



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Tunis El Manar
École Nationale d'Ingénieurs de Tunis
Département Génie électrique



Projet Réseaux locaux :

Surveillance Météorologique Connectée

Élaboré par :

- Wissal ABID
- Nadine BOUZAIDA
- Maïssa BOUZIRI

Encadré par :

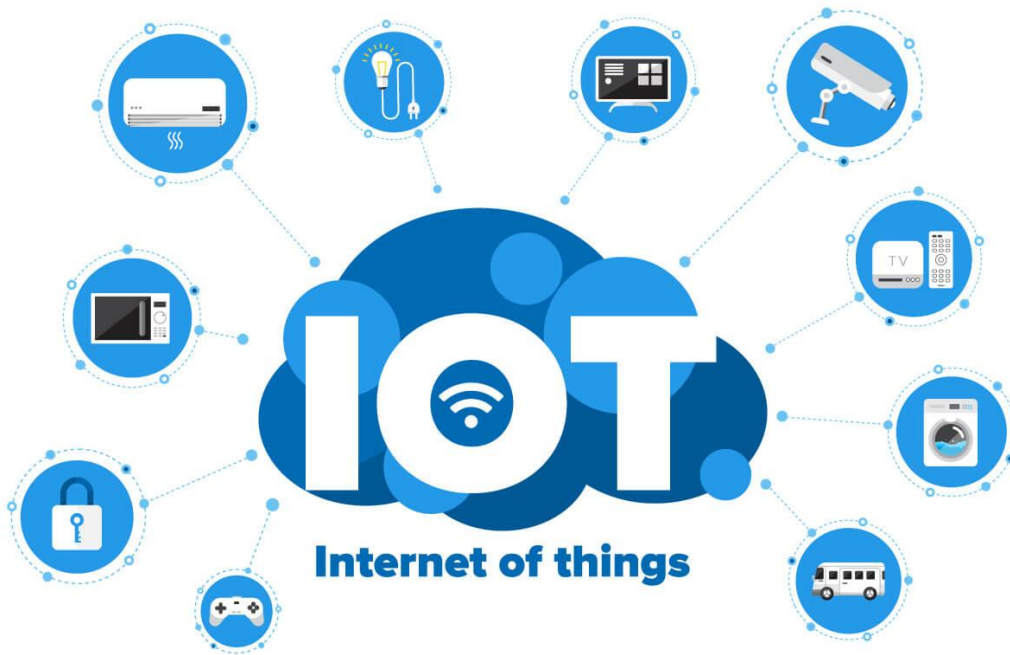
M. Khaled JELASSI

Classe : 3AGE1

Plan

- 1 Introduction
- 2 Présentation du projet
- 3 Présentation de la carte STM32L475 IoT Node
- 4 Environnement de travail
- 5 Travail Réalisé
- 6 Conclusion et Perspective

Introduction



L'Internet des objets industriel (IIoT) se réfère à l'intégration de la technologie IoT dans le domaine industriel, mettant particulièrement l'accent sur la mise en œuvre d'instruments et de systèmes de contrôle des capteurs et des terminaux. Cette intégration s'opère au travers de l'utilisation de technologies cloud, favorisant ainsi la connectivité et l'analyse de données à une échelle industrielle.

Présentation du projet

Projet de Surveillance Météorologique Connectée



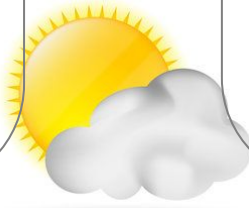
Notre projet de Surveillance Météorologique Connectée vise à révolutionner la façon dont nous collectons et interprétons les données météorologiques essentielles. À l'aide de technologies de pointe, nous avons créé un système intelligent capable de fournir des informations météorologiques précises.

Ce projet offre une approche innovante et technologiquement avancée pour la collecte, la transmission, l'affichage et le stockage des données météorologiques, offrant ainsi une solution complète et fiable pour des applications variées.

Projet de Surveillance Météorologique Connectée

1. Collecte de Données

Au cœur de notre solution se trouve la carte STM32L475 IoT Node, un microcontrôleur robuste avec une connectivité WLAN intégrée. Cela nous permet d'exploiter pleinement le module WiFi ESP8266 pour collecter des données météorologiques, notamment la température, l'humidité et la pression atmosphérique.



3. Affichage des Mesures

La plateforme Node-Red offre une interface conviviale pour visualiser les mesures provenant de divers capteurs. Les utilisateurs peuvent facilement interagir avec les données météorologiques, facilitant ainsi la compréhension des tendances et les variations.

