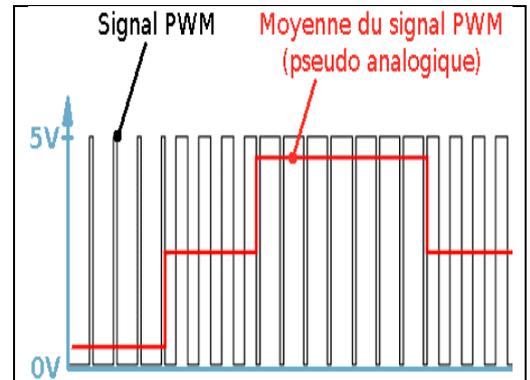


Noms : _____
 Prénoms : _____
 Classe : _____
 Date : _____

Note : /20



Objectif du TP

- Découvrir les bases de la programmation sur Arduino
- Découvrir la notion de rapport cyclique
- Découvrir le fonctionnement des diodes RGB

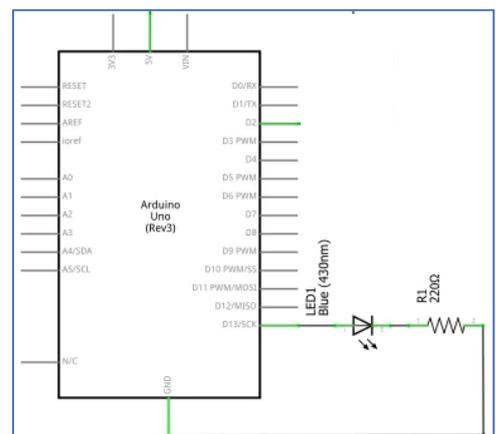
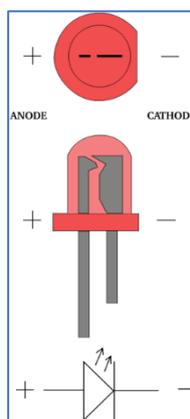
Barème

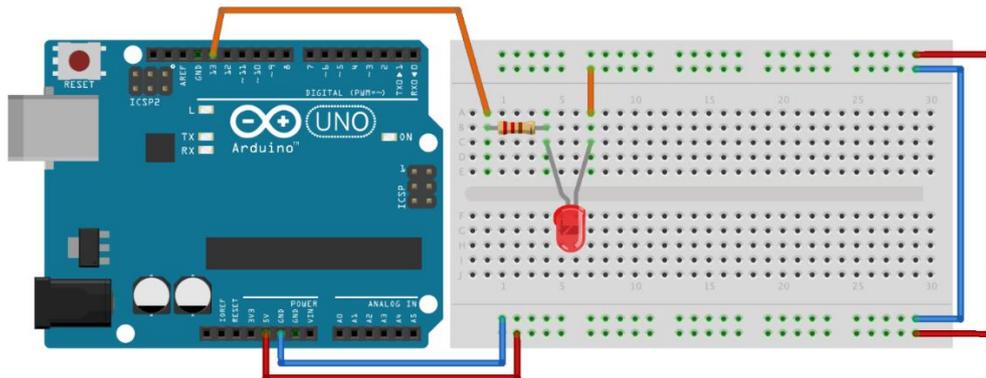
Autonomie, quantité de travail, qualité du compte-rendu	/2
Découverte d'un programme Arduino (Q1)	/3
Utilisation du moniteur série (Q2)	/2
Modifier l'intensité lumineuse d'une diode (Q3, 4, 5)	/7
Rapport cyclique du signal MLI (Q6, 7, 8, 9)	/6

1. Faire clignoter la diode branchée sur la sortie 13

Réaliser le montage suivant. Attention à bien respecter le sens de branchement de la diode

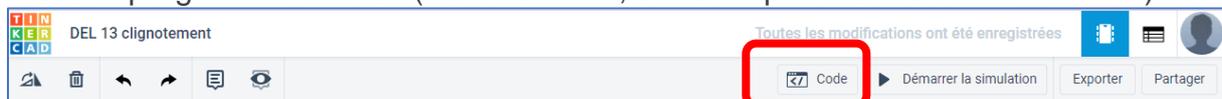
La valeur de la résistance est de 220Ω .





