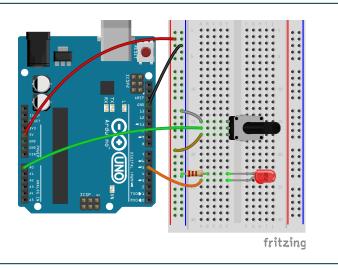
VARIATION DE LUMINOSITÉ

뢲 Premier montage

Sortez votre matériel et câblez le circuit suivant :

Matériel nécessaire:

- Arduino
- Plaque d'essai
- 1 potentiomètre (linéaire $10 \text{ k}\Omega$)
- 1 LED
- 1 résistance de 220 Ω (ou 330 Ω)
- 6 câbles



PWM ou comment simuler de l'analogique

Une LED peut être dans l'état « allumé » (HIGH) ou « éteint » (LOW). Nous avons vu que pour un composant numérique, il n'y a rien entre les deux... Une façon de faire varier la luminosité de la LED est de l'allumer et l'éteindre très rapidement.

Cette technique est appelée **PWM** (Pulse Width Modulation), ou encore MLI en français (Modulation en Largeur d'Impulsion). La durée pendant laquelle la LED est allumée est la «largeur d'impulsion».

Programme

La LED doit obligatoirement être reliée à une sortie PWM (repérée par le symbole ~).

La fonction analogWrite va régler la largeur d'impulsion. Elle nécessite un nombre compris entre **0** (toujours éteint) et **255** (toujours allumé). Ces 256 valeurs correspondent à un nombre codé sur **8 bits**. Il est alors inutile de déclarer la broche de la LED en sortie dans la procédure setup.

Il faut changer la plage de valeur du potentiomètre (nombre compris entre 0 et 1023) pour obtenir un nombre compris entre 0 et 255 : c'est la fonction map qui s'en charge (on pourrait le faire par un calcul, mais nous avons vu qu'il faut faire attention!).

Branchez l'Arduino au câble USB et *téléversez* le *croquis* ci-contre.

Appuyez sur l'icône permettant d'afficher le *moniteur série* (en haut à droite).

Faites varier le potentiomètre.

CODE ARDUINO

```
// Croquis P6-Fondu.ino
//---- CONSTANTES -----
#define Pot 0 // POT. sur la ligne ana. 0 (A0)
#define Led 5 // LED sur la ligne E/S 5 (PWM)
//---- VARIABLE ----
int valPot; // Variable du POTENTIOMETRE
int largImp; // Largeur d'impulsion (0 à 255)
//---- PROCEDURE D'INITIALISATION -----
void setup() {
  Serial.begin(9600); // Vitesse communication
//---- BOUCLE PRINCIPALE ----
void loop(){
  valPot = analogRead(Pot); // Lecture du POT.
  largImp = map(valPot, 0, 1023, 0, 255);
  analogWrite(Led, largImp); // Réglage imp.
  // Affichage sur le moniteur série
  Serial.print("Largeur d'impulsion : ");
  Serial.println(largImp);
```



🗣 Applications

- Complétez la boucle principale du croquis ci-contre pour que la largeur d'impulsion la Imp varie de 0 à 255 par pas de 5 (P6-Fondu_Q1).
- 2. Maintenant, supprimez la ligne analogWrite(Led, 0); et écrire une seconde boucle for à la suite de la première qui fera varier la largeur d'impulsion de 255 à 0 par pas de 5 (P6-Fondu_Q2).

```
CODE ARDUINO

//---- CONSTANTES ----
#define Pot 0  // POT. sur la ligne ana. 0 (A0)
#define Led 5  // LED sur la ligne E/S 5 (PWM)

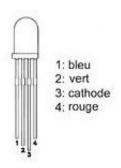
//---- PROCEDURE D'INITIALISATION ----
void setup() {} // Rien à initialiser

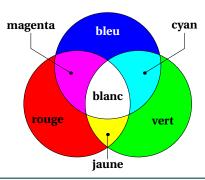
//---- BOUCLE PRINCIPALE ----
void loop(){
   analogWrite(Led, 0); // LED éteinte
   for (int laImp = ...; laImp <= ...; laImp = laImp + ...) {
      analogWrite(Led, laImp); // Réglage largeur impulsion
      delay(30); // Pour avoir le temps de voir quelquechose...
}} // } fermeture du for } fermeture du loop</pre>
```

💂 La LED RVB – Rouge Vert Bleu

Une LED RVB est en fait constituée de 3 LEDs : une ROUGE, une VERTE et une BLEUE. Vous avez peut-être vu en *cours de Physique* que si on mélange ces trois couleurs, on peut obtenir toutes les autres couleurs : c'est la **synthèse additive**.

Pour utiliser une LED RVB, il faudra donc trois résistances de protection, une pour chaque LED... Et comme nous allons faire varier la luminosité de chaque LED, il faut utiliser des broches **PWM** sur l'Arduino.



