

# Capteur de niveau d'eau dans le réservoir

## comment ça marche ?

- Ce capteur produit une tension **proportionnelle** au niveau d'eau.
- La Raspberry Pi ne peut lire que du **0V (LOW) ou 3.3V (HIGH)** sur ses GPIO.
- **Si l'eau touche les pistes**, la tension monte au-dessus de **1.8V** et la Raspberry Pi détecte **1 (HIGH)**.

## Programme:

```
import RPi.GPIO as GPIO
import time

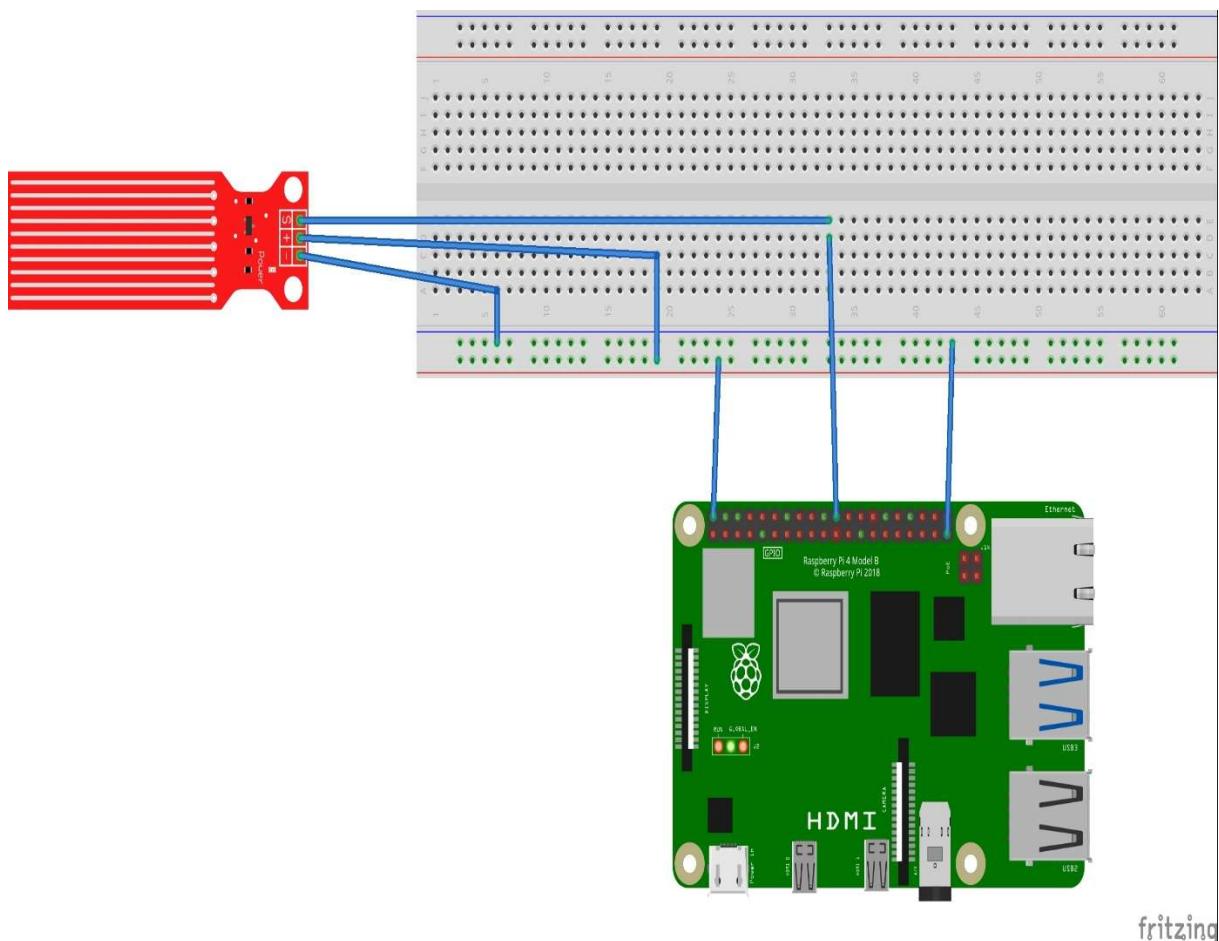
capteur_pin = 25 # GPIO utilisé

GPIO.setmode(GPIO.BCM)
GPIO.setup(capteur_pin, GPIO.IN)

try:
    while True:
        if GPIO.input(capteur_pin) == GPIO.HIGH:
            print("Présence d'eau détectée !")
        else:
            print("Pas d'eau détectée.")
        time.sleep(1)
except KeyboardInterrupt:
    GPIO.cleanup()
```

## Cablage :

- (-) → GND
- (+) → VCC
- (S) → GPIO(25)



### Démarche sur le terminal :

## 1. sudo nano nv.py

- **Objectif :** créer un nouveau script sous le nom nv.py contenant le code.

## 2. python3 nv.py

- **Objectif :** Exécute le script nv.py en utilisant l'interpréteur Python 3.

