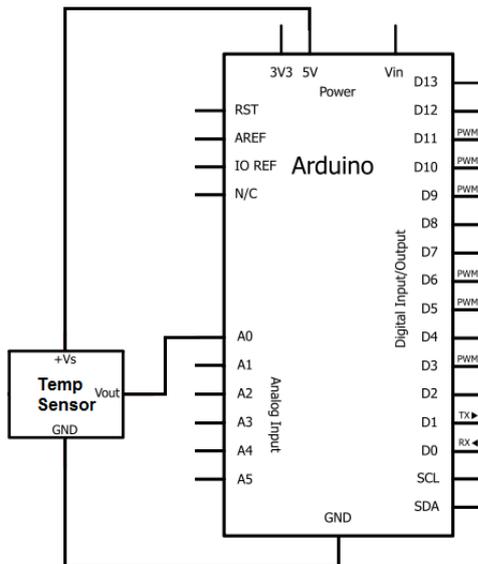
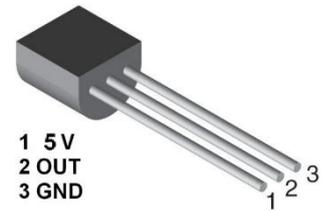


« L'informatique n'est pas plus la science des ordinateurs que l'astronomie n'est celle des télescopes. »  
Michael R. Fellows (1991) "Computer SCIENCE and Mathematics in the Elementary Schools"

- 1- Les capteurs de température sont des dispositifs permettant de transformer l'effet du réchauffement ou du refroidissement sur leurs composants en signal électrique. Le composant illustré ci-contre, une fois alimenté, va délivrer une certaine tension via la patte 2. Si vous maintenez le composant entre vos doigts alors vous augmentez la température de la sonde donc vous modifier la tension délivrée à la patte 2. Afin de vérifier le bon fonctionnement du composant, réalisez le montage suivant et le programme associé.



```
import pyfirmata
import time
port = 'COM3' #windows
board = pyfirmata.Arduino(port)
temperature_pin = board.get_pin('a:0:i')
iterator = pyfirmata.util.Iterator(board)
iterator.start()
temperature_pin.enable_reporting()
while temperature_pin.read() == None: None
try:
    while True:
        print (temperature_pin.read())
        time.sleep(1)
except:
    temperature_pin.disable_reporting()
    board.exit()
```

L'affichage correspond à une valeur comprise entre 0 et 1. Nous pouvons approximer la température en degré Celsius avec la formule suivant :  $(\text{value} * 5 - 0.5) * 100$

En vous inspirant des projets précédents, mettez dans un tableau 180 valeurs, une par seconde, soit 3 minutes de prise de mesures, et affichez le résultat avec la bibliothèque « matplotlib ».

2 – La courbe dessinée est irrégulière localement mais varie globalement assez peu en fonction l'application de vos doigts. L'irrégularité est due à du bruit. Une manière d'atténuer ce bruit consiste pour chaque valeur à calculer la moyenne avec les valeurs précédentes, autrement-dit appliquer une moyenne glissante.

La moyenne glissante est un type de moyenne statistique utilisée pour analyser des séries temporelles, en supprimant les fluctuations transitoires de façon à en souligner les tendances à plus long terme, soit la formule suivante :

$$\bar{x}_n = \frac{1}{N} \sum_{k=0}^{N-1} x_{n-k}$$

Soit l'algorithme correspondant :

```
1 fonction Moyenne_Glissante ( T : Tableau_Réels, N : Entier ) : Tableau_Réels
2 variables
3 S : Entier
4 R : Tableau_Entiers
5 début
6 pour i allant de N à longueur(T) faire
7 S ← 0
8 pour k allant de i - N + 1 à i faire
9 S ← S + T[k]
```

```

10         fin pour
11         R[i] ← S / N
12     fin pour
13     pour i allant de 1 à N-1 faire
14         R[i] ← T[i]
15     fin pour
16     retourne R
17 fin

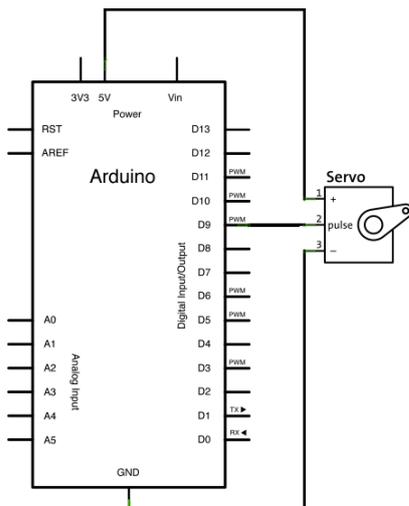
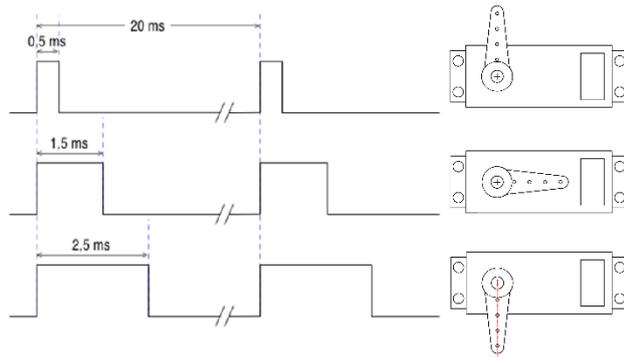
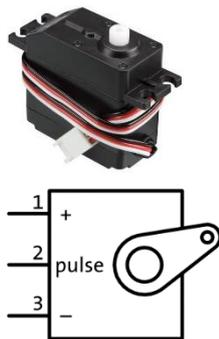
```

Traduisez cet algorithme en langage Python, testez votre programme sur les données de températures et visualisez la différence entre les deux courbes.

3- Réalisez un autre programme effectuant également une moyenne glissante basée sur la formule suivante puis testez votre programme :

$$\bar{x}_n = \bar{x}_{n-1} + \frac{x_n - x_{n-N}}{N}$$

4- Un servomoteur (provenant du latin *servus* qui signifie « esclave ») est un moteur capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure. Le servomoteur intègre dans un même boîtier, la mécanique (moteur et engrenage), et l'électronique, pour la commande et l'asservissement du moteur. Ici, la position est définie avec une limite de débattement d'angle de 180 degrés. La consigne envoyée au servomoteur n'est autre qu'un signal électronique de type PWM. pyFirmata se charge de l'émission de la commande mais ci-dessous une illustration du procédé (Attention : le fil rouge pour l'alimentation, le fil noir pour la masse et le fil blanc pour la commande) :



```

import pyfirmata
board = pyfirmata.Arduino('COM29') #windows
servo_pin = board.get_pin('d:11:s')
try :
    while True:
        angle = int(input("Entrez un angle entre 10 et 170:"))
        if angle < 10 :
            angle = 10
        elif angle > 170 :
            angle = 170
        servo_pin.write(angle)
except :
    board.exit()

```

5 – Maintenant nous souhaitons contrôler le servomoteur à l'aide d'une photorésistance en fonction de son obturation. Il faut mesurer une estimation des extrêmes en condition normale, puis projeter les valeurs dans l'espace de commande entre 10 et 170. Afin de protéger le servomoteur, les valeurs dépassant exceptionnellement les extrêmes mesurés doivent être bornées entre 10 et 170. Utilisez une fenêtre glissante pour lisser la commande. Pour le montage n'oubliez de mettre la résistance de 10kΩ pour tirer le potentiel à la masse comme sur les montages de la séance précédente.