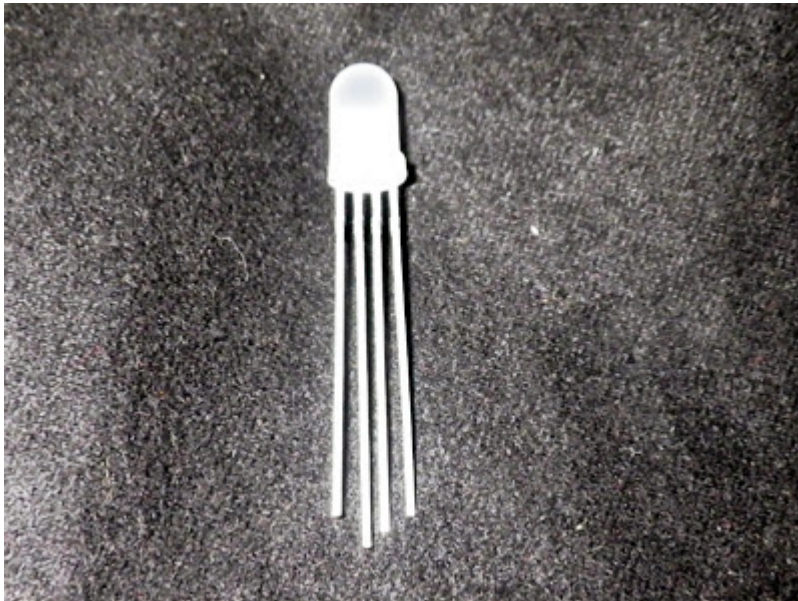


# Table des matières



Tutoriel inspiré par <http://tiptopboards.free.fr>

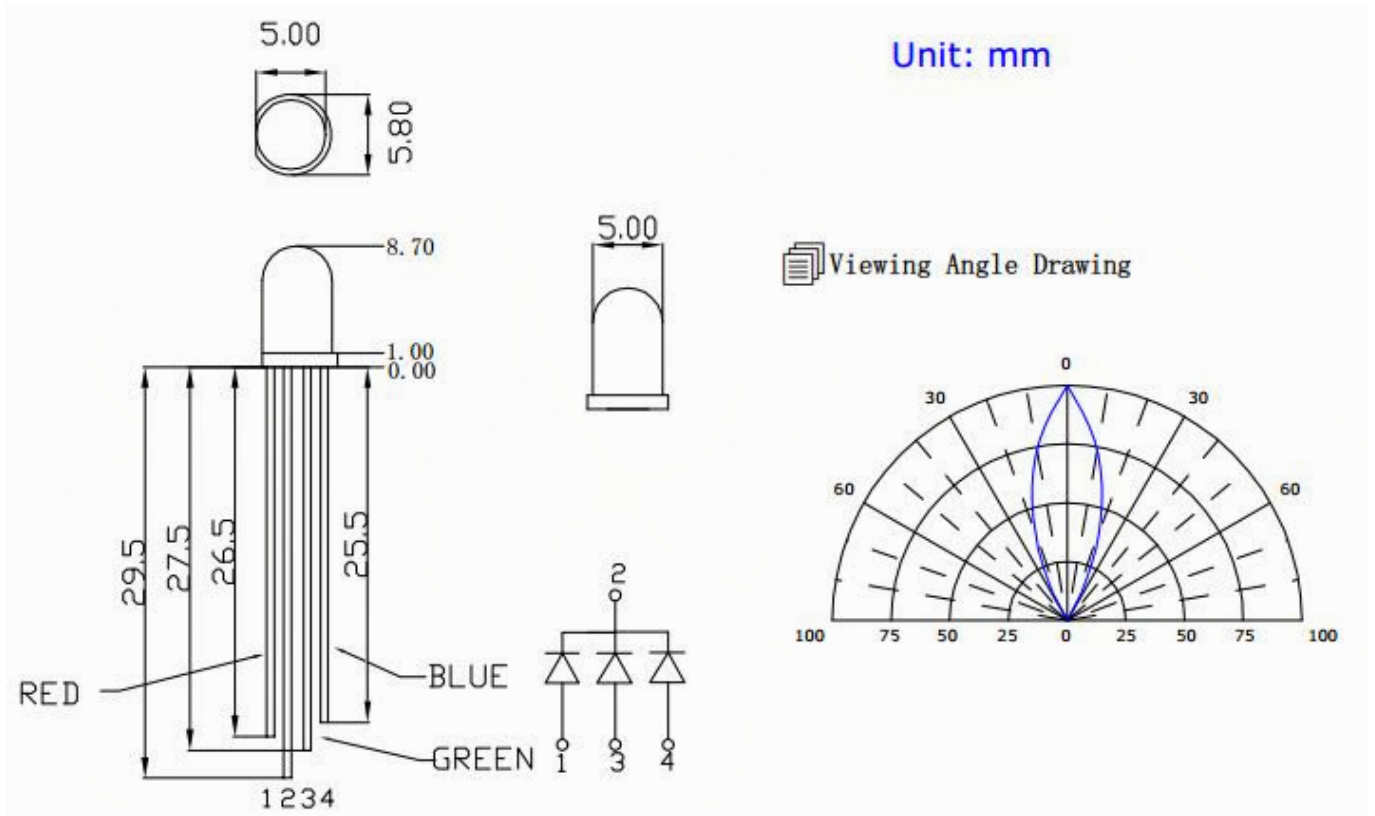
Utilisation d'une Led à trois couleurs (RGB tri-color) en modulation d'impulsions PWM avec une carte Arduino.



