Chapitre N°8.) Le capteur à ultrason pour mesurer

I. Aspects théoriques

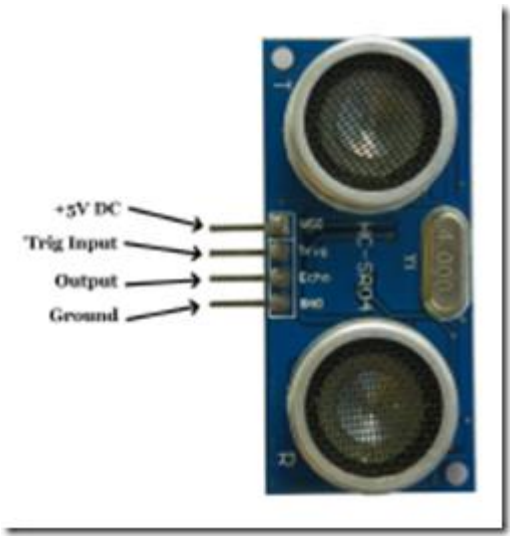
1. Caractéristique du capteur

(D'après :http://itechnofrance.wordpress.com/2013/03/12/utilisation-du-module-ultrason-hc-sr04-avec-larduino/)

Les caractéristiques techniques du module sont les suivantes :

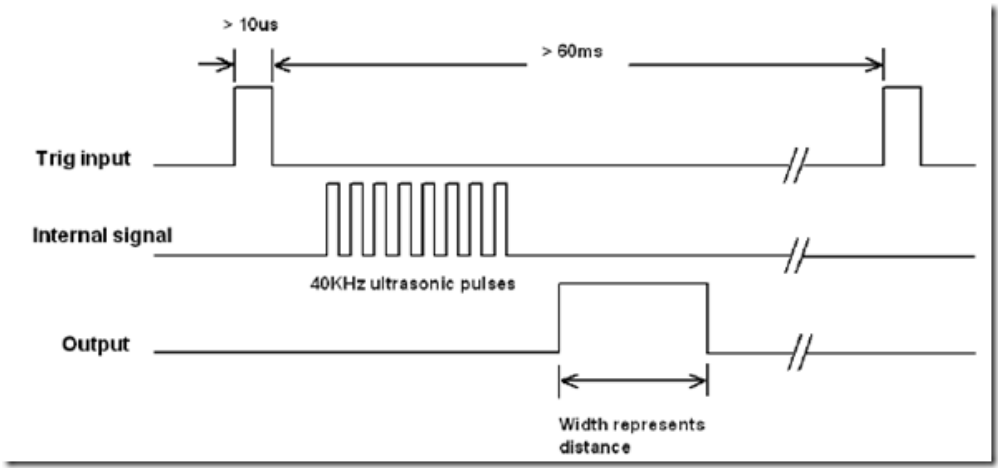
- Alimentation : 5v.
- Consommation en utilisation : 15 mA.
- Gamme de distance : 2 cm à 5 m.
- Résolution : 0.3 cm.
- Angle de mesure : < 15°.

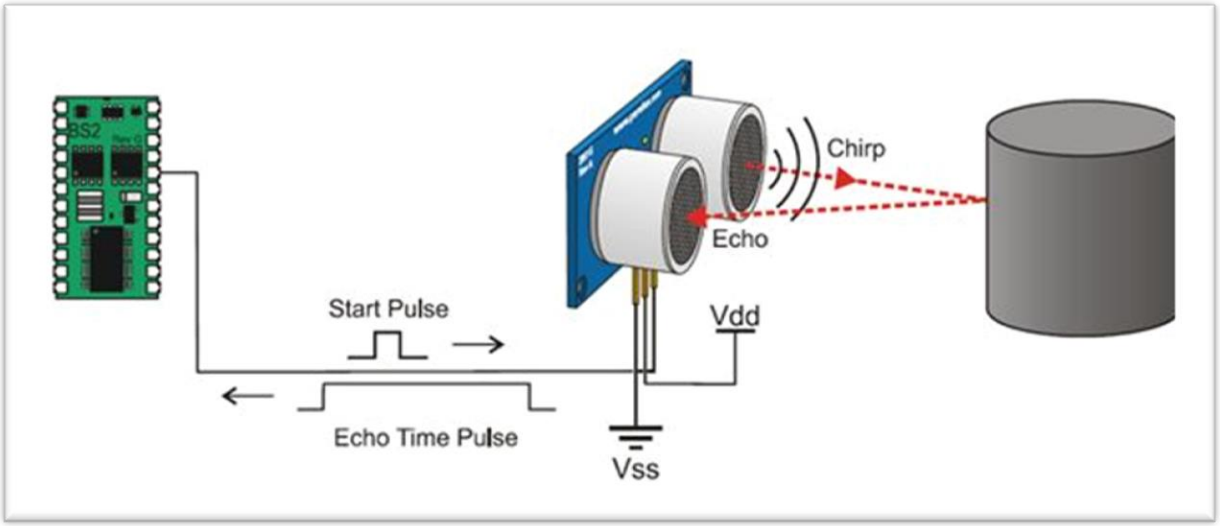
Le brochage du module est le suivant :



