

# **TP Capteurs et actionneurs avec arduino : ventilateur automatique à deux vitesses selon la température**

sécurité : le ventilateur est connecté avec des cordons non sécurisés. Ne pas utiliser ces cordons pour d'autres manipulations que celles prévues dans ce TP

## Matériel nécessaire :

- Une platine arduino UNO reliée à un PC
- Une platine LABDEC avec connecteurs latéraux
- Un ventilateur de PC
- Une diode de roue libre 1N4007
- une DEL et sa résistance de protection 470 ohms
- Un capteur de température thermistance CTN associé à une résistance 1 kohms (pont diviseur)
- Un résistor de 4700 ohms et un résistor de 100 kilohms
- Un résistor marqué (30 ohms, 1W)
- Un transistor MOSFET à canal N (référence STP16NF06)
- Deux alimentations stabilisées

## Partie 1 : mise en oeuvre d'un capteur de température : captation et affichage de la température

Rappeler ce qu'est un capteur :

.....  
.....  
.....  
.....

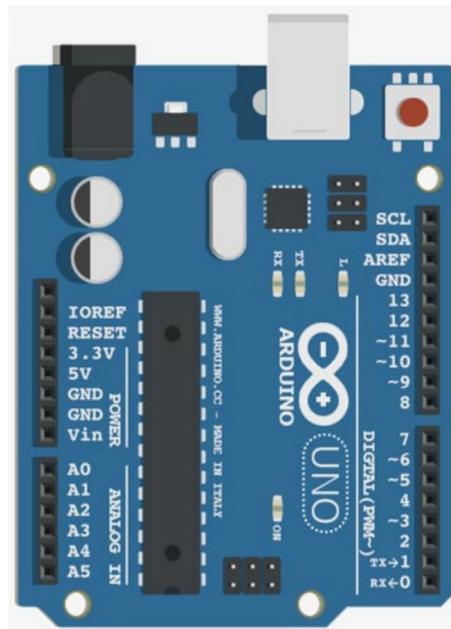
### **Utilisation de la thermistance CTN :**

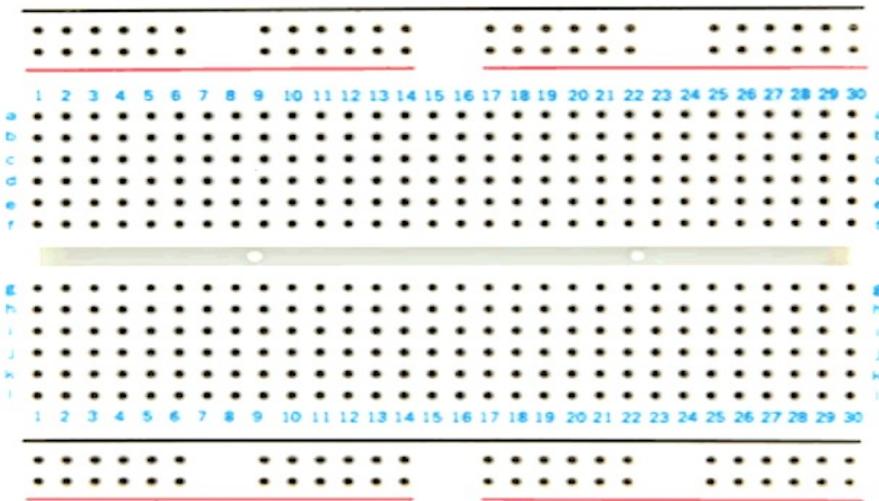
Le tracé dans un tableau de la caractéristique de la thermistance CTN a donné le résultat suivant :

$$\ln R(\text{ohms}) = 1 / T(\text{kelvins}) * 1629 + 1,440$$

En vous inspirant de la séance concernant la fabrication du luxmètre avec utilisation d'un pont diviseur de tension, écrivez le sketch arduino permettant d'utiliser la thermistance CTN.