- 1 LED multi-couleur RGB ou montée sur un PCB
- Quelques résistances
- 1 Carte Arduino Uno
- 1 plaquette d'essai ou un prototype shield pour Arduino
- Câbles de branchement mâle femelle Dupont

Une led tricolore comporte dans le même composant 3 couleurs contrôlables séparément (rouge, vert, bleu). Elle comporte 4 pins de branchement.

Par synthèse additive des 3 couleurs on obtient ainsi toutes les teintes intermédiaires voulues.



L'intensité de chaque canal se règle sur 256 niveaux (0-255).

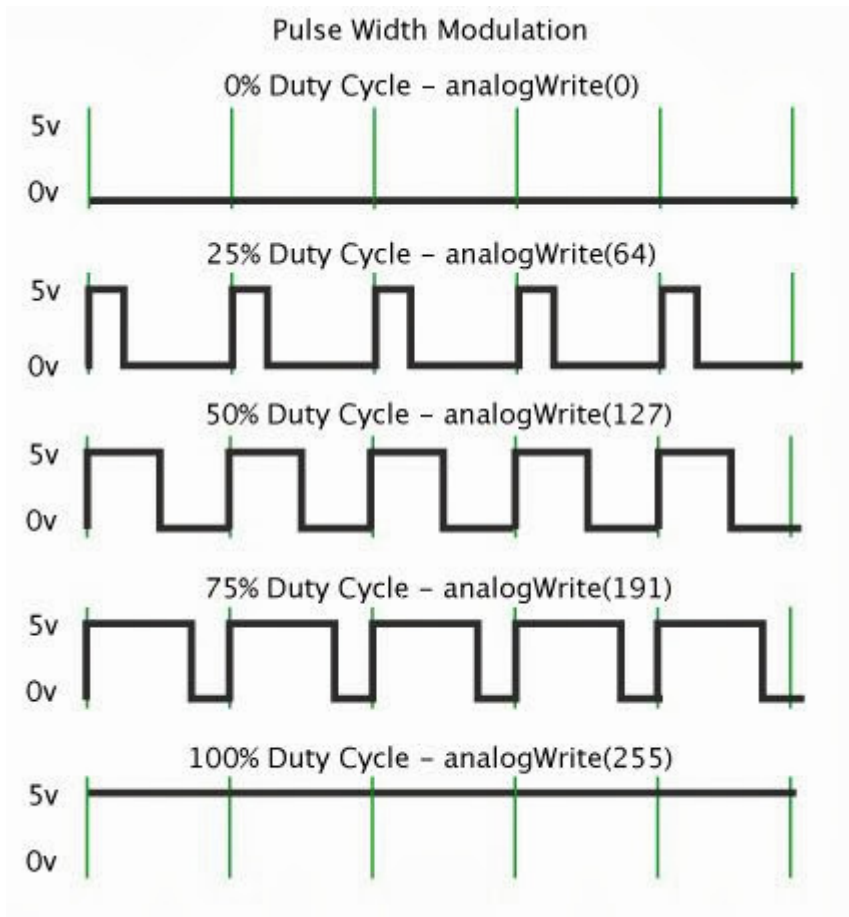
### Exemples

- $R = G = B = 0$  : tout est éteint, noir
- $R = G = B = 255$  : tout est allumé blanc maximum
- $R = 255, G = B = 0$  : allumé seulement en rouge pur
- etc...

Avec une carte Arduino, on utilise le mode PWM (Pulse Width Modulation), modulation par largeur d'impulsions, qui consiste à faire varier la durée des impulsions du courant.