Le fonctionnement du module est le suivant :

Il faut envoyer une impulsion niveau haut (à + 5v) pendant au moins 10 µs sur la broche 'Trig Input'; cela déclenche la mesure. En retour la sortie 'Output' ou 'Echo', va fournir une impulsion + 5v dont la durée est proportionnelle à la distance si le module détecte un objet.
Voici une représentation graphique de la séquence de fonctionnement du module :





http://csharpcorner.mindcrackerinc.netdna-cdn.com/UploadFile/167ad2/how-to-use-ultrasonic-sensor-hc-sr04-in-arduino/Images/HC-SR04.jpg

2. Principe du calcul

http://tiptopboards.com/69-module-%C3%A0-ultrasons-hc-sr04-capteur.html

Envoyez un signal de déclenchement IO qui fournit un signal d'au moins 10us sur le niveau haut.
Le module va alors automatiquement envoyer huit impulsions d'un signal ultra-sonore de fréquence 40 kHz de forme carrée.
Il va détecter automatiquement le signal de retour de ces impulsions en écho.
S'il y a un signal de retour en écho, la sortie passe au niveau élevé, et son temps de maintient (en s) correspond au délai de retard de réception de l'écho (modulation de largeur d'impulsion).

3. Allez un peu de math, ne me remerciez pas : calcul de la distance de l'obstacle

Soit ***T*** le temps que met le signal pour revenir au capteur : temps aller-retour de la sortie du capteur pour y revenir après avoir été réfléchi par l'obstacle.
Sachant que le son se propage à la vitesse C = 340m/s.

En déduire la distance (en mètre) entre le capteur et l'objet en fonction de T :

En déduire la distance (en cm) :

I. Mesurer les distance capteur à ultrasons

Ce capteur permet d'évaluer les distances entre un objet mobile et les obstacles rencontrés.

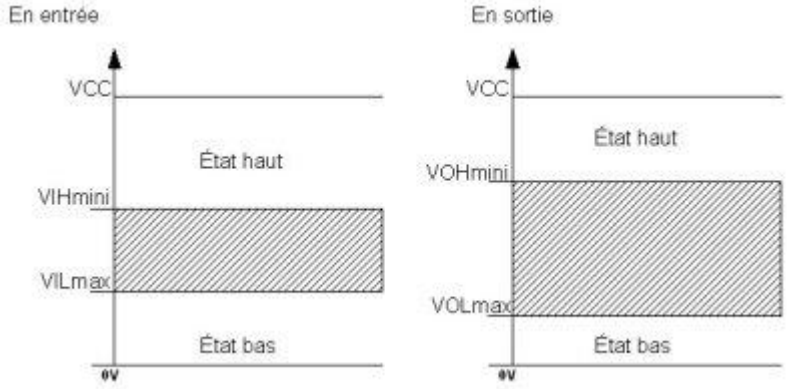
1. Montage

Ce capteur a quatre broches (la masse, ECHO, TRIG et VCC(c.f. FR)) et s'alimente en cinq volts. Le RPi émet un signal sur TRIG, et la réponse se fait par la broche ECHO MAIS en 5 volts.

C'est pourquoi il est indispensable de baisser cette tension sous 3.3V sous peine de détruire la raspberry.

