

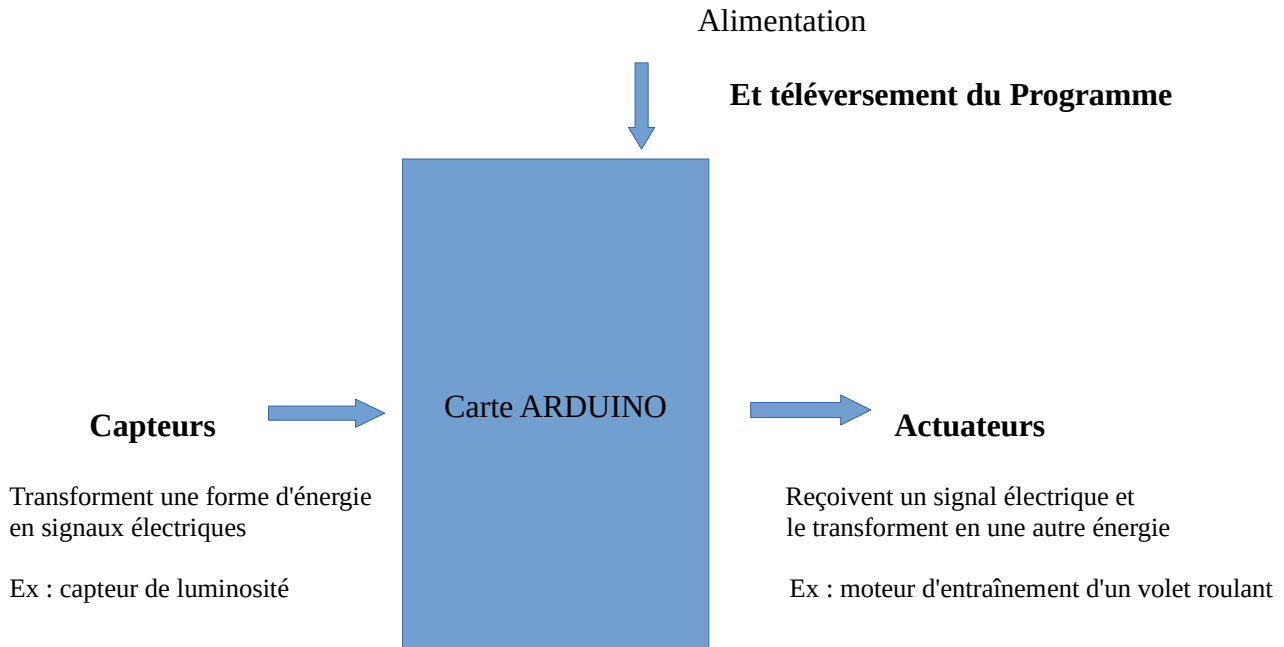
ATELIER n°2 ARDUINO

Club de Robotique et d'Électronique Programmable de Ploemeur

I- BUT de l' ATELIER :

Connaître vos outils et programmer le clignotement d'une LED.

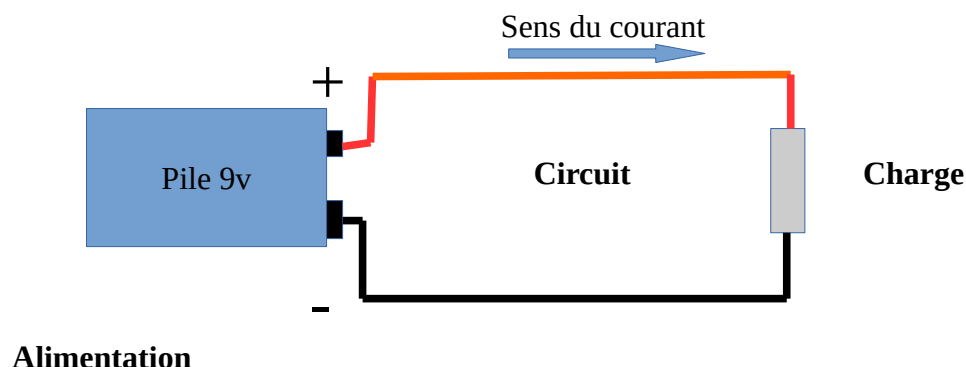
II- PRINCIPE GÉNÉRAL :



III- NOTION de CIRCUIT ÉLECTRIQUE :

On va devoir construire des CIRCUITS pour faire circuler l'électricité à travers différents COMPOSANTS.