La tension est fixe, 0 ou 5V, mais le signal est périodique (490 Hz environ) et la durée des périodes positives est modifiable par programmation.

Le composant va donc percevoir une commande modulée :



La carte Arduino Uno comporte 6 pins marquées d'un signe (3, 5, 6, 9, 10 et 11) qui permettent chacune une sortie en mode PWM. **3 sont utilisées ici : une par teinte.**

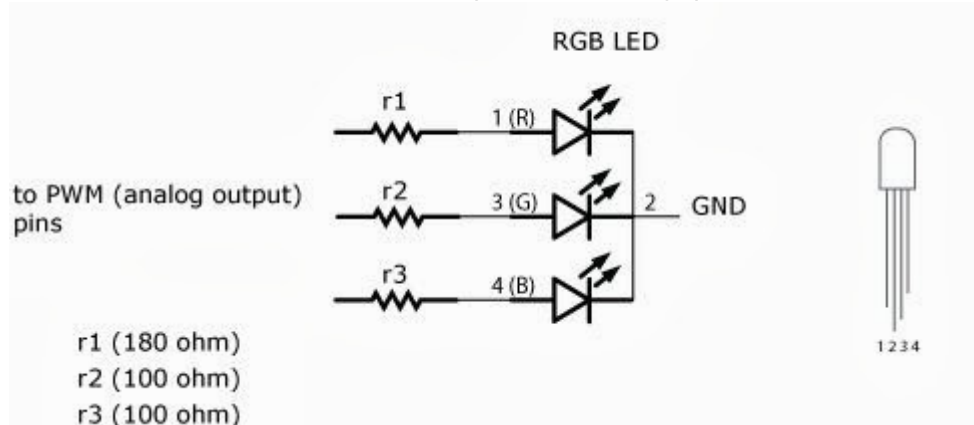
La fonction `analogWrite` permet de les contrôler.

- pin = selon le branchement réalisé
- value=0 toujours OFF, value=255 toujours ON

La LED tricolore admet une tension de **3.1 à 3.3V max** (pour une luminosité maximale), utiliser 20 mA par canal.

Elle comporte 4 pins : 3 anodes (R, G, B) et 1 cathode commune.