Indiquer maintenant les connections entre les différents éléments ci-dessous en ajoutant la CTN et le résistor de 1 kohm :



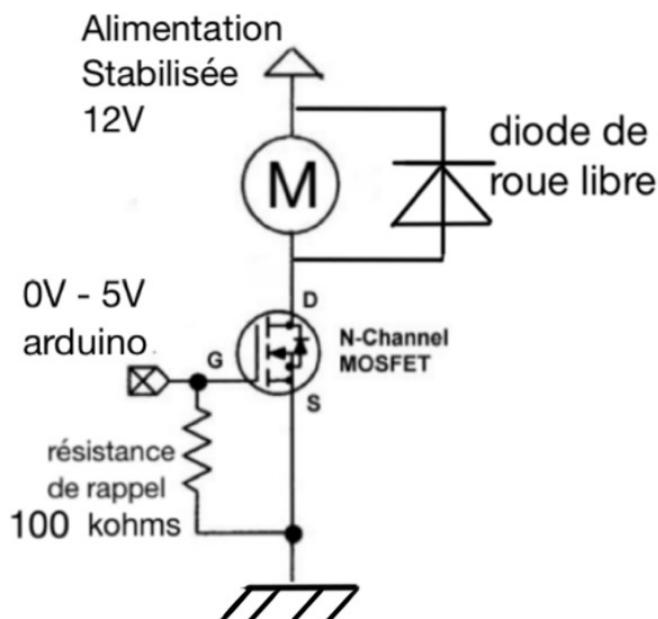


Faire le montage et vérifier le bon fonctionnement. *Ne pas défaire le montage, il sera complété par le montage qui suit dans la partie 2*

#### Partie 2 : mise en oeuvre de la variation de vitesse d'un moteur à courant continu

Objectif : comprendre et utiliser un sketch arduino (donné) qui permet de régler la vitesse d'un moteur de ventilateur de PC en saisissant le rapport cyclique. Le moteur du ventilateur peut être vu comme un actionneur.

Le moteur du ventilateur est un moteur à courant continu piloté en PWM par la carte Arduino. On rappelle que la vitesse d'un tel moteur est proportionnelle à la valeur moyenne de la tension à ses bornes et donc au rapport cyclique.



Le transistor a 3 électrodes :  
G la grille  
D le drain  
S la source

On ne peut brancher le moteur directement sur la carte Arduino car le courant de sortie de la carte est insuffisant (20 mA) par rapport au courant appelé par le moteur (200 mA), ce qui pourrait détruire la sortie de la carte.

On rappelle que pour un moteur à courant continu, le courant est proportionnel au couple moteur. On alimente donc le moteur avec une source de tension de 12V en intercalant un transistor à effet de champ MOSFET qui se comporte entre le drain D et la source S comme un interrupteur commandé en tension par la tension de commande  $v_{GS}$  entre grille et source issue de la carte Arduino (0V ou 5V).

Le MOSFET utilisé est le STP16NF06. Parmi les caractéristiques de bases du STP16NF06, nous relèverons :

**Vds : 60 Volts** : tension maximale supportée par le transistor entre le drain et la source

**Id : 16A** : courant maximum supporté par le transistor.

Si  $v_{GS} = 5V$ , le transistor est équivalent à un interrupteur fermé entre D et S

Si  $v_{GS} = 0V$ , le transistor est équivalent à un interrupteur ouvert entre D et S

La diode de roue libre sert à protéger le transistor des surtensions à l'ouverture liées à la nature inductive du moteur.

Une résistance de rappel de 100 kohms est nécessaire pour le fonctionnement dans le cas où on déconnecte la carte arduino. Dans ce cas de figure, la grille (isolée côté transistor) est flottante et se comporte comme une antenne susceptible de capter une tension pouvant provoquer une mise en route intempestive du moteur. La résistance résout le problème en faisant le lien avec la masse pour fixer le potentiel à zéro volt (moteur immobile).

### Les broches de la PWM

Sur votre carte Arduino, vous disposez de 6 broches qui sont compatibles avec la génération d'une tension PWM. Elles sont repérées par le symbole tilde ~ . Voici les broches générant une PWM : 3, 5, 6, 9, 10 et 11.