Les CIRCUITS sont des boucles de fils fermées avec une ALIMENTATION (pile 9v 6LR61 par ex.) et quelque chose qui utilise intelligemment l'énergie, une CHARGE (ampoule électrique par ex.).



L'électricité circule du point ayant le plus fort potentiel énergétique + (**plus**) vers un point ayant un potentiel énergétique plus faible - (**moins**). Le - est aussi appelé **masse ou GND** (ground). Dans les circuits que nous construisons l'électricité ne circule que dans un seul sens c'est un COURANT CONTINU ou encore Direct Current (DC).

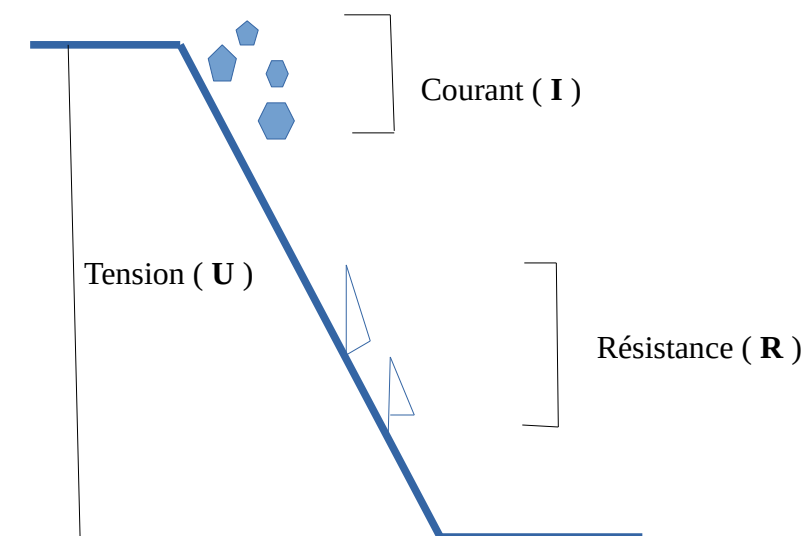
Remarque : La fourniture d'énergie électrique domestique et industrielle se fait en courant alternatif. Le courant change de sens 50 fois par seconde (50 Hz).

Le **COURANT (I)** se mesure en **Ampères** (symbole **A**) c'est la quantité de charge électrique circulant à un certain point du circuit.

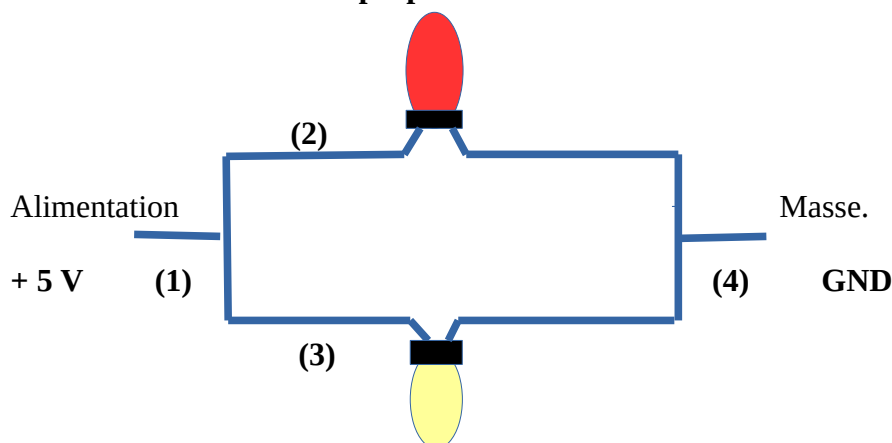
La **TENSION (U)** se mesure en **Volts** (symbole **V**) c'est la différence d'énergie – différence de potentiel entre 2 points du circuit.

La **RÉSISTANCE (R)** se mesure en **Ohms** (symbole Ω) correspond à la capacité d'un composant à résister au flux électrique lorsqu'il travaille.