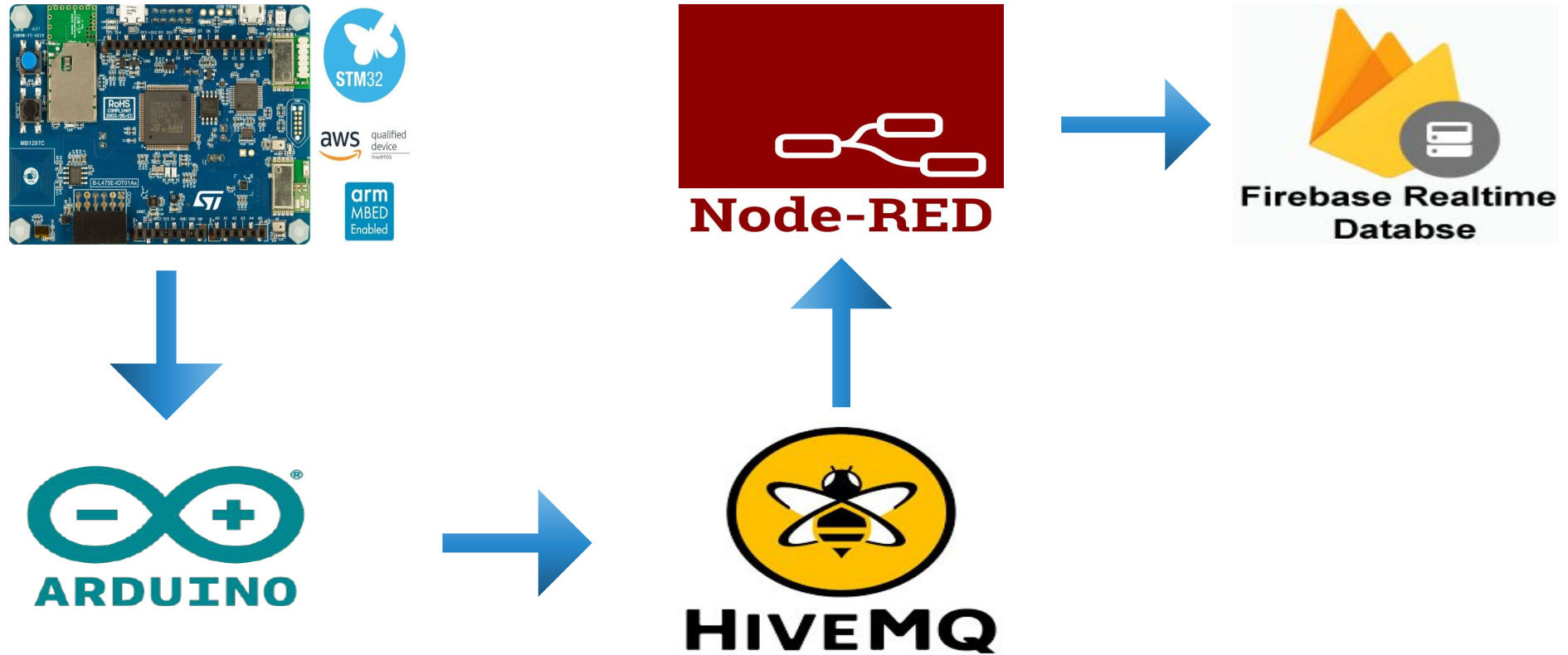
2. Transmission des Données

Nous avons choisi le protocole MQTT pour la transmission efficace des mesures. Cela garantit une communication fiable entre la carte STM32L475 IoT Node et la plateforme Node-Red, où les données sont traitées et prêtes à être visualisées.

4. Stockage des Données

Pour assurer la pérennité des informations météorologiques collectées, toutes les données sont stockées de manière sécurisée et persistante dans la base de données Firebase. Cela permet un accès ultérieur aux données historiques et facilite l'analyse à long terme.

