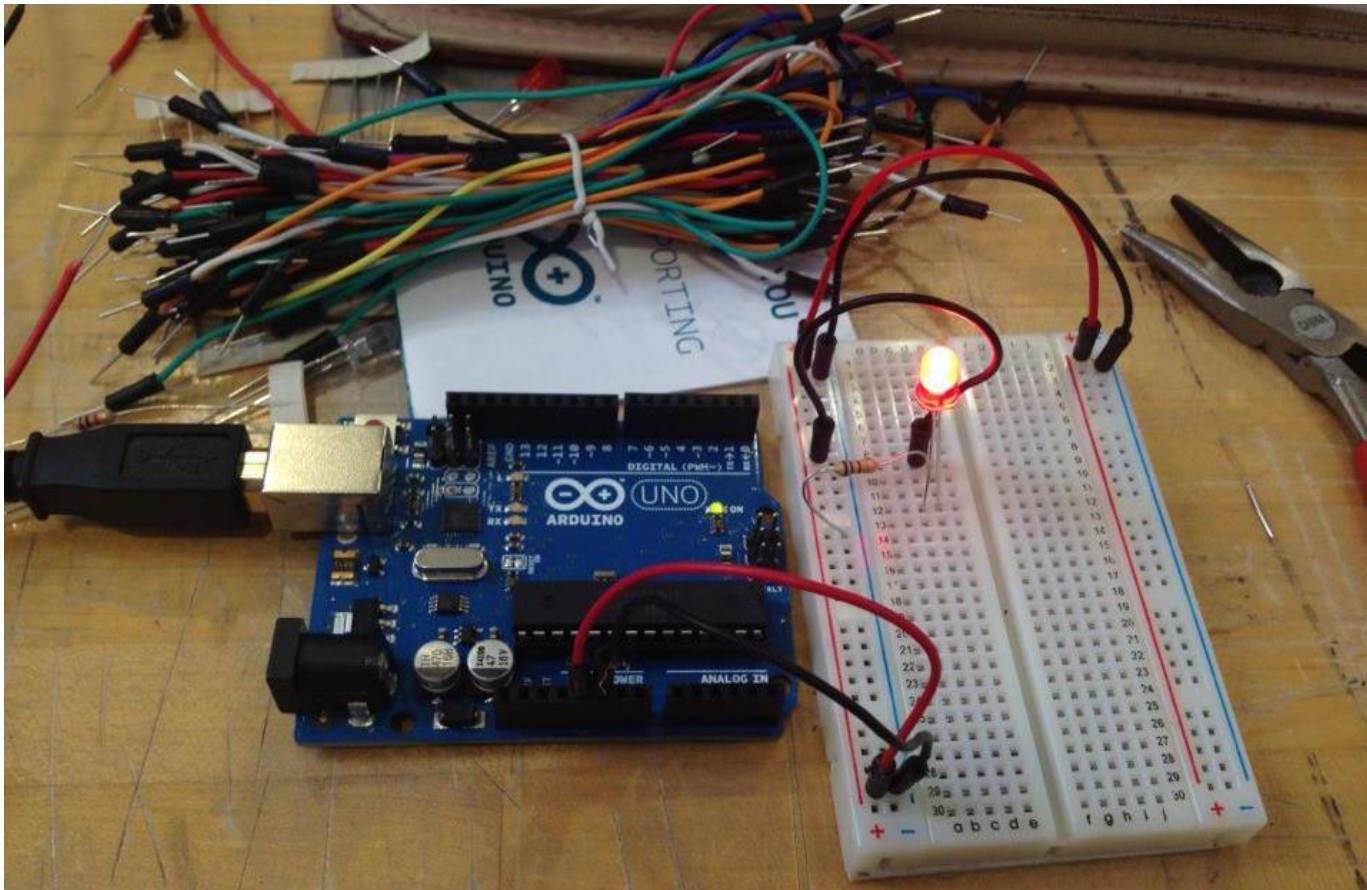
- Un signal qui va dans la carte
- Exemples:
 - Bouton-poussoir
 - Interrupteurs
 - Capteurs de lumière/humidité/température

- Les Sorties

- Un signal qui sort de la carte
- Exemples:
 - DELs
 - Moteurs
 - Buzzers piezo



Créons un circuit!



Sommaire

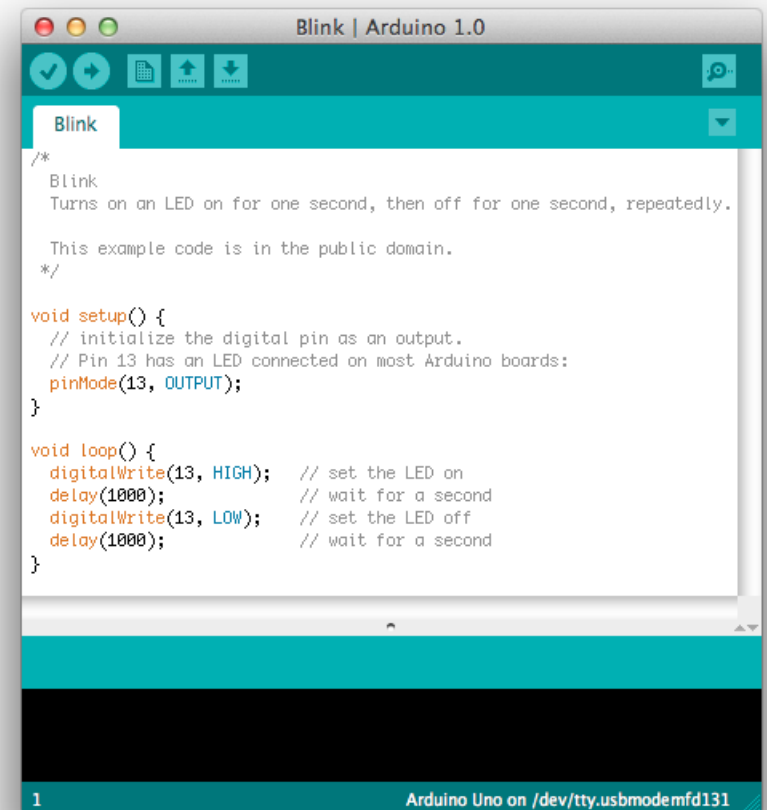
- Installation
- L'électronique de base
- « Hello World! » électronique
- Programmation de base
- Utilisation des composants

Le « Hello World! » de l'Electronique

- Une DEL qui clignote!
- Après cette étape, on peut avancer vers les concepts (encore) plus amusants, où on écrit nos propres programmes (« sketch »)

Le « Hello World! » de l'Electronique

- Ouvrez un workspace Arduino avec l'environnement de développement que l'on vient d'installer
- Ouvrez le projet « Blink »:
- File -> Examples -> 1.Basics -> Blink



The screenshot shows the Arduino IDE interface with the 'Blink' example code loaded. The title bar reads 'Blink | Arduino 1.0'. The code is as follows:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