2. Rappel TTL

Les entrées et les sorties des fonctions logiques ont des marges pour qualifier un état logique :



(VCC = tension d'alimentation)

Technologie TTL : (VCC = 5V)

$$VIHmini = 2V$$
$$VILmax = 0.8V$$
$$VOHmini = 2.4V$$
$$VOLmax = 0.4V$$

Technologie CMOS : (dépend de VCC)

$$VIHmini = 0.55 \times VCC$$
$$VILmax = 0.45 \times VCC$$
$$VOHmini = 0.95 \times VCC$$
$$VOLmax = 0.05 \times VCC$$

La solution la plus simple est d'utiliser un pont diviseur
Pour faire simple : TTL est un « vieux » standard tandis que le 3.3 V CMOS utilisé par le rapberry pi est plus récent.

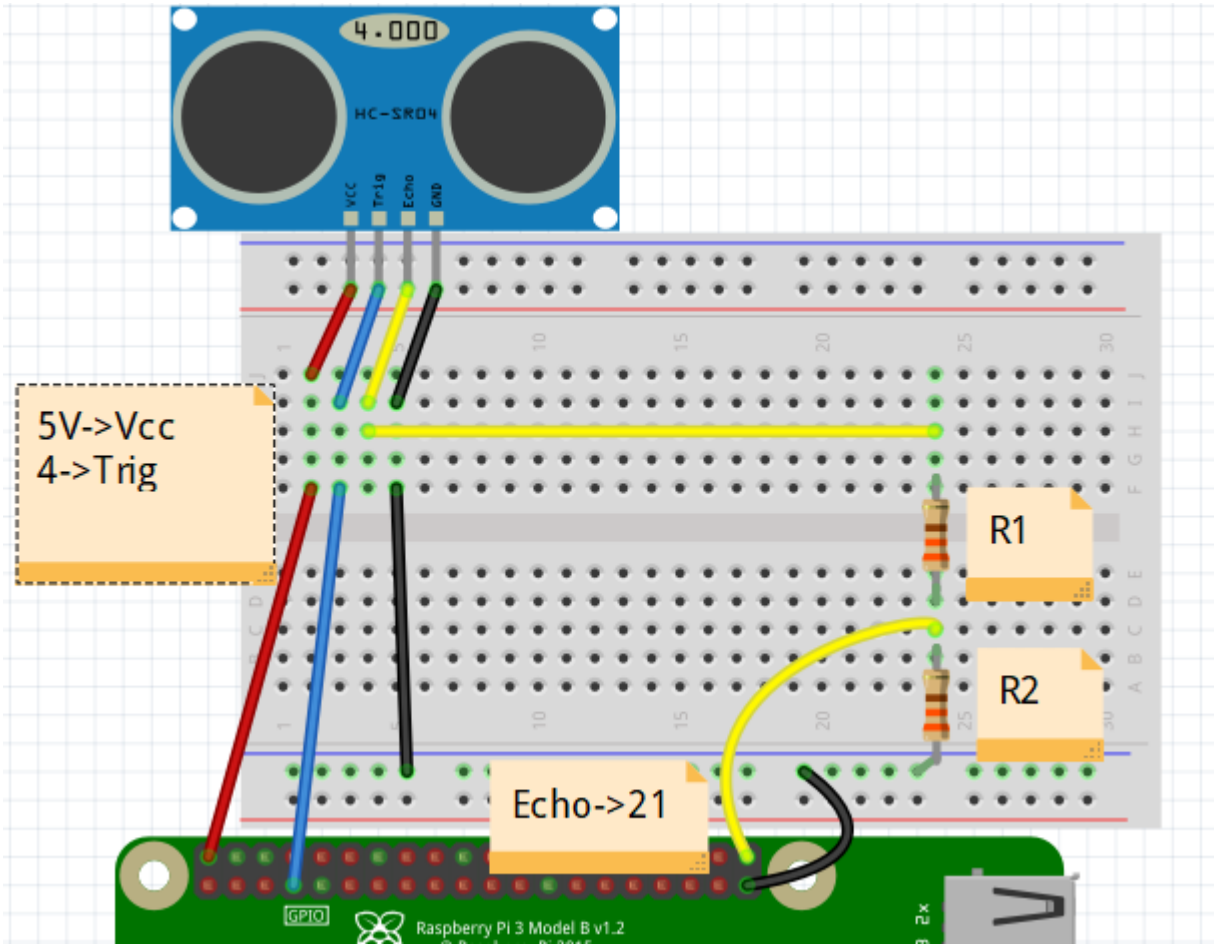
Complétez les tableaux suivants :

