

Cours Internet des Objets (IoT) et Cloud Computing

CH5: Plateformes de contrôle et de monitoring IoT

Par:

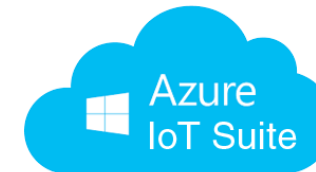
Hassan EL FADIL

elfadil.h@gmail.com

1. Les plateformes IoT

- Les plateformes de contrôle et de monitoring IoT les plus répandues varient en fonction de plusieurs critères tels que la taille de l'entreprise, les besoins spécifiques en matière de surveillance, les intégrations requises, etc.
- Voici quelques-unes des plateformes de monitoring IoT:

- **AWS IoT Core**
- **Microsoft Azure IoT Suite**
- **Google Cloud IoT Core**
- **IBM Watson IoT Platform**
- **ThingSpeak**
- **Node-RED**
- ...



Cloud IoT Core



IBM Watson



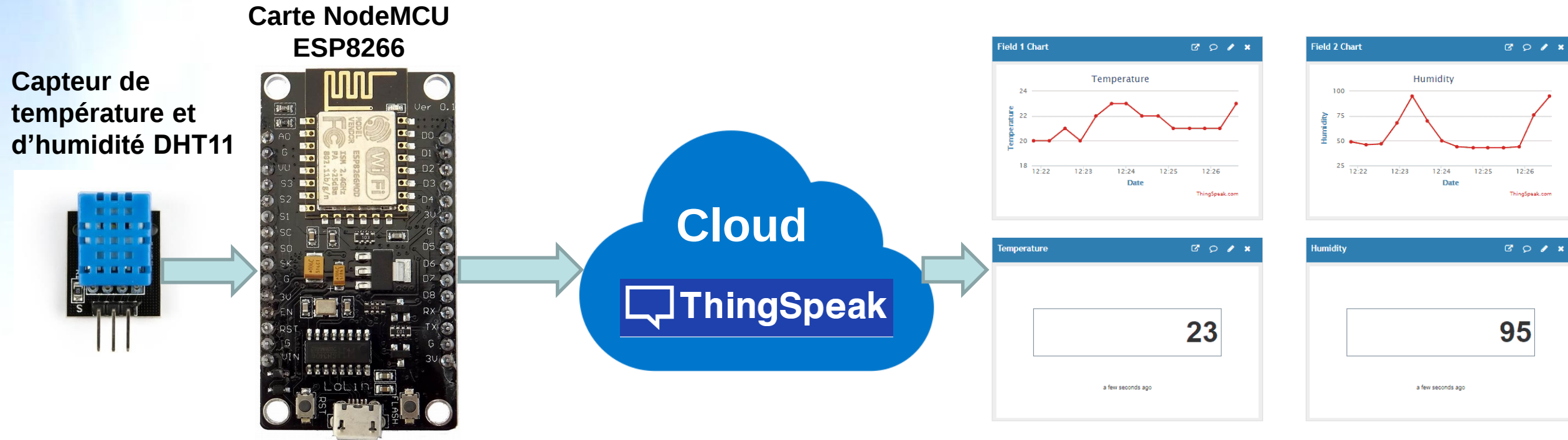
Node-RED



2. Exemple d'application IoT avec la plateforme ThingSpeak

2.1. Objectif

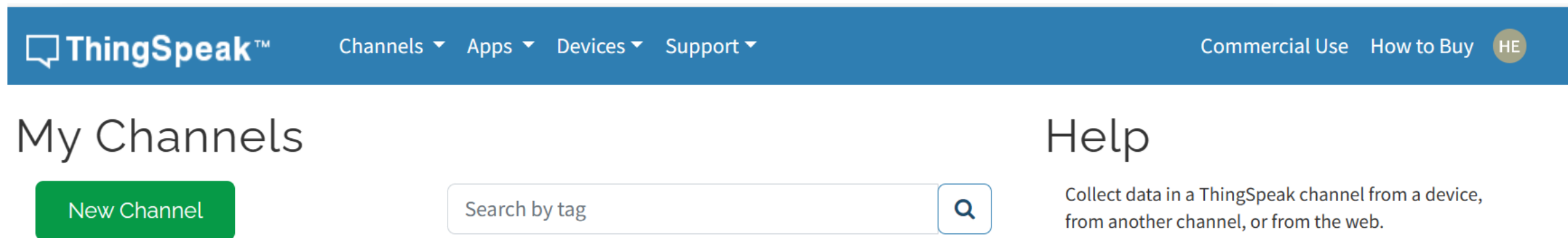
- Monitoring de la température et de l'humidité.
- Utilisation de la carte **NodeMCU ESP8266** et la plateforme **ThingSpeak**



2.2. Configuration de ThingSpeak:

1 Créez un compte ThingSpeak si vous n'en avez pas déjà un. <https://thingspeak.com/>
Il s'agit d'un compte Mathworks

- *ThingSpeak a été conçu et développé par MathWorks (celle qui a développé Matlab/Simulink).*
- *ThingSpeak est une plateforme polyvalente pour la collecte, le stockage, la visualisation et l'analyse de données IoT, offrant des outils puissants pour surveiller et contrôler des appareils connectés à distance, analyser des tendances, et prendre des décisions basées sur les données en temps réel.*



2 Créez un nouveau canal ThingSpeak et définissez deux champs pour la température et l'humidité :

My Channels

a)

Search by tag

- a) Cliquer sur « **New Channel** »
- b) Saisir le nom du canal et une description (pas vraiment obligatoire)
- c) Créer deux champs (**Field**): **Température** et **Humidité**.
- d) En bas de l'écran, cliquer sur « **Save Channel** » pour sauvegarder.

New Channel

b)

Name	Monitoring de la Température et de l'Humidité	
Description	Donées du Cpateur de Température et d'Humidité DHT11	

c)

Field 1	Température	☒
Field 2	Humidité	☒
Field 3		☐

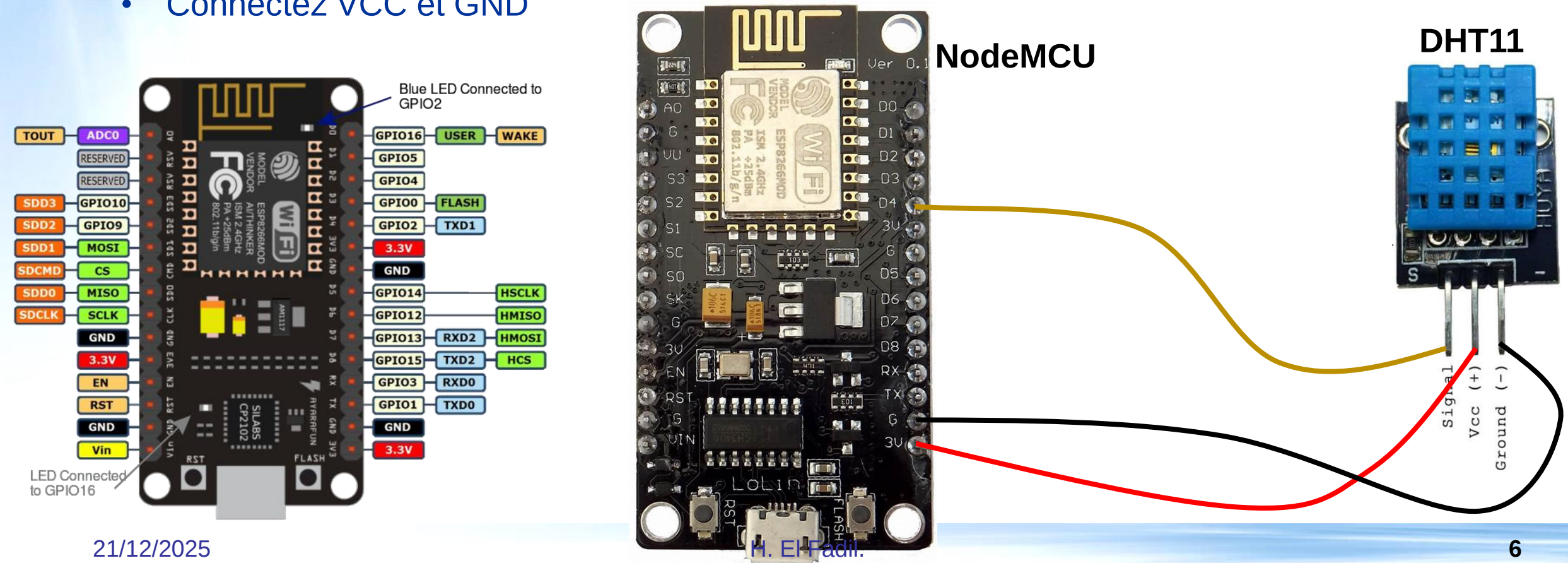
Show Status ☐

d)