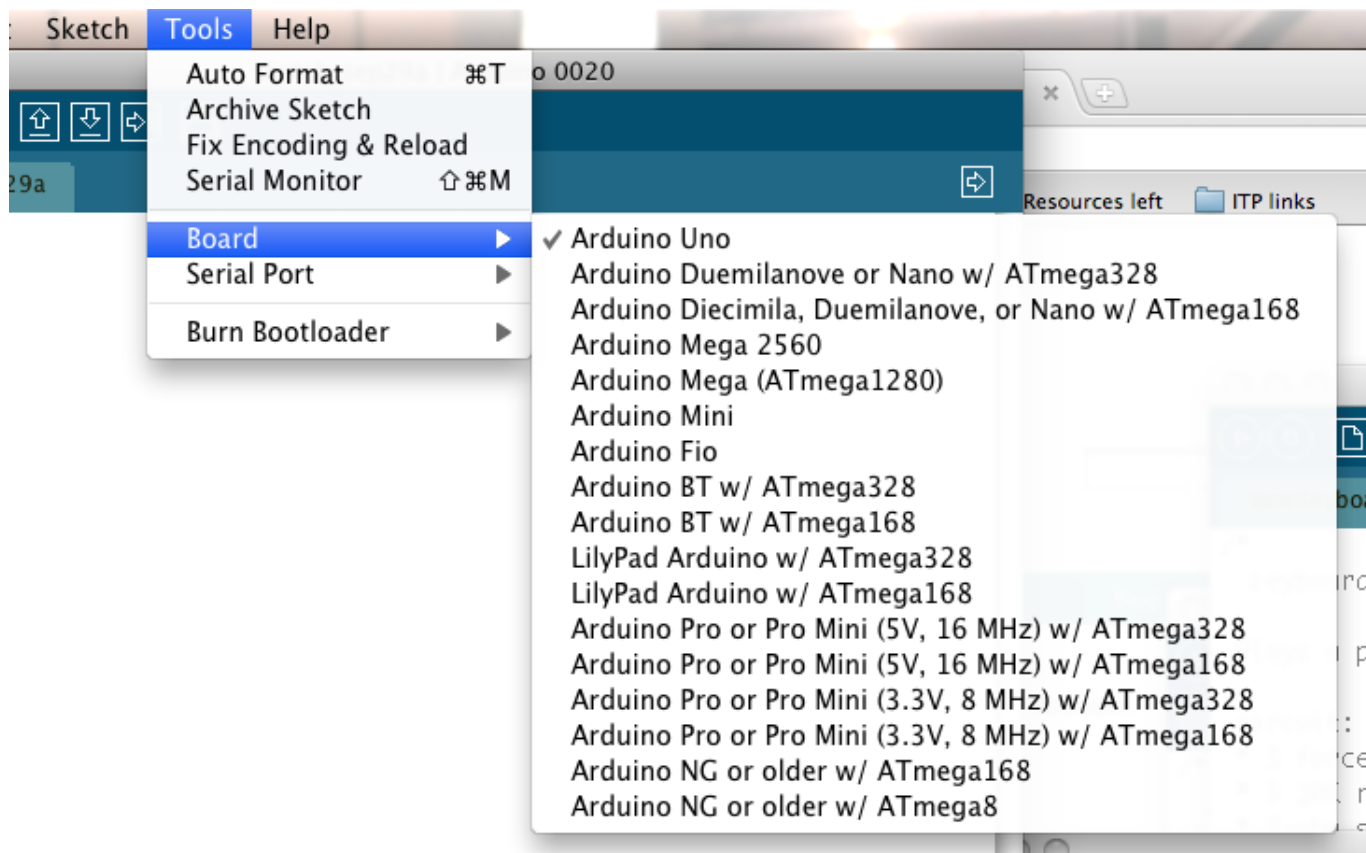
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);             // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);             // wait for a second
}
```

The status bar at the bottom indicates '1' and 'Arduino Uno on /dev/tty.usbmodemfd131'.

Le « Hello World! » de l'Electronique

- Sélectionnez la carte
 - Il faut configurer le logiciel Arduino pour votre type de carte Arduino



Le « Hello World! » de l'Electronique

- Selectionnez le port série:
 - Tools -> Serial Port
- Ce port devrait etre COM3 ou plus
- Pour etre sur, déconnectez votre Arduino et rouvrez le menu, le port qui disparaît devrait etre le port pour la carte Arduino
- Reconnectez la carte et choisissez ce port

Le « Hello World! » de l'Electronique



Code pour
programmer la
carte

Status

Console de
notification

Le « Hello World! » de l'Electronique

- La première étape est de vérifier et compiler le code
 - C'est-à-dire vérifier qu'il n'y a pas de fautes et
 - Le traduire dans une application qui est compatible avec le matériel Arduino
- Choisissez Sketch -> Verify/Compile
- Si tout va bien un message « Done compiling » devrait être affiché dans le status et puis le « Binary sketch size » dans le console de notification

Le « Hello World! » de l'Electronique

- Téléchargez le program sur la carte Arduino avec le bouton « upload » dans l'environnement de développement
- Attendez un peu, les DELs RX et TX devraient clignoter
- Si le téléchargement a été un succès, un message « Done uploading » devrait etre affiché dans le status de l'environnement de développement
- La DEL 13 sur la carte devrait commencer à clignoter (orange)

Sommaire

- Installation
- L'électronique de base
- « Hello World! » électronique
- Programmation de base
- Utilisation des composants

Programmation de Base

```
/* * Blink
*
* L'exemple Arduino de base. Allume une DEL pour une seconde,
* puis l'éteint pour une seconde, en boucle... On utilise la broche 13
* parce que, en fonction de votre carte Arduino, il a soit une DEL embarquée
* ou de la place pour une DEL.
*
* http://www.arduino.cc/en/Tutorial/Blink */

int ledPin = 13;          // DEL connectée à la broche numérique 13

void setup()              // s'exécute une seule fois, quand le programme commence
{
    pinMode(ledPin, OUTPUT); // définit la broche numérique en tant que sortie
}

void loop()               // tourne en boucle
{
    digitalWrite(ledPin, HIGH); // allume la DEL
    delay(1000);                // attend une seconde
    digitalWrite(ledPin, LOW);  // éteint la DEL
    delay(1000);                // attend une seconde
}
```

Des Commentaires

```
/* * Blink
*
* L'exemple Arduino de base. Allume une DEL pour une seconde,
* puis l'éteint pour une seconde, en boucle... On utilise la broche 13
* parce que, en fonction de votre carte Arduino, il a soit une DEL embarquée
* ou de la place pour une DEL.
*
* http://www.arduino.cc/en/Tutorial/Blink */
```

- Aide les utilisateurs à comprendre ce que fait le code
- Commence avec `/*` et finit avec `*/` ou commence simplement avec `//`
- Tout ce qui est entre ces symboles est ignoré par l'Arduino
- Indispensable pour bien expliquer ton code!

Des Variables

```
int ledPin = 13;
```

```
// DEL connectée à la broche numérique 13
```

- Les déclarations finissent avec ;
- **int** veut dire « integer » (entier)
- **ledPin** est le nom du variable
- Le variable est initialisé avec la valeur **13**

Type de la boite	Nom de la boite	=	Ce que le boite contient
Entier (int)	ledPin	=	13

Des Fonctions

```
void setup()           // s'exécute une seule fois, quand le programme commence
{
    pinMode(ledPin, OUTPUT); // définit la broche numérique en tant que sortie
}
```

- Une séquence de tâches à suivre qui sont exécutées l'une après l'autre de haut en bas

Valeur de retour	Nom de la fonction	Paramètres (d'entrée)	{Des déclarations}
void	setup	()	{ pinMode(ledPin, OUTPUT); }

- **void** veut dire « rien ». Il n'y a pas de valeur de retour quand la fonction se termine
- On n'a qu'une ligne dans cette fonction, et c'est un appel à une autre fonction qui s'appelle **pinMode**

Des Fonctions