Analogie : Un éboulement comme métaphore pour le passage du courant électrique



3-1 Une ou deux choses à propos des circuits :



Courant en (1) = courant en (2) + courant en (3) = courant en (4)

- Il faut qu'il y ait un chemin complet depuis la source d'énergie (alimentation) jusqu'au point de moindre énergie (masse). Sans cela, le circuit ne fonctionnera pas.

- Toute l'énergie électrique d'un circuit est consommée par les composants le constituant. Elle est alors convertie en une autre forme d'énergie (lumière, chaleur, force motrice....).

- La quantité de courant entrant en un point spécifique d'un circuit sera toujours égale à la quantité de courant sortant de ce même point.

- Le courant électrique cherchera toujours à emprunter le chemin ayant la plus petite résistance.
- **En reliant l'alimentation directement à la masse, sans résistance, on fait un court circuit, le courant n'est donc plus limité et l'énergie électrique est transformée en chaleur, étincelles ou explosion, c'est très dangereux.**

3-2 Comprendre la Loi d'Ohm :

Le courant (I), la tension (U) et la résistance (R) **sont tous liés. Le changement sur l'une de ces grandeurs affectera aussi les autres.**

La relation entre elles est connue sous le nom de la **Loi d' Ohm**

Tension (U) = Résistance (R) x Courant (I) **$U=RI$**

Exemples (mesures faites en atelier avec un multimètre)

Valeurs connues : $U = 9 \text{ V}$, $R = 220 \Omega$ Valeur mesurée : $I = 0,049 \text{ A}$ soit 49 mA

Connaissant 2 valeurs on peut en déduire la 3ème par la transformation de la formule $U = R I$ soit :