2.3. Configuration matérielle et logicielle

1 Câblage: Connectez le DHT11 au NodeMCU.

- Connectez la broche de données du DHT11 (Signal) au GPIO 02 (broche D4) du NodeMCU.
- Connectez VCC et GND



- 2 Câblage: Connecter le NodeMCU avec le PC (vérifier que le carte est bien détectée. Sinon installer ses drivers):

Voir Chapitre 4

- 3 Préparation de l'environnement de développement (IDE Arduino):

*Assurez-vous d'avoir les bibliothèques: **ESP8266WiFi**, **ThingSpeak**, **WiFiClient** et **DHT**, sont installées dans votre environnement Arduino IDE*

Voir Chapitre 4

2.4. Code IDE Arduino:

1 Localiser l'ID de votre chaîne ThingSpeak :

Monitoring de la Température et de l'Humidité

Channel ID: 2523123

Author: mwa0000013081713

Access: Private

Données du Capteur de Température et d'Humidité DHT11

2 Localiser la clé API d'écriture de votre chaîne ThingSpeak (Write API Key) :

Private View Public View Channel Settings Sharing API Keys Data Import / Export

+ Add Visualizations + Add Widgets Export recent data

Write API Key

Key 664NAF9S3WKE4S41

Generate New Write API Key

3 Code complet Arduino 1/3:

```
#include <ESP8266WiFi.h>
#include <ThingSpeak.h>
#include <WiFiClient.h>
#include <DHT.h>
```

ThingSpeak utilise le protocole HTTP et ne pas MQTT

```
// Remplacez ces valeurs par les informations de votre réseau WiFi et ThingSpeak
const char* ssid = "AUTOMATIC_LAB_2G"; // Remplacer par le nom du Wifi
const char* password = "ACET-LAB"; // Remplacer par le mot de passe du Wifi
```

```
WiFiClient client;
```

Remplacer par l'ID de votre chaîne ThingSpeak!

```
// Remplacez ces valeurs par les informations de votre chaîne ThingSpeak
unsigned long CHANNEL_ID = 2365623; // ID de votre chaîne ThingSpeak sans les guillemets
const char* thingspeakApiKey = "664NAF9S3WKE4S41"; // Write API Key de votre chaîne ThingSpeak
```

Remplacer par le Write API Key de votre chaîne ThingSpeak!

3 Code complet Arduino 2/3:

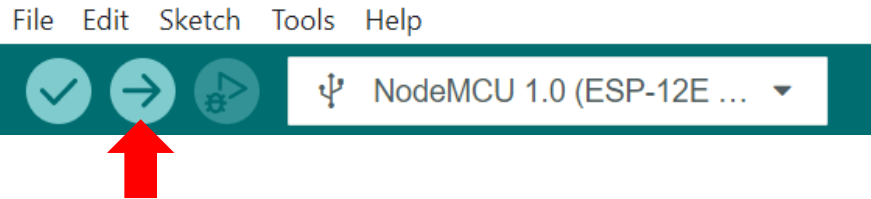
```
#define DHTPIN 2          // Broche du DHT connectée au GPIO 2 (D4)
#define DHTTYPE DHT11    // Type de capteur DHT
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  delay(10);
  // Se connecter au Wi-Fi
  Serial.println();
  Serial.print("Connexion au WiFi : ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(250);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connecté");
  dht.begin();
  ThingSpeak.begin(client);
}
```

3 Code complet Arduino 3/3:

```
void loop() {  
  
    // Lire les données du capteur  
    float humidity = dht.readHumidity();  
    float temperature = dht.readTemperature();  
    // Afficher les données sur la console série  
    Serial.print("Humidité: ");  
    Serial.print(humidity);  
    Serial.print("% - Température: ");  
    Serial.print(temperature);  
    Serial.println("°C");  
    // Envoyer les données sur un serveur (ThingSpeak)  
    ThingSpeak.writeField(CHANNEL_ID, 1, temperature, thingspeakApiKey);  
    ThingSpeak.writeField(CHANNEL_ID, 2, humidity, thingspeakApiKey);  
    delay(100);  
  
}
```

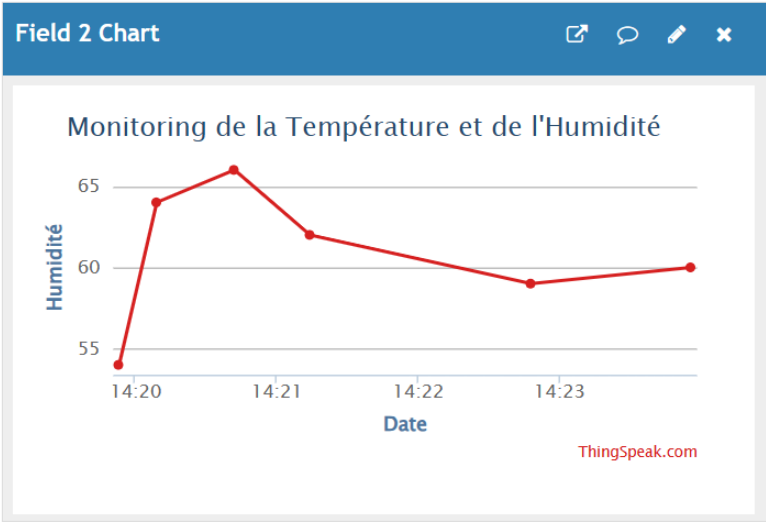
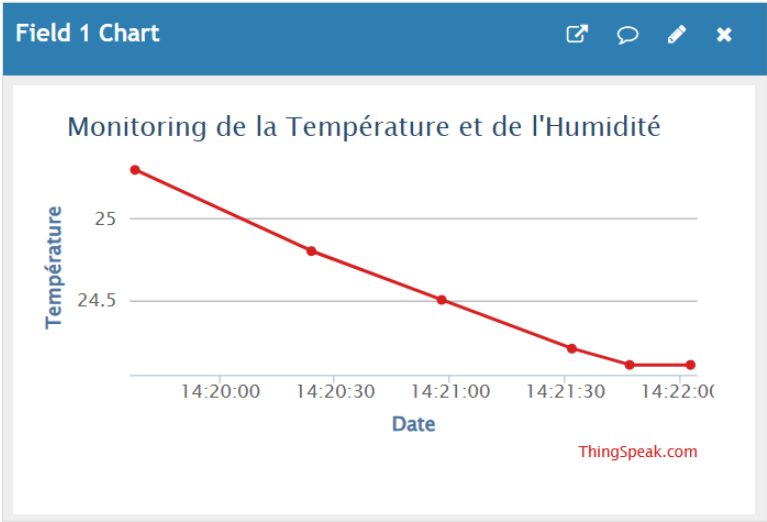
2.5. Test et validation:

- 1 La carte étant connectée au port USB du PC, **Téléverser**, alors, le programme (UPLOAD):



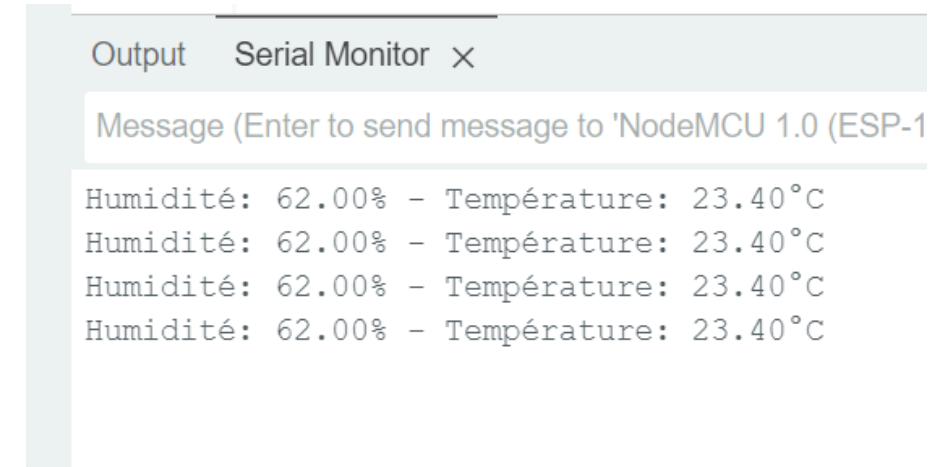
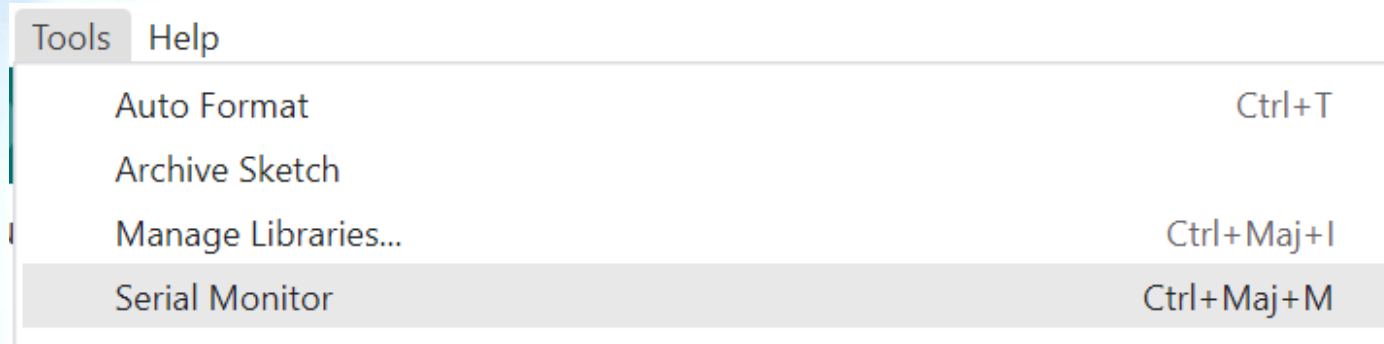
- 2 Observer l'évolution de la température et l'humidité sur votre Canal ThingSpeak:

