

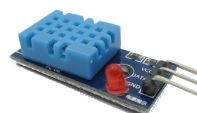
TP : création d'un objet connecté

Dans ce TP nous allons utiliser un microcontrôleur **ESP8266** avec un capteur **DHT11** pour lire la température et l'humidité. Le but finale est de pouvoir récupérer ces informations grâce à une API qui nous permettra de les insérer dans une application web pour pouvoir les stocker dans une base de données.

1. Les composants

Pour cela nous aurons besoin :

- d'un microcontrôleur : l'ESP8266 ;
- d'un capteur de température et d'humidité : le DHT11

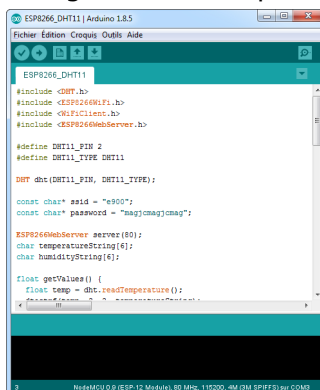


- de câbles dupont ;
- d'un câble micro USB.





2. Découverte du logiciel Arduino

Pour programmer notre microcontrôleur, nous allons avoir besoin d'un Environnement de Développement Intégré (EDI) ou Integrated Development Environment (IDE) en anglais : le logiciel Arduino.



Ce logiciel permet principalement trois actions :

1. l'écriture du code, que l'on appelle le développement d'un programme ;
2. la vérification/compilation du programme, c'est à dire sa transformation en langage machine ; 
3. son téléversement sur le microcontrôleur. 

Il permet également d'ouvrir un terminal sur le port console du microcontrôleur, ce qui sert principalement au débogage du programme. 

3. Installation des librairies / bibliothèques

Pour pouvoir utiliser notre microcontrôleur il nous faut installer du « code » qui va permettre à l'IDE Arduino, prévu à la base pour l'Arduino, de pouvoir dialoguer avec un ESP8266.

Le code qui permet la gestion d'une fonctionnalité, ici le dialogue avec l'ESP8266, s'appelle une librairie ou bibliothèque.

Pour ce faire, nous allons indiquer l'emplacement de cette librairie au logiciel Arduino.

Dans le menu Fichier → Préférences, indiquez l'URL suivante :

http://arduino.esp8266.com/staging/package_esp8266com_index.json



Il faut télécharger cette librairie en allant dans le menu :

Outils → Type de carte → Gestionnaire de carte

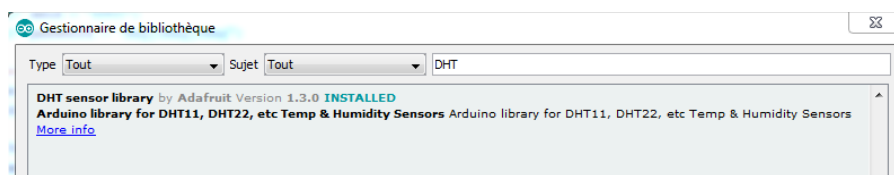
Dans le champ de recherche tapez « ESP »

Installez la dernière version de la librairie « esp8266 by ESP8266 Community »

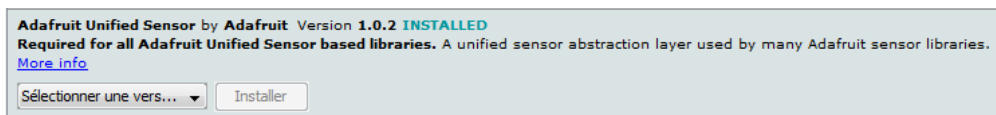


Il faut maintenant ajouter la librairie qui va nous permettre d'utiliser le module DHT11. Dans le menu Croquis → Inclure une bibliothèque → Gérer les bibliothèques, installez les librairies :

- DHT sensor library by Adafruit



- Adafruit Unified Sensor by Adafruit

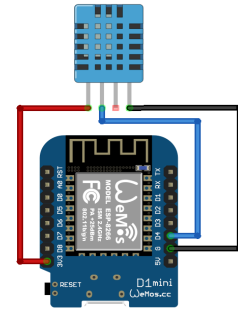


4. Récupération des valeurs du capteur

a) Montage électronique

Le montage électronique est très simple, il suffit de faire les connexions suivantes :

- VCC du DHT11 sur 3.3v de l'ESP
- GND du DHT11 sur GND de l'ESP (G)
- DATA du DHT11 sur D4 de l'ESP



b) Récupération des valeurs dans le port console

Dans un premier temps, nous allons uniquement récupérer les valeurs qui nous intéressent et les afficher sur le port console de l'IDE Arduino.

