

Noms :

Prénoms :

Classe :

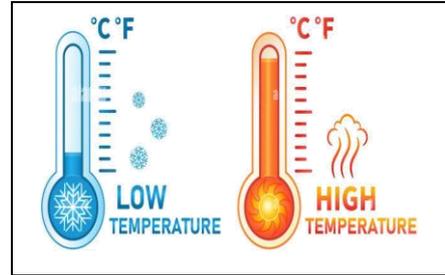
Date :

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_



**Note : /20**

## Compétences abordées :

- Traduire le comportement attendu ou observé d'un objet : comportement séquentiel, structures algorithmiques (variables, fonctions, structures séquentielles, itératives, répétitives, conditionnelles)
- Analyser le traitement de l'information : algorithmes, programmes, langages informatiques
- Traduire un algorithme en un programme exécutable
- Numération changement de bases
- Précision d'un capteur

## Problématique

**Contexte de départ :** on souhaite faire varier la couleur d'une diode RGB en fonction de la température relevée par un capteur

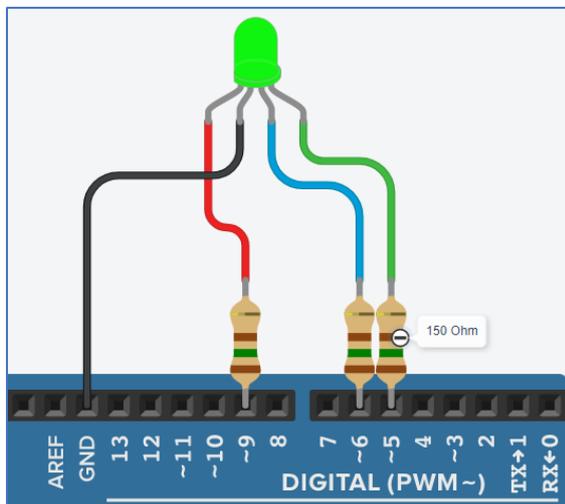
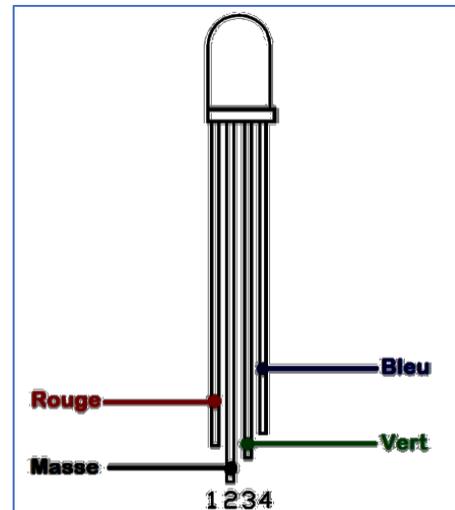
## Critères d'évaluation et barème

|   |    |
|---|----|
| Présentation / Soin / quantité de travail | /2 |
| Etude de la diode                         | /8 |
| Etude du capteur                          | /8 |
| Réalisation du programme global           | /2 |

## 1. La diode RGB

Une LED RGB est constituée de trois LEDs de couleurs Rouge-Vert-Bleu (Red-Green-Blue en anglais → RGB) réunies dans un même petit boîtier. Le mélange de ces trois couleurs permet d'obtenir une grande variété de teintes.

Elles sont facilement reconnaissables par le fait qu'elles possèdent 4 pattes (une masse + une patte de commande par couleur).



Pour brancher ce type de diode sur Arduino, il faut brancher les pattes pilotant les couleurs sur les broches numériques permettant de générer un rapport cyclique (celles précédées d'une petite vague : 3, 5, 6, 9, 10, 11) et ne pas oublier d'ajouter une résistance d'une valeur comprise entre 150 et 220Ω sur chaque patte, sauf la masse.