Vous pouvez souffler sur le capteur DHT11 pour observer la variations des deux grandeurs mesurées.



3 Observer mesures envoyées depuis le moniteur série de l'IDE Arduino:

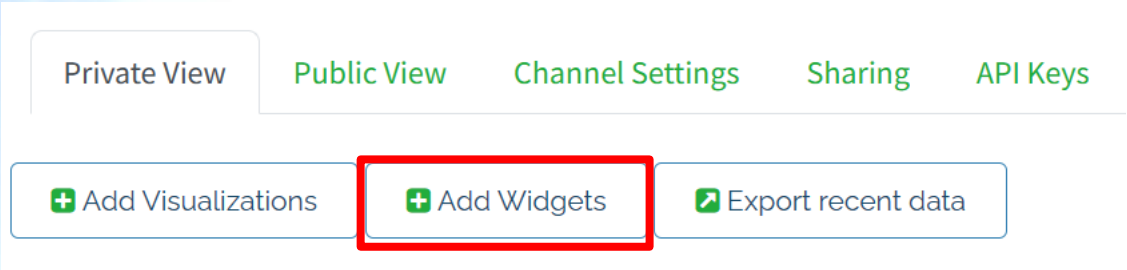
Tools > Serial Monitor



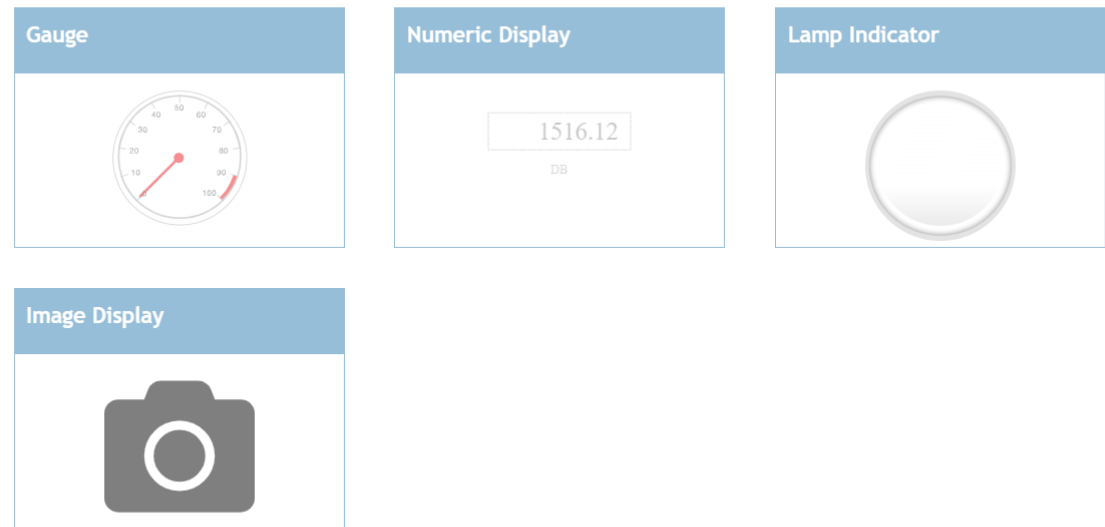
Remarque: pour un compte ThingSpeak gratuit (free), les données sont envoyées avec un pas de 15 secondes à peu près.

Pour des vitesses rapides vous devez disposer d'un compte professionnel (Payant)!

4 Vous pouvez ajouter d'autres éléments graphiques (Add Widgets) à votre Canal ThingSpeak :



Click on a widget to add it to the Channel



ENJOY!

Exemple: Insérer
une jauge
(Gauge) pour la
mesure de la
température

Température Options

Name

Température

Field

Field 1

Min

0

Max

50

Display Value

☒

Units

°C

Tick Interval

10

Update Interval

15

second(s)

Range

40

50

30

40

20

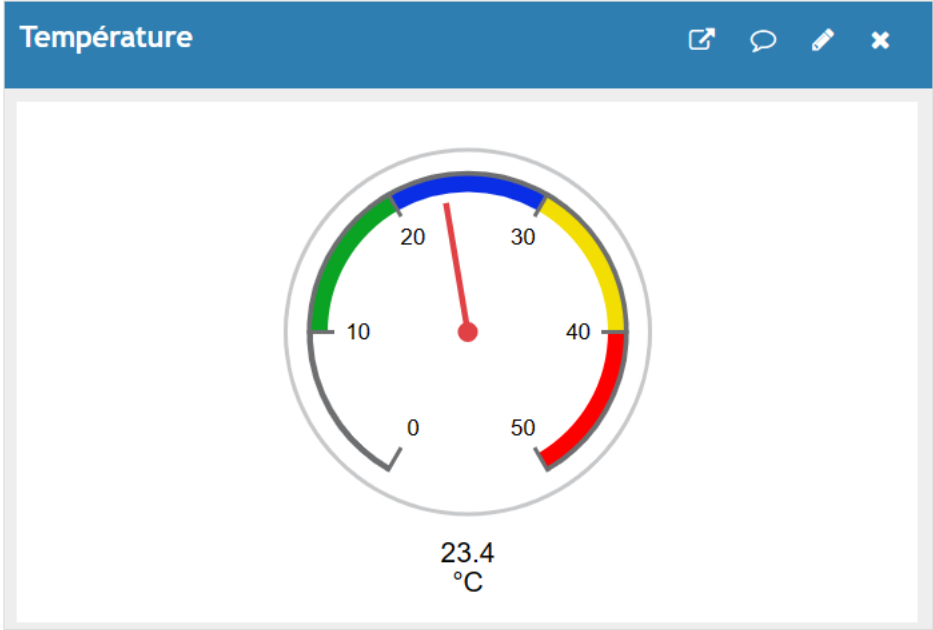
30

10

20

Save

Cancel



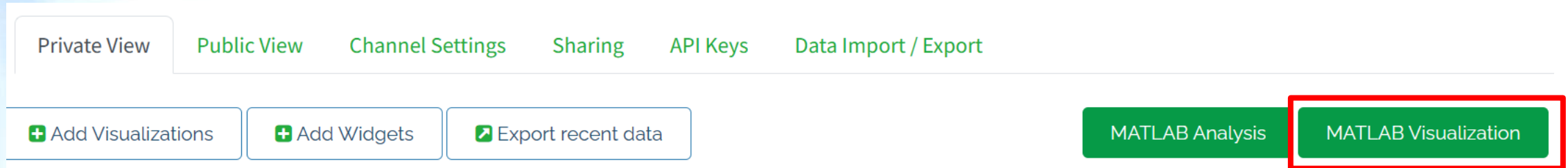
4 Vous pouvez partager votre affichage avec d'autres personnes. Pour cela, vous devez rendre publique votre chaîne :

- a) Cliquer sur « **Sharing** »
- b) Sélectionner « **Share channel view with everyone** »
- c) Partager le lien **URL** de votre chaîne.

The screenshot shows the ThingSpeak web interface. At the top, there are tabs: 'Private View', 'Public View', 'Channel Settings', 'Sharing' (highlighted with a red box and labeled 'a)'), 'API Keys', and 'Data Import / Export'. Below these tabs are three buttons: '+ Add Visualizations', '+ Add Widgets', and 'Export recent data'. The 'Channel Sharing Settings' section is visible, with three radio button options: 'Keep channel view private', 'Share channel view with everyone' (selected and highlighted with a red box and labeled 'b)'), and 'Share channel view only with the following users:'. Below this is an 'Email Address' input field with a placeholder 'Enter email here' and an 'Add User' button. At the bottom, a browser address bar shows the URL 'https://thingspeak.com/channels/2523123' (highlighted with a red box and labeled 'c)'). The ThingSpeak logo and navigation menu (Channels, Apps, Devices, Support) are at the very bottom.