### La fréquence de la PWM

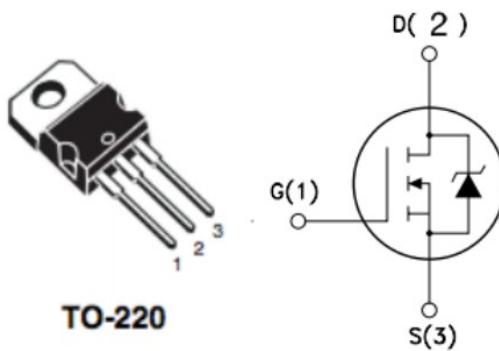
Cette fréquence est fixe, elle ne varie pas au cours du temps. Pour votre carte Arduino elle vaut 490Hz.

### La fonction analogWrite()

La carte Arduino intègre une fonction spécifique pour utiliser la PWM. En fait, les commandes données par le microcontrôleur sont les paramètres passés dans la fonction générant la PWM, c'est-à-dire la fonction : analogWrite() . Elle prend deux arguments :

- Le premier est le numéro de la broche où l'on veut générer la PWM
- Le second argument représente la valeur du rapport cyclique à appliquer. On n'exprime pas cette valeur en pourcentage, mais avec un nombre entier compris entre 0 et 255 (sur 8 bits). Ainsi, pour un rapport cyclique de 0% nous enverrons la valeur 0, pour un rapport de 50%, nous enverrons 128 et pour 100% ce sera 255.

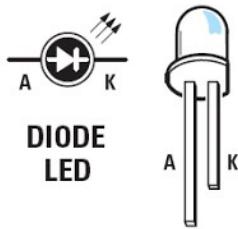
Le câblage du boîtier du transistor est le suivant :



Ajouter une Diode qui s'allume pour signaler l'état du ventilateur :

- Rapport cyclique 50% : diode éteinte
- Rapport cyclique 100% : diode allumée

La DEL témoignant du rapport cyclique de 100% sera connectée entre la broche 7 (côté anode) et la masse avec une résistance en série de 470 ohms.



Le code est donné et doit être compris :