Environnement de travail

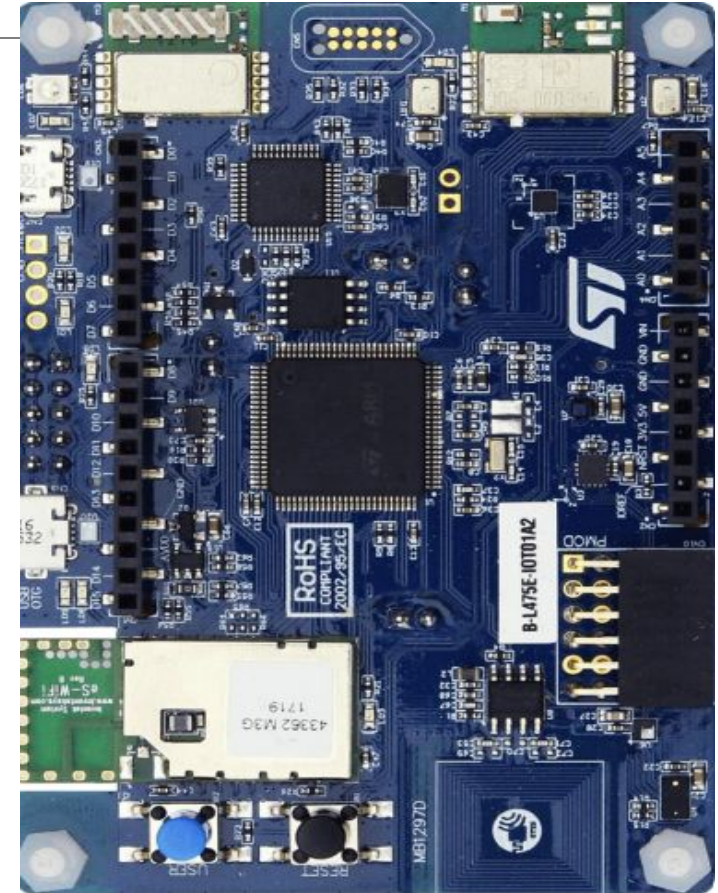


Présentation de la carte STM32L475 IoT Node

La carte STM32L475 IoT Node est un dispositif électronique conçu pour faciliter le développement d'applications IoT (Internet des objets).

Ce microcontrôleur appartient à la famille STM32L4, qui est spécifiquement conçue pour offrir une faible consommation d'énergie tout en fournissant des performances élevées.

Connectivité : La STM32L475 IoT Node est équipée de plusieurs interfaces de connectivité, ce qui en fait un choix idéal pour les applications IoT. Elle peut prendre en charge des protocoles de communication sans fil tels que Bluetooth et Wi-Fi, facilitant ainsi la connectivité avec d'autres dispositifs IoT et les réseaux.



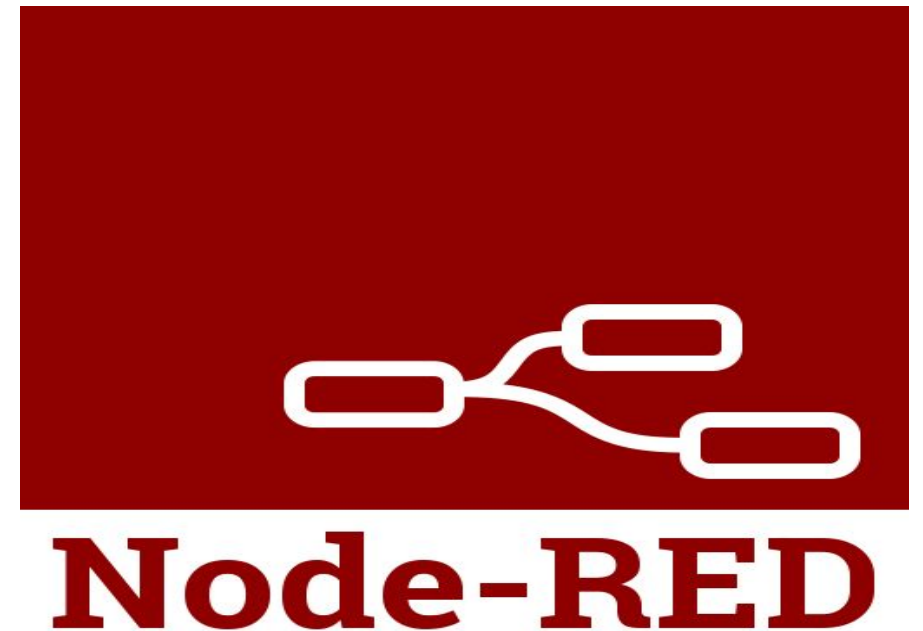
Présentation de HIVEMQ

HiveMQ est une plateforme de messagerie MQTT (Message Queuing Telemetry Transport) conçue pour faciliter la communication entre les appareils dans le contexte de l'Internet des objets (IoT). MQTT est un protocole de messagerie léger et basé sur la publication/abonnement, souvent utilisé pour le transfert de données entre appareils connectés.