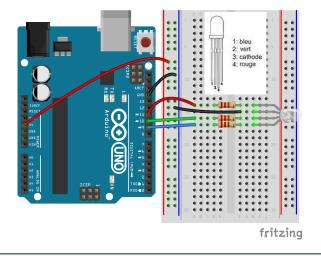


Second montage

Sortez votre matériel et câblez le circuit suivant :

Matériel nécessaire:

- Arduino
- Plaque d'essai
- 1 LED RVB
- 3 résistances de 220 Ω (ou 330 Ω)
- 6 câbles





Programme

Pour ce programme, pas besoin de broche PWM, nous allons simplement allumer certaines LEDs afin d'obtenir les 7 couleurs possibles.

La procédure couleursPrincipales() est définie **après** la boucle <code>loop()</code>. Elle est par contre appelée **dans** la boucle <code>loop()</code>.

Cela permet de clarifier le programme et d'écrire plusieurs procédures, en appelant uniquement celles voulues.

```
CODE ARDUINO
 // Croquis P6-RVB.ino
 //---- CONSTANTES -----
 #define Rouge 11 // Couleur ROUGE sur la ligne d'E/S 11
 #define Vert 10 // Couleur VERTE sur la ligne d'E/S 10
 #define Bleu 9 // Couleur BLEUE sur la ligne d'E/S 9
 //---- PROCEDURE D'INITIALISATION -----
 void setup() {
   pinMode(Rouge, OUTPUT); // Ligne de la LED ROUGE en sortie
   pinMode(Vert, OUTPUT); // Ligne de la LED VERTE en sortie
   pinMode(Bleu, OUTPUT); // Ligne de la LED BLEUE en sortie
 }
 //---- BOUCLE PRINCIPALE ----
 void loop(){
   couleursPrincipales();
 }
 //---- PROCEDURE SUPPLEMENTAIRE ----
 void couleursPrincipales(){
   // Eteint
   digitalWrite(Rouge, LOW);
   digitalWrite(Vert, LOW);
   digitalWrite(Bleu, LOW);
   delay(1000);
   // Rouge
   digitalWrite(Rouge, HIGH);
   digitalWrite(Vert, LOW);
   digitalWrite(Bleu, LOW);
   delay(1000);
   // Vert
   digitalWrite(Rouge, LOW);
   digitalWrite(Vert, HIGH);
   digitalWrite(Bleu, LOW);
   delay(1000);
   // Bleu
   digitalWrite(Rouge, LOW);
   digitalWrite(Vert, LOW);
   digitalWrite(Bleu, HIGH);
   delay(1000);
   // Jaune
   digitalWrite(Rouge, HIGH);
   digitalWrite(Vert, HIGH);
   digitalWrite(Bleu, LOW);
   delay(1000);
   // suite page suivante
```



💂 Suite du programme

Branchez l'Arduino au câble USB et *téléversez* le *croquis* cicontre (en entier!).

```
CODE ARDUINO
   // Cyan
   digitalWrite(Rouge, LOW);
   digitalWrite(Vert, HIGH);
   digitalWrite(Bleu, HIGH);
   delay(1000);
   // Magenta
   digitalWrite(Rouge, HIGH);
   digitalWrite(Vert, LOW);
   digitalWrite(Bleu, HIGH);
   delay(1000);
   // Blanc
   digitalWrite(Rouge, HIGH);
   digitalWrite(Vert, HIGH);
   digitalWrite(Bleu, HIGH);
   delay(1000);
 }
```

💂 Un dernier programme

D'après un programme de SparkFun Electronics.

Nous allons ici utiliser les broches PWM afin d'obtenir beaucoup plus de couleurs (768 pour être exact!)

Branchez l'Arduino au câble USB et téléversez le croquis P6-RVB PWM.

Ce programme est un peu compliqué, mais il est intéressant au niveau de la programmation :

La procédure showSpectrum() appelle elle même une autre procédure showRGB() en lui passant un paramètre (x) compris entre 0 et 767.

Dans la procédure showRGB(), les intensités du rouge, du vert et du bleu vont varier de 0 à 255, selon trois zones dépendantes de x.

On peut aussi remarquer la structure conditionnelle qui utilise plusieurs if pour les trois zones, comme ci-contre.

```
CODE ARDUINO
  if (...){
     ...
}
  else if (...){
     ...
}
  else {
     ...
}
```

📮 Dans le monde réel...

De nombreuses LEDs sont en fait des LEDs RVB (LEDs de smartphone qui change de couleur suivant les notifications, l'état de charge, etc.).

Chaque pixel d'une image en couleur est défini par 3 sous-pixels : un rouge, un vert et un bleu. Chaque couleur prend la valeur d'un nombre codé sur 8 bits (un nombre compris entre 0 et 255).

Les broches PWM servent aussi à faire varier la vitesse de rotation d'un moteur électrique.

