

Electronique des Systèmes EmbarquésTP3

TP3

Manipulation 1 : Contrôle d'un Servomoteur avec Raspberry Pi

Matériel requis :

- Raspberry Pi avec Raspbian OS installé
- Un servomoteur
- Une plaque d'essai (breadboard)
- Câbles de raccordement (mâle-femelle)
- Résistance de 330 ohms (ou similaire)
- Alimentation externe pour le servomoteur (le Raspberry Pi seul ne peut pas fournir suffisamment de puissance)

Objectif : L'objectif de ce TP est de contrôler un servomoteur en utilisant Raspberry Pi et Python. Nous allons créer un programme qui permettra au servomoteur de tourner à différents angles.

Tâches :

1. Connexion du matériel :

- Branchez le servomoteur sur la plaque d'essai (breadboard) en utilisant des câbles de raccordement.
- Connectez la broche de signal (généralement de couleur blanche ou jaune) du servomoteur à une broche GPIO du Raspberry Pi (par exemple, GPIO 17).
- Connectez la broche d'alimentation (généralement rouge) du servomoteur à l'alimentation externe (3,3 à 5 V).
- Connectez la broche de masse (généralement noire ou brune) du servomoteur à la masse du Raspberry Pi.

2. Écrire le code Python :

- Ouvrez un éditeur de texte sur le Raspberry Pi.
- Écrivez un programme Python qui utilise la bibliothèque RPi.GPIO pour contrôler le servomoteur. Voici un exemple de code :

```
import RPi.GPIO as GPIO
import time
```

```
# Configuration des broches GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT)
```

```
# Configuration du PWM pour le servomoteur
pwm = GPIO.PWM(17, 50) # Fréquence de 50 Hz
```

```
# Fonction pour définir l'angle du servomoteur
```

```
def set_angle(angle):
    duty = (angle / 18) + 2
    GPIO.output(17, True)
    pwm.start(duty)
    time.sleep(1)
    GPIO.output(17, False)
    pwm.ChangeDutyCycle(0)
```

```
try:
```

```
    while True:
        set_angle(0) # Rotation à 0 degrés
        time.sleep(1)
        set_angle(90) # Rotation à 90 degrés
        time.sleep(1)
        set_angle(180) # Rotation à 180 degrés
        time.sleep(1)
```

```
except KeyboardInterrupt:
```

```
    pwm.stop()
    GPIO.cleanup()
```

3. Exécution du code :

- Enregistrez le code Python sous un nom tel que "servo_control.py".
- Exécutez le code en utilisant la commande suivante dans le terminal : **python servo_control.py**.

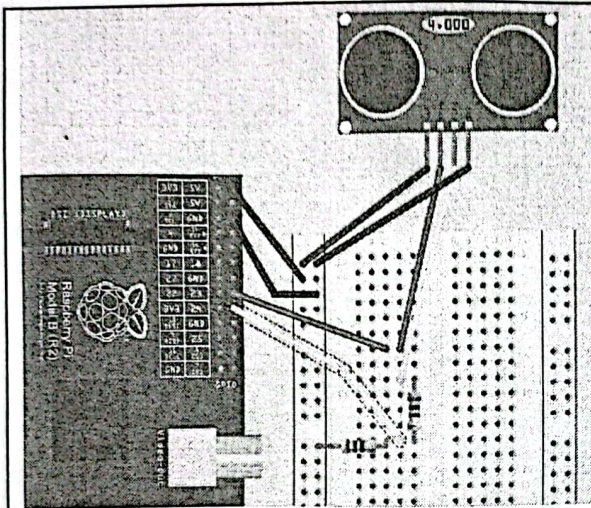
4. Observation :

- Le servomoteur devrait tourner à différents angles (0, 90, 180 degrés) à intervalles d'une seconde.

Manipulation 2 : Mesure de distance avec HC-SR04

Matériels :

- Un module ultrasons HC-SR04
- Une résistance de 1 k Ω
- Une résistance de 2 k Ω
- Une plaquette de câblage rapide



```
import time

TRIG=23

ECHO=24

print "Distance Measurement In Progress"
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)
GPIO.output(TRIG, False)
print "Waiting For Sensor To Settle"
time.sleep(2)
GPIO.output(TRIG, True)
time.sleep(0.00001)
GPIO.output(TRIG, False)
while GPIO.input(ECHO) == 0:
    pulse_start = time.time()
while GPIO.input(ECHO) == 1:
    pulse_end = time.time()
pulse_duration = pulse_end - pulse_start
distance = pulse_duration * 17150
distance = round(distance, 2)
print "Distance: ", distance, " cm"
GPIO.cleanup()
```

Sauvegardez votre code en le nommant « range_sensor.py » et lancez-le en utilisant la commande suivante :

```
sudo python range_sensor.py
```