5 Visualisation des données sous Matlab :

a) *Directement à partir de la plateforme ThingSpeak (MATLAB Visualization). Utiliser l'ID et le ReadAPIKey de votre chaîne*



b) *Ecriture d'un code depuis Matlab. Exemple:*

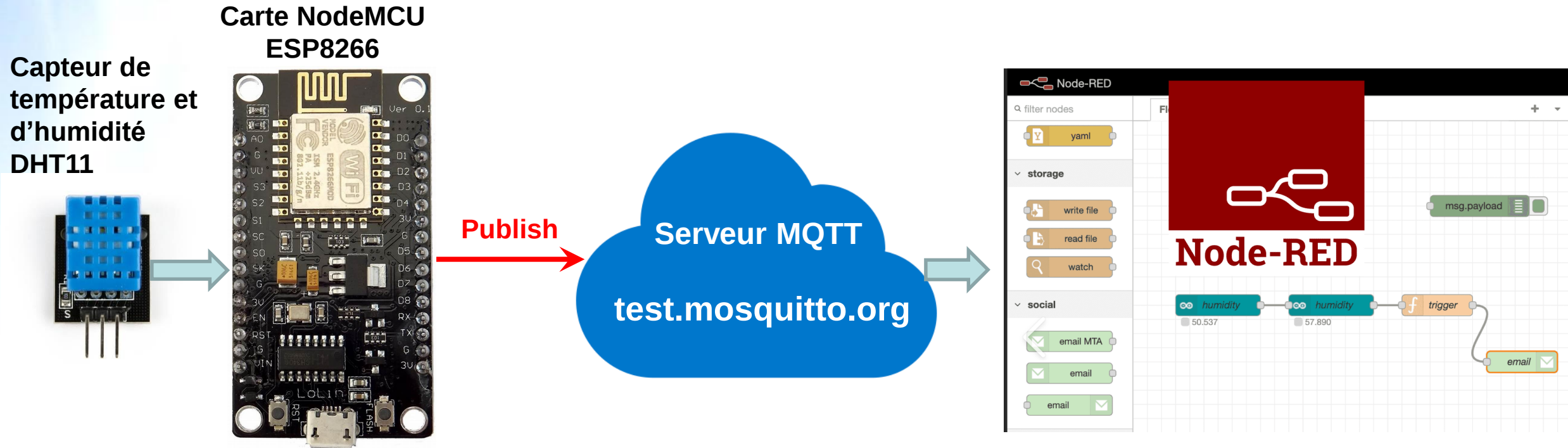
```
% Remplacez 'VotreCléAPI' par votre propre clé d'API
channelID = 2523123;
readAPIKey = '664NAF9S3WKE4S41';
numPoints = 100; % Nombre de points de données à récupérer

data = thingSpeakRead(channelID, 'ReadKey', readAPIKey, 'NumPoints', numPoints);
```

3. Application avec la plateforme Node-RED

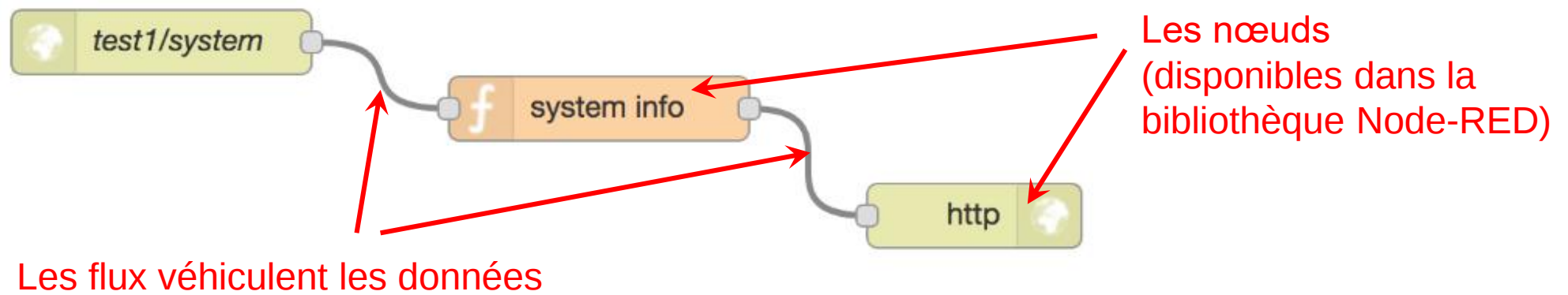
3.1. Objectif

- Monitoring de la température et de l'humidité.
- Utilisation de la carte **NodeMCU ESP8266** et la plateforme **Node-RED**



3.2. Présentation de Node-RED:

- Node-RED est une plateforme open source de développement visuel qui permet de connecter facilement des appareils, des services web et des API pour créer des flux de données et des applications IoT.
- Avec Node-RED, vous pouvez créer des flux de traitement en utilisant des nœuds pré-construits et en les reliant par glisser-déposer dans une **interface graphique conviviale**.
- Cela facilite la création d'applications IoT sans nécessiter une programmation complexe, ce qui en fait un outil populaire pour les développeurs des applications IoT.



- Node-RED a été développé par IBM, depuis 2013.
- Outil open source
- Disponibles sous Windows, macOS et Linux
- Gestion de plusieurs protocoles:

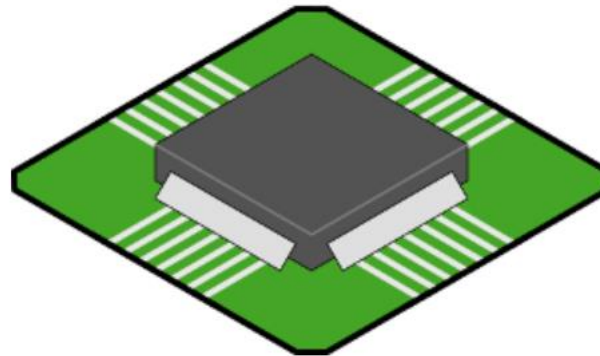


- Node-RED peut être utilisé:



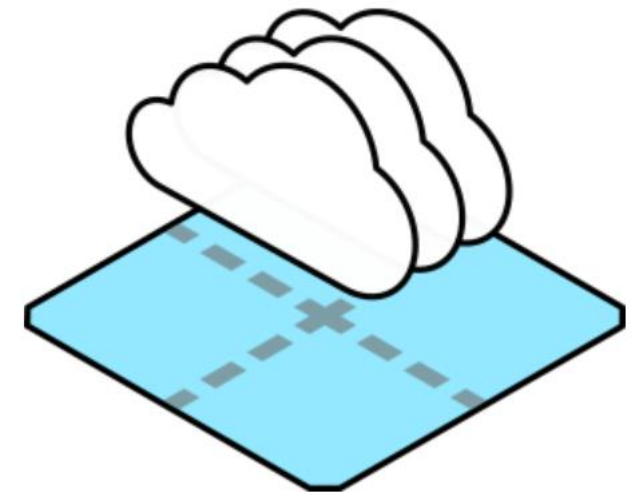
Localement

Installé sur votre ordinateur (Windows, macOS ou Linux)



Sur un système embarqué

Carte Raspberry Pi, BeagleBone, Arduino, Android et autres.



Sur un Serveur distant (Cloud)

IBM Cloud
Amazon Web Services
Microsoft Azure
Etc.

3.3. Installation de Node-RED:

Pour installer Node-RED sur Windows, suivez les étapes suivantes:

1 Installer Node.js

*Node-RED nécessite **Node.js** pour fonctionner. Si vous n'avez pas déjà Node.js installé, téléchargez et **installez** la dernière version depuis le site officiel de Node.js :*
<https://nodejs.org>

Une fois installé, ouvrez une invite de commande (cmd) et exécutez la commande suivante pour vous assurer que **Node.js** et **npm** sont correctement installés:

node --version && npm --version

Download Node.js®

Download Node.js the way you want.

[Prebuilt Installer](#) [Prebuilt Binaries](#) [Package Manager](#) [Source Code](#)

I want the version of Node.js for running

[Download Node.js v20.12.2](#)

Node.js includes npm (10.5.0) ↗.