## Ajoutation d'un affichage sur LCD

**Programme :**

```
import time
import RPi.GPIO as GPIO
import drivers # Pour l'écran LCD

# Initialisation du LCD
display = drivers.Lcd()

# Configuration du capteur d'eau
capteur_pin = 25 # GPIO utilisé
GPIO.setmode(GPIO.BCM)
GPIO.setup(capteur_pin, GPIO.IN)

try:
    while True:
        # Lire l'état du capteur
        if GPIO.input(capteur_pin) == GPIO.HIGH:
            etat = "Présence d'eau"
        else:
            etat = "Pas d'eau"

        # Afficher les valeurs sur le LCD
        display.lcd_display_string("Capteur d'eau", 1)
        display.lcd_display_string(etat, 2)

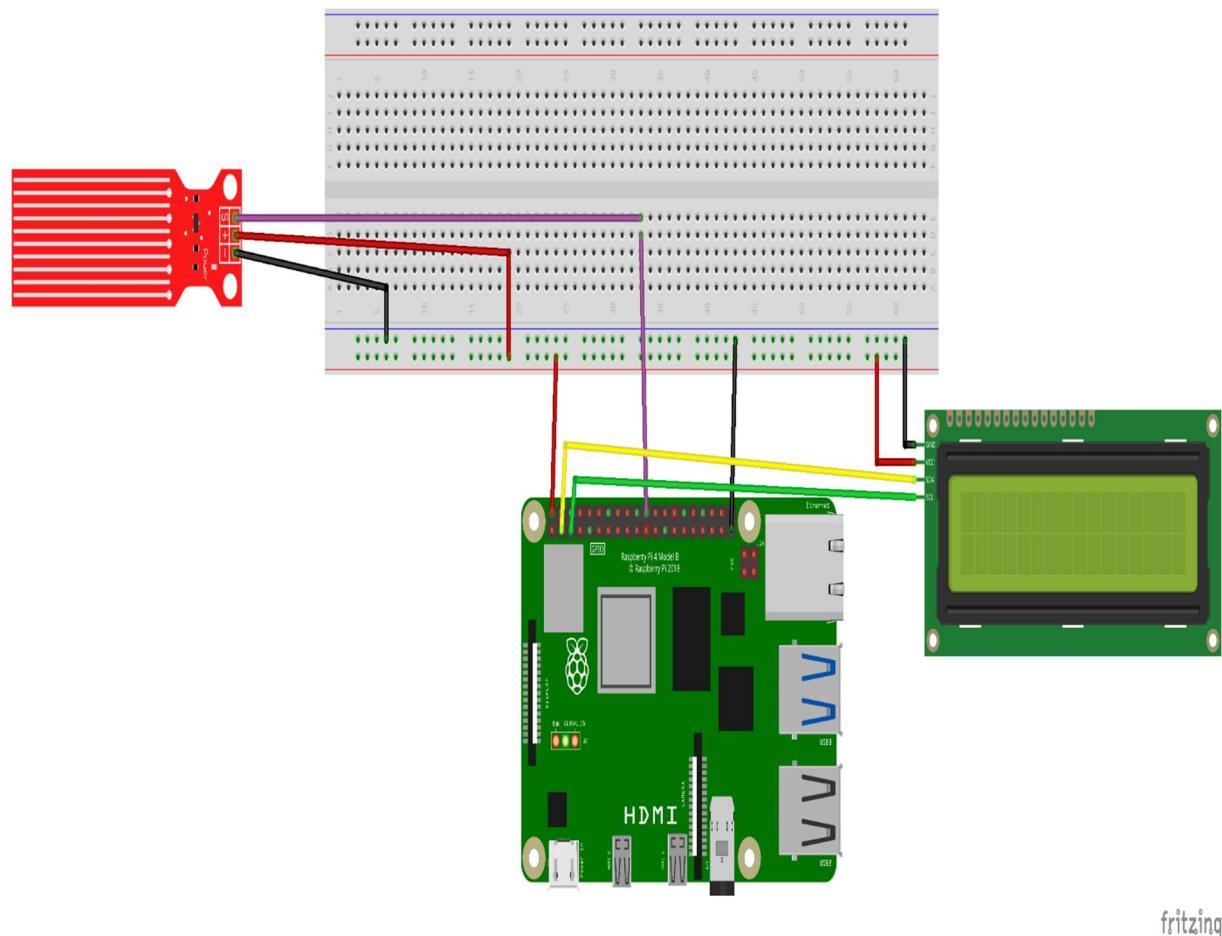
        # Afficher dans le terminal
        print(etat)

        time.sleep(2) # Pause pour éviter une mise à jour trop rapide

except KeyboardInterrupt:
```

```
print("Programme arrêté.")  
display.lcd_clear() # Nettoyer l'écran LCD avant de quitter  
GPIO.cleanup()
```

## Câblage :



fritzing

- **De capteur :**  
(-) → GND  
(+) → VCC  
(S) → GPIO(25)
- **De l'afficheur LCD :**  
VCC • 5V (Pin 2)  
GND • GND (Pin 6)  
SDA • GPIO (SDA, Pin 3)  
SCL • GPIO3 (SCL, Pin 5)

### Démarche sur le terminal :

#### 1. cd lcd

- **Objectif :** Change le répertoire courant pour accéder au dossier lcd car la bibliothèque de l'afficheur se trouve dans ce dossier.

#### 2. sudo nano reservoir.py

- **Objectif :** créer un nouveau script sous le nom reservoir.py contenant le code.

#### 3. python3 reservoir.py

- **Objectif :** Exécute le script reservoir.py en utilisant l'interpréteur Python 3.

The screenshot shows a terminal window with a light gray header bar containing the text "sifaoui@raspberrypi: ~/lcd". Below the header is a menu bar with "File", "Edit", "Tabs", and "Help". The main area of the terminal is black and displays the following text:

```
sifaoui@raspberrypi:~ $ cd lcd
sifaoui@raspberrypi:~/lcd $ sudo nano reservoir.py
sifaoui@raspberrypi:~/lcd $ python3 reservoir.py
    Pas d'eau
    Pas d'eau
    Pas d'eau
    Pas d'eau
    Pas d'eau
Presence d'eau
Presence d'eau
Presence d'eau
Presence d'eau
^CProgramme arrêté.
sifaoui@raspberrypi:~/lcd $
```