```
chat propre laver le chat (chat sale)    // une fonction pour laver le chat
{
    ouvre le robinet de la douche.
    va trouver le chat.
    prends le chat.
    mets le chat sous la douche.
    attends 3 minutes.                    // attends que le chat soit propre
    libère le chat
}
```

- Comme faire une tâche étape par étape
- Cette fonction est pour laver le chat. Le nom de la fonction est « laver le chat », ça prend en entrée un « chat sale » et a comme sortie un « chat propre »
- Les crochets { } indiquent le début et la fin de la fonction
- Dans la fonction il y a une liste de tâches qui décrivent le procédé de lavage du chat

Des Fonctions

```
void loop()           // tourne en boucle
{
    digitalWrite(ledPin, HIGH); // allume la DEL
    delay(1000);           // attend une seconde
    digitalWrite(ledPin, LOW);  // éteint la DEL
    delay(1000);           // attend une seconde
}
```

- Une autre fonction **loop**, qui ne prend pas d'entrée et qui n'a pas de sortie
- A l'intérieur, 4 appels aux fonctions
- **digitalWrite** et **delay** sont des fonctions, avec leurs commentaires à côté

Des Fonctions

- Sur l'Arduino, les fonctions **setup()** et **loop()** sont des fonctions spéciales:
 - Après chaque reset, l'Arduino fait ce qu'il a à faire dans la fonction **setup()**
 - Et puis il fait ce qu'il est précisé dans la fonction **loop()** en boucle (à l'infini)

Modifions le programme

- La fonction `delay(1000)` attend 1000 millisecondes
- Changez l'appel à la fonction `delay()` pour que l'attente soit 500 millisecondes
- Sauvegardez le programme dans un nouveau fichier
- Compilez et téléchargez le programme pour la mettre sur l'Arduino
- La DEL devrait clignoter plus rapidement




Modifions le programme

- Exercise 1:
 - Modifiez le code pour que la DEL soit allumée pour 100ms et éteint pour 900ms
- Exercise 2:
 - Modifiez le code pour que la DEL soit allumée pour 50ms et éteint pour 50ms
- Exercise 3:
 - Modifiez le code pour que la DEL soit allumée pour 10ms et éteint pour 10ms

Sommaire

- Installation
- L'électronique de base
- « Hello World! » électronique
- Programmation de base
- Utilisation des composants

Des Composants

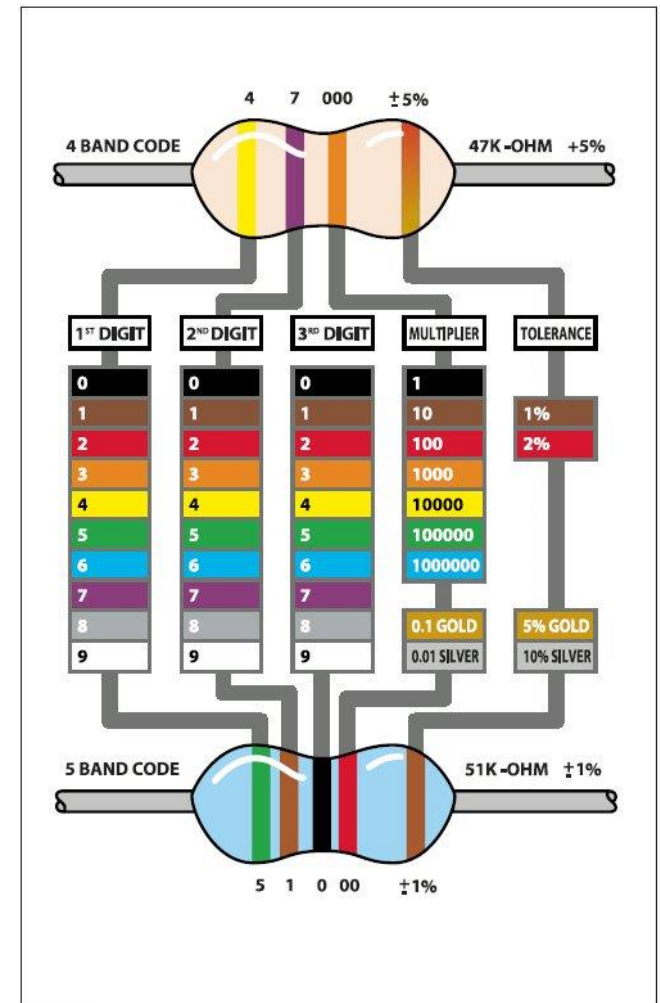
Nom	Image	Fonction
Résistance		Pour contrôler le courant
DEL (diode électroluminescente)		Crée de la lumière
