

1. Objectif

Dans ce TP, nous allons faire varier la **luminosité** d'une LED.

Pour commencer, nous l'allumerons avec une intensité prédéfinie :

- 1- Eteinte ;
- 2- A moitié allumée ;
- 3- Allumée « plein gaz ».

Ce problème n'est pas anodin sachant qu'une sortie de carte Arduino n'est sensée fournir que 2 états pour la LED :

- 4- Eteinte ;
- 5- Allumée.

Ce que nous allons découvrir, nous sera utile dans nos futures projets pour modifier la vitesse de rotation d'un moteur, l'angle de rotation d'un servo-moteur, la température de chauffe d'une résistance...

Nous allons pousser plus loin ce principe, en pilotant une LED RGB capable de restituer n'importe quelle couleur à partir du moment où le sujet est maîtrisé.

2. Préparation

2.1 Hardware

Nous allons allumer une LED câblée sur O2 du shield Tinkerkit et faire varier son intensité à l'aide de différents programmes.

2.2 Software

Pour rédiger ces programmes, en plus des instructions des programmes précédents, il nous faudra :

- Comprendre le fonctionnement d'une LED avec le document ressource : **RES_Diode_&_LED.pdf**
- Apprendre le comportement d'une nouvelle instruction : ***analogWrite()***

Syntaxe `analogWrite(broche, RCN)`

broche : n° de la broche sur laquelle il faut appliquer le signal PWM.

RCN : variable de type byte (codée sur un octet) comprise entre 0 et 255.

```
// --- Fonction LOOP ---
analogWrite(pinLED, RCN) ;

// cette instruction sera placée dans le LOOP
// allumer la LED avec une luminosité définie
// par la valeur 0 <= RCN <= 255
```

Remarque : RCN est le **Rapport Cyclique Numérique** basé non pas sur un pourcentage mais sur une valeur numérique codée sur un octet donc entre 0 et 255 :

- 6- RCN = 0 (**correspondant à 0% de luminosité**), la LED est éteinte ;
- 7- RCN = 255 (**correspondant à 100% de luminosité**), la LED est allumée « plein gaz » ;
- 8- Pour les **254** autres valeurs, **suspense...**

Remarque : Cette instruction utilisant un signal **PWM** (*Pulse Width Modulation* ou MLI Modulation de Largeur d'Impulsion) pour moduler la luminosité de la LED et aussi utilisée pour moduler :

- la vitesse de rotation des moteurs électriques ;
- l'angle de rotation d'un servo-moteur ;
- le chaleur émise par une résistance chauffante électrique.

Remarque : Sur la carte Arduino, un signal PWM ne peut être fourni que par les sorties PWM. Celles-ci sont bien différenciées des autres sorties **TOR** (Tout Ou Rien ⇔ 1 ou 0) qui ne bénéficient pas de la possibilité offerte par l'instruction ***analogWrite()***. Ces sorties PWM sont marquées d'un « ~ ». Par contre elles, elles peuvent être utilisées comme des sorties TOR.

Remarque : La carte Arduino/UNO que nous utilisons comporte 6 sorties qui peuvent être PWM. Ce sont les broches : 3, 5, 6, 9, 10 et 11.

► Utiliser une structure de boucle nouvelle : ***while*** (tant que...)

Syntaxe `while(expression){ ... }`
expression : Une instruction (booléenne) qui renvoie un résultat VRAI ou FAUX.

Description

Les boucles ***while*** bouclent indéfiniment jusqu'à ce que la condition ou l'expression entre les parenthèses () devienne fausse. Quelque chose doit modifier la variable testée, sinon la boucle ***while*** ne se terminera jamais. Cela peut être dans votre code, soit une variable incrémentée, ou également une condition externe, soit le test d'un capteur.

3. Travail à faire

Après avoir câblé la LED sur la carte Arduino/UNO et le shield Tinkerkit, trois programmes seront à l'étude :

- ① Allumer la LED avec une luminosité fixée directement dans le programme ;
- ② Faire varier automatiquement la luminosité de manière progressive puis régressive ;
- ③ Faire varier la couleur d'une LED tricolore.

3.1 Premier programme

Algorithme

```
Définir une valeur luminosité : RCN = 127 // pour une luminosité à 50%
Début Algorithme Prog_TP06_Grada_LED_Fixe
    FaireToujours
        Allumer la LED à l'aide d'un signal PWM de RCN = 127
    FinFaireToujours
```


Code du programme « Prog TP06 Grada LED Fixe » :


```
/* Programme v1.0
 * TP06 pour allumer une LED (PWM9) en utilisant la fonction analogWrite()
 * le 27 oct 2020
 * Yann Langrée
 */

const int pinLED = 9;
byte RCN = 127;

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

void loop() {
  analogWrite(pinLED, RCN);
}
```

 **Copiez** ce programme dans l'EDI/Arduino et **testez-le**.

 **Q1. Expliquez** le comportement de ce programme en y ajoutant des commentaires et en l'intégrant dans la rédaction de votre compte-rendu (CR).


