# Capteur de température et humidité DHT22

PAR CFAURY · PUBLIÉ 7 MAI 2019 · MIS À JOUR 6 JUIN 2020

**Matériel testé :** capteur de température et d'humidité DHT22

**Bibliothèque pour Python :** Adafruit_Python_DHT


Le **capteur DHT22** est un capteur numérique de température et d'humidité. Il utilise un capteur d'humidité capacitif et une thermistance pour mesurer l'air ambiant et génère un signal numérique sur la broche de données.

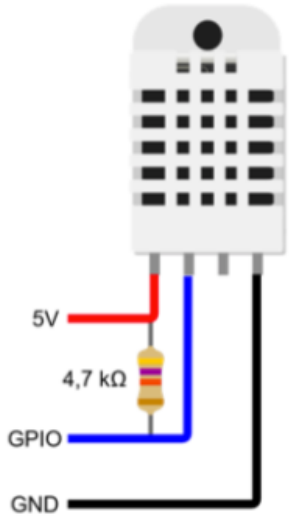
Le seul inconvénient réel de ce capteur est sa période de mesure : une fois toutes les 2 secondes.

**Caractéristiques techniques :**

- Alimentation : 3V à 5V
- Consommation : 2.5mA max
- Gamme d »humidité : 0-100% (précision 2% à 5%)
- Gamme de température : -40°C à 80°C (précision ±0.5°C)
- Période de mesure : 2 secondes

# Câblage

Il faut alimenter le capteur (5V) et utiliser une résistance de *Pull-Up* (4,7kΩ à 10kΩ) pour le port GPIO de communication avec le capteur.



Quel que soit votre choix de capteur DHTxx, le câblage est le même.

Les capteurs DHTxx communiquent avec le microcontrôleur via une unique broche d'entrée / sortie, dont on verra ensemble le protocole de communication dans le chapitre suivant.

Le brochage du capteur est le suivant :

- La broche n°1 est la broche d'alimentation (5 volts ou 3.3 volts).
- La broche n°2 est la broche de communication. Celle-ci doit impérativement être reliée à l'alimentation via une résistance de tirage de 4.7K ohms (il s'agit d'une sortie à collecteur ouvert).
- La broche n°3 n'est pas utilisée et ne doit pas être câblée.
- La broche n°4 est la masse du capteur (GND).

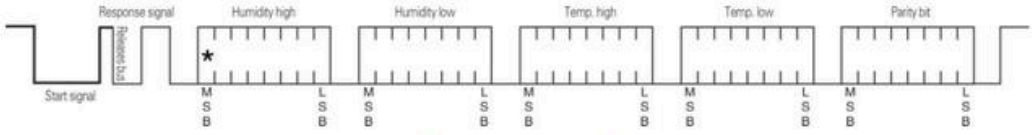
**N.B. Un condensateur de 100nF est requis entre les broches n°1 et n°4 pour que le capteur fonctionne correctement. On verra cela lors de la réalisation du montage de démonstration.**

## Le protocole de communication

Comme précisé précédemment, les capteurs DHTxx ont la particularité de communiquer avec le microcontrôleur via une unique broche d'entrée / sortie.

Bien que cela soit marqué "One Wire" un peu partout sur le document constructeur du capteur, il ne s'agit pas d'un véritable bus de communication 1-Wire. Il s'agit simplement d'un protocole de communication propriétaire, utilisant un seul fil et nécessitant des timings très précis.

## Le déroulement d'une communication




**Pic5:** AM2302 Single-bus communication protocol

Format d'une trame de communication

La communication avec un capteur DHTxx se fait en 3 étapes :

- Tout d'abord, le microcontrôleur maître (la carte Arduino dans notre cas) réveille le capteur en plaçant la ligne de données à `LOW` pendant au moins 800µs (au moins 18ms pour le DHT11). Durant ce laps de temps, le capteur va se réveiller et préparer une mesure de température et d'humidité. Une fois le temps écoulé, le maître va libérer la ligne de données et passer en écoute.
- Une fois la ligne de données libérée, le capteur répond au maître (pour montrer qu'il est bien réveillé) en maintenant la ligne de données à `LOW` pendant 80µs puis à `HIGH` pendant 80µs.
- Le capteur va ensuite transmettre une série de 40 bits (5 octets). Les deux premiers octets contiennent la mesure de l'humidité. Les deux octets suivants contiennent la mesure de la température et le cinquième octet contient une somme de contrôle qui permet de vérifier que les données lues sont correctes.

*N.B. La broche de communication des capteurs DHTxx est de type "collecteur ouvert". La sortie du capteur ne génère pas de tension. Elle ne fait que commuter (via un transistor) la tension au niveau de la résistance de tirage sur la ligne de données. Dans ce contexte, `HIGH` est la tension de la résistance de tirage et `LOW` la tension à la masse (0 volt). Cela peut paraitre complexe, mais sans cette sortie à collecteur ouvert, il ne serait pas possible d'alimenter le capteur en 5 volts tout en ayant une résistance de tirage à 3.3 volts (dans le cas d'un montage avec une carte Arduino en 3.3 volts derrière).*