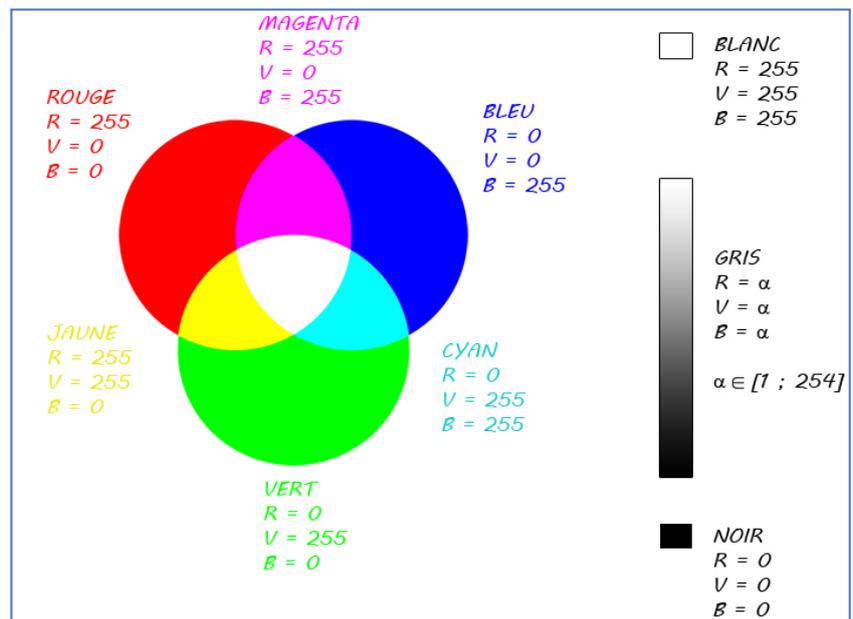
Réalisez le montage sur Tinkercad et programmez la séquence suivante :



**Q1)** Faites constater le bon fonctionnement par votre professeur et mettez une capture d'écran du montage et du programme dans votre compte-rendu.

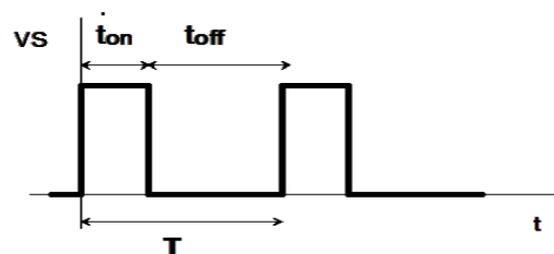
On allume une LED de couleur en appliquant aux bornes de celle-ci une valeur moyenne de tension comprise entre 0V (LED éteinte) et la tension maximale (LED à 100% de son intensité lumineuse).

Pour générer cette tension moyenne, on programme un rapport cyclique sur les boches pilotant chaque couleur. Sur Arduino, le rapport cyclique est codé sur 1 octet (8 bits)



**Rappel** : rapport cyclique :  $\alpha = t_{on}/T$

$t_{on}$  : temps à l'état haut sur une période ( $t_{on} < T$ )  
T : période du signal



L'œil humain peut distinguer environ 20 images par seconde. C'est à dire qu'au-dessus de ce seuil, il ne verra pas de clignotement entre les images, en dessous la succession d'images est perceptible.

- Q2)** Donner la période T maximale du signal de commande des LEDs pour qu'un utilisateur ne puisse pas percevoir de clignotement.
- Q3)** Sachant que la fréquence du PWM est de 490 Hz sur les boches 3, 9, 10, 11 et de 980 Hz sur les broches 5 et 6, calculer la période des signaux et vérifier qu'ils sont inférieurs à la période qui permettrait de voir un clignotement.
- Q4)** Sachant que chaque couleur est codée sur un octet, combien de couleurs différentes peut-on programmer ?

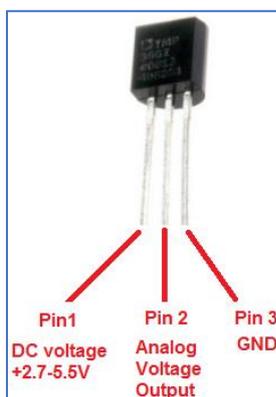
**Objectif** : On souhaite que toutes les 2 secondes la diode change de couleur pour afficher précisément les couleurs ci-dessous :



Le code de chaque couleur est exprimé en base 16

- Q5)** Convertir chaque code couleur en base 10
- Q6)** Convertir chaque code couleur en base 2
- Q7)** Sur Tinkercad, programmez cette séquence. Pour cela partez, de votre programme précédent, dans « code », cliquez sur « Texte » pour pouvoir modifier les valeurs des rapports cycliques sur les boches. Insérez une capture d'écran dans votre compte-rendu.

## 2. Le capteur de température



Le TMP36 est un capteur de température de précision basse tension. La tension de sortie est proportionnelle à la température dans la plage de  $-40^{\circ}\text{C}$  à  $+125^{\circ}\text{C}$ .

