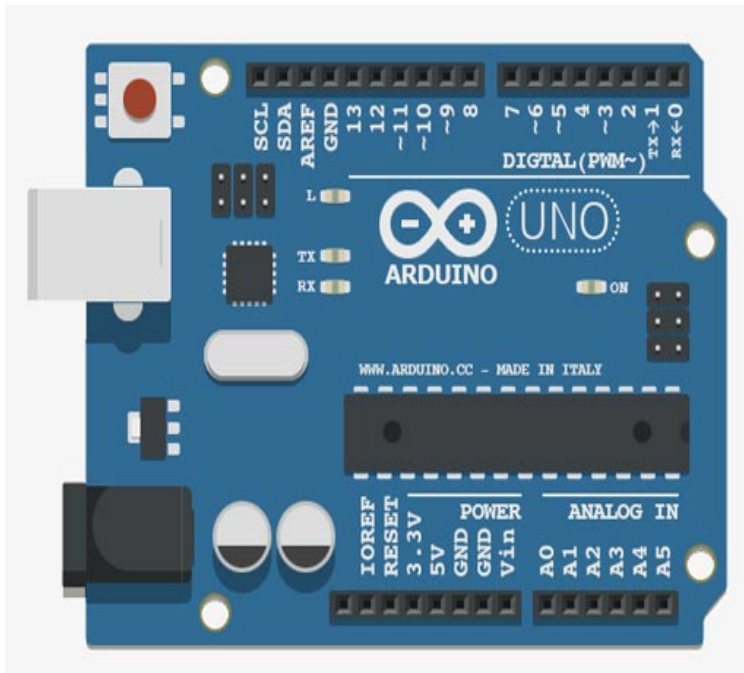


TP N° 1 (2 SÉANCES DE 2H) : PREMIÈRES MANIPULATIONS

I- Matériel et environnement de développement

- **Arduino**



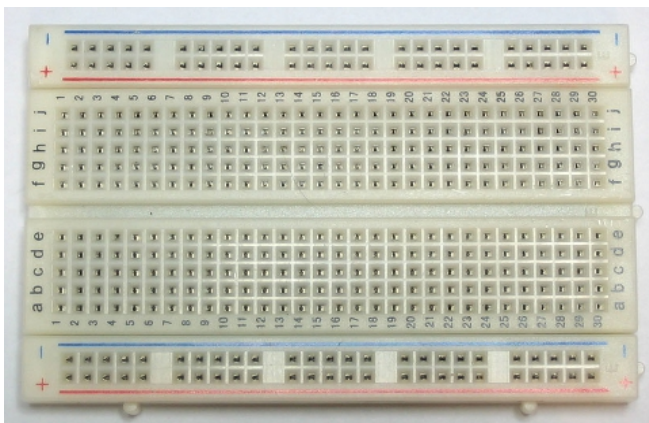
Vous disposez d'un Arduino Uno que vous brancherez à votre pc grâce au connecteur USB. Cette connexion permettra d'alimenter électriquement l'Arduino mais aussi de communiquer avec le logiciel qui compile puis téléverse vers Arduino les programmes que vous allez exécuter.

Sur le haut de l'image vous voyez 14 entrées/sorties numérotées de 0 à 13 que nous allons utiliser dans ce tp

Vous utiliserez également les 2 broches 5V et GND (Masse) qui vous permettront d'alimenter vos circuits électriques

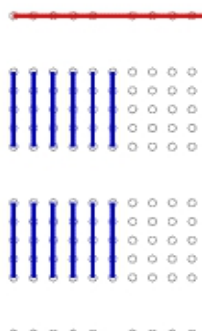
- **La plaque d'essai**

Afin de réaliser vos montages (avec des LEDs et des boutons poussoirs dans un premier temps), vous disposez d'une plaque d'essai :



La plaque d'essai s'utilise avec des bouts de fils en cuivre de taille et de longueur différentes pour relier les différents composants (LEDs, résistances, ...)

