

Guide d'utilisation des capteurs de distance à ultrasons HC-SR04 avec une carte Raspberry Pi.



Introduction :

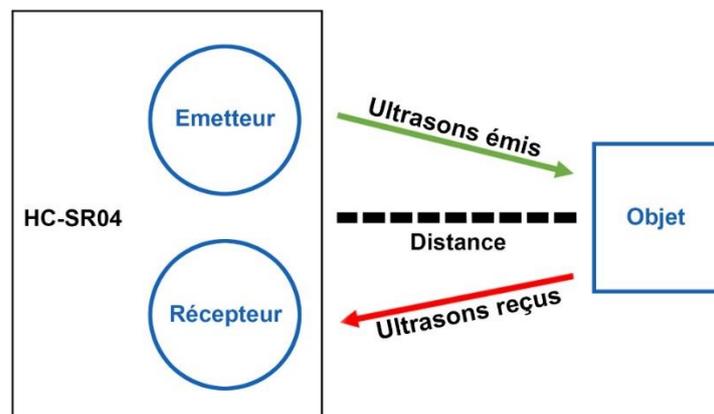
Les différents télémètres HC-SR04, proposés sur le site Gotronic.fr, utilisent les ultrasons pour mesurer la distance à laquelle se trouve un objet. Ces capteurs offrent de très bonnes plages de mesures et une bonne réactivité.

Dans cet exemple, les distances mesurées pourront être visualisées grâce à une carte Raspberry Pi et un programme Python.

Fonctionnement :

Un signal de 10 μ S est envoyé sur la broche **Trigger** pour créer des ultrasons à 40 kHz via la capsule émettrice du module. Ce signal est généré via une sortie digitale de la carte Raspberry Pi.

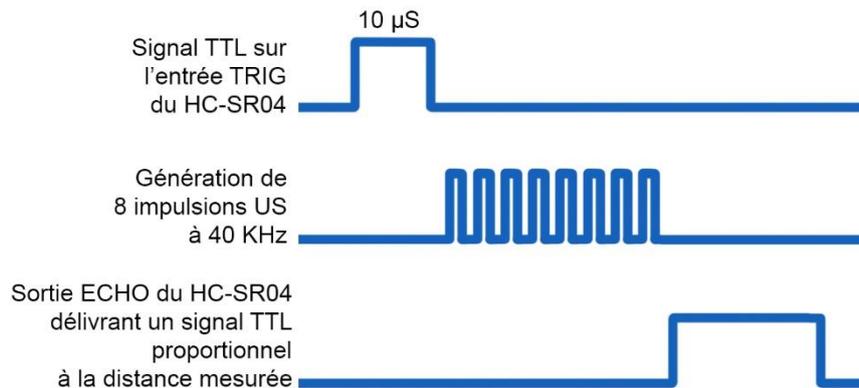
La seconde capsule, installée sur le module, permet de réceptionner les ultrasons retournés par l'objet. Cette information est envoyée vers une E/S digitale de la carte microcontrôleur via la broche **Echo**.



Remarque : Ce capteur délivre un signal de 5 Vcc sur la broche ECHO. Les broches de la carte Raspberry sont incompatibles avec ce signal. Ce dernier pourrait endommager la carte Raspberry Pi. La Raspberry Pi est seulement compatible avec un niveau logique haut de 3,3 Vcc.

Cette restriction impose l'utilisation d'un pont diviseur de tension composé de deux résistances (330 et 470 Ω) raccordées sur la broche ECHO, voir schéma plus bas.

Signal de sortie sur Trigger :



Matériel requis :

- 1 x capteur à ultrasons [HC-SR04](#)
- 1 x carte Raspberry Pi (Pi 2B, 3B, 3B+, 4B ou Zero)
- 1 x résistance de $330\ \Omega$
- 1 x résistance de $470\ \Omega$
- 1 x jeu de cordons mâles-femelles type [BBJ21](#).

Ces composants et modules sont donnés à titre indicatif et peuvent être remplacés par d'autres modèles équivalents.

Schéma de câblage :