Invite de commandes

```
Microsoft Windows [version 10.0.19045.4170]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Dell>node --version && npm --version
v20.10.0
10.2.3

C:\Users\Dell>
```

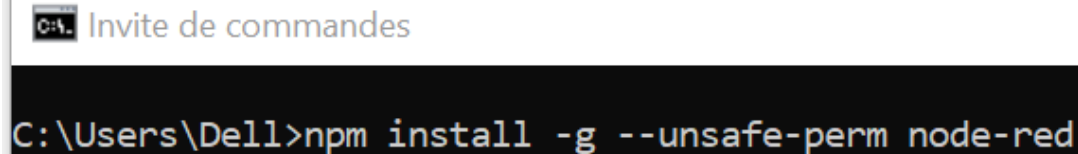


- **Node.js** est un environnement d'exécution **JavaScript** côté serveur.
- Il est optimisé pour la construction d'applications web rapides, évolutives et en temps réel, tout en simplifiant le processus de développement grâce à **npm** et à son approche asynchrone
- **npm** (**Node Package Manager**) est le gestionnaire de paquets officiel pour Node.js.
- Son rôle principal est de faciliter l'installation, la gestion et la mise à jour des bibliothèques, des modules et des dépendances JavaScript utilisées dans les projets Node.js.

2 Installer Node-RED:

Une fois Node.js installé, ouvrez une fenêtre de terminal (invite de commandes: **cmd**) et tapez la commande suivante pour installer Node-RED via npm :

```
npm install -g --unsafe-perm node-red
```

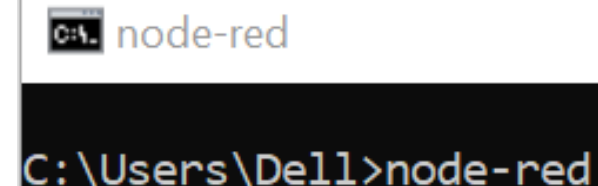


```
C:\Users\Dell>npm install -g --unsafe-perm node-red
```

3 Démarrer Node-RED:

Une fois l'installation terminée, vous pouvez démarrer Node-RED en utilisant la commande suivante dans votre terminal (**cmd**):

```
node-red
```



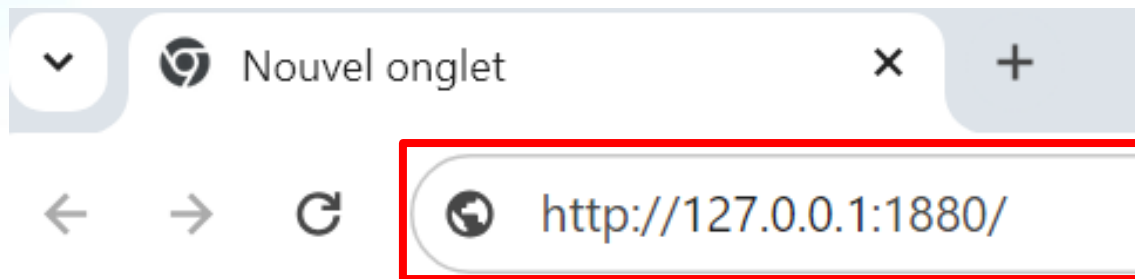
```
C:\Users\Dell>node-red
```

```
CA: Sélection node-red

Welcome to Node-RED
=====

26 Apr 13:06:25 - [info] Node-RED version: v3.1.9
26 Apr 13:06:25 - [info] Node.js version: v20.10.0
26 Apr 13:06:25 - [info] Windows_NT 10.0.19045 x64 LE
26 Apr 13:06:25 - [info] Loading palette nodes
26 Apr 13:06:26 - [info] Dashboard version 3.6.2 started at /ui
26 Apr 13:06:27 - [info] Settings file : C:\Users\Dell\.node-red\settings.js
26 Apr 13:06:27 - [info] Context store : 'default' [module=memory]
26 Apr 13:06:27 - [info] User directory : \Users\Dell\.node-red
26 Apr 13:06:27 - [warn] Projects disabled : editorTheme.projects.enabled=false
26 Apr 13:06:27 - [info] Flows file
26 Apr 13:06:27 - [info] Server now running at http://127.0.0.1:1880/
26 Apr 13:06:27 - [warn]
```

- Le serveur est en fonctionnement à l'adresse web indiquée ici.
- Copier cette adresse et la coller dans votre navigateur web (Google chrome, Microsoft Edge, Mozilla Firefox, etc.)



Navigateur web.

Le N°1880 correspond au port Node-RED
127.0.0.1: adresse locale.

Peut être remplacée par **localhost**:
http://localhost:1880/

...Exemple d'application IoT avec la plateforme Node-RED / Installation de Node-RED

Pour le déploiement **Menu général**

The screenshot shows the Node-RED web interface. At the top, there's a header with the Node-RED logo and a 'Déployer' button. Below the header, there's a search bar and a row of tabs labeled 'Flux 1', 'Flux 2', 'Flux 3', and 'Flux 4'. A red circle highlights a '+' button next to 'Flux 4'. On the left side, there's a palette of nodes grouped into 'Commun' and 'Fonction'. A red box highlights this palette. In the center, there's a workspace with a grid. A red box highlights the workspace area. On the right side, there's an 'Aide' panel showing details for the 'mqtt in' node. A red box highlights this panel. Annotations in French are overlaid on the image:

- Ajouter autant de pages que vous souhaitez** (Add as many pages as you want) - points to the '+' button.
- Zone d'informations textuelles, d'aide, de débogage et de configuration** (Textual information zone, help, debugging, and configuration) - points to the 'Aide' panel.
- Insérer Ici les différents nœuds de votre application** (Insert here the different nodes of your application) - points to the workspace.
- Zone des différents nœuds disponibles (Palette). Les nœuds sont regroupés en dossiers: Commun, Fonction, Réseau, dashboard, etc.** (Zone of the different available nodes (Palette). The nodes are grouped into folders: Common, Function, Network, dashboard, etc.) - points to the node palette.

3.4. Réalisation de l'application IoT avec Node-RED:

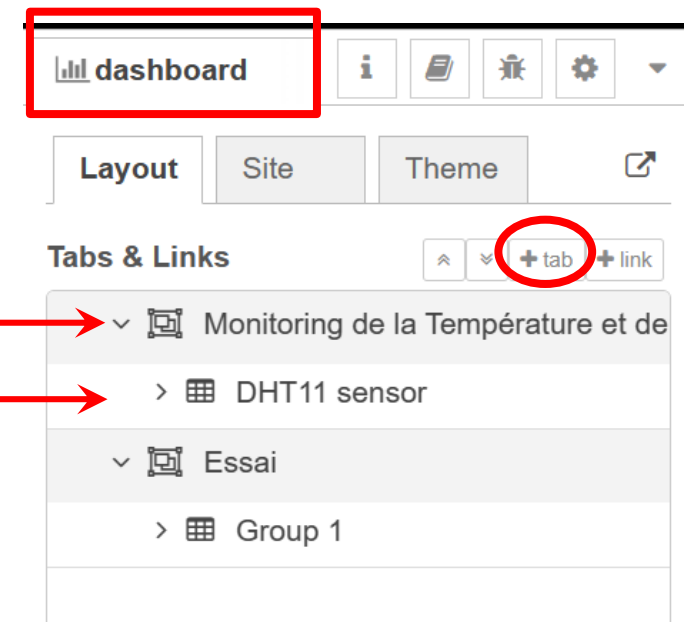
1 Sur l'interface Node-RED, Menu « Dashboard », créer un onglet (Tab) et ajouter un groupe (Group):

- Nom de l'Onglet (page web): « **Monitoring de la Température et de l'Humidité** »
- Nom du Groupe: « **DHT11 Sensor** »