Présentation de Node-Red

Node-RED est une plateforme open source de développement visuel basée sur Node.js, permettant la création d'applications par la connexion graphique de blocs fonctionnels appelés "nœuds". Principalement utilisé dans le contexte de l'Internet des objets (IoT), Node-RED simplifie la création d'applications en facilitant l'intégration de différents appareils et services.



Présentation de Firebase

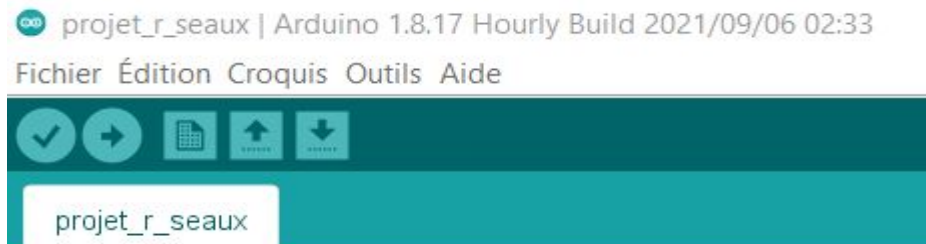
Firebase Realtime Database is a cloud-hosted NoSQL database provided by Google as part of the Firebase platform. It is designed to store and synchronize data in real-time between clients, making it particularly useful for building collaborative and responsive applications.



Travail Réalisé

1. Préparation des environnements de travail:

La configuration de l'IDE Arduino



Installation de Node-red

```
C:\Users\HP>node-red
16 Jan 13:20:43 - [info]

Welcome to Node-RED
=====

16 Jan 13:20:43 - [info] Node-RED version: v3.1.0
16 Jan 13:20:43 - [info] Node.js version: v20.9.0
```

Installation du Broker MQTT Mosquitto

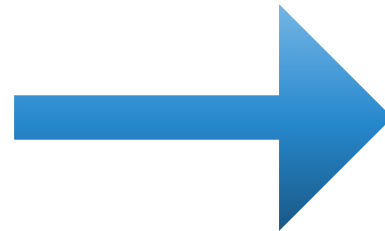


Création d'une base de données dans firebase



2. Développement du code pour la carte IoT et affichage des données des capteurs sur un terminal via l'Arduino IDE.

```
projet_r_seaux $  
#include <stm32l475e_iot01.h>  
#include <stm32l475e_iot01_tsensor.h>  
#include <stm32l475e_iot01_hsensor.h>  
#include <stm32l475e_iot01_psensor.h>  
#include <SPI.h>  
#include <WiFiST.h>  
#include <WiFiUdpST.h>  
#include <MQTTClient.h>  
unsigned long lastMillis = 0;  
float sensor_value_T = 0;  
float sensor_value_H = 0;  
float sensor_value_P = 0;  
String string_MQTT;  
int16_t topic=0;  
SPIClass SPI_3(PC12, PC11, PC10);  
WiFiClass WiFi(&SPI_3, PE0, PE1, PE8, PB13);  
int status = WL_IDLE_STATUS;  
char ssid[] = "Ooredoo_M30_83E8";  
char pass[] = "585266ED";  
WiFiClient net;  
MQTTClient client;  
void setup() {  
    BSP_TSENSOR_Init();// temperature  
    BSP_HSENSOR_Init();// Humidité  
    BSP_PSENSOR_Init();// Pression
```

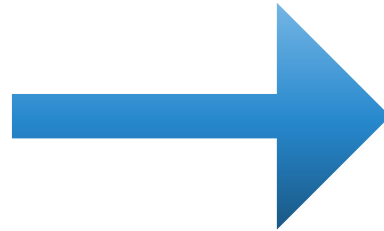


```
14:08:20.736 -> *****TEMPERATURE***  
14:08:20.736 -> TEMPERATURE = 25.50  
14:08:20.776 -> *****  
14:08:20.776 -> *****HUMIDITY*****  
14:08:20.776 -> Humidite = 58.08  
14:08:20.776 -> *****  
14:08:20.776 -> *****PRESSURE*****  
14:08:20.776 -> Pression = 1006.25  
14:08:20.776 -> *****  
14:08:21.776 -> *****TEMPERATURE***  
14:08:21.776 -> TEMPERATURE = 25.49  
14:08:21.776 -> *****  
14:08:21.776 -> *****HUMIDITY*****  
14:08:21.776 -> Humidite = 58.08  
14:08:21.776 -> *****  
14:08:21.776 -> *****PRESSURE*****  
14:08:21.776 -> Pression = 1006.24  
14:08:21.776 -> *****
```