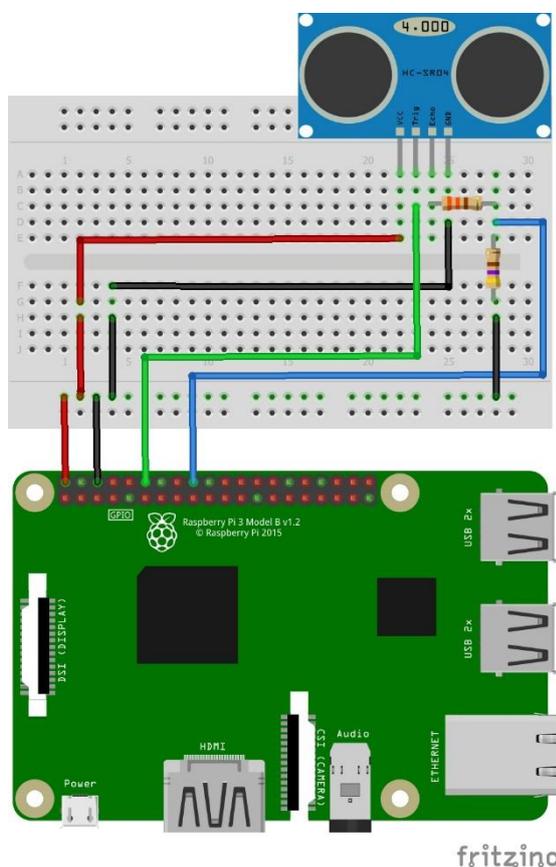


Tableau de correspondance :

Raspberry Pi	HC-SR04
5 V	Vcc
GND	Gnd
GPIO24	Echo*
GPIO18	Trig

* La sortie Echo comporte un diviseur de tension basé sur une résistance de $330\ \Omega$ (orange, orange, marron et or) et une de $470\ \Omega$ (jaune, violet, marron et or).

Ce pont diviseur permet d'atténuer la tension de 5 Vcc en sortie du capteur afin de ne pas endommager l'entrée digitale de la carte Raspberry Pi (uniquement compatible 3,3 Vcc).

Préparation de la carte Raspberry Pi :

Il est recommandé d'exécuter la dernière version de Raspberry Pi OS avec les dernières mises à jour des différents paquets installés.

Téléchargement Raspberry Pi OS.

La mise à jour des paquets s'effectue via quelques lignes de commandes à exécuter dans un terminal (LXTerminal par défaut sous Pi OS) :



Mise à jour des informations des paquets disponibles pour votre distribution :

```
sudo apt-get update
```

Mise à jour complète de tous les paquets :

```
sudo apt-get dist-update
```

```

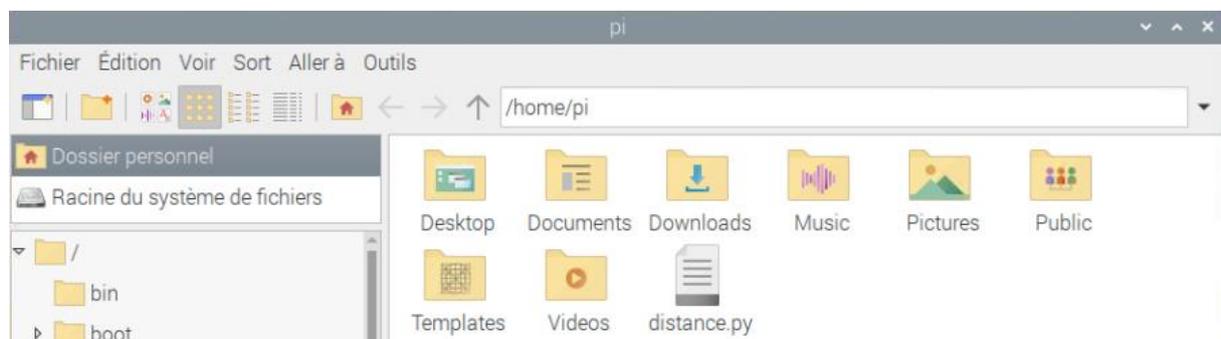
pi@raspberrypi: ~
Fichier  Édition  Onglets  Aide
pi@raspberrypi:~ $ sudo apt-get update
Réception de :1 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0
kB]
Atteint :2 http://archive.raspberrypi.org/debian buster InRelease
15,0 ko réceptionnés en 1s (11,6 ko/s)
Lecture des listes de paquets... Fait
pi@raspberrypi:~ $ sudo apt-get dist-upgrade

```

Exemple de code :

L'exemple de code Python détaillé et expliqué est à télécharger directement à [cette adresse](#). Vous retrouverez également ce code en [page 5/6](#).