Bouton-poussoir		Interrupteur dans le circuit (ouvre ou ferme le circuit)

La Résistance

- Les résistances sont codées avec des bandes de couleur pour signifier leur valeur
- Elles n'ont pas de « sens » de connexion

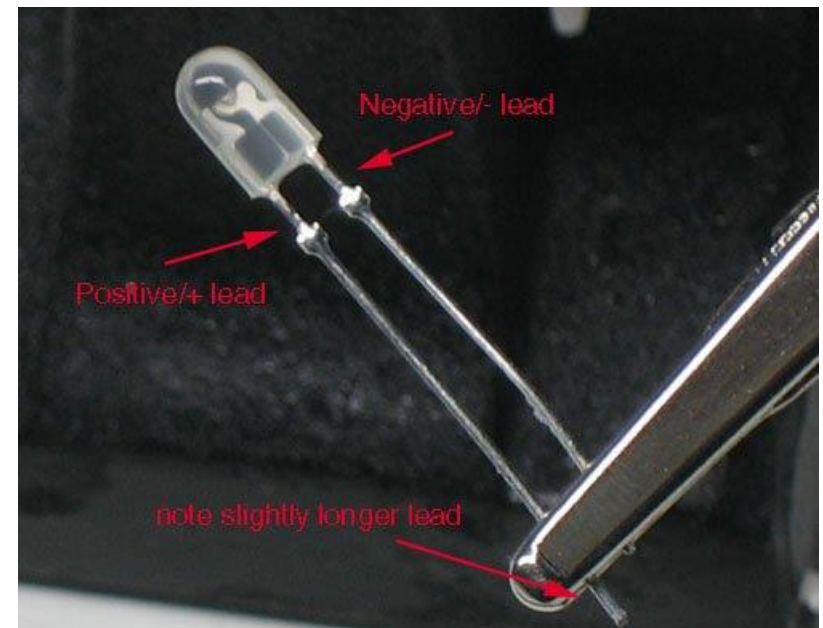
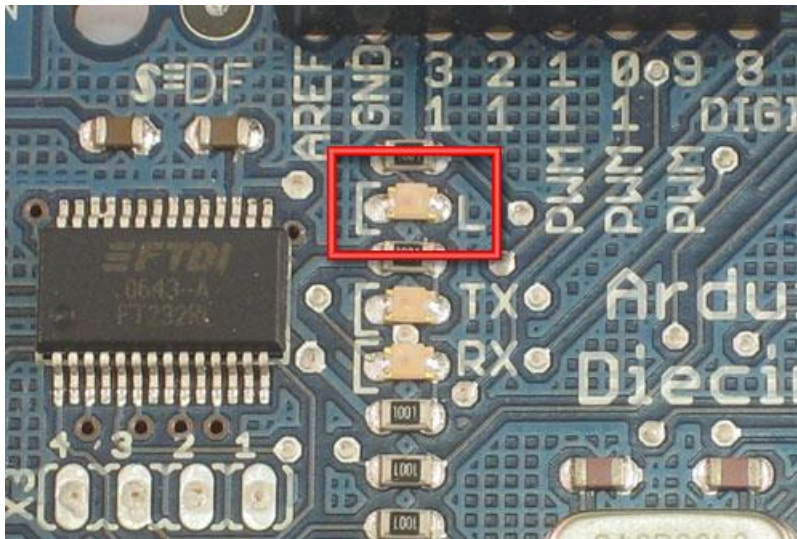


Ou un site web qui fait le calcul pour vous:
<http://www.dannyg.com/examples/res2/resistor.htm>



La DEL (Diode Electroluminescente)

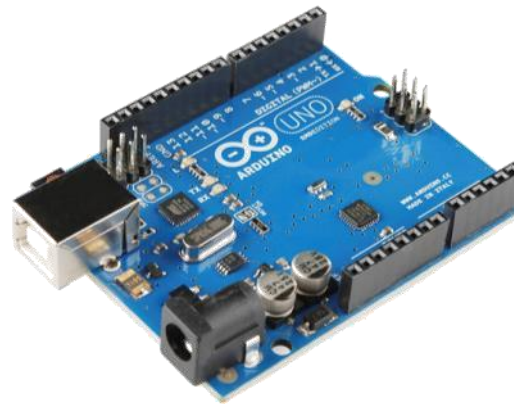
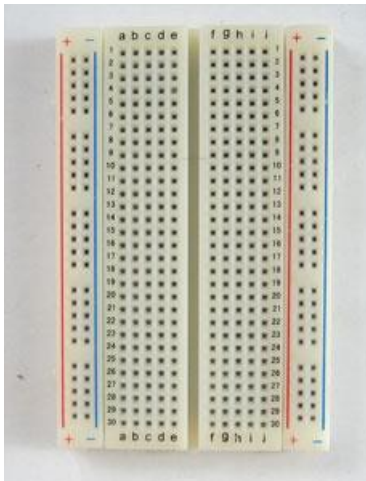
- Convertit du courant en lumière
- La diode est un chemin sens unique pour le courant
- Le courant va du côté positif au côté négatif
- Le côté positif est le pied plus long



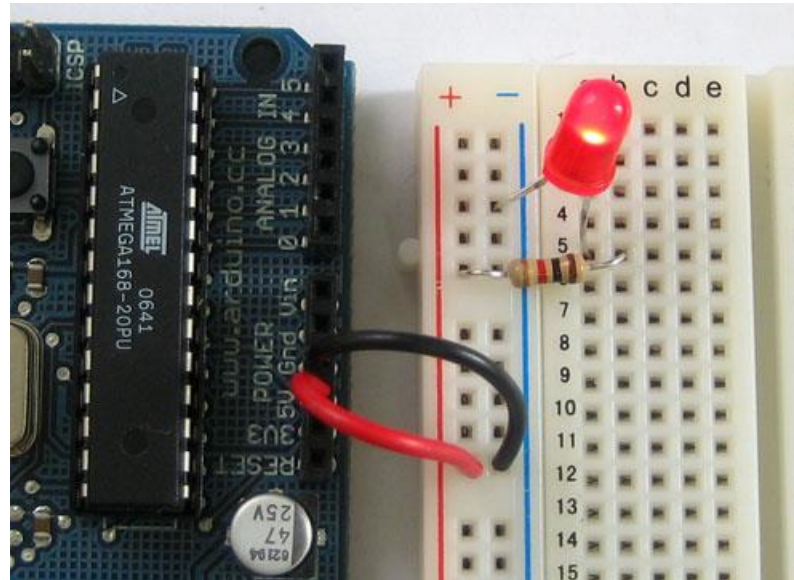
Construire un Circuit



1K Ω



Construire un Circuit



Construire un Circuit

- Il faut toujours mettre une résistance avec une DEL
- Mettez la résistance entre la ligne « + » du breadboard et un trou numérotée
- Positionnez la DEL entre la résistance et une ligne « - » du breadboard, comme montré dans le dessin. Le pied plus longue de la DEL doit etre connecté à la résistance, et le pied le plus court doit etre connecté au « - »
- Connectez une ligne « + » du breadboard au 5V de l'Arduino
- Connectez une ligne « - » du breadboard au GND de l'Arduino
- Allumez l'Arduino. La DEL devrait s'allumer

Symboles d'un Schéma du Circuit



Une résistance

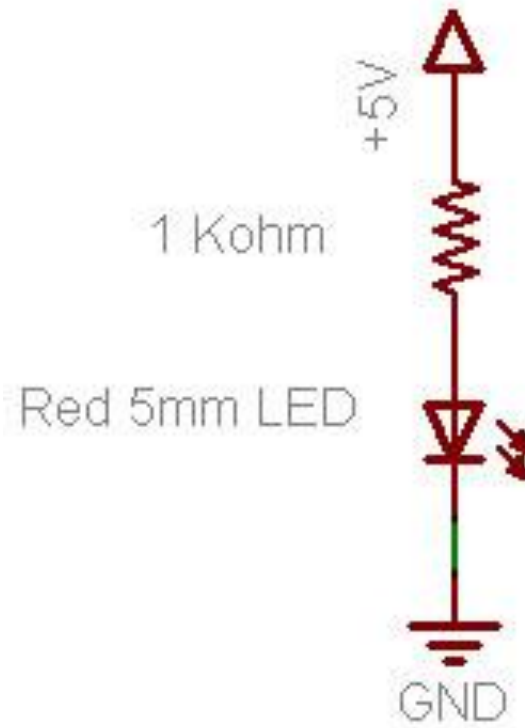


Une DEL



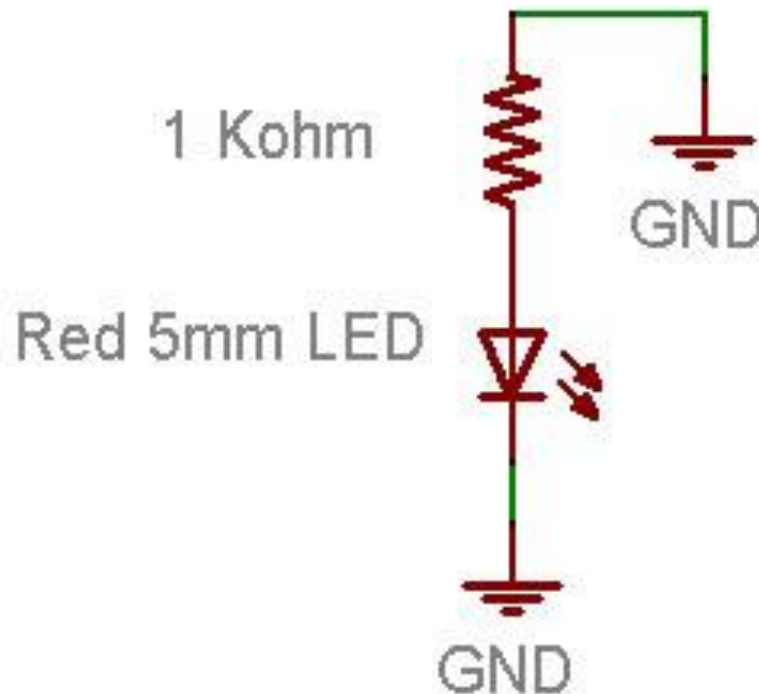
L'alimentation électrique
et la terre

Le Schéma du Circuit



Modifions le Circuit 1

- On va modifier le circuit pour ressembler au circuit ci-dessous. Qu'est-ce qui va se passer?
- Changez le circuit
- La DEL devrait rester éteint

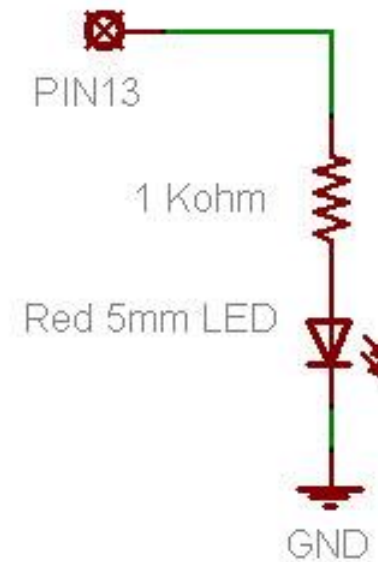


Pour Résumer

- Si on connecte le coté résistance à l'alimentation 5V, la DEL s'allume
- Si on connecte le coté résistance à la terre, la DEL s'éteint
- Et si on tournait ce procédure dans une boucle?
 - => la DEL devrait clignoter

Modifions le Circuit 2

- Ouvrez l'environnement de développement Arduino sur le programme « Blink ». Vérifiez que les délais entre l'éclairage et l'extinction de la DEL sont 500ms
- Téléchargez le programme sur votre Arduino
- Modifiez le circuit pour qu'il se ressemble au circuit suivant:
- La DEL devrait clignoter



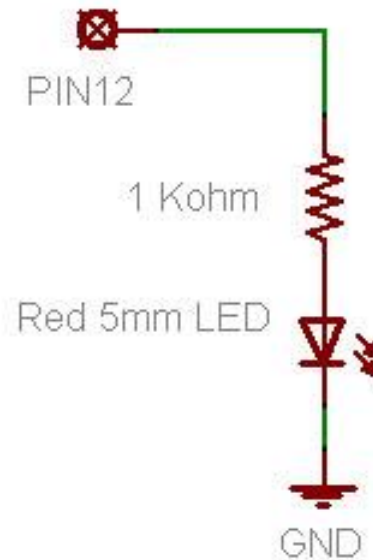
digitalWrite