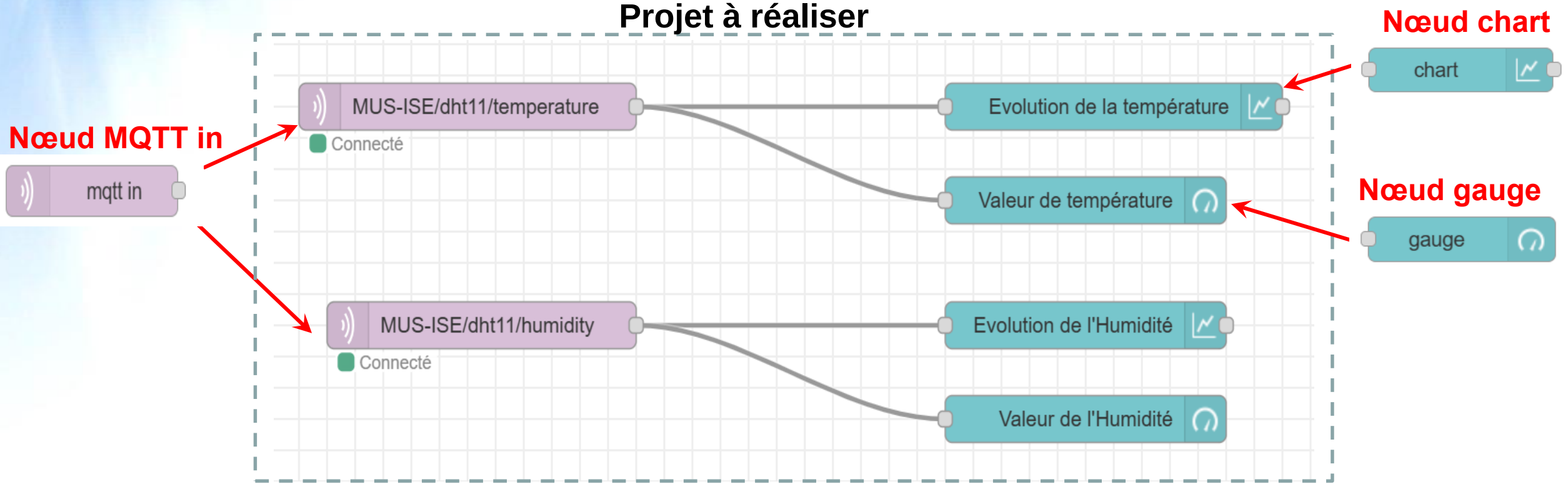
- Les onglets sont des zones dans l'interface Node-RED qui vous permettent d'organiser vos flows en groupes logiques. Chaque onglet peut contenir un ou plusieurs flows et représente généralement une partie spécifique de votre application ou de votre système.
- Les groupes sont des boîtes visuelles dans l'interface Node-RED qui vous permettent de regrouper des nodes (nœuds) liés les uns aux autres. Vous pouvez créer des groupes pour regrouper des nodes qui ont une fonctionnalité similaire ou qui font partie du même processus.

Nom de l'onglet

Nom du Groupe



2 Réaliser un projet node-RED suivant :



3 Configurer les différents nœuds (cliquer deux fois sur chaque nœud):

Nœud MQTT in 

Modifier le noeud mqtt in

Supprimer Annuler Terminer

Propriétés

Serveur test.mosquitto.org:1883

Action S'abonner à un seul sujet

Sujet MUS-ISE/dht11/temperature

QoS 2

Sortie détection automatique

Nom Nom

Indiquer le broker

Topic:
Pour la température: MUS-ISE/dht11/temperature
Pour l'humidité: MUS-ISE/dht11/humidity

Modifier le noeud mqtt in

Supprimer Annuler Terminer

Propriétés

Serveur test.mosquitto.org:1883

Action S'abonner à un seul sujet

Sujet MUS-ISE/dht11/humidity

QoS 2

t JSON analysé, chaîne ou

3 Configurer les différents nœuds :

Nœud chart



Modifier le noeud chart

Supprimer Annuler Terminer

Propriétés

Group [Monitoring de la Température et de l'Humidité] DHT11 sensor

Size [Essai] Group 1

Label Evolution de la température

Type Line chart

X-axis last 1 hours OR 1000 points

Nœud gauge (cas de température)



Modifier le noeud gauge

Supprimer Annuler Terminer

Propriétés

Group [Monitoring de la Température et de l'Humidité] DHT11 sensor

Size 6 x 2

Type Gauge

Label Valeur de température

Value format {{value}}

Units °C

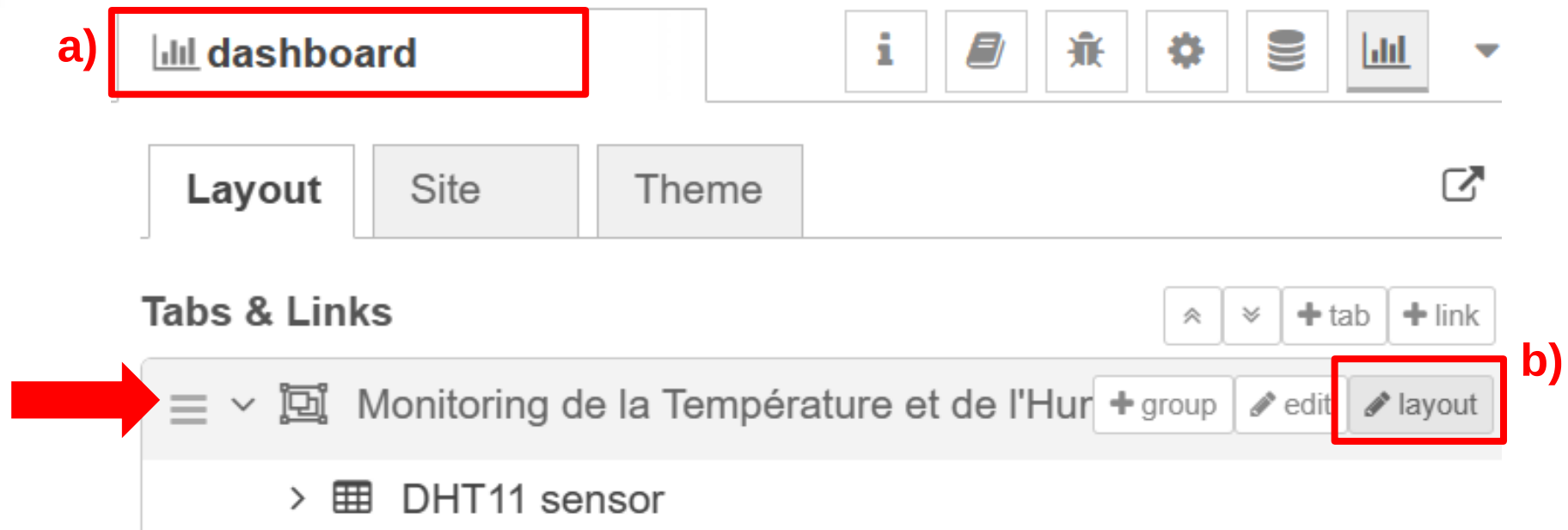
Range min 0 max 50

Colour gradient

Sectors 0 10 30 50

3 Organiser votre Dashboard :

- a) Dans l'onglet « Dashboard » sélectionner l'onglet « Monitoring de la température et de l'Humidité »
- b) Appuyer sur « Layout »



Remarque importante – Dashboard Node-RED

Le Dashboard Node-RED n'est pas installé par défaut.

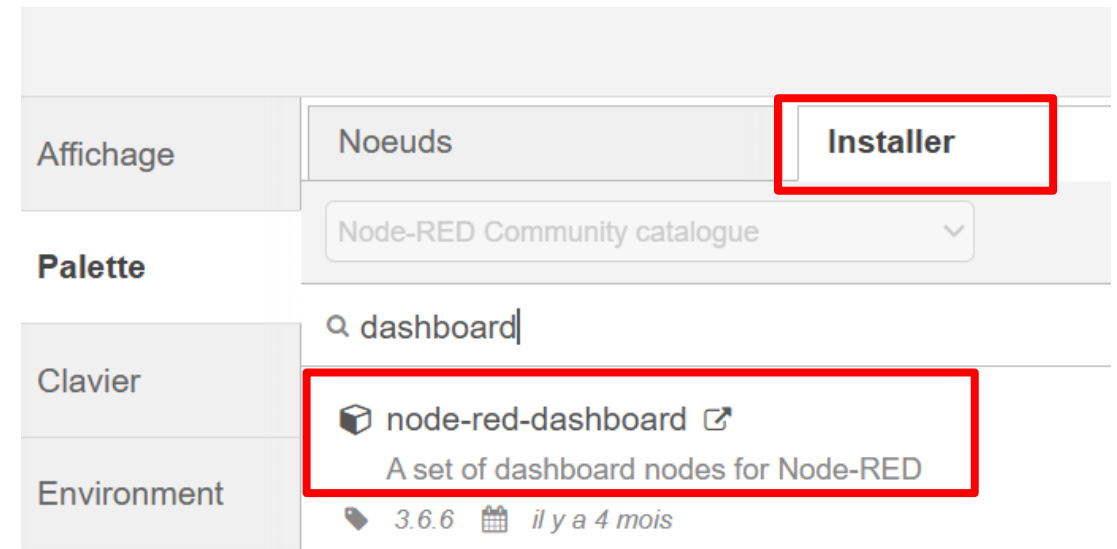
Pour utiliser les interfaces graphiques (boutons, jauges, graphiques, etc.), il est indispensable d'installer le module **node-red-dashboard**.

a) Accéder au gestionnaire de palettes

- Cliquer sur le menu ☰ (en haut à droite)
- Sélectionner **Manage palette**

b) Installer le module Dashboard

- Aller dans l'onglet **Install**
- Rechercher : node-red-dashboard
- Cliquer sur **Install**



c) Organiser l’affichage des éléments (taille et disposition)