Pour cela, nous allons tout d'abord faire appel aux librairies dont nous avons besoin :

```
// Code relatif à l'ESP
#include <ESP8266WiFi.h>
#include <WiFiClient.h>

// Code propre au DHT11
#include <DHT.h>
#define DHT11_PIN D4
#define DHT11_TYPE DHT11

DHT dht(DHT11_PIN, DHT11_TYPE);
```

Nous allons ensuite modifier la fonction `setup()` :

```
void setup() {
  Serial.begin(115200);
  delay(1000);
  Serial.println("Starting...");
}
```

Enfin, dans la fonction `loop()` nous allons afficher la température :

```
void loop() {
  Serial.print("Température : ");
  Serial.println(dht.readTemperature());
  Serial.print("Humidité : ");
  Serial.println(dht.readHumidity());
  delay(1000);
}
```

Une fois le code téléversé sur la puce, vous devriez voir de l'activité sur le port série.

5. Connexion de l'objet

a) Utilisation du Wi-Fi

La particularité de l'ESP8266 est d'embarquer la partie radio nécessaire au **Wi-Fi**. Elle peut agir comme un point d'accès, comme un client ou comme les deux.

Pour configurer l'ESP8266 comme un point d'accès ou comme un client Wi-Fi nous allons déclarer les variables suivantes :

```
const char* ssid = "mon_ssid_unique_attention_aux_autres_groupes";  
const char* password = "mon_mot_de_passe";
```

1 Configuration en point d'accès

Si vous voulez utiliser l'ESP comme point d'accès et vous y connecter, dans la fonction setup() il suffit d'utiliser le code suivant :

```
boolean result = WiFi.softAP(ssid, password);  
if(result == true) {  
    Serial.println("OK");  
}else{  
    Serial.println("Echec!");  
}
```

2 Configuration en client Wi-Fi

Si vous voulez que l'ESP se connecte à un point d'accès (comme votre téléphone), dans la fonction setup() il suffit d'utiliser le code suivant :

```
WiFi.begin(ssid, password);  
Serial.println("");  
while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
}  
Serial.println("");  
Serial.print("Connecté à ");  
Serial.print(ssid);  
Serial.print(" avec l'IP: ");  
Serial.println(WiFi.localIP());
```

6. Utilisation du serveur web embarqué

Notre objet connecté est presque terminée, il affiche bien la température et l'humidité mais sur le port série. Maintenant que nous l'avons raccordée au réseau Wi-Fi, nous allons utiliser le serveur web intégré pour afficher ces informations sur une page accessible par un navigateur.

a) Écriture d'une page Web

Il faut, dans un premier temps, déclarer la librairie qui va nous permettre d'utiliser le serveur web, déclarer que le serveur utilise le port 80 et déclarer une variable qui nous servira à contenir le HTML :

```
#include <ESP8266WebServer.h>
ESP8266WebServer server(80);
String html = "";
```

Dans la fonction *setup()* :

- nous allons créer une page en HTML que nous allons mettre dans un objet de type chaîne de caractères :

```
html += "<html><head><title>Station météo</title>";
html += "<meta charset=\"utf-8\" />\n";
html += "</head><body>ESP8266</body></html>";
```

- Nous allons spécifier l'URL que notre serveur utilisera dans la fonction *setup()* :

```
server.on("/", [] () {
    server.send(200, "text/html", html);
});
```

- Enfin, nous allons démarrer le serveur pour qu'il écoute les connections des clients :

```
server.begin();
```

Dans la fonction *loop()* nous allons nous occuper des connections des navigateurs web :

```
void loop() {
    server.handleClient();
}
```

Vous pouvez maintenant modifier la page affichée pour que la température et l'humidité y figure !