	TTL 5 Volts en Sortie	Raspberry 3.3 Volts en Entrée
Minimum pour avoir un '0'	$V_{OutLowMiniTTL} = 0\ Volt$	$V_{InLowMini3.3V} = 0$ Volt
Maxi pour un '0'	$V_{OutLowMaxiTTL} = 0.4$ V	$V_{InLowMaxi3.3V} = 1.485$ V
Mini pour un '1'	$V_{OutHighMiniTTL} =$	$V_{InHighMini3.3V} =$
Maxi pour un '1'	$V_{OutHighMaxiTTL} =$	$V_{InHighMaxi3.3V} =$

	Raspberry 3.3 Volts en Sortie	TTL 5 Volts en Entrée
Minimum pour avoir un '0'	$V_{OutLowMin3.3V} =$	$V_{InLowMiniTTL} =$
Maxi pour un '0'	$V_{OutLowMaxi3.3V} =$	$V_{InLowMaxiTTL} =$
Mini pour un '1'	$V_{OutHighMini3.3V} =$	$V_{InHighMiniTTL} =$
Maxi pour un '1'	$V_{OutHighMaxi3.3V} =$	$V_{InHighMaxiTTL} =$

Quels sont les problèmes ?

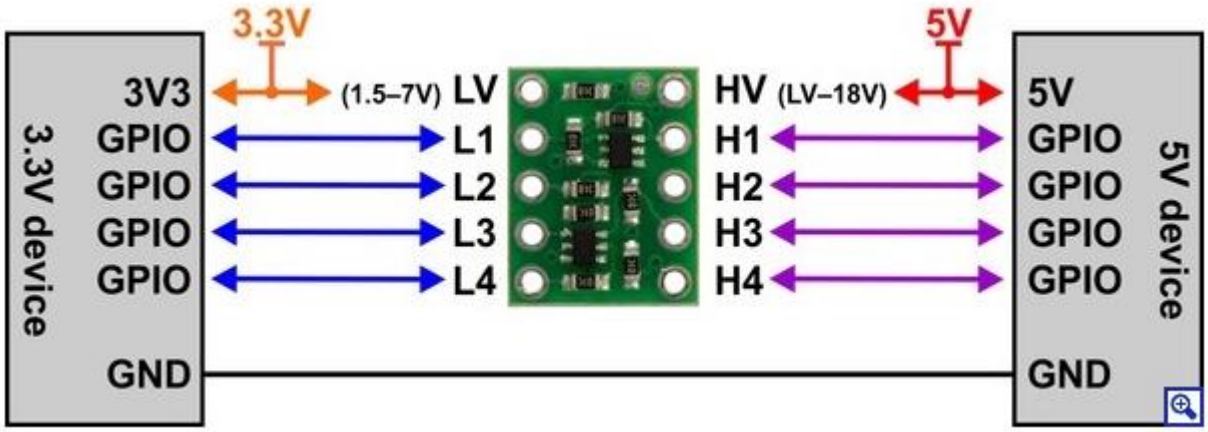
3. Montage



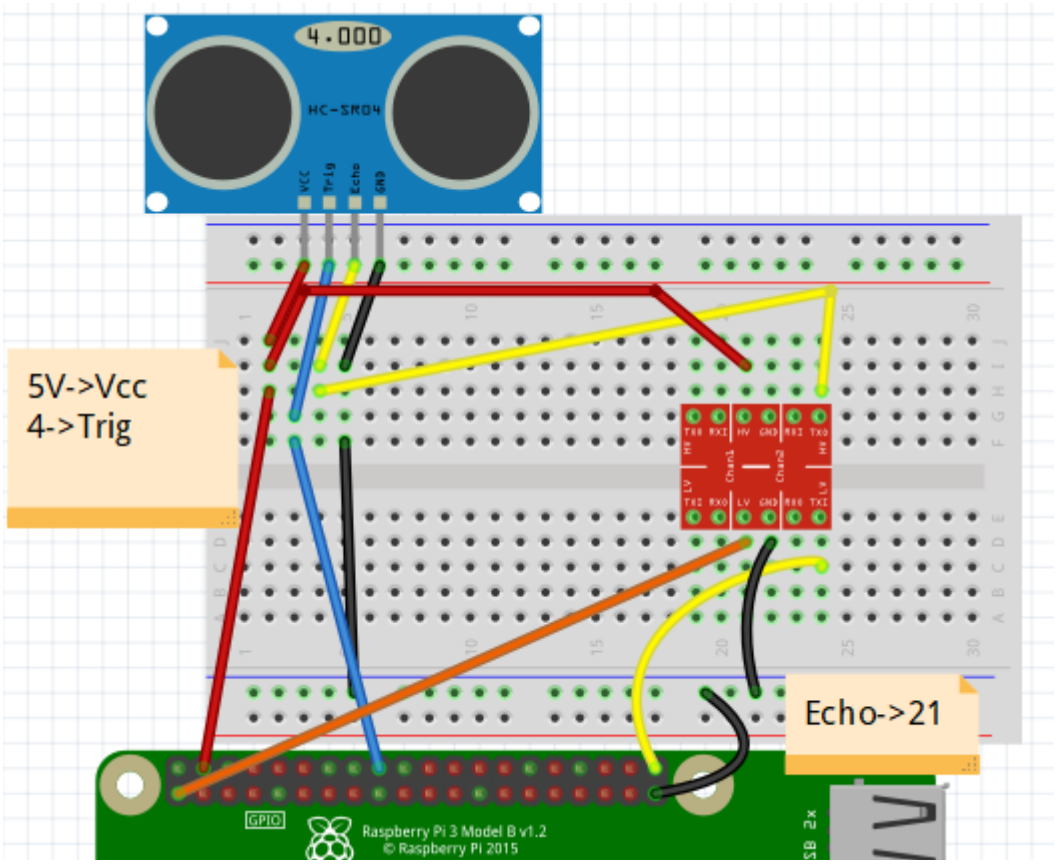
- Résumez par un scéma un pont diviseur de tension : sur papier libre, dessinez un le pont diviseur de tension : avec Vecho tension de la broche Echo du capteur, V21 la tension de la broche GPIO21 et les résistances R1 et R2.
- Rappelez la formule d'un pont diviseur
- Calculez maintenant R2 si R1 vaut 330Ω.
- Quelle résistance R2 avez-vous trouvé ? Recalculez la nouvelle tension
- Vérifiez vos calculs à l'aide de http://www.ohmslawcalculator.com/voltage-divider-calculator
- Refaites le calcul avec une résistance R1 de 1k Ω. Est-ce préférable de prendre une résistance de 1k Ω ou de 330 Ω