fritzing

Saisir le programme suivant (sur Tinkercad, il faut cliquer sur le bouton « code ») :



```

1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(13, HIGH);
9   delay(1000); // Wait for 1000 millisecond(s)
10  digitalWrite(13, LOW);
11  delay(1000); // Wait for 1000 millisecond(s)
12 }

```

Q1 : Indiquer le (ou les) numéros de lignes associée(s) à chaque élément :

N° de lignes	Elements
	Attendre 1 seconde
	Lignes lues et exécutées en boucle
	Lignes correspondant à l'initialisation
	Initialiser la broche 13 comme une sortie
	Allumer la diode
	Eteindre la diode

Tester le programme, soit :

- en utilisant l'IDE Arduino en double cliquant sur : , puis en téléversant le programme sur la carte
- en lançant la simulation sur le site Tinkercad.

2. Utiliser le moniteur série

Pour visualiser des valeurs sur le moniteur série, il faut :

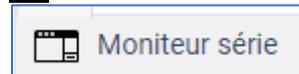
- Modifiez le programme en écrivant dans le set up :
`Serial.begin(9600)` ;

Cela permet d'initialiser la vitesse de communication entre le PC et la carte à $9600\text{bits}\cdot\text{s}^{-1}$

- Modifiez le programme en écrivant à deux endroits (judicieusement choisis) dans la boucle « Loop » :
`Serial.println(digitalRead(13))` ;

Cela permet d'afficher la valeur lue sur la broche 13 et de passer à la ligne.

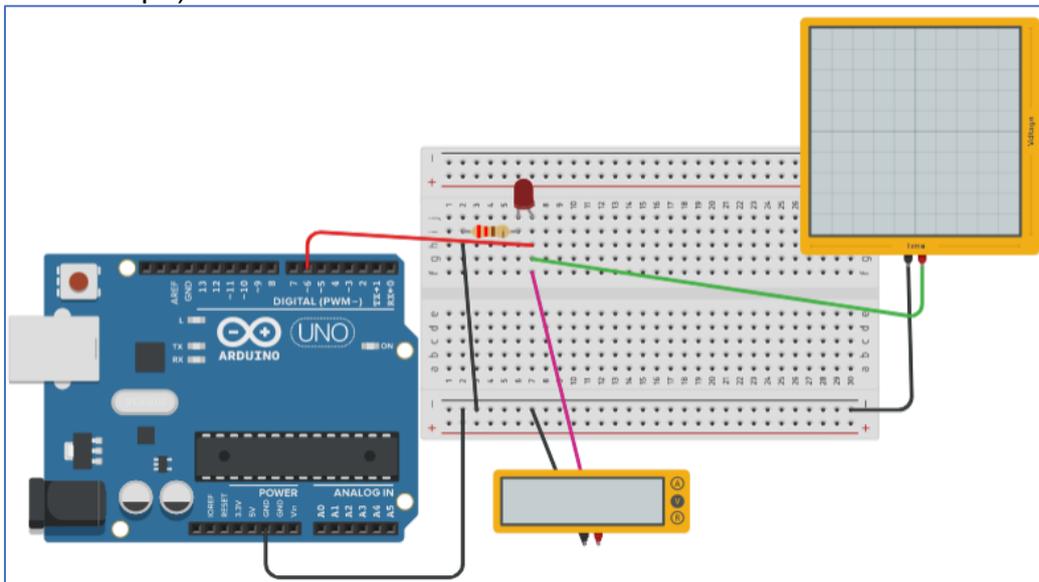
Q2 : Lancez la simulation, cliquez sur code puis sur Moniteur série (en bas de l'écran) :



. Que lisez-vous ? comment interprétez-vous ces valeurs ?

3. Modifier l'intensité lumineuse de la diode

Réaliser le circuit ci-dessous (une diode, une résistance de 220Ω , un voltmètre et un oscilloscope) :



Saisir le code ci-dessous :

```

1  int led_pin=6;
2  void setup()
3  {
4      pinMode(led_pin, OUTPUT);
5      Serial.begin(9600);
6  }
7
8  void loop()
9  {
10     for(int i=0; i<255; i++){
11         Serial.println(i);
12         analogWrite(led_pin, i);
13         delay(10);
14     }
15     for(int i=255; i>0; i--){
16         Serial.println(i);
17         analogWrite(led_pin, i);
18         delay(10);
19     }
20 }

```

Q3 : Notez ce que vous observez concernant l'intensité lumineuse de la diode.