- Tous les points d'une même ligne du bus d'alimentation (en rouge et en bleu sur le schéma) sont connectés entre eux. Vous penserez à connecter le +5v au bus d'alimentation rouge et la masse (GND) au bus d'alimentation bleu



Sur le second schéma ci -contre

- Tous les points d'une demi-colonne (en bleu) sont connectés entre eux
- Les colonnes sont coupées en deux par le rail central qui permet de mettre des composants "à cheval"

- **Le logiciel Arduino**

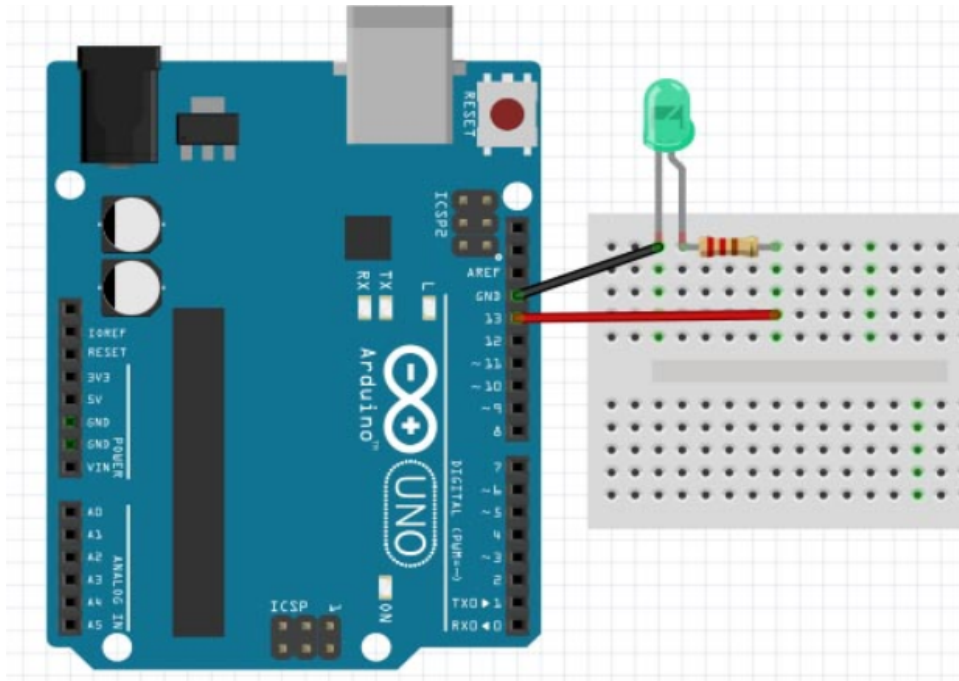
Le logiciel Arduino est installé sous linux et Windows dans les salles de tp informatique.

Il permet d'éditer vos programmes (sous forme de sketch) de les compiler puis de les téléverser vers Arduino sur lequel ils s'exécutent. Pensez à sauvegarder vos sketches sur votre espace de stockage ou une clé usb à la fin de chaque séance de TP.

II- Premier exemple : Allumer une LED

Notre premier circuit comportera une LED avec une résistance et un programme qui allume cette LED.

- **Schéma**



- **Programme**

```
int rouge=2;

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

void loop() {
    digitalWrite(rouge,HIGH);
}
```

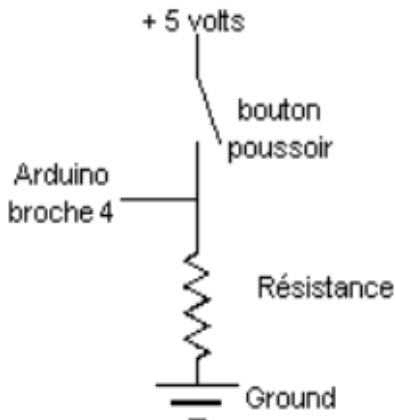
Modifier le programme ci-dessus pour faire clignoter la LED.

III- Avec un bouton poussoir

1. Un bouton poussoir qui allume/éteint une diode

Notre second circuit comportera en plus de la LED précédente un bouton poussoir avec une résistance *pull*down que l'on prendra d'une valeur de 10 k Ω

- **Schéma**



- **Programme**

Écrire le programme qui allume la LED lorsque le bouton poussoir est enfoncé, et éteint la LED lorsqu'il est relâché.

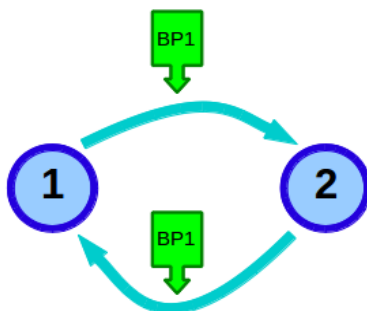
2. Un bouton poussoir qui change l'état d'une diode

Nous allons maintenant écrire un programme qui lorsque l'on appuie sur le bouton poussoir (BP1), change l'état de la diode.