$$U = R I$$

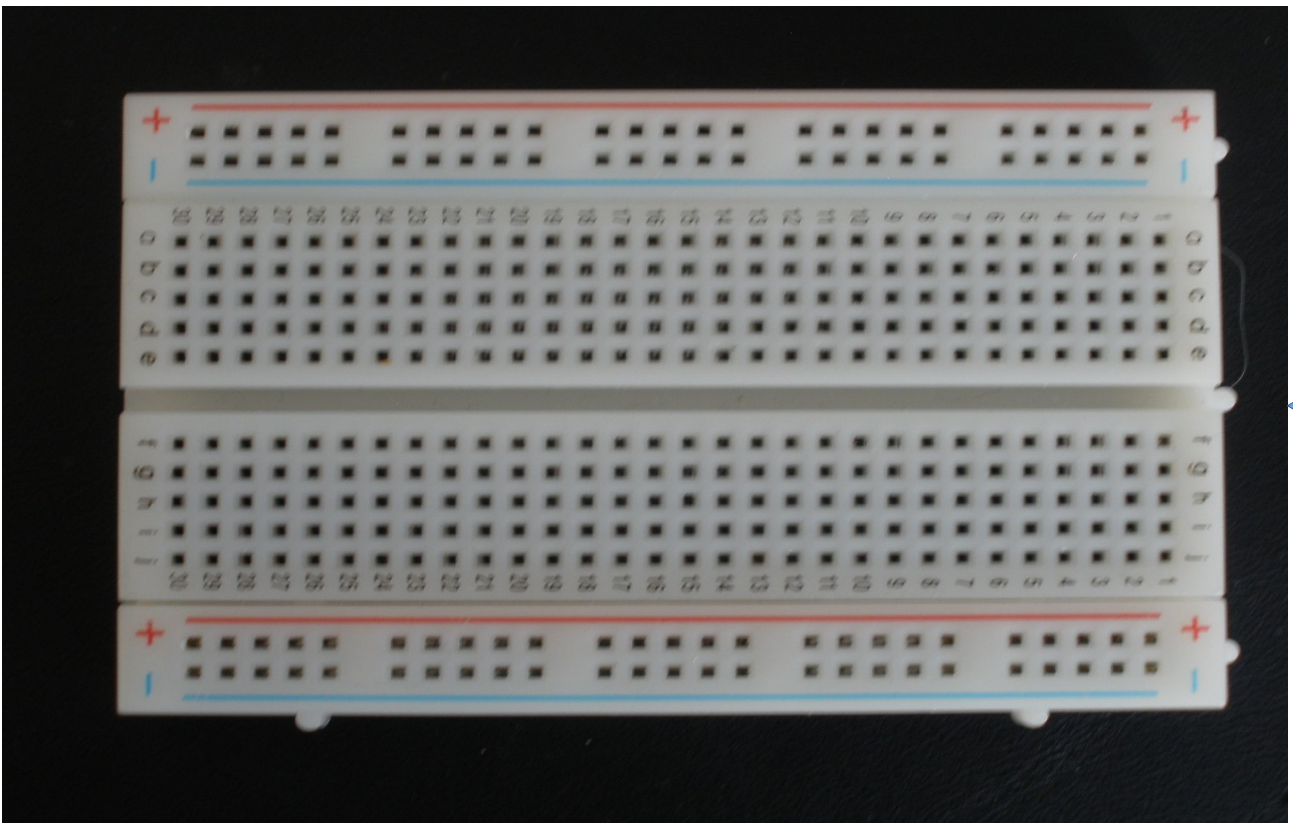
$$I = U / R$$

$$R = U / I$$

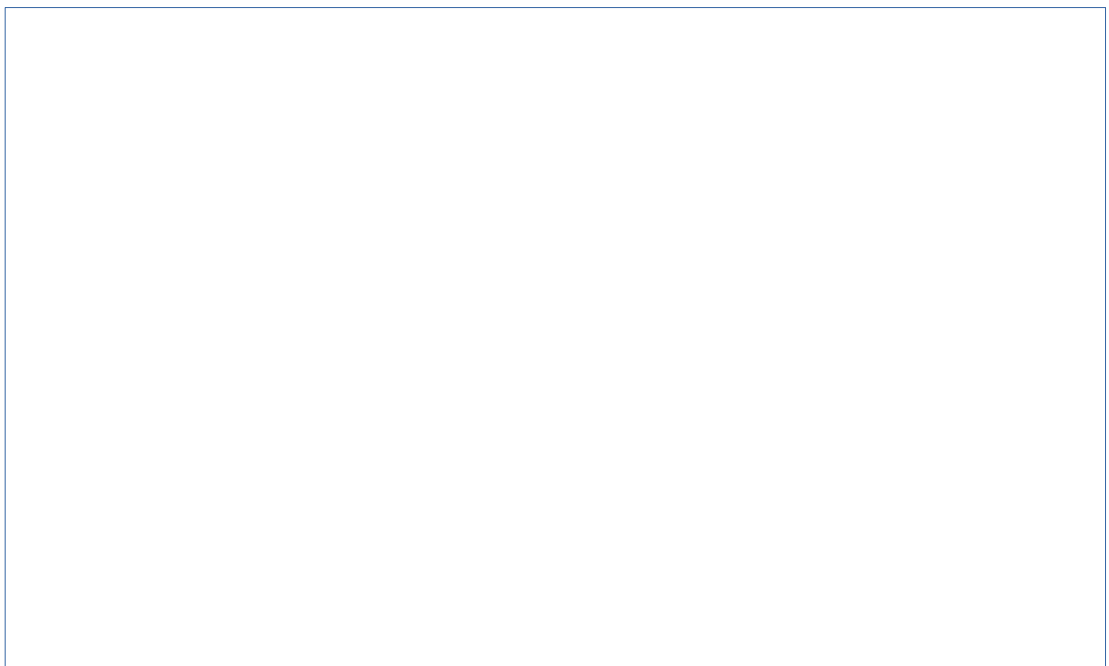
IV QU' EST CE QU' UNE BREADBOARD ?

La breadboard est le premier endroit où vous construirez vos circuits. Votre breadboard (plaque de prototypage) est dite « sans soudures », elle vous permettra de faire vos premières réalisations sans soudeuse comme des LEGO avec des fils adaptés (jumpers) au format électronique.

Les lignes horizontales et verticales de la breadboard conduisent l'électricité via des petits connecteurs en métal situés sous le plastique et accessible via les trous.



- (1) Les 5 trous de chaque ligne verticale sont connectés électriquement grâce à des bandes de métal à l'intérieur de la breadboard
- (2) La colonne centrale coupe la connexion entre les 2 côtés de la breadboard.
- (3) Les lignes horizontales qui courent le long de chaque côté de la breadboard sont connectées électriquement. Elles sont généralement utilisées pour les connexions à l'alimentation et à la masse. Ces 2 lignes sont appelées BUS.