3.2 Second programme


Nous voulons que la luminosité de la LED (O2) varie automatiquement, de façon progressive, puis régressive, sans arrêt. La luminosité variera par palier (15 paliers dans un sens puis 15 paliers dans l'autre sens).

Algorithme

```
Début Algorithme TP9_Grada_LED_Varie
  FaireToujours
    Initialiser la luminosité à 0
    Tant que la luminosité est inférieure à 255
      Allumer la LED à l'aide d'un signal PWM de RCN = luminosité
      Augmenter la luminosité de 17
      Attendre 100 ms pour que le phénomène soit visible
    FinTantque
  FinFaireToujours
```

Code du programme « Prog TP06 Grada LED Var » :

 **Q2. Rédigez** un programme respectant l'algorithme précédent.

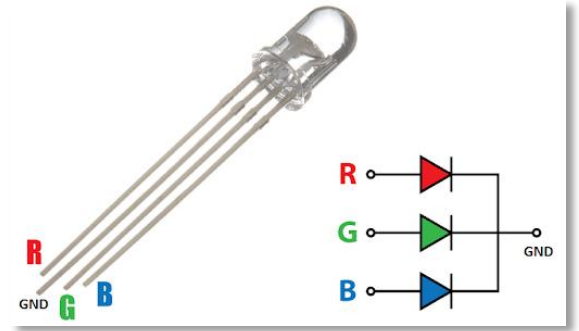
 **Copiez-le** dans l'EDI/Arduino, **testez-le**.

 **Q3. Validez** son comportement dans votre CR.

 **Validez** votre travail auprès du Prof 🙋😊 ...

3.3 Projets

Nous voulons aller plus loin en faisant varier la couleur d'une LED RGB utilisant le principe de réglage de luminosité sur les trois couleurs primaires : **Red, Green and Blue**.



🔧 **Q4. Dessinez** le câblage de la LED RGB pour tester son comportement sur une Breadboard.

🔧 **Testez** le fonctionnement de la LED RGB.

🔧 **Q5. Proposez** un schéma de câblage de la LED RGB sur la carte Arduino/UNO.

🔧 **Testez** le fonctionnement de votre montage.

🔧 **Q6. Validez** vos tests dans votre CR.

Projet 1

Algorithme

🔧 **Q7. Rédigez** un algorithme permettant d'allumer la LED RGB dans une couleur de votre choix.

Code du programme « Prog TP06 Projet1 » :

🔧 **Q8. Rédigez** un programme respectant l'algorithme précédent.

🔧 **Copiez-le** dans l'EDI/Arduino et **testez-le**.

🔧 **Q9. Validez** son comportement dans votre CR.

Projet 2

Algorithme

🔧 **Q10. Rédigez** un algorithme permettant de faire varier les couleurs progressivement.

Code du programme « Prog TP06 Projet2 » :

🔧 **Q11. Rédigez** un programme respectant l'algorithme précédent.

🔧 **Copiez-le** dans l'EDI/Arduino et **testez-le**.

🔧 **Q12. Validez** son comportement dans votre CR.

🔧 **Validez** votre travail auprès du Prof 🙋😊 ...

Projet 3

La carte Arduino Uno possède six entrées analogiques notées A0 à A5. Nous pouvons appliquer sur ces entrées des tensions comprises entre 0 et 5V et le convertisseur analogique-numérique (à approximations successives des ATmega) transforme ces tensions en mots binaires de 10 bits.

L'instruction permettant de lancer la conversion analogique-numérique est : **analogRead(broche)**
où broche représente le numéro de la broche concernée.

Extrait d'un programme utilisant cette instruction :

```
int pinPOT = A0 ;
int pinLED = 3 ;
int valCAN = 0;
int val = 0 ;

void setup(){
    pinMode(pinPOT, INPUT);
    pinMode(pinLED, OUTPUT);
}

void loop(){
    valCAN=analogRead(pinPOT);           //conversion AN
    val=map(valCAN, 0, 1023, 0, 255);    //mise à l'échelle de val
    analogWrite(pinLED, val);            //le rapport cyclique de la sortie pinLED
                                        //   varie en fonction de la variable val
}
```

Nous allons donc utiliser un potentiomètre branché sur une de ces entrées analogiques pour faire varier la tension de cette entrée.

Algorithme

- 🔗 **Q13.** En vous inspirant de l'extrait de programme précédent, **récrivez** un algorithme permettant de faire varier l'intensité lumineuse d'une LED branchée sur la broche 3 en fonction de la valeur de la tension appliquée sur la broche A0.

Code du programme « Prog TP06 Projet1 » :

- 🔗 **Q14. Rechercher** sur internet l'action réalisée sur la variable **val** par l'instruction **map()**.
- 🔗 **Q15. Rédigez** un programme respectant l'algorithme précédent.
- 🔗 **Copiez-le** dans l'EDI/Arduino et **testez-le**.
- 🔗 **Q16. Validez** son comportement dans votre CR.
- 🔗 **Validez** votre travail auprès du Prof 🙌😊 ...