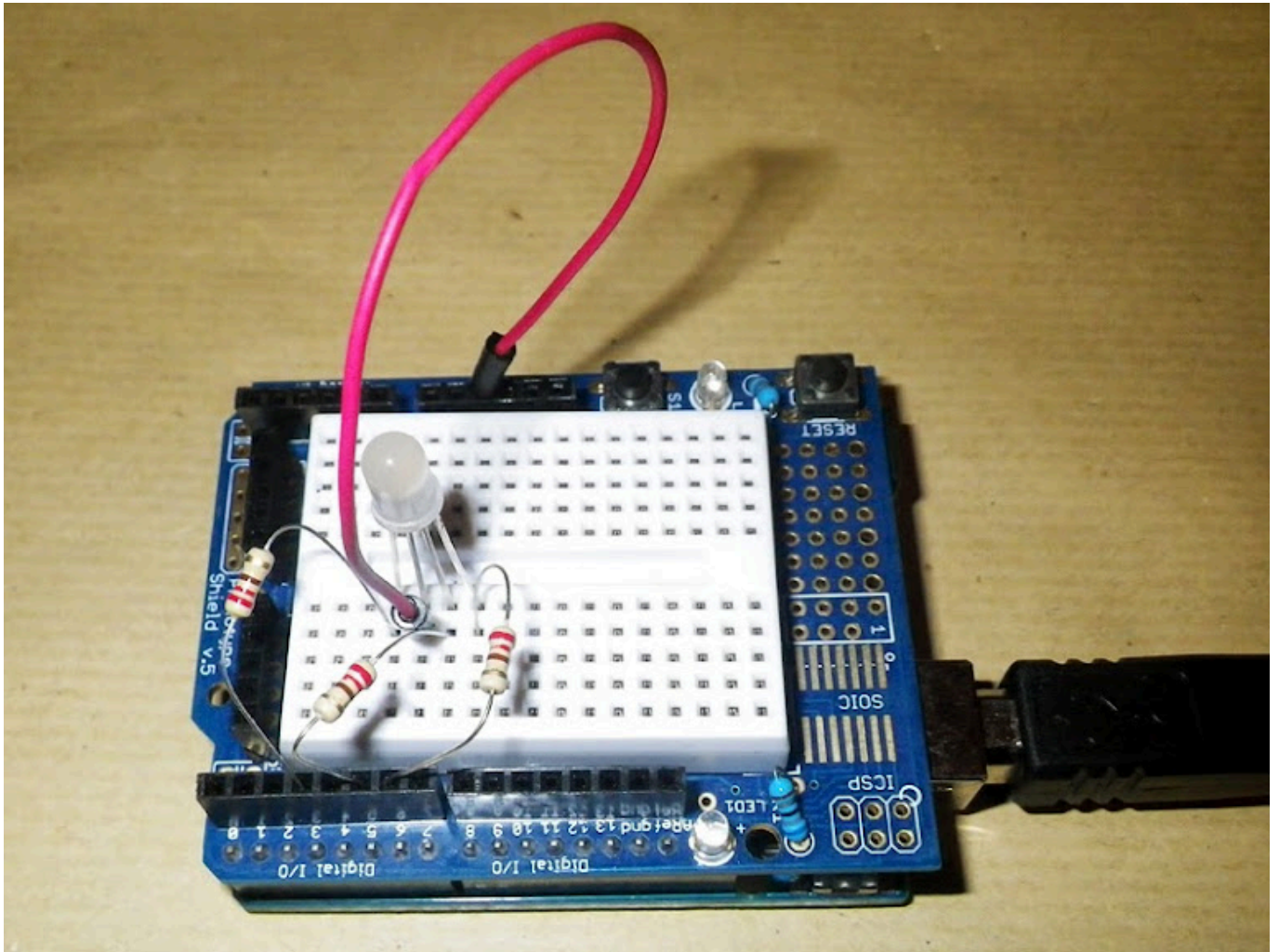
La connexion à cathode commune permet une logique directe : allumée à l'état haut.



- 1 Rouge -> pwm pin3
- 2 La grande patte est la cathode commune -> GND Arduino

- 3 Vert -> pwm pin 5
- 4 Bleu -> pwm pin 6

On utilise sur chaque couleur une résistance de limitation de courant de 200 Ohm.



Rouge : 625 nm

Vert : 520 nm

Bleu : 467 nm

Courant direct 20 mA, 30 mA en pic.