```
void loop()           // tourne en boucle
{
    digitalWrite(ledPin, HIGH);    // allume la DEL
    delay(1000);                  // attend une seconde
    digitalWrite(ledPin, LOW);     // éteint la DEL
    delay(1000);                  // attend une seconde
}
```

- La fonction digitalWrite connecte la broche défini par le premier paramètre (« ledPin » dans notre exemple) soit à l'alimentation 5V soit à la terre, suivant le deuxième paramètre « HIGH » ou « LOW »

Modifions le Circuit 3

- Connectez la résistance à la broche 12 (au lieu de 13)
- La DEL ne clignote plus. Pourquoi?



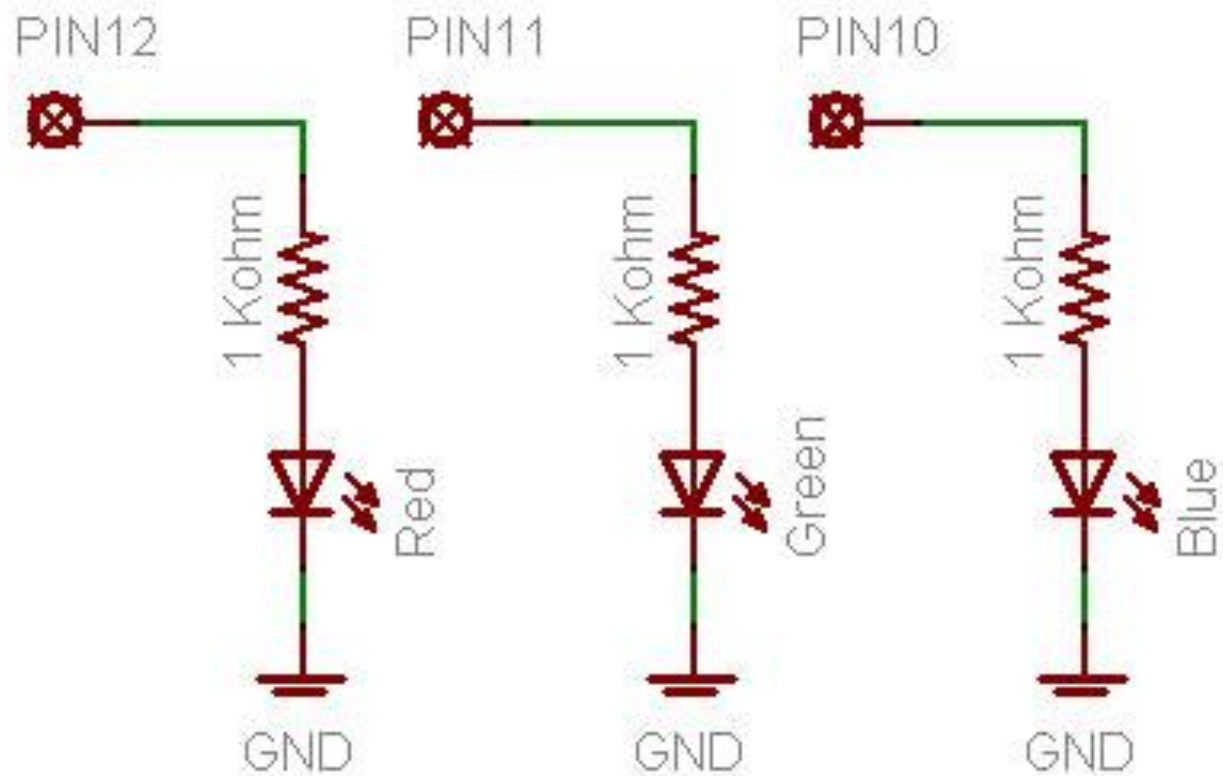
Modifions le Circuit 3