- Appelez le professeur
- Effectuez le montage sans alimentation
- Faites vérifier par le professeur

4. Autre montage – extension –

Au lieu d'utiliser des résistances, il existe des convertisseurs de tension (Level Shifter en anglais).



Et le schéma :



Attention : vérifier le sens des signaux en effectuant des mesures de tension <u>avant</u> de le brancher sur la Rpi.

5. Programmation

- La classe s'appelle DistanceSensor
- Exemple de déclaration d'un objet de type DistanceSensor
  - `HC_SR04 = DistanceSensor(echo=18, trigger=17)`
- `HC_SR04.distance` renvoie la distance en mètre

Exemple

```
from gpiozero import DistanceSensor
sensor = DistanceSensor(echo=18, trigger=17)
print('Distance: ', sensor.distance * 100)
```

- Ecrivez un programme qui affiche toutes les secondes la distance mesurée par le capteur
- Vérifiez la mesure à l'aide d'une règle. Il y-a-t-il une différence entre le réel et le simulé ? Quelles sont les raisons de cette différence ?
- Appelez le professeur

D'après la documentation *gpiozero* :

max_distance

The maximum distance that the sensor will measure in meters. This value is specified in the constructor and is used to provide the scaling for the `value` attribute. When `distance` is equal to `max_distance`, `value` will be 1.