**LED Chip Typical Electircal & Optical Characteristics: (Ta=25 °C)**

ITEMS	Color	Symbol	Condition	Min.	Typ.	Max.	Unit
Forward Voltage	Red	V <sub>F</sub>	I <sub>F</sub> =20mA	1.8	2.0	2.2	V
	Green			3.0	3.2	3.4	
	Blue			3.0	3.2	3.4	
Luminous Intensity	Red	I <sub>v</sub>	I <sub>F</sub> =20mA	---	---	800	mcd
	Green			---	---	4000	
	Blue			---	---	900	
Wavelength	Red	Δλ	I <sub>F</sub> =20mA	620	623	625	nm
	Green			515	517.5	520	
	Blue			465	466	467.5	
Light Degradation after 1000 hours	Red	-4.68% ~ -8.27%					
	Green	-11.37% ~ -15.30%					
	Blue	-8.23% ~ -16.81%					

allume les 3 teintes à la fois, avec un effet progressif (PWM croissant puis décroissant) donc fait varier une lumière blanche modulée.

=====

tiptopboards.com

C. Rolland 07 09 2013

Utilisation d'une LED tri color RGB en mode PWM

Lumière blanche modulée avec fading périodique

=====

```
int RougePin = 3; R sur pin 3 pwm Arduino UNO
int VertPin = 5; G sur pin 5 pwm
int BleuPin = 6; B sur pin 6 pwm
int brightness = 0; LED brightness
int increment = 5; brightness increment
```

```
void setup()
{
pinMode(RougePin, OUTPUT);
pinMode(VertPin, OUTPUT);
pinMode(BleuPin, OUTPUT);
Serial.begin(9600);
}
```

```
void loop()
{
=== Effet de lumière blanche (R=G=B) et fading progressif
brightness = brightness + increment; augmenter l'intensité lumineuse

if (brightness == 0 || brightness >= 255) inverser le sens du fading
{
increment = -increment;
}
```

```

brightness = constrain(brightness, 0, 255);
analogWrite(RougePin, brightness);
analogWrite(VertPin, brightness);
analogWrite(BleuPin, brightness);
delay(30); pause 20 ms pour ralentir les effets de lumière

}

```

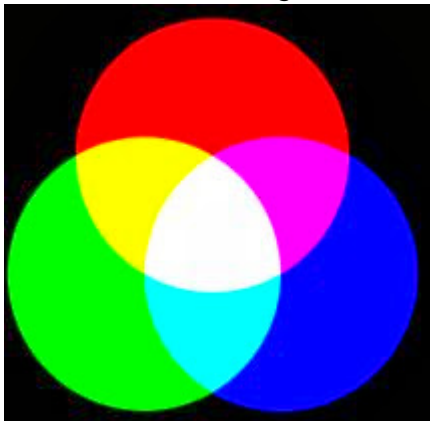
: la première fonction du programme mélange les 3 couleurs primaires de façon simple. Les 8 teintes de base sont ici stockées dans un simple tableau à 3 colonnes (r, g, b), le premier élément porte le numéro 0. Ces .

```

{0,0,0}, noir (0 teinte)
{255,0,0}, rouge (1 teinte)
{0,255,0}, vert (1 teinte)
{0,0,255}, bleu (1 teinte)
{255,255,0}, rouge + vert = jaune (2 teintes)
{255,0,255}, magenta = rouge + bleu (2 teintes)
{0,255,255}, cyan = vert + bleu (2 teintes)
{255,255,255} blanc = rouge + vert + bleu (3 teintes)

```

On fait ici des mélanges de teintes (R, G, B) d'intensité maximale 255 chacune.



Le seconde fonction du programme produit un du ROUGE → Jaune → VERT →Cyan → BLEU → Magenta → ROUGE

```

=====
tiptopboards.com
C. Rolland 07 09 2013
Utilisation d'une LED tri color RGB en mode PWM

```

- 1 - Utilisation d'un tableau de 8 couleurs en mémoire
- 2 - Exemple de mélange graduel de teintes adidtive

```

=====
int RougePin = 3; R sur pin 3 pwm Arduino UNO
int VertPin = 5; G sur pin 5 pwm
int BleuPin = 6; B sur pin 6 pwm