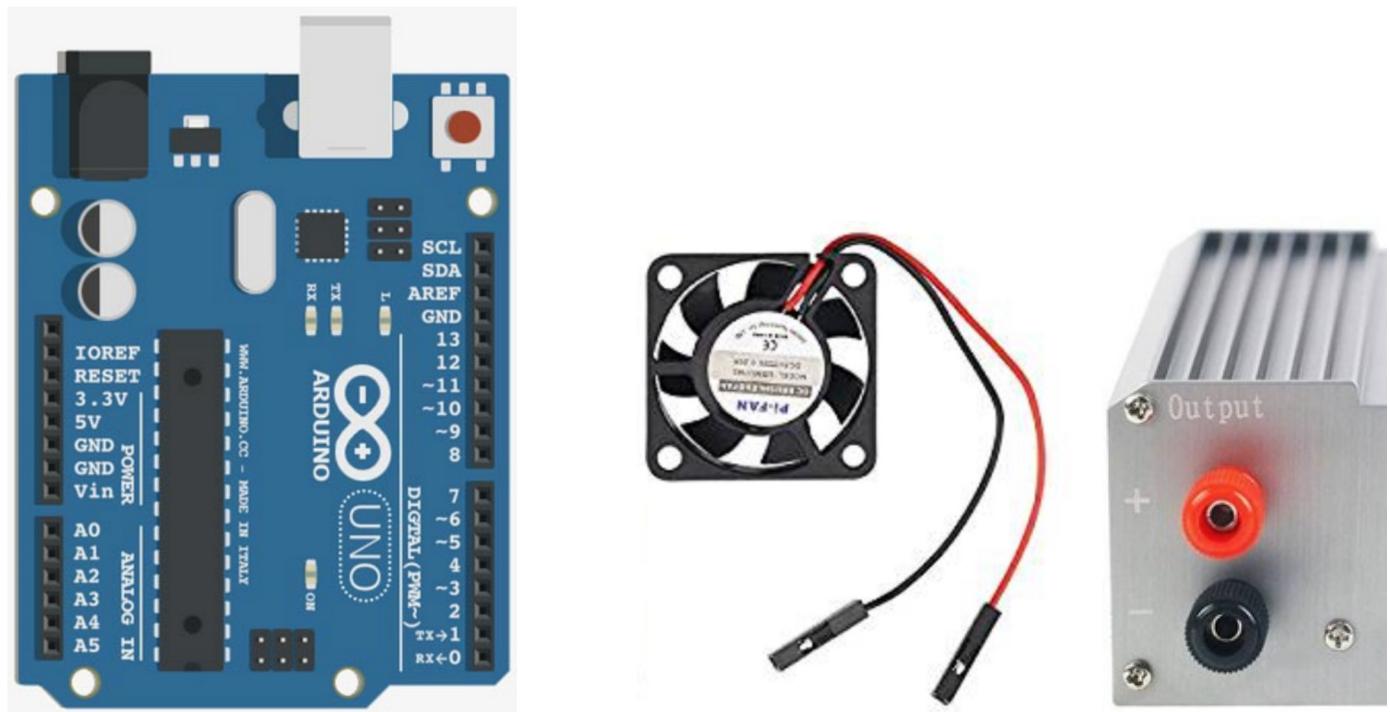
```
*****  
// Pilotage du moteur du ventilateur en PWM  
*****  
// sortie PWM sur la broche 9  
const int sortiePWM = 9;  
  
void setup()  
{  
    pinMode(sortiePWM, OUTPUT); // pour piloter le MOSFET du ventilateur  
    pinMode(7, OUTPUT) ; // pour la diode servant de témoin de la vitesse de rotation du ventilateur  
}  
  
void loop()  
{  
  
    analogWrite(sortiePWM, 0); // on met la vitesse nulle  
    digitalWrite(7, LOW) ; // diode éteinte  
    delay(5000); // pause de 5 secondes  
  
    analogWrite(sortiePWM, 128); // on met un rapport cyclique de 128/255 = 50 %  
    digitalWrite(7, LOW); // diode éteinte  
    delay(5000); // pause de 5 secondes  
  
    analogWrite(sortiePWM, 255); // on met un rapport cyclique de 255/255 = 100 %  
    digitalWrite(7, HIGH) ; // diode allumée  
    delay(5000); // pause de 5 secondes  
}
```

