

Sortie numérique PWM simulant un signal analogique - `analogWrite()`

<https://docs.arduino.cc/built-in-examples/basics/Fade/>

💡 Application : générer un signal pseudo-analogique

Instructions

analogWrite(pin, niveau)

Ecrire la valeur *niveau* sur la broche *pin* (~3, ~5, ~6, ~9, ~10, ~11 PWM = symbole ~).
`analogWrite()` accepte des valeurs entre 0 et 255.

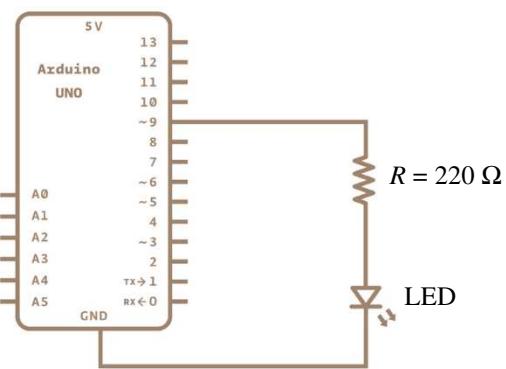
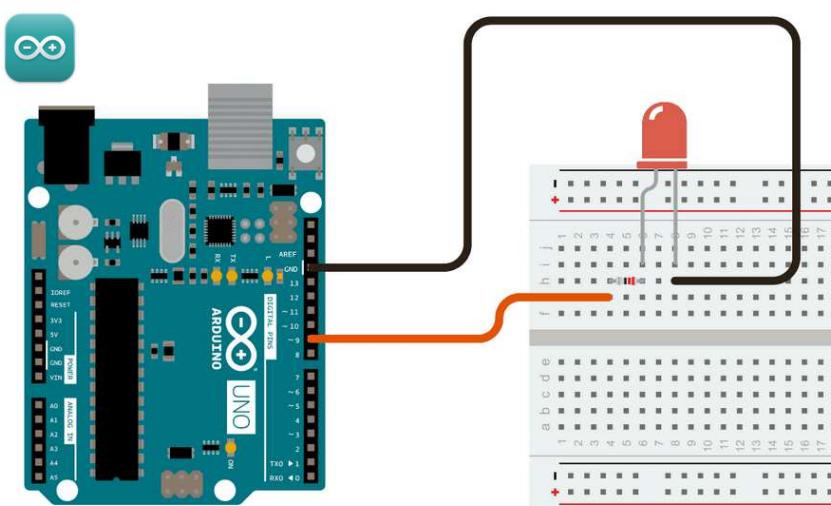
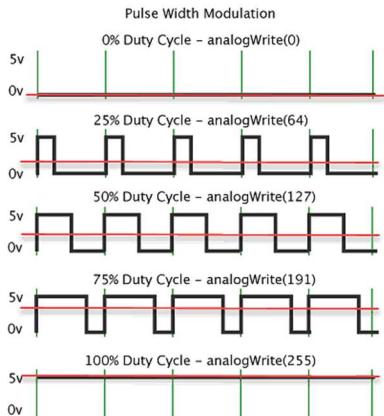
Modulation de largeur d'impulsion (schéma ci-contre) : en faisant varier le *rapport cyclique* (durée du niveau haut / durée du niveau bas) des impulsions, on fait varier la *valeur moyenne* du signal.

C'est cette valeur moyenne qui simule un signal analogique. Le rapport cyclique pouvant varier rapidement, la valeur moyenne peut également évoluer au cours du temps et simuler un signal analogique variable dans le temps.

 On fait varier la luminosité d'une LED en utilisant une sortie numérique PWM à modulation de largeur d'impulsion.

Schéma du circuit

La broche PWM utilisée est la broche ~9.



Sketch

The C++ logo is a blue hexagon containing a white 'C' and two white '+' symbols.

```
1 int led = 9;           // Nom de variable : n° de la broche PWM connectée à la LED
2 int intensite = 0;     // Niveau de luminosité
3 int variationI = 5;    // Pas = ΔI de variation de l'intensité
4
5 void setup() {
6     // Configure la broche 9 = led en sortie (OUTPUT)
7     pinMode(led, OUTPUT);
8 }
9
10 void loop() {
11     // Fixe le niveau de luminosité de la led en écrivant ce niveau à la broche 9
12     analogWrite(led, intensite);
13
14     //
15     intensite = intensite + variationI;
16
17     //
18     if (intensite <= 0 || intensite >= 255) {
19         variationI = -variationI;
20     }
21     delay(30);
22 }
```

💡 Expliquer le code (lignes 15, 18 et 19).