Dashboard layout editor : Monitoring de la Température et de l'Humidité

Annuler Terminer

DHT11 sensor

Valeur de température gauge	Evolution de la température chart
Valeur de l'Humidité gauge	Evolution de l'Humidité chart

Width: 21

Vous pouvez ajuster la largeur

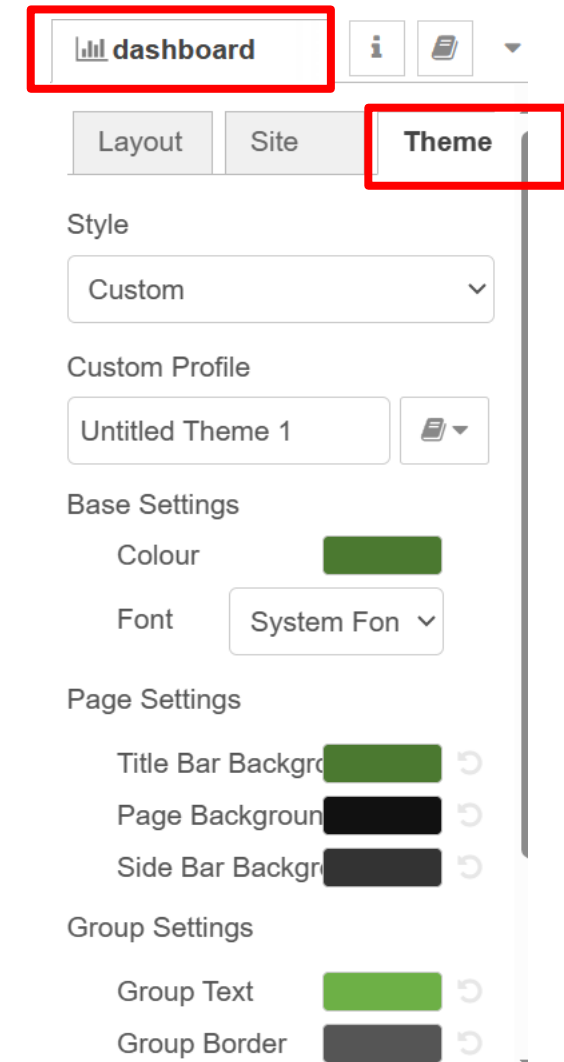
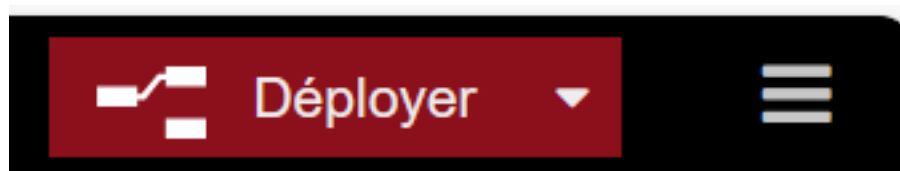
Quand c'est verrouillé, cela permet un redimensionnement manuel

d) Choisir les thèmes de la page:

Dashboard > Theme

- Choix des couleurs du fond du texte, etc
- Style de texte
- Etc.

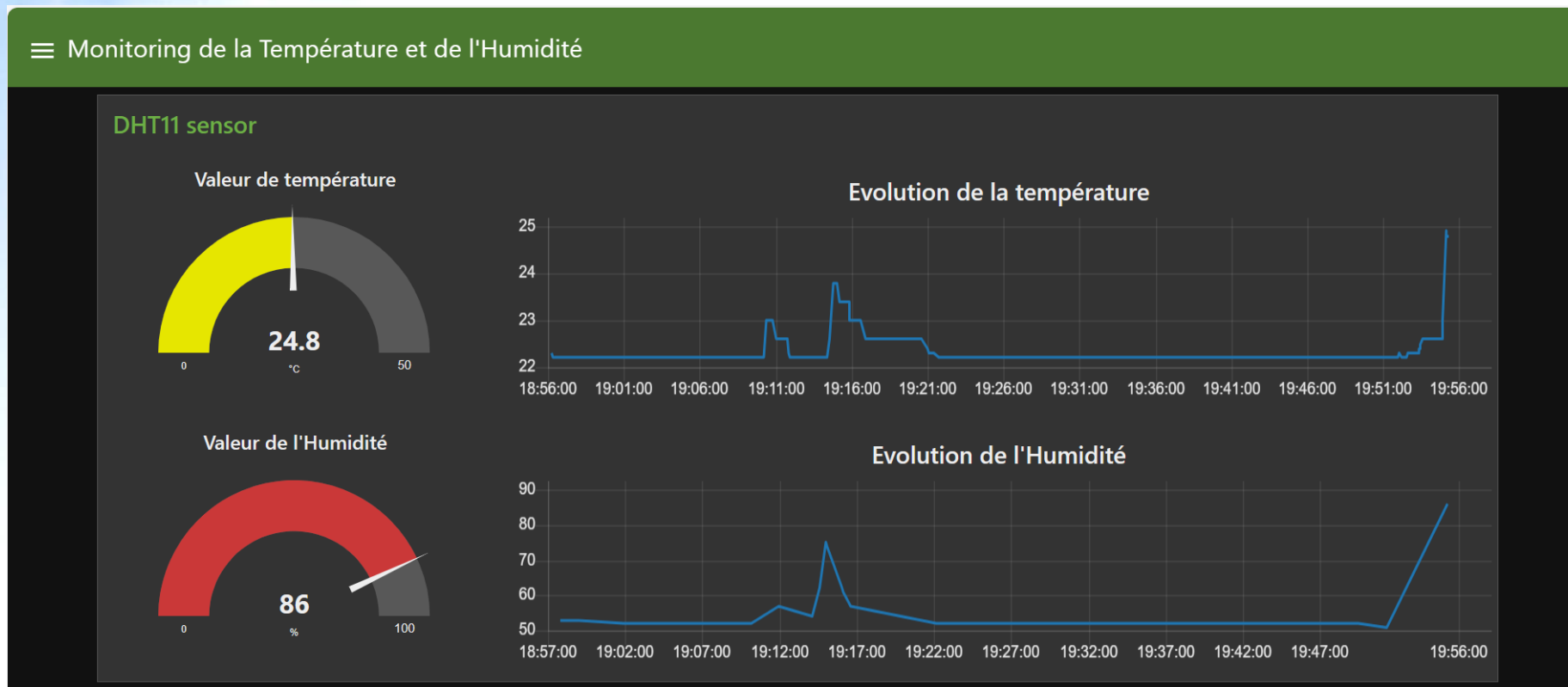
4 Une fois les ajustements réalisés, appuyer enfin sur
« **Déployer** »



4 Pour visualiser le Dashboard réalisé: taper dans le navigateur web:

127.0.0.1:1880/ui

ui: user interface

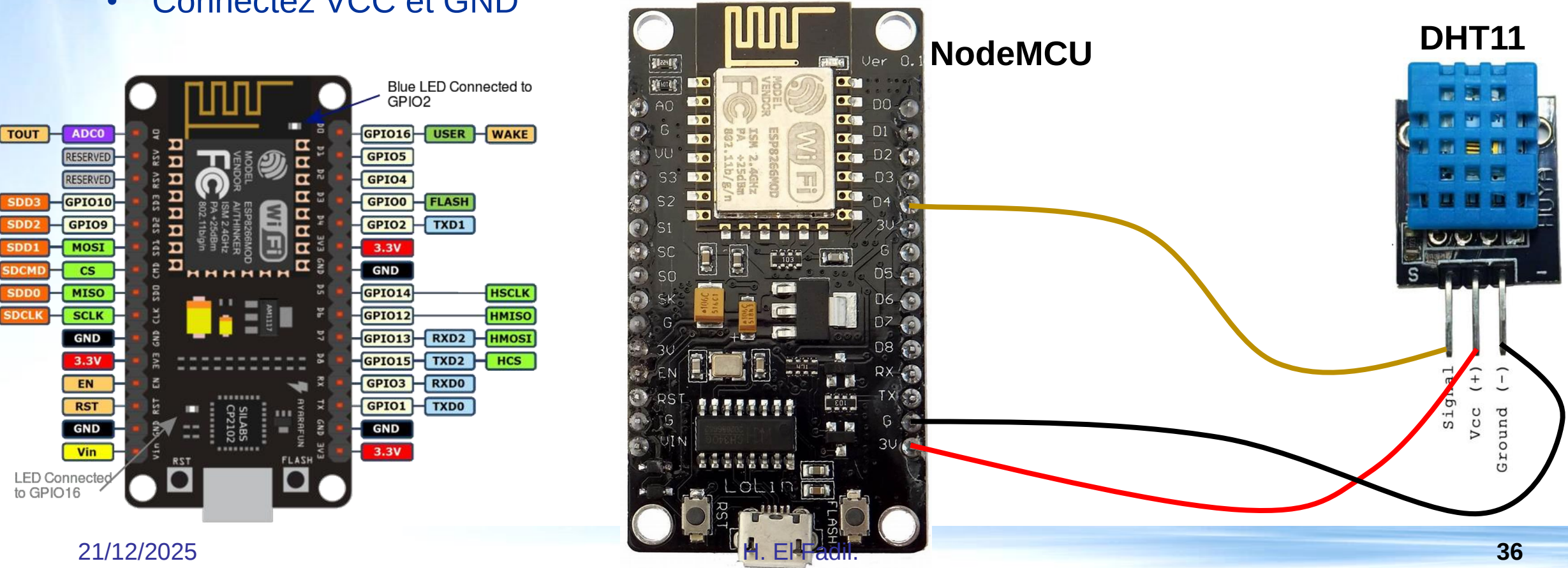


Il ne vous reste maintenant qu'à programmer le NodeMCU pour lire les mesures issues de DHT11 et les envoyer au broker test.mosquitto en utilisant les topics.