On considère donc le diagramme d'état suivant :

1=LED Allumée

2=LED Eteinte



Écrire le programme de la façon suivante :

- utiliser une variable globale `etat` (qui vaut un nombre pair si la LED est éteinte et un nombre impair si elle est allumée) initialisée à 0. La variable est incrémentée à chaque changement d'état.
- dans la fonction `loop()` :
 - lire l'état du bouton poussoir. S'il est enfoncé alors changer la valeur de `etat`.
 - En fonction de la valeur de `etat` allumer ou éteindre la LED

Tester le programme obtenu. Quel est le problème rencontré ?

Pour visualiser le problème, on peut utiliser la console pour afficher l'état chaque fois que l'on détecte le bouton enfoncé. Pour cela ajouter dans la fonction `setup()` `Serial.begin(9600)` ; et ajouter `Serial.println(etat)` ; à chaque fois que l'on détecte le bouton enfoncé.

3. Utiliser des interruptions

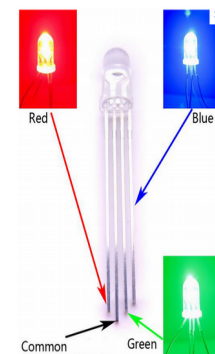
Pour résoudre le problème précédent on va utiliser une interruption. On va brancher le bouton poussoir sur une broche dédiée aux interruptions. Cela se fait grâce à la commande `attachInterrupt(num, traitement, RISING)` ; où `num` est le numéro de l'interruption et `traitement` est la fonction qui sera appelée lorsque l'interruption sera déclenchée, `RISING` signifie que l'interruption sera déclenchée lorsque la broche passera de la valeur 0 à la valeur 1. Attention sur Arduino UNO seules deux broches peuvent déclencher une interruption : la valeur `num=0` correspond à la broche 2 et la valeur `num=1` correspond à la broche 3.

4. Faire clignoter la LED

Réaliser maintenant un montage avec 1 LED et 2 BP qui a le comportement suivant : la LED clignote infiniment (1 s allumée, 1 s éteinte). Si on appuie sur BP1 on multiplie la vitesse par 2. Si on appuie sur BP2 on divise la vitesse par 2.

IV- Diode RVB

On souhaite utiliser maintenant une diode RVB avec anode commune. Elle fonctionne de la façon suivante : on connecte l'anode commune à la masse et on alimente chacune des autres broches avec une tension comprise entre 0 et 5V. La lumière obtenue est alors le résultat de la synthèse additive des 3 couleurs. Le problème principal est que les 3 broches de couleur ne sont pas alimentées par 0 ou 5V.



- **sortie analogique**

Heureusement Arduino nous fournit une solution à ce problème grâce aux sorties analogiques. En effet certaines broches sur lesquelles figure le signe \sim (les broches 3,5,6, 10 et 11) permettent de fournir une tension de sortie comprise entre 0 et 5V. (enfin pas exactement : pour obtenir 1V on envoie 5V pendant t secondes puis 0V pendant $4t$ secondes donc en moyenne, si t est très petit, on envoie 1V)

Pour utiliser ces sorties analogiques il faut utiliser la fonction

`analogWrite(sortie, val);`

où `sortie` est le numéro de la sortie utilisée et `val` la valeur analogique que l'on veut obtenir ; enfin pas tout à fait : pour obtenir 5V `val=255`, donc pour obtenir xV il faut choisir `val=255 * x/5`

1. Vous êtes maintenant en mesure d'écrire une fonction `couleur (int r, int v, int b)` qui permet d'allumer une LED RVB et de réaliser le schéma électrique associé. (n'oubliez pas de mettre une résistance sur chacun des 3 entrées R, V et B)
2. Écrivez maintenant un programme qui allume votre LED en faisant varier la couleur chaque seconde parmi 8 couleurs que vous choisirez.

V- Jeu 3 diodes / 2 poussoirs

On dispose de deux diodes Jaune (J) et Rouge (R), d'une LED RVB et de deux Boutons Poussoir BP1 et BP2)

Réaliser le jeu dont le fonctionnement est le suivant : un code de 4 couleurs (J-R-R-J) est généré au hasard (utiliser la fonction **random** et stocker le résultat dans un tableau : un tableau se déclare comme en JAVA : `int tab[4] ;`) et affiché sur les 2 LEDs J et R (chaque diode reste allumée 1s).

Le joueur doit ensuite répéter le code en appuyant sur 2 boutons poussoirs (BP1 correspond à J et BP2 à R) . En cas de succès : la diode multicolore s'affiche en vert . En cas d'erreur : la diode multicolore s'affiche en rouge. On repart au début du jeu lorsqu'on appuie sur BP1 ou BP2

1. Établir le diagramme d'états du jeu
2. Le réaliser avec Arduino.