V- VOS PREMIERS COMPOSANTS :

Une LED (light emitting diode ou DEL, diode électroluminescente), est un composant qui convertit l'énergie électrique en énergie lumineuse. Les LEDs sont des **composants polarisés** ce qui signifie qu'ils ne laissent passer l'électricité que dans un seul sens. La patte la plus longue de la LED est appelée **anode** et sera branchée à l'alimentation, La patte la plus courte est la **cathode** et sera connectée à la masse (GND).

Quand on applique une tension à l'anode et que la cathode est à la masse la LED émet de la lumière.

Un **résistor** (également appelé résistance par abus de langage) est un composant qui résiste au passage du courant électrique (voir le document plastifié de votre kit pour le code des couleurs). Il convertit une partie de l'énergie électrique en chaleur.

Si vous mettez une résistance **en série** avec un composant comme une LED, le résistor utilisera une partie de l'énergie électrique et la LED recevra moins d' énergie

Ceci permet d'alimenter des composants avec la quantité d'énergie dont ils ont besoin.

On utilise une résistance en série avec une LED pour l'empêcher de subir une tension trop élevée. Sans la résistance la LED serait plus lumineuse pendant un court instant, mais grillerait rapidement.

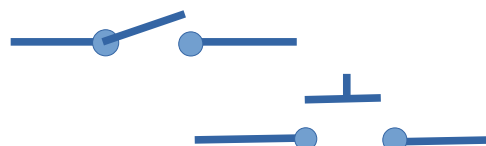
Un interrupteur interrompt le passage de l'électricité, coupant le circuit lorsqu'il est ouvert.

Il existe plusieurs types d'interrupteurs, ceux de votre kit sont des boutons poussoirs. On les appelle aussi interrupteurs instantanés .

5-1 Le bouton poussoir de votre kit :

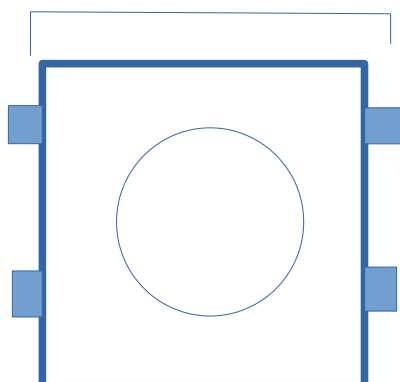
On distingue :

- Les interrupteurs à commutation :



- Les boutons poussoir :

Les 2 pattes sont reliées



Ne sont pas reliées

Forment l'interrupteur

VI – CONSTRUISEZ le CIRCUIT :

La carte Arduino sert juste d'alimentation dans ce circuit. Quand il est branché à un port USB ou à une batterie de 9v l'Arduino délivre 5v accessibles entre les broches 5 volts et GND.

Débranchez votre Arduino avant de monter votre circuit.

Connectez un fil rouge au 5 volts de l'Arduino et un fil noir au GND de l'Arduino. Le fil rouge sera connecté au + du bus de la breadboard, le fil noir sera connecté au - du bus de la breadboard.

Disposez fils et composants suivant figure ci-contre

VII - CIRCUIT SÉRIE :

Les composants en série se mettent l'un après l'autre. Ici les **boutons poussoirs sont en série**



Ces 2 éléments
sont en série

Après avoir **enlevé votre source d'alimentation**, mettez un bouton poussoir à côté de celui déjà présent sur votre breadboard.

Branchez les en série comme le montre la figure 11 ci-dessus. Connecter l'anode (patte longue) de la LED au second bouton poussoir.

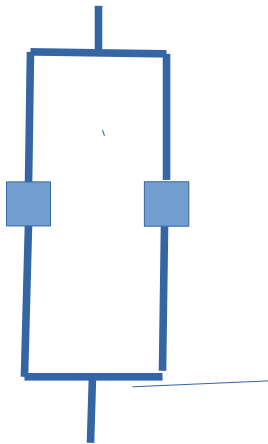
Connecter la cathode de la LED à la masse (-). Alimenter l' Arduino à nouveau.

Maintenant, pour allumer la LED, il vous faut **appuyer** sur les 2 boutons poussoirs **en même temps**. Comme ils sont placés en série, ils doivent tous les deux être fermés pour que le circuit soit complet.