3.5. Configuration matérielle et logicielle

1 Câblage: Connectez le DHT11 au NodeMCU.

- Connectez la broche de données du DHT11 (Signal) au GPIO 02 (broche D4) du NodeMCU.
- Connectez VCC et GND



- 2 Câblage: Connecter le NodeMCU avec le PC (vérifier que la carte est bien détectée. Sinon installer ses drives):

Voir Chapitre 4

- 3 Préparation de l'environnement de développement (IDE Arduino):

*Assurez-vous d'avoir les bibliothèques: **ESP8266WiFi**, **WiFiClient** et **DHT**, sont installées dans votre environnement Arduino IDE*

Voir Chapitre 4

4 Code complet Arduino 1/4:

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <WiFiClient.h>
#include <DHT.h>

// Données Wi-Fi;
const char* ssid = "AUTOMATIC_LAB_2G";
const char* password = "ACET-LAB";
// Données Broker
const char* mqttServer = "test.mosquitto.org"; /// 91.121.93.94
const int mqttPort = 1883;
const char* mqttUser = "";
const char* mqttPassword = "";
// Créez une instance pour le capteur DHT11
#define DHTPIN 2 // Broche GPIO02 (D4)
#define DHTTYPE DHT11 // Type de capteur DHT11
DHT dht(DHTPIN, DHTTYPE);

WiFiClient espClient;
PubSubClient client(espClient);
```

4

Code complet Arduino 2/4:

```
void setup() {  
  Serial.begin(9600);  
  delay(10);  
  
  dht.begin();  
  
  WiFi.begin(ssid, password);  
  while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
  }  
  Serial.println("");  
  Serial.println("WiFi connecté");  
  Serial.println("Adresse IP: ");  
  Serial.println(WiFi.localIP());  
  
  client.setServer(mqttServer, mqttPort);  
  // Connexion au Broker  
  while (!client.connected()) {  
    Serial.println("Tentative de connexion MQTT...");  
    if (client.connect("ESP8266Client", mqttUser, mqttPassword)) {  
      Serial.println("Connecté au broker MQTT");  
    } else {  
      Serial.print("Échec, rc=");  
      Serial.print(client.state());  
      Serial.println(" Réessayez dans 5 secondes");  
      delay(5000);  
    }  
  }  
}
```

4

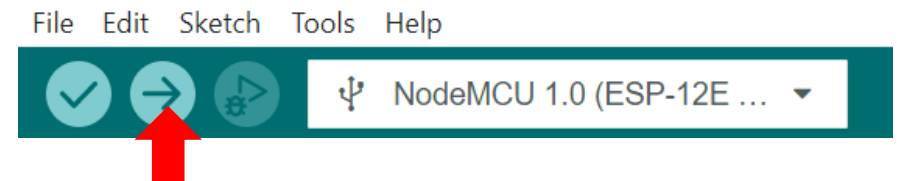
Code complet Arduino 3/4:

```
void loop() {  
  if (!client.connected()) {  
    reconnect();  
  }  
  client.loop();  
  
  float humidity = dht.readHumidity();  
  float temperature = dht.readTemperature();  
  
  if (isnan(humidity) || isnan(temperature)) {  
    Serial.println("Erreur de lecture du capteur DHT");  
    return;  
  }  
  
  Serial.print("Humidité: ");  
  Serial.print(humidity);  
  Serial.print("% - Température: ");  
  Serial.print(temperature);  
  Serial.println("°C");  
  
  char tempString[8];  
  dtostrf(temperature, 1, 2, tempString); // convertir nombre en string  
  char humString[8];  
  dtostrf(humidity, 1, 2, humString); // convertir nombre en string  
  
  client.publish("MUS-ISE/dht11/temperature", tempString); // publier la donnée température  
  client.publish("MUS-ISE/dht11/humidity", humString); // publier la donnée humidité  
  
  delay(1000); // Attendez 1 seconde avant de lire à nouveau le capteur  
}
```

4 Code complet Arduino 4/4:

```
void reconnect() {  
  while (!client.connected()) {  
    Serial.println("Tentative de reconnexion MQTT...");  
    if (client.connect("ESP8266Client", mqttUser, mqttPassword)) {  
      Serial.println("Connecté au broker MQTT");  
    } else {  
      Serial.print("Échec, rc=");  
      Serial.print(client.state());  
      Serial.println(" Réessayez dans 5 secondes");  
      delay(5000);  
    }  
  }  
}
```

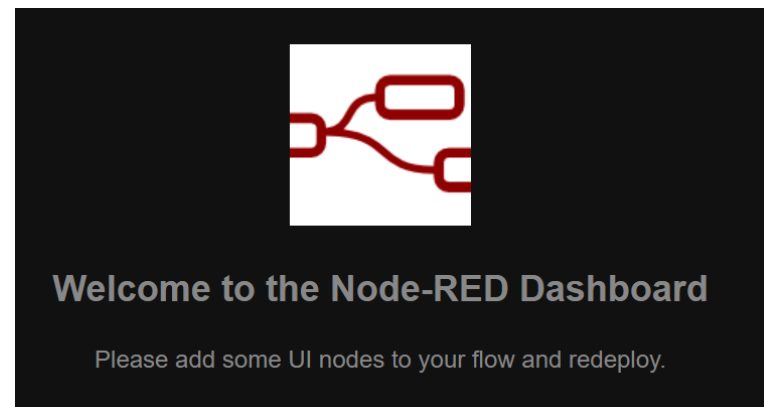
5 La carte étant connectée au port USB du PC, **Téléverser**, alors, le programme (UPLOAD). visualiser de nouveau votre Dashboard Node-RED (observer l'évolution des mesures):

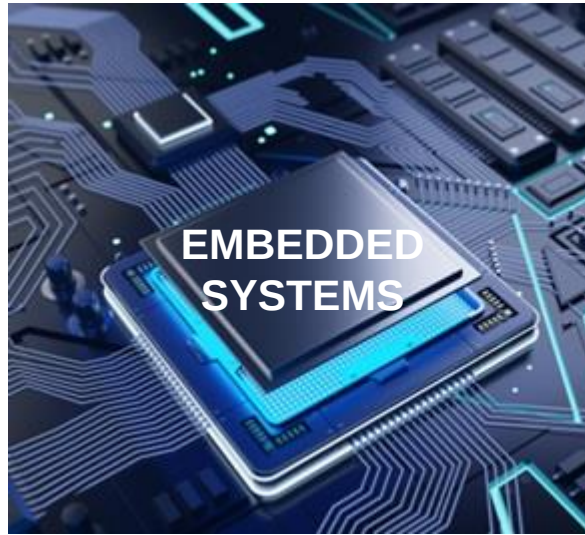


6

Pour arrêter le Serveur Node-RED, appuyer sur **Ctrl-C** dans l'invite de commande:

```
node-red
26 Apr 20:38:26 - [info] Updated flows
26 Apr 20:38:26 - [info] Starting flows
26 Apr 20:38:26 - [info] Started flows
26 Apr 20:38:26 - [info] [mqtt-broker:4d42d185593cd1fb] Connected to broker: mqtt://test.mosquitto.org:1883
26 Apr 21:06:09 - [info] Stopping flows
26 Apr 21:06:09 - [info] [mqtt-broker:4d42d185593cd1fb] Disconnected from broker: mqtt://test.mosquitto.org:1883
26 Apr 21:06:09 - [info] Stopped flows
26 Apr 21:06:09 - [info] Updated flows
26 Apr 21:06:09 - [info] Starting flows
26 Apr 21:06:09 - [info] Started flows
26 Apr 21:06:10 - [info] [mqtt-broker:4d42d185593cd1fb] Connected to broker: mqtt://test.mosquitto.org:1883
```





FIN !
Questions ?