Enregistrez ou déplacez le code dans votre dossier */home/pi*



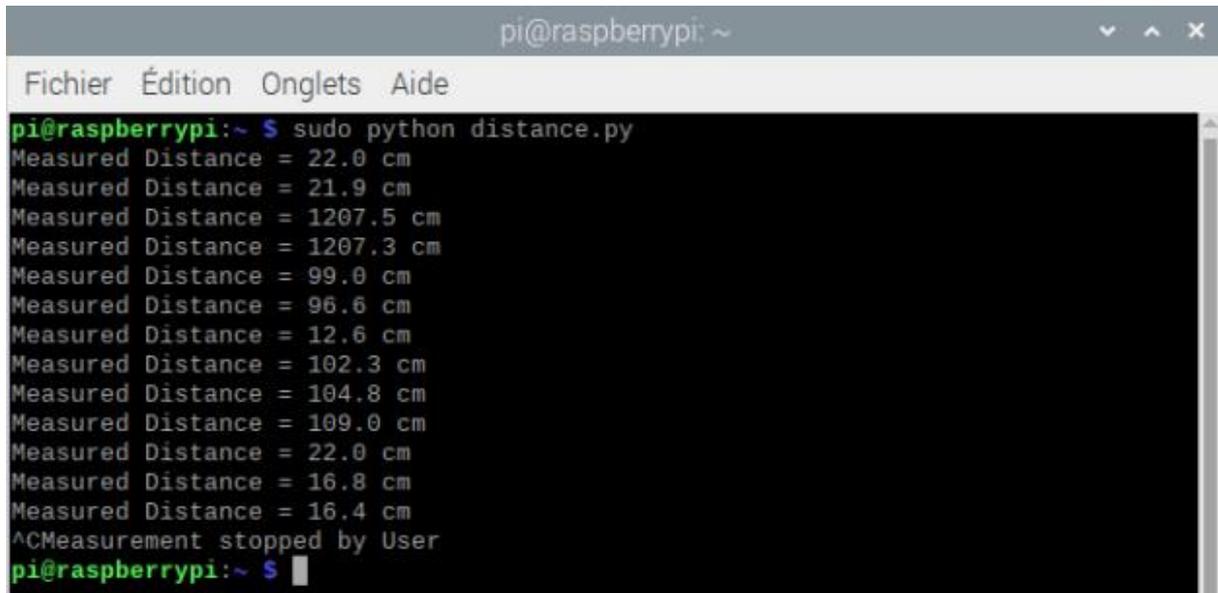
Maintenant vous pouvez exécuter le code dans un terminal :



```
sudo python distance.py
```

Ce programme permet la mesure de la distance entre votre capteur HC-SR04 et un objet, un mur, etc.

Cette distance est affichée directement dans le terminal de Raspberry Pi OS.



```
pi@raspberrypi: ~  
Fichier Édition Onglets Aide  
pi@raspberrypi:~ $ sudo python distance.py  
Measured Distance = 22.0 cm  
Measured Distance = 21.9 cm  
Measured Distance = 1207.5 cm  
Measured Distance = 1207.3 cm  
Measured Distance = 99.0 cm  
Measured Distance = 96.6 cm  
Measured Distance = 12.6 cm  
Measured Distance = 102.3 cm  
Measured Distance = 104.8 cm  
Measured Distance = 109.0 cm  
Measured Distance = 22.0 cm  
Measured Distance = 16.8 cm  
Measured Distance = 16.4 cm  
^CMeasurement stopped by User  
pi@raspberrypi:~ $
```

Pour arrêter la mesure, effectuez CTRL+C.

Exemple de code détaillé et expliqué :

```
# Import des librairies GPIO et time (temps et conversion) #
import RPi.GPIO as GPIO
import time

# Module GPIO: BOARD ou BCM (numérotation comme la sérigraphie de la carte ou comme
le chip) #
GPIO.setmode(GPIO.BCM)

# Définition des broches GPIO #
GPIO_TRIGGER = 18
GPIO_ECHO = 24

# Définition des broches en entrée ou en sortie #
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)

def distance():
    # Mise à l'état haut de la broche Trigger #
    GPIO.output(GPIO_TRIGGER, True)

    # Mise à l'état bas de la broche Trigger après 10 µs #
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)

    StartTime = time.time()
    StopTime = time.time()

    # Enregistrement du temps de départ des ultrasons #
    while GPIO.input(GPIO_ECHO) == 0:
        StartTime = time.time()

    # Enregistrement du temps d'arrivée des ultrasons #
    while GPIO.input(GPIO_ECHO) == 1:
        StopTime = time.time()

    # Calcul de la durée de l'aller-retour des US #
    TimeElapsed = StopTime - StartTime
    # On multiplie la durée par la vitesse du son: 34300 cm/s #
    # Et on divise par deux car il s'agit d'un aller et retour. #
    distance = (TimeElapsed * 34300) / 2

    return distance

if __name__ == '__main__':
    try:
        while True:
            dist = distance()
            print ("Distance mesurée = %.1f cm" % dist)
            time.sleep(1)

        # On reset le programme via CTRL+C #
    except KeyboardInterrupt:
        print("Mesure stoppée")
        GPIO.cleanup()
```

GO TRONIC

ROBOTIQUE ET COMPOSANTS ÉLECTRONIQUES

Si vous rencontrez des problèmes, merci de nous contacter par courriel à :

sav@gotronic.fr