```

```
void setup()
{
pinMode(RougePin, OUTPUT);
pinMode(VertPin, OUTPUT);
pinMode(BleuPin, OUTPUT);
Serial.begin(9600);
}

void loop()
{
=== Succession de 8 teintes fixes
Serial.println("8 couleurs successives");
huitCouleurs(); Succession de 8 teintes fixes

=== Transition R > G > B progressif avec fondu
Serial.println("3 couleurs en fondu");
couleursFondu();
}

=== Affichage de 8 teintes successives
void huitCouleurs(){
tableau de 8 teintes en mémoire
int duree = 400; delai en ms entre changements
int mesCouleurs[8][3] = {
{0,0,0},
{255,0,0},
{0,255,0},
{0,0,255},
{255,255,0},
{255,0,255},
{0,255,255},
{255,255,255}
};
0 noir
1 2 3 rouge vert bleu (pur 1 teinte)
4 5 6 jaune magenta cyan (mélange de 2 teintes)
7 blanc (mélange de 3 teintes)

for (int i=1; i<=7; i++)
{
int r = mesCouleurs[i][0];
int g = mesCouleurs[i][1];
int b = mesCouleurs[i][2];
analogWrite(RougePin, r);
analogWrite(VertPin, g);
analogWrite(BleuPin, b);
Serial.print("couleur ");
Serial.print(i);
Serial.print(" ");
}
```

```
Serial.print®;
Serial.print(" ");
Serial.print(g);
Serial.print(" ");
Serial.println(b);
delay(duree); delai entre les couleurs simples qui défilent
} Boucle i des 8 couleurs
} fin de la fonction HuitCouleurs
```

```
==== 3 couleurs pures en dégradé fondu
void couleursFondu(){
int duree = 5; intervale en ms entre transitions
Départ sur rouge 255
Serial.println("RED ...GREEN ");
Ajouter progressivement du vert et retirer du rouge
for( int g = 0; g < 255; g++ ) {
int r = 255 - g;
analogWrite(RougePin, r); rouge décroissant
analogWrite(VertPin, g); vert croissant
analogWrite(BleuPin, 0);
Serial.println®;
delay(duree); progressif
} Le vert est maintenant à 255
```

```
Serial.println("GREEN ... BLUE");
Ajouter progressivement du bleu et retirer du vert
for( int b = 0; b < 255; b++ ) {
int g = 255 - b;
analogWrite(RougePin, 0); rouge éteint
analogWrite(VertPin, g); vert décroissant
analogWrite(BleuPin, b); bleu croissant
Serial.println(g);
delay(duree);
} boucle g
```

```
Ajouter progressivement du rouge et retirer du bleu
Serial.println("BLUE ... RED");
for( int r = 0; r<255; r++ ) {
int b = 255 - r;
analogWrite(RougePin, r); rouge progressif
analogWrite(VertPin, 0); vert éteint
analogWrite(BleuPin, b); bleu décroissant
Serial.println(b);
delay(duree);
} boucle r
} Fin de la fonction couleursFondu
```

Pour obtenir la référence d'une couleur précise ,on peut utiliser le mélangeur en ligne

## Convertisseur Hexadécimal (HEX) / Décimal (RGB)

Canal RGB	Hexadécimal	Décimal RGB
- <b>Red</b> +	F8	248
- <b>Green</b> +	C7	199
- <b>Blue</b> +	60	96
- <b>Tout</b> +	Montrer	Montrer

<http://www.proftnj.com/RGB3.htm>

Exemple (248,199,96)

```
analogWrite(RougePin, 248);
```

```
analogWrite(VertPin, 199);
```

```
analogWrite(BleuPin, 96);
```

Référence du PWM sur Arduino

<http://arduino.cc/en/Tutorial/PWM>

Fonction analogWrite sur Arduino

<http://arduino.cc/en/Reference/AnalogWrite>

Le système de mélange de couleur additif rouge, vert, bleu

[http://fr.wikipedia.org/wiki/Rouge\\_vert\\_bleu](http://fr.wikipedia.org/wiki/Rouge_vert_bleu)

From:

<https://wiki.libreeduc.cc/> - **LibrEduc**

Permanent link:

[https://wiki.libreeduc.cc/fr:arduino:led\\_rgb](https://wiki.libreeduc.cc/fr:arduino:led_rgb)

Last update: **2025/01/16 20:24**