VIII- CIRCUIT PARALLÈLE :

Les composants en parallèle se mettent côte à côte. Ici les **boutons poussoirs sont en parallèle**.

9



Ces 2 éléments sont en parallèle.

Gardez les boutons poussoirs et la LED où ils sont mais enlevez la connexion entre ces boutons poussoirs

Reliez les 2 boutons poussoirs à la résistance. Branchez l'autre côté de chaque bouton poussoir à la LED (voir fig 13 ci-dessus). Désormais lorsque vous appuyez sur l'un ou l'autre des boutons le circuit est fermé et la LED s'allume.

IX- LE MICRO-PROCESSEUR ARDUINO :

- (1)- Port USB utilisé pour **alimenter** votre carte Arduino, y **transférer** vos programmes et pour **communiquer** avec ces derniers (via *Serial.Println ()* etc....).
- (2)- **Connecteur d'alimentation** : C'est par ici que vous connectez votre Arduino lorsqu'il n'est pas branché à un port USB. Accepte une **tension comprise entre 7v et 12v**.
- (3)- **Broches numériques** : Utiliser ces broches avec *digitalRead()*, *analogWrite()*
- (4)- **Entrées analogiques** : Utiliser ces broches avec *analogRead()*.
- (5)- **Broches 5v et GND** : Utiliser ces broches pour **alimenter vos circuits** avec du +5v et une masse.
- (6)- **Micro-contrôleur Atmega** : Le coeur de votre Arduino
- (7)- **LEDs Tx et Rx** : Ces LEDs indiquent une communication entre votre Arduino et votre ordinateur. Elles clignotent pendant le transfert de vos programmes ainsi que pendant une communication série. Utile pour le debug.
- (8)- **LED broche 13** : Le seul actionneur présent sur votre Arduino. Outre son utilité pour le premier sketch, peut-être utile pour le debug.
- (9)- **Témoin d'alimentation** : Indique que votre Arduino est bien alimenté. Utile pour le debug.
- (10)- **Bouton reset** : Redémarre le micro-contrôleur Atmega.

X- **COMMUNIQUER avec L' ARDUINO** grâce à l' IDE :

But : Faire clignoter la led de la carte Arduino.

10-1 Raccorder votre carte Arduino à votre ordinateur avec le câble USB.

10-2 Ouvrir la fenêtre Arduino (IDE) : **Sketch_oct12a Arduino 1.8.9** en italique, date et ordre du sketch.

10-3 Nous allons **téléverser** dans l' Arduino le programme d' exemple Blink qui fera clignoter la LED 13 de votre carte Arduino.

10-4 Trouvez le sketch d' exemple Blink (sketch est le nom donné aux programmes Arduino), il est situé dans : **Fichiers** ⇒ **Exemples** ⇒ **01.Basics** ⇒ **Blink**

10-5 Une fenêtre avec du texte devrait s'être ouverte. Laissez la ouverte pour le moment et sélectionner votre carte dans : **Outils** ⇒ **Type de carte** (Arduino Genuino Uno)

10-6 Choisissez le port série auquel est branché votre Arduino dans le menu **Outils** ⇒ **Port Série**
Dans windows **Outils** ⇒ **Port** ⇒ **COM3** (Arduino Genuino Uno)

10-7 Pour mettre le sketch Blink dans votre Arduino,
cliquez sur le bouton **Téléverser**

Vous devriez voir une barre indiquant la progression du téléversement dans le coin en bas à gauche de l'IDE Arduino et les LEDs Tx et Rx sur votre carte devraient clignoter.
Si l'opération s'est bien passée l'IDE indiquera le message **Téléversement terminé**