Q4 : Indiquer le (ou les) numéros de lignes associée(s) à chaque élément :

N° de lignes	Elements
	Attendre 10 millisecondes
	Définir la variable « led_pin » (type et valeur)
	Initialiser la broche utilisée comme une sortie
	Allumer la diode avec la valeur « i »
	Initialisation la vitesse de communication avec le moniteur série
	Définir un compteur pour incrémenter une variable « i »
	Définir un compteur pour décrémenter une variable « i »
	Faire en boucle

Q5 : noter les valeurs minimales et maximales de la tension moyenne aux bornes de la diode. Noter les valeurs minimales et maximales lues sur le moniteur série.

Pour visualiser la variation de « i » au cours du temps en cliquant, en bas à droite de l'écran, sur : 

Q6 : Pour faire varier l'intensité lumineuse de la diode, on l'alimente par une tension en créneaux. La période du signal étant faible (20ms), la diode se comporte comme si elle était alimentée par la tension moyenne du signal.

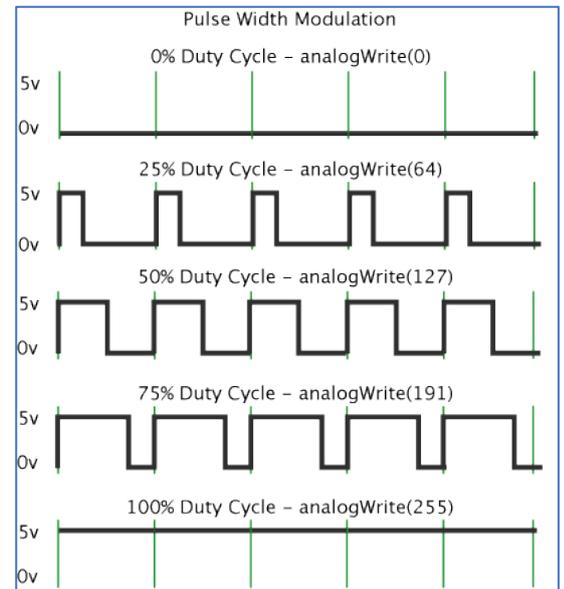
Nom de cette technique : modulation de largeur d'impulsion (MLI) – ou PWM en anglais pour Pulse Width Modulation

Le rapport cyclique permet de quantifier le temps au niveau haut par rapport à la période du signal. Il est souvent exprimé en pourcent.

$$\alpha = \frac{\tau}{T} = \frac{V_{moy} - V_{min}}{V_{max} - V_{min}}$$

rapport cyclique (sans unité) temps à l'état haut τ / période du signal T valeur moyenne V_{moy} - valeur minimale V_{min} / valeur maximale V_{max} - valeur minimale V_{min}

même unité de temps (s, min, etc.) même unité d'amplitude (V, A, W, etc.)



Sachant que le rapport cyclique est codé sur 8 bits, et que $2^8 = 256$, calculer la plus petite variation de tension moyenne que l'on peut lire aux bornes de la carte Arduino.

Q7 : Pour générer une tension moyenne de 4V, quel mot faut-il envoyer (valeur comprise entre 0 et 255), que vaut le rapport cyclique exprimé en pourcent ?

```

10 for(int i=0; i<255; i++){
11   Serial.println(i);
12   analogWrite(led_pin, i);
13   delay(10);
14 }
15 for(int i=255; i>0; i--){
16   Serial.println(i);
17   analogWrite(led_pin, i);
18   delay(10);
19 }

```

Dans le code, on lit :

Q7 : Trouver la signification de i++ et de i--

Q8 : Si l'on avait voulu incrémenter la valeur par pas de 5, qu'aurait-il fallu écrire ?

4. La diode RGB (Red Green Blue)

Q9 : Cherchez le fonctionnement d'une diode RGB. La tester sur Tinkercad (utiliser les broches pwm (symbole : ) de la carte ; ne pas oublier d'ajouter les résistances).