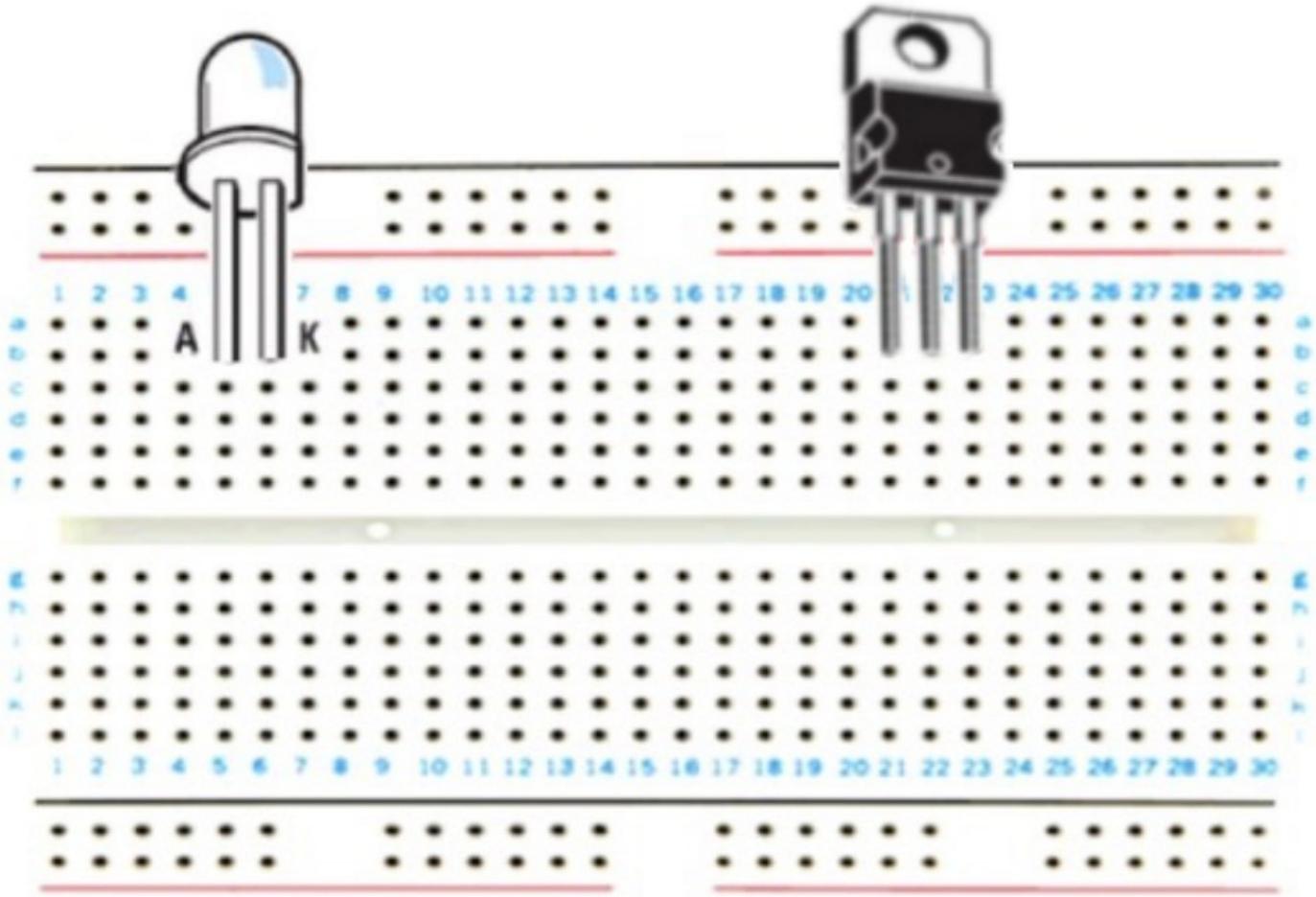
Indiquer ci-après les connections entre les différents éléments ci-dessous en ajoutant :

- les résistors de 100 kohms et 470 ohms
- la diode DRL

et en veillant à fixer :

- une ligne horizontale comme ligne de masse sur la platine labdec
- une ligne horizontale pour la tension 5V
- une ligne horizontale pour la tension 12V





Réaliser le montage (Ne pas oublier de connecter à nouveau la masse de la carte arduino pour la sortie PWM) et vérifier le bon fonctionnement du sketch arduino donné.

### Partie 3 : mise en oeuvre du ventilateur automatique

Soit une température seuil valant la température ambiante + 2°C.

L'objectif est de baser sur les deux sketchs arduino précédents pour écrire un sketch de façon à ce que le ventilateur soit piloté en vitesse avec un rapport cyclique de 50% si la température est inférieure à la température seuil , et 100% sinon.

Ecrire ce code, veiller à la mise en forme avec indentation (sélectionner tout le code puis faire bouton de droite et formatage automatique).

Veiller à bien orienter le ventilateur dans le sens où le capteur de température est exposé au flux d'air.

Pour modifier la température au-delà de la température seuil, il est possible de créer une source de chaleur de deux façons :

- soit avec le doigt
- soit on peut placer au voisinage du capteur un résistor marqué (30 ohms, 1W) branchée à une alimentation stabilisée pouvant fournir 2,5A. Ce résistor est déjà brûlant au doigt sous 4V (attention à ne pas vous blesser). On ne dépassera pas 5V aux bornes de ce résistor.

Calculer la tension maximale  $U_{max}$  et le courant maximal  $I_{max}$  si  $R = 30$  ohms et  $P_{max} = 1W$ .

.....  
.....  
.....  
.....  
.....

Mettre en oeuvre le montage et vérifier le bon fonctionnement du sketch arduino. Contrôler le rapport cyclique à l'oscilloscope en visualisant la tension  $v_{GS}(t)$ .