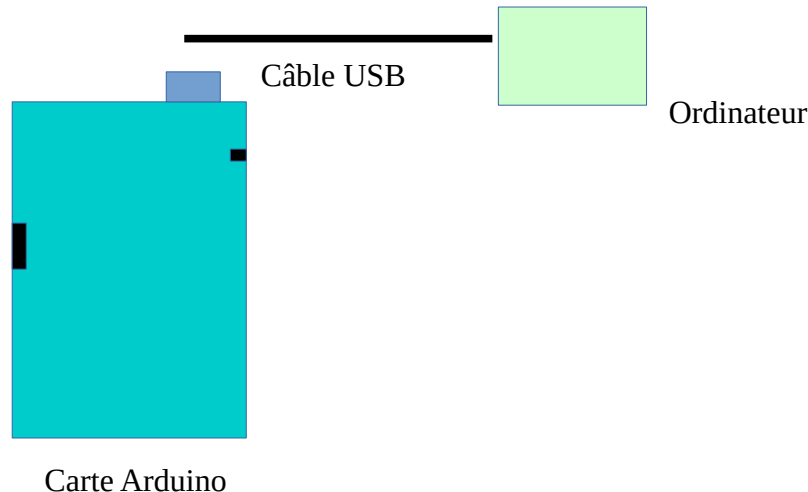
Quelques secondes après que le téléversement soit fini
vous devriez voir la LED jaune avec un L à côté
commencer à clignoter.

XI- ÉCRIRE le PREMIER PROGRAMME :

But : Faire clignoter une LED.

11-1 En respectant le circuit électrique ci-dessous, monter **1 LED** en série avec une **Résistance** de $220\ \Omega$ sur votre breadboard et :

- Raccorder l' anode de la LED (patte la plus longue) à la broche 13 de votre carte Arduino
- Raccorder la cathode de la LED (patte la plus courte) à une entrée de la résistance,
- Raccorder l 'autre côté de la résistance à une broche GND de la carte Arduino.



Une fois le montage réalisé, raccorder la carte Arduino à votre ordinateur.

11-2 Ouvrir la fenêtre Arduino :

- vérifier que votre carte Arduino est bien sélectionnée **Outils** \Rightarrow **Type de carte**,
- choisissez votre port série **Outils** \Rightarrow **Port** \Rightarrow **COM3** (windows)

Avant d'écrire un programme il est important de connaître le déroulement de tout programme :

- * après avoir pris en compte les instructions de la **partie déclarative**,
- * Puis après avoir exécuté la partie configuration (fonction **setup()**),
- * Le programme bouclera sans fin (fonction **loop()**) exécutant de façon répétitive le code compris dans la boucle sans fin.

11-3 Écriture du programme pour faire clignoter une LED :

```
// ---Entête déclarative ---

void setup () { // Ne prend en compte les informations qu'une seule fois au démarrage du
                programme

    pinMode ( 13, OUTPUT ) ; // déclaration d'état de la broche 13, ici en sortie.

}

void loop() { // Instructions exécutées en boucle sans fin

    digitalWrite ( 13, HIGH ) ; // Valeur haute attribuée à la broche, ici 5v
    delay ( 1000 ) ;             // durée d'allumage de la LED en ms
    digitalWrite ( 13, LOW ) ; // valeur basse attribuée à la broche 13, ici 0v
    delay ( 1000 ) ;             // durée d'extinction de la LED en ms

}
```

Remarques :

1- Structure, fonctions de base : `void setup ()` et `void loop ()` sont 2 **fonctions obligatoires** dans tout programme Arduino.

2- Syntaxe de base ; point virgules
 { } accolades
 // commentaire sur une ligne
 /* */ commentaire sur plusieurs lignes

3- Constantes prédéfinies sont des valeurs particulières ayant une signification spécifique :

HIGH	LOW	Valeur haute et basse	
INPUT	OUTPUT	Mode entrée	Sortie
true	false	Vrai	Faux

4- Fonctions Entrées / sorties numériques :

```
pinMode ( broche, mode )
digitalWrite( broche, valeur )
int digitalRead ( broche )
```

cf : Langage Arduino

11-4 Vérifier votre programme :

- Dans la fenêtre IDE, cliquer sur la 1ère icône en haut à gauche « Vérifier »
 - * Dans le bas de la fenêtre apparaît, pour un nouveau sketch, « Sauvegarder le sketch »
 - * Puis « Compilation de croquis »
 - * Puis « Compilations terminée »
 Dans le cas où il y aurait une erreur dans l'écriture du sketch, les informations apparaîtront dans la partie inférieure de la fenêtre.
- Dans la fenêtre IDE, cliquer sur la 2ème icône en haut à gauche « Téléverser »

« Téléversement terminé » dans la partie inférieure de la fenêtre IDE indique que le téléversement a réussi.

La LED clignote toutes les secondes

Vous pouvez faire varier le temps du clignotement de la LED en modifiant la valeur de **delay**.

=====