```
int ledPin = 13;
```

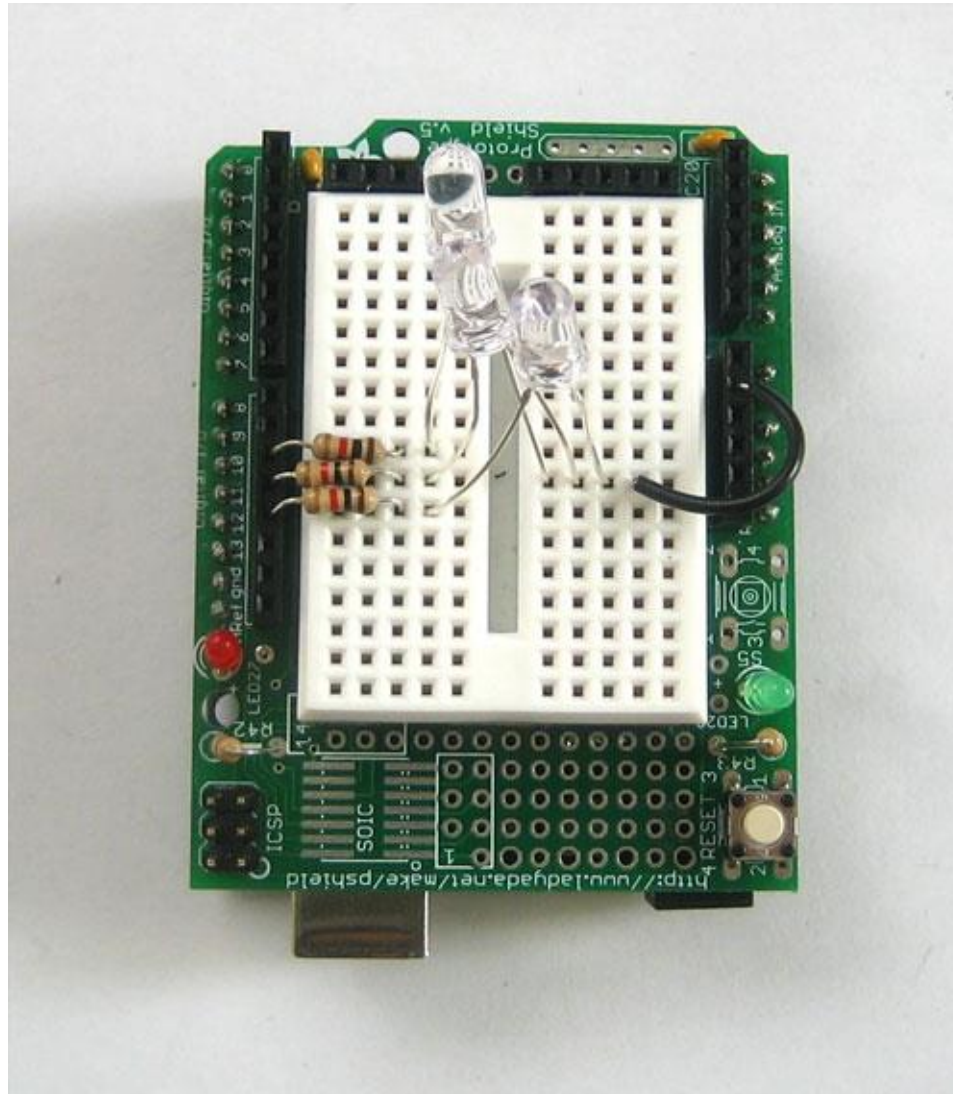
```
// DEL connectée à la broche numérique 13
```

- Revenez à la ligne ci-dessus de le programme
- Cette ligne précise à quelle broche est connectée la DEL
- Changez cette valeur à 12 pour l'aligner avec les changements faits au circuit
- Recompilez le code et téléchargez-le sur l'Arduino. La DEL devrait clignoter

Ajouter des DEL (pour avoir R, V, et B)



Ajouter des DEL RVB



Circuit RV

```
int brocheRouge = 12;           // DEL rouge connectée à la broche numérique 12
int brocheVerte = 11;          // DEL verte connectée à la broche numérique 11

void setup()                    // s'exécute une seule fois, quand le programme commence
{
    pinMode(brocheRouge, OUTPUT); // définit la broche numérique en tant
    // que sortie
    pinMode(brocheVerte, OUTPUT); // définit la broche numérique en tant que
    // sortie
}

