

# Capteur de température et d'humidité DHT22

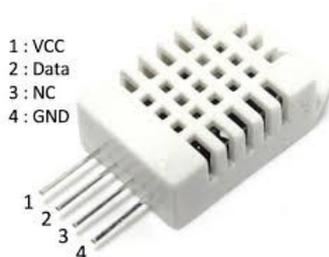
## Sommaire :

I - Présentation du capteur DHT22.....	1
II - Mise en œuvre du DHT22 avec un ESP32.....	1
III - Mise en œuvre du DHT22 avec un Raspberry Pi.....	1

## I - Présentation du capteur DHT22

La sonde de type **DHT22** permet de mesurer la **température** et l'**humidité**. Alimentée en **3.3V** ou **5V**, elle a les spécifications suivantes :

- Lecture de l'humidité entre **0** et **100%** avec une précision allant de **2%** à **5%** ;
- Lecture de la température de **-40°C** à **80°C** avec une précision d'environ **0.5°C**.



## II - Mise en œuvre du DHT22 avec un ESP32

Pour pouvoir utiliser le **DHT22** avec un **ESP32**, il faut ajouter la librairie **DHT** et le relier par exemple via la broche **Data** à **D4**, la broche **VCC** à **3.3V** et la broche **GND** à **GND**.

Exemple de programmation :

```
#include "DHT.h" // Librairie des capteurs DHT
#define DHTPIN 4 // Pin sur lequel est branché le DHT
#define DHTTYPE DHT22 // DHT 22
DHT dht(DHTPIN, DHTTYPE); // Instance de DHT
dht.begin(); // Initialisation du DHT22
float t = dht.readTemperature(); // Lecture température
float h = dht.readHumidity(); // Lecture humidité
```

## III - Mise en œuvre du DHT22 avec un Raspberry Pi

Pour pouvoir utiliser le **DHT22** avec un **Raspberry Pi**, il faut le relier par exemple via la broche **Data** à **GPIO6** (ligne **31**), la broche **VCC** à **3.3V** ou **5V** et la broche **GND** à **GND**.

Si l'on regarde le fichier **/boot/overlays/README**, on peut voir que le module **overlays dht11** est pris en charge :

```
Name: dht11
Info: Overlay for the DHT11/DHT21/DHT22 humidity/temperature sensors
      Also sometimes found with the part number(s) AM230x.
Load: dtoverlay=dht11,<param>=<val>
Params: gpiopin GPIO connected to the sensor's DATA output. (default 4)
```

Il suffit maintenant de modifier le fichier **/boot/config.txt** en rajoutant la ligne suivante, et de redémarrer le Raspberry Pi :

**dtoverlay=dht11,gpiopin=6**

On peut maintenant lire les informations de notre capteur dans les fichiers suivants :

**le nom du capteur : cat /sys/bus/iio/devices/iio:device0/name**

**la température en °C\*1000 : cat /sys/bus/iio/devices/iio:device0/in\_temp\_input**

**l'humidité en %\*1000 : cat /sys/bus/iio/devices/iio:device0/in\_humidityrelative\_input**

Si on veut rajouter un autre capteur **dht22** sur **GPIO18**, on peut exécuter la commande suivante :

**sudo dtoverlay dht11 gpiopin=18**

Et le nouveau capteur sera accessible via le fichier : **/sys/bus/iio/devices/iio:device1/**.

**Remarques** : Si ça ne fonctionne pas, on peut faire les 3 modifications suivantes :

1. Afin de connaître la version du **Raspberry Pi** utilisé, on peut exécuter la commande :

```
pi@raspberrypi:~ $ cat /proc/cpuinfo
processor       : 0
model name     : ARMv7 Processor rev 5 (v7l)
BogoMIPS      : 38.40
Features       : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae
                evtstrm
CPU implementer : 0x41
CPU architecture: 7
CPU variant    : 0x0
CPU part       : 0xc07
CPU revision   : 5
Hardware       : BCM2835
Revision       : a01041
Serial         : 000000006f60762b
Model          : Raspberry Pi 2 Model B Rev 1.1
```

2. En fonction du modèle du Raspberry Pi, il faut charger le bon fichier **dtb** :

```
bcm2708-rpi-0-w.dtb:  Pi Zero W
bcm2708-rpi-b.dtb:   Pi Model B and Model A
bcm2708-rpi-b-plus.dtb: Pi B+, A+ and Zero
bcm2708-rpi-cm.dtb:  Compute Module (a minimal dtb, intended to be a starting
point)
bcm2709-rpi-2-b.dtb:  Pi 2B
bcm2710-rpi-3-b.dtb:  Pi 3B
bcm2710-rpi-3-b-plus.dtb: Pi 3B+ and 3A+
bcm2710-rpi-cm3.dtb:  Compute Module 3
```

3. Il suffit maintenant de modifier le fichier **/boot/config.txt** en rajoutant les 3 lignes suivantes, et de redémarrer le Raspberry Pi :

**device\_tree=bcm2709-rpi-2-b.dtb**  
**dtparam=i2c\_arm=on**  
**dtoverlay=dht11,gpiopin=6**

Le programme en **langage C**, permettant de récupérer les informations du **DHT22** :

```
#include <stdio.h>
#include <errno.h>
#include <unistd.h>
#include <fcntl.h>

int main(int argc, char **argv)
{
    char buffer[25];
    int fd = open("/sys/bus/iio/devices/iio:device0/name",O_RDONLY);
    read(fd, buffer, 25);
    close(fd);
    printf("%s",buffer);
    fd = open("/sys/bus/iio/devices/iio:device0/in_temp_input", O_RDONLY);
    read(fd, buffer, 25);
    close(fd);
    int temp;
    sscanf(buffer, "%d", &temp);
    float temperature = temp / 1000.0;
    printf("temperature = %2.1f\n", temperature);
    fd=open("/sys/bus/iio/devices/iio:device0/in_humidityrelative_input",O_RDONLY);
    read(fd, buffer, 25);
    close(fd);
    int hum;
    sscanf(buffer, "%d", &hum);
    float humidite = hum / 1000.0;
    printf("humidite = %2.1f\n", humidite);
}
```

Résultat :

```
pi@raspberrypi:~/dht22/c $ sudo ./appli
dht11@6
temp int = 25000
temperature = 25.000000
hum int = 59100
humidite = 59.099998
```

Le programme en **shell bash**, permettant de récupérer les informations du **DHT22** :

dht11\_info.sh :

```
#!/bin/bash
# Get information from DHT11 Temperature and Humidity Sensor
# /sys/bus/iio/devices/iio:device0
while true
do
TEMP=`cat /sys/bus/iio/devices/iio\:device0/in_temp_input`
echo "Current TEMP is: `expr $TEMP / 1000` C"
sleep 10
HUMIDITY=`cat /sys/bus/iio/devices/iio\:device0/in_humidityrelative_input`
echo "Current Humidityrelative is: `expr $HUMIDITY / 1000` %"
sleep 10
clear
done
```