

## Internet of Things TP n° 5

### Objectifs

- Mesurer la température et l'humidité en utilisant le capteur Dht11/Dht22
- Utiliser une plateforme IoT pour la visualisation des données.
- Utiliser une plateforme IoT pour le contrôler à distance des GPIO

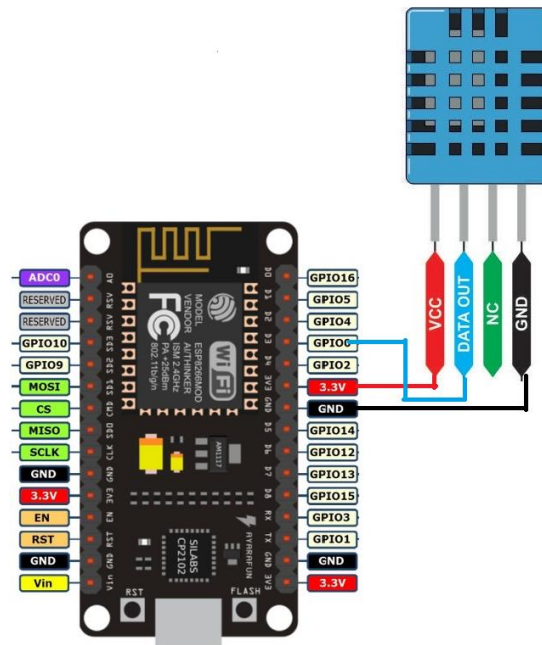
### Matériel Nécessaire

- Un ordinateur avec accès internet
- Carte ESP8266 NodeMCU V3 (ou ESP32) avec Câble Micro USB
- Capteur DHT11/DHT22

### Travaux

#### Exercice 1 (Mesurer la température et l'humidité )

Réaliser le montage suivant permettant d'utiliser un capteur de température DHT11.



- Ecrire le code Arduino permettant de mesurer la température et l'niveau d'humidité de la salle puis l'afficher sur le Moniteur Série (vous devez installer la librairie DHT11).

- i. A partir du gestionnaire des librairies de votre IDE :
  1. Installez la librairie "DHT.h"
  2. Installez la librairie "Adafruit Unified Sensor"
- ii. Le code devra contenir les éléments suivants :
  1. Un objet de type DHT dht(DHTPIN, DHTTYPE)
  2. La méthode dht.begin()
  3. Les méthodes readHumidity() et readTemperature();

## Exercice 2 (visualiser les données sur une plateforme IoT)

Modifiez le programme précédent afin d'envoyer les données collectées par le capteur DHT11 vers une plate-forme IOT dans le Cloud pour la visualisation (Utilisez la plate-forme ThingSpeak).

- b. Pour cela vous devez :
  - i. Créer un compte sur le site [https://thingspeak.com/users/sign\\_up](https://thingspeak.com/users/sign_up)
  - ii. Remplir les champs qui sont demandé
  - iii. Valider votre compte en cliquant sur le lien reçu dans votre boite de messagerie, puis connectez-vous.
  - iv. Créer un « channel » nommé BDSas-IoT contenant deux champs :  
Température °C et Humidité %  
« Un channel représente un ensemble de données regroupées entre elles et qui proviennent d'un objet connecté. Un channel est composé de champs (field). Il peut y en avoir jusqu'à 8. Et chaque champ représente une donnée ».

Il faut faire des relevés de température et humidité à intervalles réguliers et d'envoyer ces valeurs sur le channel ThinkSpeak créé à cet usage.

Le programme devra donc :

- Se connecter en wifi à un point d'accès local connecté à internet
  - Bibliothèque <ESP8266WiFi.h>
- Lire les données sur le capteur
  - Bibliothèque <DHT.h>
- Créer une connexion TCP pour envoyer les données à la plateforme Thingspeak
  - Utilisez la classe WiFiClient() pour créer une connexion TCP
  - La méthode connect(host, httpPort) pour se connecter au serveur Thingspeak
  - La méthode print() pour envoyer les données au niveau de la plateforme de visualisation.
- Attendre un moment (6 sec) et recommencer la lecture des données.

### Exercice 3 (Contrôler une LED à distance via Thingspeak )

L'objectif de cet exercice est de contrôler (allumer ou éteindre) une LED connectée à un ESP32 à distance en utilisant la plateforme ThingSpeak.

Pour se faire vous devez utiliser :

- La bibliothèque « WiFi.h » pour faire connecter l'ESP32 à un point d'accès wifi
- la classe WiFiClient pour créer un objet (client) qui peut se connecter à une adresse IP et un port Internet spécifiés.
- La bibliothèque « ThingSpeak.h » pour
- La méthode ThingSpeak.begin() pour initialiser la communication avec thingspeak
- La méthode ThingSpeak.readFloatField() pour récupérer l'état du field ( champ) au niveau de thingspeak