Alimentation: 2,7 à 5,5 Vcc

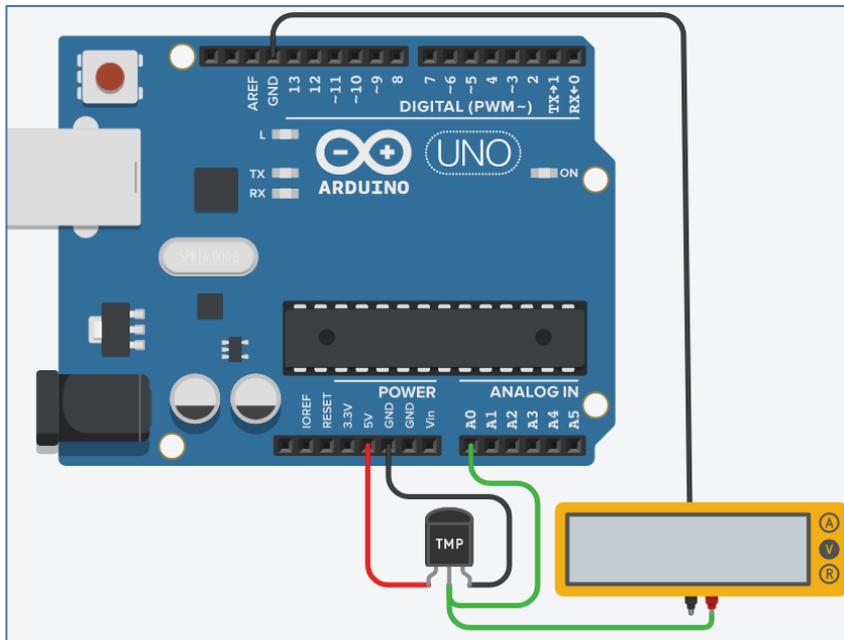
Plage de mesure:  $-40^{\circ}\text{C}$  à  $+125^{\circ}\text{C}$

Précision:  $\pm 1^{\circ}\text{C}$

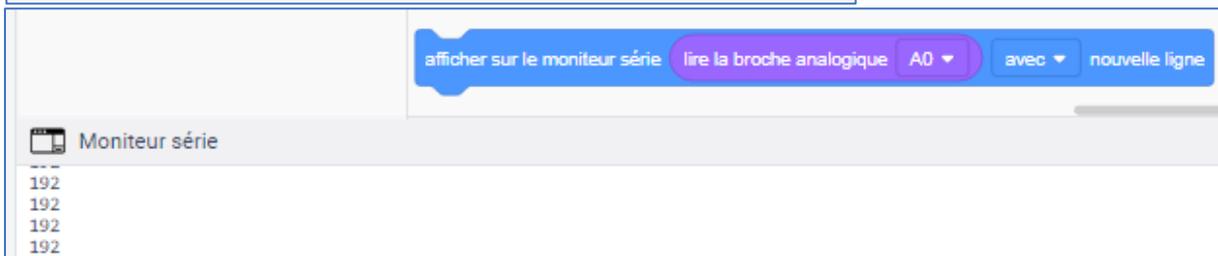
Conversion: 10 mV/ $^{\circ}\text{C}$



- Q8)** Réaliser le montage sur Tinkercad. Programmez le fait d'afficher la valeur lue sur la broche A0 sur la console. Insérez votre capture d'écran dans votre compte-rendu.



Lorsque vous lancez la simulation et que vous cliquez sur le capteur de température, vous pouvez faire varier la température de  $-40^{\circ}\text{C}$  à  $+125^{\circ}\text{C}$



Faites vérifier le bon fonctionnement de votre montage à votre professeur.

**Q9)** Sur combien de bits à codée la valeur lue sur la broche analogique A0 ?

**Q10)** Complétez le tableau suivant :

| Temperature ( $^{\circ}\text{C}$ ) | Tension (V) | Valeurs lues sur A0 |
|------------------------------------|-------------|---------------------|
| -40                                |             |                     |
| 0                                  |             |                     |
| 40                                 |             |                     |
| 80                                 |             |                     |
| 125                                |             |                     |

**Q11)** A partir de vos valeurs, calculer la précision du capteur en  $\text{mV}/^{\circ}\text{C}$ . Cette valeur est-elle conforme à celle indiquée dans la documentation technique ?

**Q12)** A partir de vos valeurs, calculer le quantum du convertisseur de la carte Arduino. Vérifiez, à partir de la documentation technique de la carte Arduino, que votre résultat est correct.

### 3. Pilotage de la couleur en fonction de la température

Programmer la carte pour que le système fonctionne de la façon suivante :

| Plage de températures | Couleur de la diode  |
|-----------------------|--|
| -40°C à 0             |  <span style="margin-left: 20px;">HEX #33FFDD</span>  |
| 0 à 40                |  <span style="margin-left: 20px;">HEX #33FF46</span>  |
| 40 à 80               |  <span style="margin-left: 20px;">HEX #FF9333</span>  |
| 80 à 125              |  <span style="margin-left: 20px;">HEX #FF0C04</span> |

**Q13)** Faites une capture d'écran de votre programme et insérez-le dans votre compte-rendu