void loop()                    // tourne en boucle
{
    digitalWrite(brocheRouge, HIGH); // allume la DEL
    digitalWrite(brocheVerte, HIGH); // allume la DEL
    delay(500);                    // attend une seconde
    digitalWrite(brocheRouge, LOW);  // allume la DEL
    digitalWrite(brocheVerte, LOW);  // allume la DEL
    delay(500);                    // attend une seconde
}
```

Circuit RV

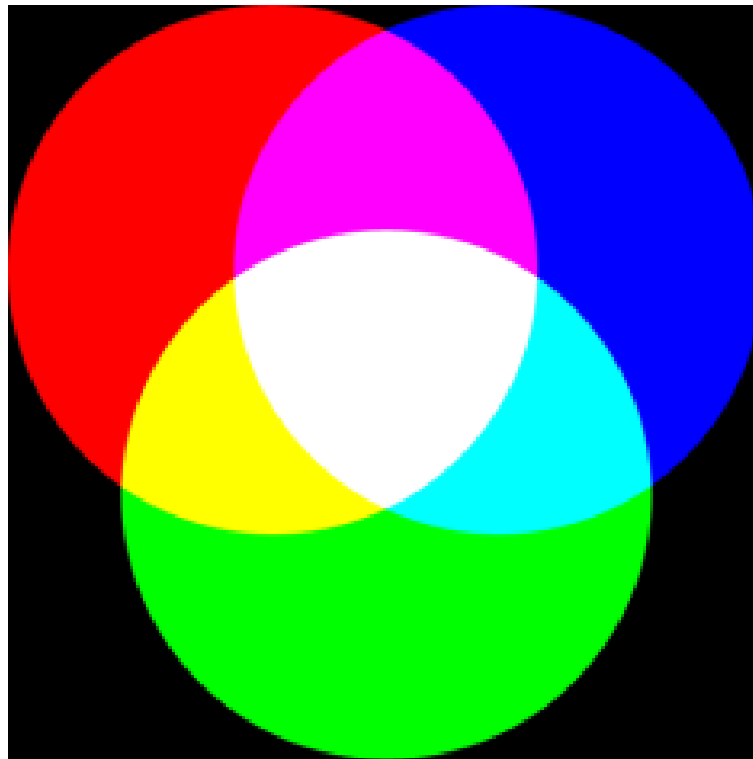
- Exercice:
 - Changez le code pour que les DELs clignotent l'un après l'autre (quand la DEL rouge est allumée, la DEL verte devrait être éteinte, et vice versa)
- Exercice:
 - Changez le code dans la fonction « loop » au programme suivant, avec chaque tâche l'une après l'autre, pour tourner en boucle:
 - Tous les deux DELs allumées pour 500ms, puis
 - Seule la DEL rouge allumée pour 500ms, puis
 - Toutes les deux DELs éteintes pour 500ms, puis
 - Seule la DEL verte allumée pour 500ms

Circuit RVB

- Revenons au code du début avec les DELs rouge et verte
- Exercice: Ajoutez le code pour la DEL bleue vous-mêmes:
 - Créez un variable pour la broche bleue. A quelle broche doit-il etre connecté? Examinez le circuit
 - Ajoutez une ligne pour dire à l'Arduino que la broche bleue va etre une sortie numérique
