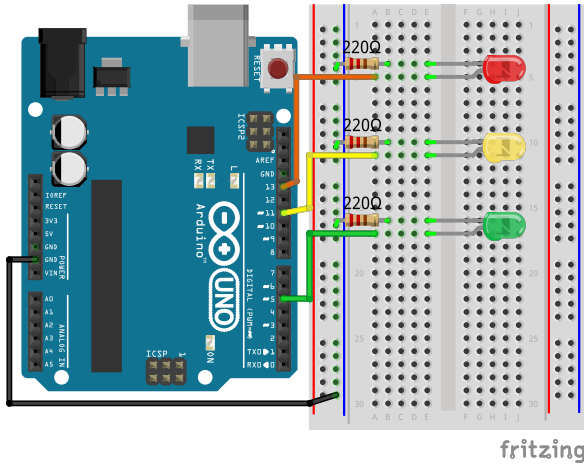
TROIS LEDS

Montage

Sortez votre matériel et câblez le circuit suivant :



Matériel nécessaire :

- Arduino
- Plaque d'essai
- 3 LEDs
- 3 résistances de 220 Ω (ou 330 Ω)
- 4 câbles

Branchez ensuite l'Arduino au câble USB.

Programme

Téléversez le *croquis* ci-contre.

C'est joli, mais ça va un peu vite !
Pour augmenter le temps d'allumage et d'extinction des LEDs, on peut augmenter le paramètre de la fonction `delay` . . . ce qui fait 6 lignes de code à modifier !
On peut le faire en une seule fois en utilisant une **variable**.

CODE ARDUINO

```
// Croquis P2-3_Leds.ino
//----- CONSTANTES -----
#define Led0 5  // LED 0 sur la ligne d'E/S 5
#define Led1 11 // LED 1 sur la ligne d'E/S 11
#define Led2 13 // LED 2 sur la ligne d'E/S 13
//----- PROCEDURE D'INITIALISATION -----
void setup() { // Les 3 lignes en sortie
  pinMode(Led0, OUTPUT);
  pinMode(Led1, OUTPUT);
  pinMode(Led2, OUTPUT);
}
//----- BOUCLE PRINCIPALE -----
void loop(){ // Allumage...
  digitalWrite(Led0, HIGH); // LED 0 allumée
  delay(100); // Attente de 100 millisecondes
  digitalWrite(Led1, HIGH); // LED 1 allumée
  delay(100); // Attente de 100 millisecondes
  digitalWrite(Led2, HIGH); // LED 2 allumée
  delay(100); // Attente de 100 millisecondes
  // Extinction...
  digitalWrite(Led2, LOW);  // LED 2 éteinte
  delay(100); // Attente de 100 millisecondes
  digitalWrite(Led1, LOW);  // LED 1 éteinte
  delay(100); // Attente de 100 millisecondes
  digitalWrite(Led0, LOW);  // LED 0 éteinte
  delay(100); // Attente de 100 millisecondes
}
```

Constantes et variables

Un arduino, comme un ordinateur, effectue les calculs en **binaire** : les nombres ne s'écrivent qu'avec deux chiffres, **0** et **1**.
0 correspond à une tension de **0 V** et **1** à une tension de **5 V**.
Un chiffre binaire est appelé un **bit**, de l'anglais **B**inary dig**IT** . Pour écrire les nombres, les bits sont regroupés. Un groupe de 8 bits s'appelle un **octet**. On peut aussi les regrouper par 16 (2 octets), 32 (4 octets), 64 (8 octets). Vous avez surement entendu parlé de « processeur 64 bits »… l'Arduino UNO a un micro-contrôleur 8 bits.

Pour l'instant vous avez utilisé le nombre 5 (pour la broche n° 5), mais vous avez aussi utilisé `LOW` qui est un autre nom pour 0 et `HIGH` qui est un autre nom pour 1 ! Ces nombres ne sont pas amenés à changer de valeur : ce sont des **constantes**.

Contrairement à une constante, une **variable** peut-être amenée à changer de valeur au cours du programme. Pour définir une variable, il faut donner son **type** à l'Arduino. Il existe plusieurs types, par exemple :

- `int` : abréviation de *integer* (nombre entier en anglais). La taille d'une constante ou d'une variable de type `int` est de 2 octets. Les valeurs possibles vont de −32 768 à 32 767.
- `byte` : désigne un octet en anglais. La taille d'un `byte` est donc de… 1 octet ! Les valeurs possibles vont de 0 à 255.
- `word` : nombre entier sur deux octets. Les valeurs possibles vont de 0 à 65 535.
- `long` : nombre entier sur quatre octets. Les valeurs possibles vont de −2 147 483 648 à 2 147 483 647.
- `unsigned long` : nombre entier sur quatre octets. Les valeurs possibles vont de 0 à 4 294 967 295.
- `float` : nombre décimal sur quatre octets (flottant), de $-3{,}402\,823\,5 \times 10^{38}$ à $3{,}402\,823\,5 \times 10^{38}$.

Applications

1. Dans votre programme (*croquis*) en dessous de la déclaration des constantes, rajoutez les deux lignes suivantes, et remplacer les 100 par `tempsLed` :

CODE ARDUINO

```
//----- VARIABLES -----
int tempsLed = 200; // Temps d'attente
```

Le temps d'allumage et d'attente est désormais de 200 millisecondes.
Faites plusieurs essais en modifiant la valeur de `tempsLed`.

2. Créez maintenant une deuxième variable `tempsEteint` de type `int` pour le temps d'*extinction*, et faites plusieurs essais en changeant les valeurs des deux variables.
3. Maintenant modifiez le programme pour que la LED 0 s'allume pendant `tempsLed` millisecondes, s'éteigne en même temps que la LED 1 s'allume, ceci pendant `tempsLed` millisecondes, ensuite la LED 1 s'éteint en même temps que la LED 2 s'allume, etc. Enregistrez votre *croquis* sous le nom `P2-3_Leds_Q3`.
4. Même chose, mais lorsque la LED 2 s'éteint, c'est elle qui se rallume, ensuite elle s'éteint et c'est la LED 1 qui s'allume, etc. Enregistrez votre *croquis* sous le nom `P2-3_Leds_Q4`.
5. Cette fois, la LED 0 et la LED 2 s'allument en même temps pendant `tempsLed` millisecondes, puis quand elles s'éteignent, le LED 1 s'allume pendant `tempsLed` millisecondes, et lorsqu'elle s'éteint, les LED 0 et 2 se rallument, etc. Enregistrez votre *croquis* sous le nom `P2-3_Leds_Q5`.

Dans le monde réel…

La plupart des panneaux d'affichage utilisent de nombreuses LEDs pour diffuser les informations, il est important de pouvoir programmer leur allumage et leur extinction de manière efficace. Nous verrons dans la prochaine fiche comment encore améliorer vos programmes !