



Contrôle d'intensité - Arduino

1. Objectifs :

- Contrôler l'intensité de la lumière émise par une ampoule DEL.
- Ajouter un commentaire dans un programme.
- Utiliser une boucle for.
- Utiliser un opérateur de contrôle.

2. Matériel :

- Arduino UNO
- Résistance de 220 Ω
- DEL de la couleur désirée
- Ordinateur + fil de branchement pour la carte
- Quelques fils
- Platine d'essai

3. Ce que fait le programme

Ce programme permet de moduler l'intensité de la lumière émise par une DEL. L'intensité passera de nulle à maximum à plusieurs reprises.

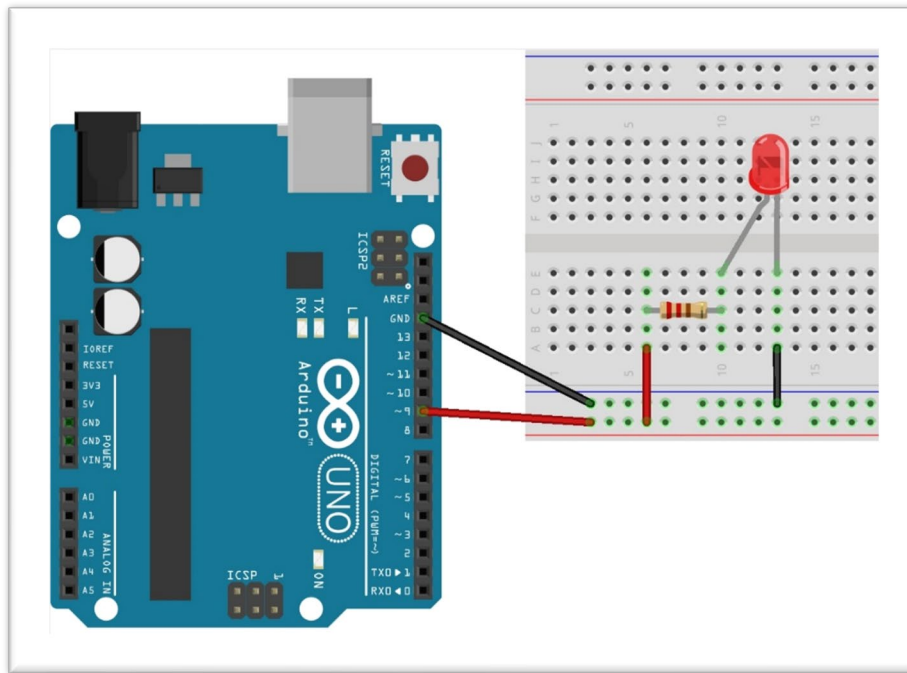
4. Étapes de réalisation

1. Brancher la carte Arduino UNO dans l'ordinateur avec le fil approprié. La DEL (verte) d'alimentation de la carte devrait être allumée.
2. Ouvrir le logiciel Arduino. Il est possible d'utiliser la version en ligne disponible au www.arduino.cc
3. On doit ensuite s'assurer que le bon type de carte et le bon port de communication sont utilisés. Dans l'onglet *Outils*, modifier le *Type de carte* dans la liste déroulante si « Arduino/Genuino UNO » n'est pas sélectionné. Sélectionner aussi le bon port de communication.
4. Réaliser le circuit demandé (voir plus bas).
 - a. La carte Arduino doit rester branchée à l'ordinateur.
 - b. La patte plus longue de la DEL se branche du côté de la résistance.



- c. Les fils n'ont pas à être de la couleur qui est indiquée sur le schéma.
5. Copier ou écrire le code dans le sketch.
6. Compiler et téléverser le programme dans la carte. S'il y a des erreurs, elles seront indiquées en orange/rouge en bas de la fenêtre du programme.

5. Schéma du circuit



6. Code

```
int delPin = 9;
int controle = 5;

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

void loop() {
  for(int intensite=0; intensite<=255 ; intensite=intensite+controle){
    //Changer l'intensité de la DEL
    analogWrite(delPin, intensite);
    //Attendre 30 ms pour voir la variation d'intensité
    delay(30);
  }
}
```

7. Fonctions utilisées

```
for(int i =0 ; condition ; i = i +increment){  
}
```

La fonction **for** (« pour » en français) est une fonction très importante et très utilisée pour réaliser des boucles de contrôle dans un programme.

Elle a trois arguments :

1. Le premier est la valeur d'**initialisation**. Ici, on déclare la variable *i* qui n'existera donc que dans la boucle *for*. On lui attribue sa valeur de départ dans la boucle. C'est la valeur de la variable *i* au premier passage dans la boucle.
2. Le deuxième argument est la **condition**. La boucle sera exécutée tant que la condition est vraie. Si la condition devient fausse, on sort de la boucle et on passe aux commandes suivantes.
3. Le troisième argument est appelé l'**incrément**. Il indique comment la variable *i* sera modifiée à chaque exécution du bloc d'instructions de la boucle. L'incrément peut être fait avec une soustraction, une multiplication ou une division.

Le déroulement de l'exécution de la boucle sera ainsi :

Début de la boucle :

1^{er} passage :

i=0

la condition est vraie, on exécute le bloc de commandes

i=0+1 =1

2^e passage :

i=1

La condition est vraie, on exécute le bloc de commandes

i=1+1=2

3^e passage :

i=2

La condition est vraie, on exécute le bloc de commandes

i=2+1=3

....