 - Ajoutez du code pour dire que la DEL bleue va etre allumée au meme temps que les DELs rouge et verte

Circuit RVB

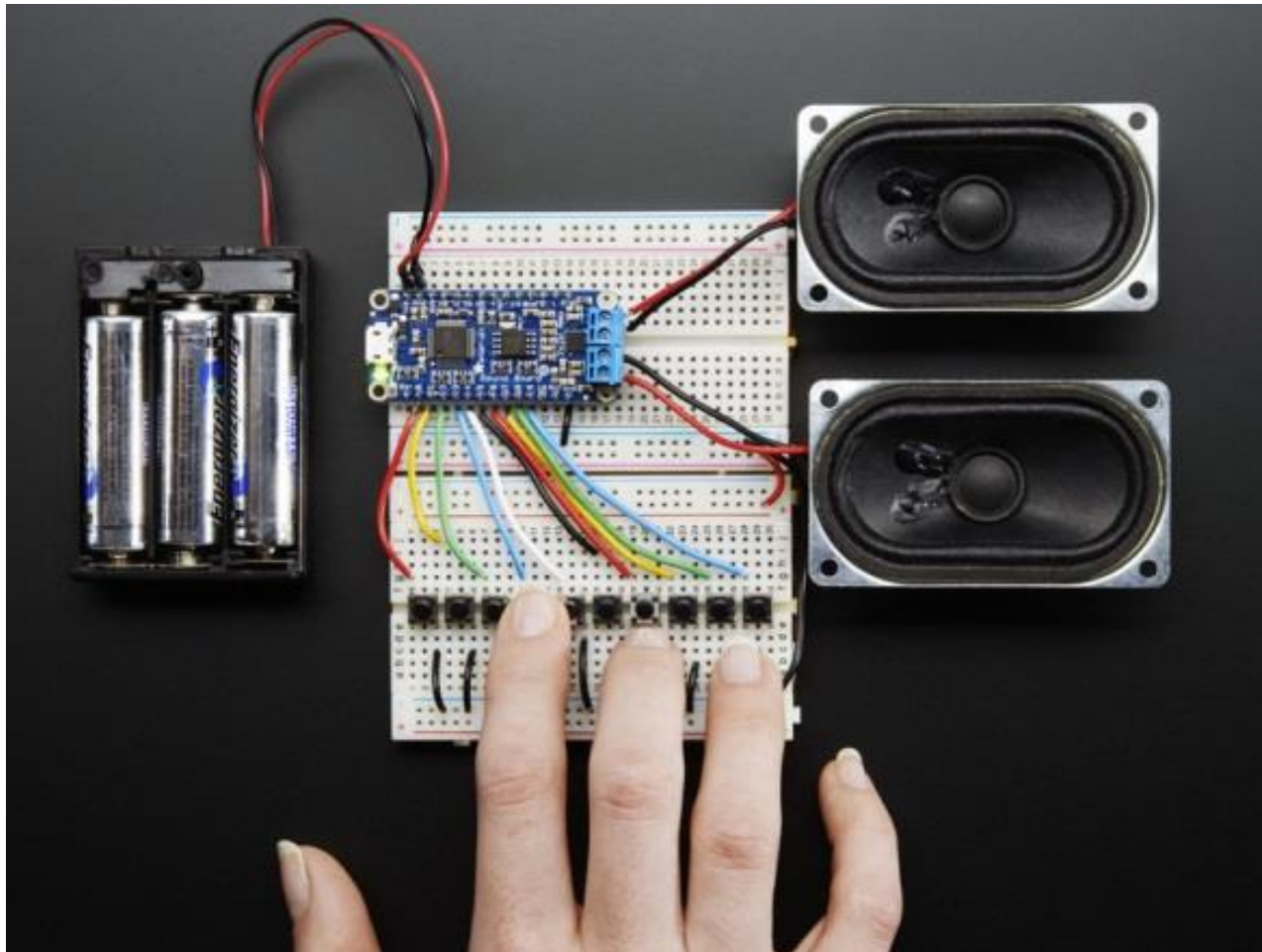
- Mélanger des couleurs
- Vos yeux mélange des couleurs pour en créer d'autres



Circuit RVB

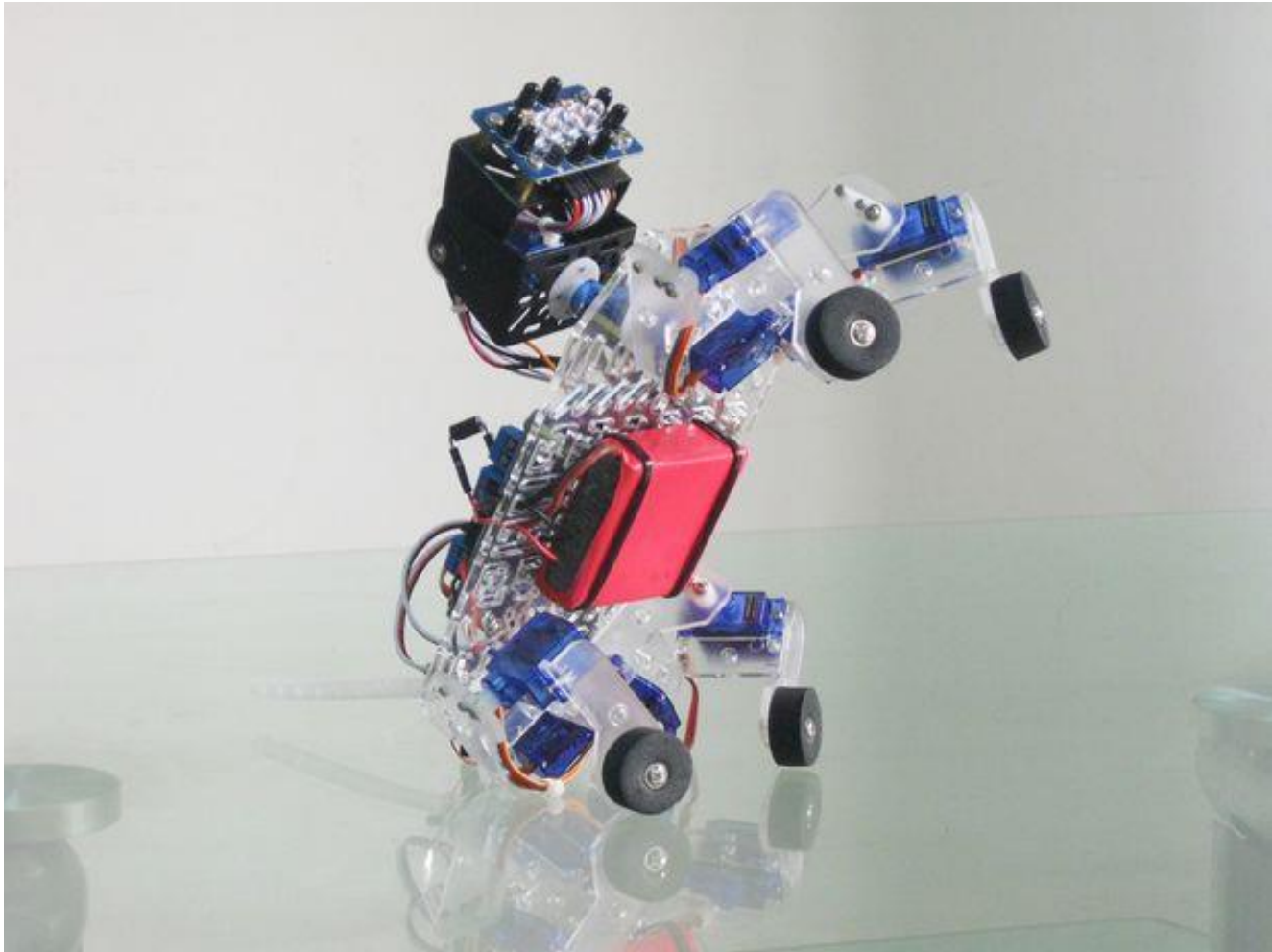
- Exercice: Modifiez le code pour créer des couleurs suivantes:
 - Violet (rouge et bleu)
 - Turquoise (bleu et vert)
 - Jaune (vert et rouge)
- Exercice:
 - Modifiez le code pour que les DELs changent de couleur entre rouge, jaune, vert, turquoise, violet et encore rouge (dans l'ordre et en boucle)
 - Les DELs devraient attendre une demi-seconde entre chaque changement

Projets Arduino



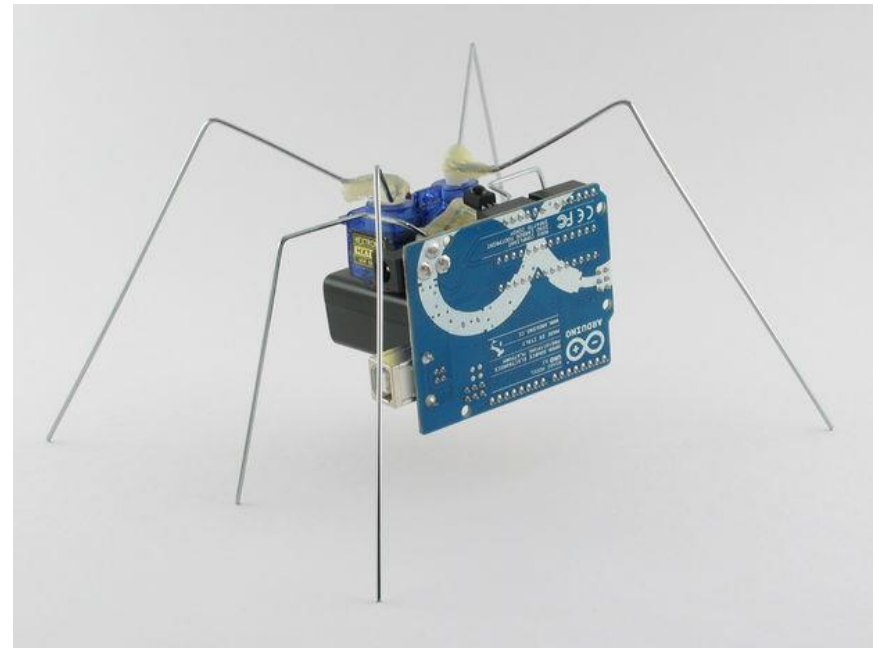
<https://learn.adafruit.com/adafruit-audio-fx-sound-board>

Projets Arduino



<http://www.instructables.com/id/Playful-Puppy-Robot/>

Projets Arduino

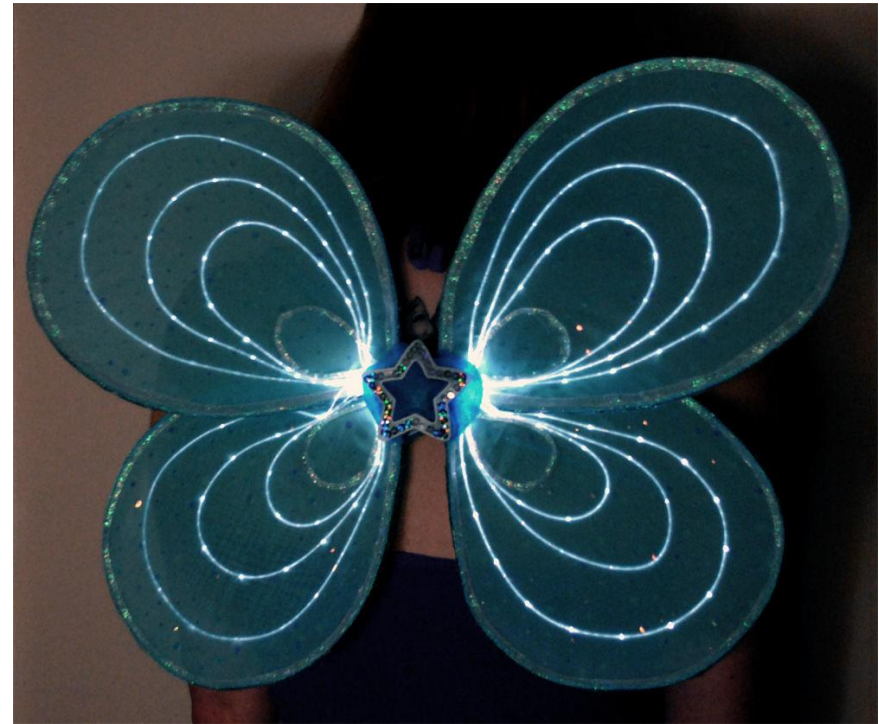
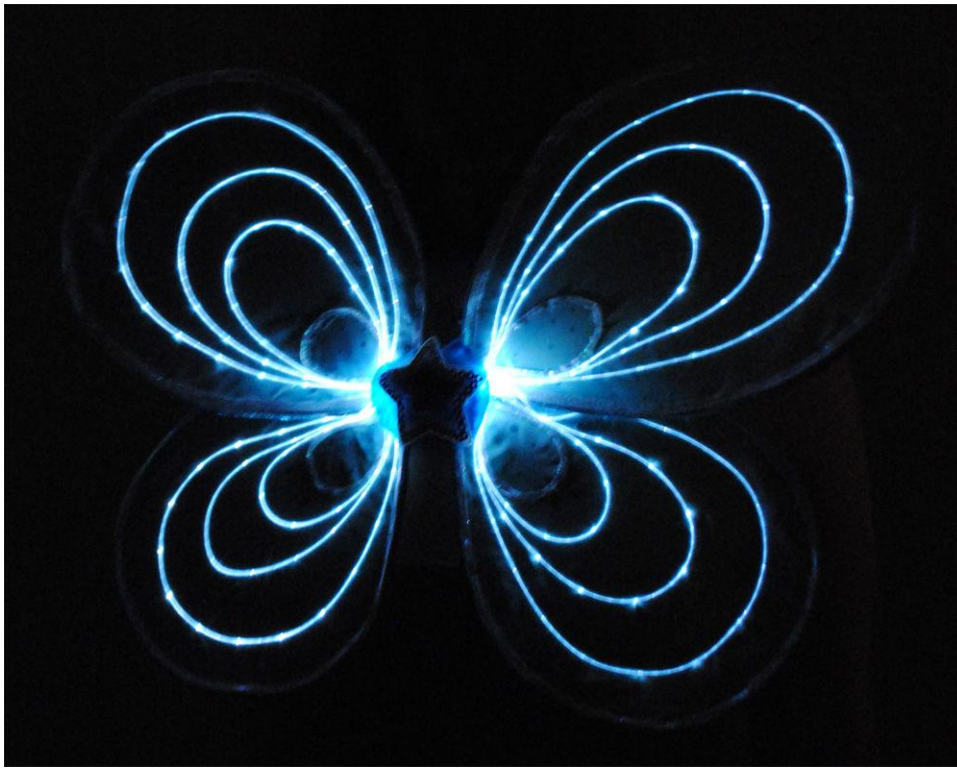


Les E-Textiles



<https://users.soe.ucsc.edu/~emme/>

Les E-Textiles



<http://www.instructables.com/id/Lightwings-Fiber-Optic-Fairy-Wings/>

Les E-Textiles



<http://makezine.com/2008/06/23/led-turn-signal-bike-jack/>

Des Circuits sur Papier