Et d'après la documentation du capteur HC-SR4 :

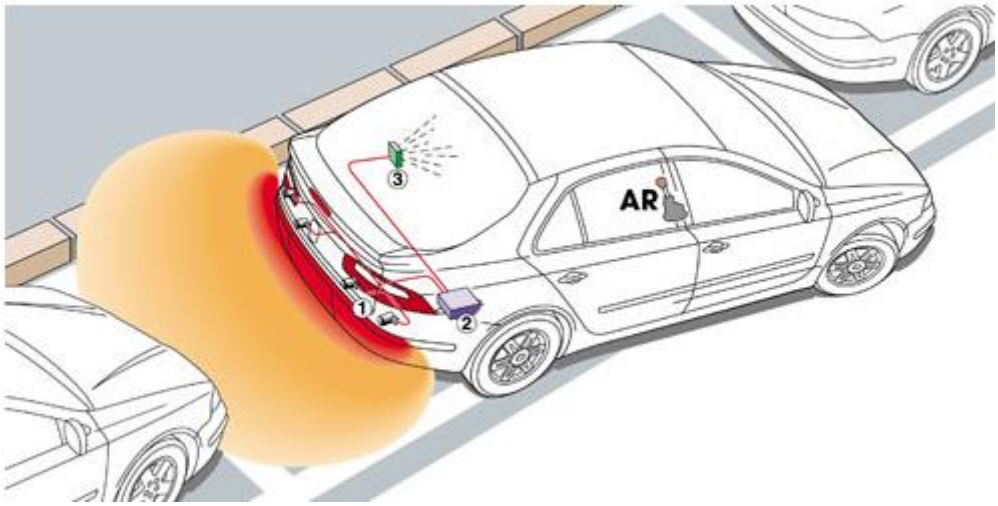
Tension d'alimentation	5V DC
Courant d'alimentation	15mA
Fréquence de travail	40Hz
Distance maximale de détection	4m
Distance minimale de détection	2cm
Angle de détection	15 degrés
Signal d'entrée de l'émetteur	Impulsion à l'état haut de 10µs
Signal de sortie du récepteur	Signal numérique à l'état haut et la distance proportionnellement
Dimension	45*20*15mm

Modifiez le constructeur de l'objet en mettant max_distance à bonne distance. Les mesures sont-elles plus proches de la réalité ?

Le constructeur de l'objet c'est DistanceSensor(echo=21, trigger=4,max_distance=4)

Testez le code en annexe et voyez si comparez les résultats

6. L'aide au stationnement



Source http://vendre.autobiz.fr/blog/dictionnaire/aide-au-stationnement/

7. Qu'est ce que c'est ?

Le **système d'aide au stationnement** informe en continu le conducteur de la **distance restant entre le pare-chocs et l'éventuel obstacle** pendant la manœuvre.

8. Comment ça fonctionne ?

La distance est mesurée par 4 ou 6 **capteurs ultrasons** placés sur la largeur du pare-chocs. Le système émet un « BIP » lorsqu'un objet est à moins de 1,20 mètre et sa fréquence sonore augmente au fur et à mesure que l'objet se rapproche : par exemple trois paliers rien au dessus de 20 cm, 20-10, 10-5 et en-dessous de 3 centimètres, le « BIP » est continu.

9. Système à réaliser

En utilisant le buzzer simuler un tel système à l'aide d'un capteur ultrason (faire une boucle et utiliser la méthode *beep* avec notamment comme paramètre n=1) ; Mettez un BP pour simuler la marche arrière

II. Annexe

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
print( "+-----------------------------------------------------------+")
print ("|   Mesure de distance par le capteur ultrasonore HC-SR04   |")
print ("+-----------------------------------------------------------+")

Trig = 4            # Entree Trig du HC-SR04 branchee au GPIO 23
Echo = 21           # Sortie Echo du HC-SR04 branchee au GPIO 24
debutImpulsion =0
GPIO.setup(Trig,GPIO.OUT)
GPIO.setup(Echo,GPIO.IN)

GPIO.output(Trig, False)

try :
   while True:
      time.sleep(1)        # On la prend toute les 1 seconde
      GPIO.output(Trig, True)
      time.sleep(1)
      GPIO.output(Trig, False)

      while GPIO.input(Echo)==0:  ## Emission de l'ultrason
        debutImpulsion = time.time()

      while GPIO.input(Echo)==1:   ## Retour de l'Echo
        finImpulsion = time.time()

      distance = round((finImpulsion - debutImpulsion) * 340 * 100 / 2, 1)  ## Vitesse
du son = 340 m/s

      print( "La distance est de : ",distance," cm")

except :
   GPIO.cleanup()
```

Chapitre N°9.) Détecter les mouvements