257^e passage :

i=256

La condition est fausse, on n'exécute pas le bloc de commandes et on sort de la boucle *for*.

Fin de la boucle

```
i <= 255
```

Dans la boucle *for*, c'est la façon dont la condition a été écrite. Le symbole « <= » est un opérateur de comparaison. Il signifie simplement « plus petit ou égal ». Donc, tant que la valeur de *i* est plus petite ou égale à 255, la boucle va s'exécuter. Elle s'arrête lorsque *i* a une valeur supérieure ou égale à 255. Il existe d'autres opérateurs de comparaison.

```
//Attendre 30 ms pour voir la variation d'intensité
```

Toute ligne qui débute par « // » ne sera pas prise en compte lors de la compilation du programme. On se sert de ces lignes pour écrire des commentaires. Ceux-ci permettent d'indiquer des informations sur le déroulement d'une fonction ou d'une partie du programme. Il est d'usage et de bonne pratique d'écrire assez de commentaires pour que quelqu'un d'autre puisse en comprendre les étapes.

```
analogWrite( nomDePin , valeur);
```

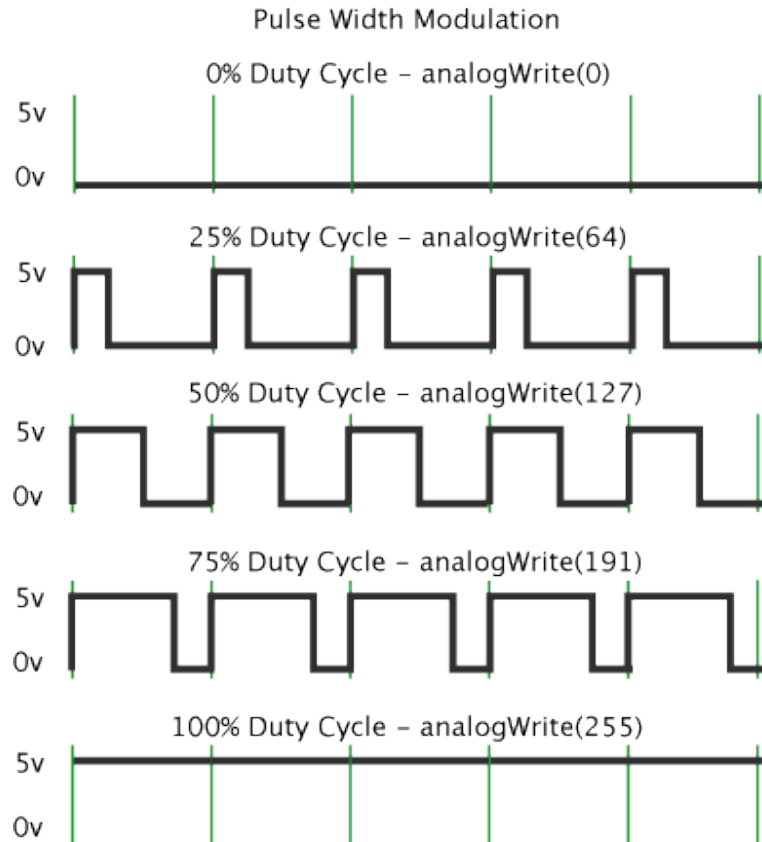
La fonction ***analogWrite*** est utilisée pour contrôler l'intensité d'une lumière ou encore la vitesse d'un moteur. Elle ne fonctionne que sur les broches digitales qui ont un symbole « ~ ». Sur une carte de type Arduino UNO, les broches sont 3, 5, 6, 9, 10 et 11. Les broches analogiques A0 à A5 ne le permettent pas.

La valeur est un nombre entier compris entre 0 et 255.

Comme vu précédemment, les broches digitales peuvent envoyer un signal de 5V (HIGH) ou de 0V (LOW). Il n'est pas possible d'envoyer un signal intermédiaire (2.5 V par exemple). Pour réussir à avoir un résultat qui simule l'effet d'une tension intermédiaire, on procède en modulant la largeur d'une impulsion carrée (« Pulse Width Modulation » ou PWM).

La carte Arduino génère une onde carrée à une fréquence d'environ 500 Hz. Cela correspond à 500 cycles par seconde ou encore un cycle de 0,002 s.

La *valeur* détermine la fraction du cycle pendant laquelle la valeur de la tension sera à 5V. La figure de la page suivante montre le principe de base.



The arduino.cc team [CC BY-SA 3.0 (<https://creativecommons.org/licenses/by-sa/3.0/>)]

Il est à noter que l'œil humain ne peut pas percevoir le clignotement d'une DEL à une fréquence de 500 Hz. On perçoit moins de lumière provenant de la DEL lorsque le temps où la tension est à 5V est plus court.

Défis :

- Ajouter votre propre commentaire à une partie du programme.
- Faire passer l'intensité de la lumière de maximum à minimum.
- Modifier le programme pour faire d'autres rythmes de modulation de l'intensité.
- Faire passer l'intensité de la lumière de maximum à minimum et ensuite à maximum en répétition.
- Faire clignoter la lumière en plus d'en moduler l'intensité
- Tout autre chose que vous voudriez essayer!