3. Transmission des données de température, pression et humidité vers un serveur cloud, avec les points d'accès "/température", "/pression" et "/humidite", en utilisant "broker.hivemq.com"

```
client.begin("broker.hivemq.com", net);
client.onMessage(messageReceived);
Serial.print("\nconnecting...");

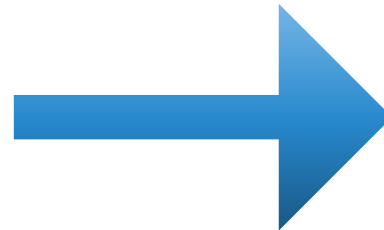
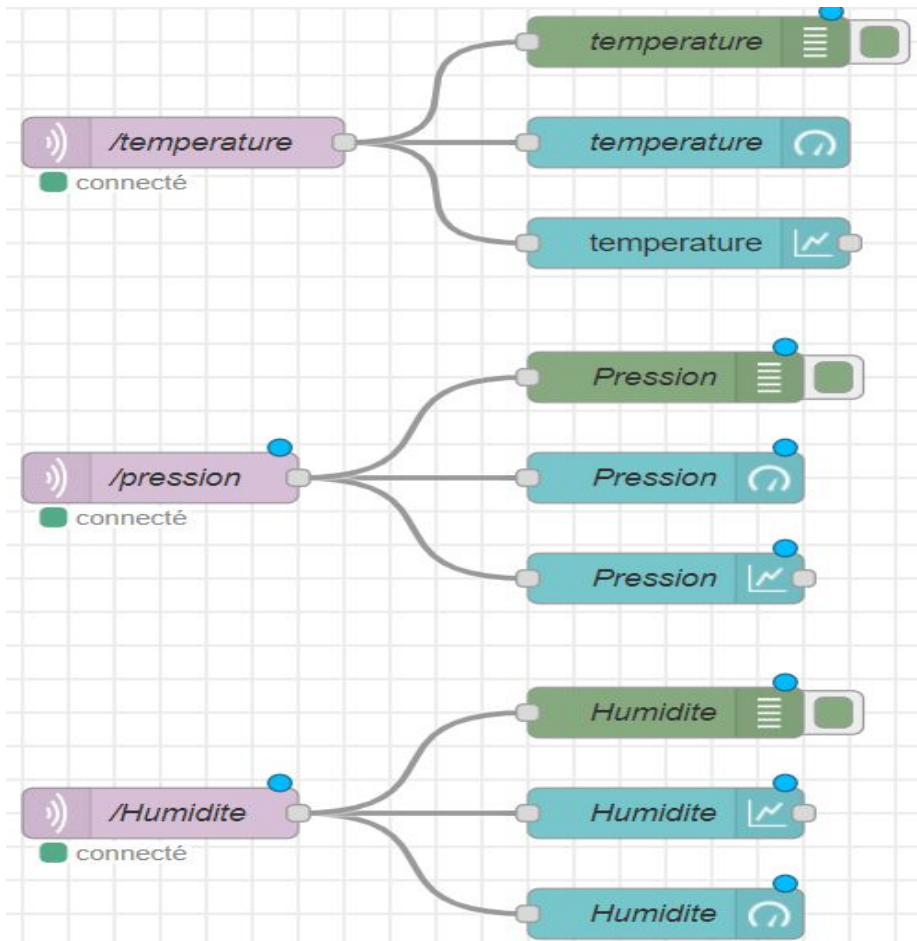
while (!client.connect("Nedin", "Nedin bouzaida", "xx"))
{
  Serial.print(".");
  delay(1000);
}
Serial.println("\n connected!");
```



```
SSID: Ooredoo _M30_83E8
IP Address: 192.168.0.148
signal strength (RSSI):-25 dBm

connecting...
connected!
```


4. Développement d'une interface graphique avec Node-RED pour la lecture des sujets MQTT et leur visualisation en mode débogage.



```
16/01/2024 14:28:26  noeud: temperature
/temperature : msg.payload : number
25.9

16/01/2024 14:28:26  noeud: Humidite
/Humidite : msg.payload : number
57.8

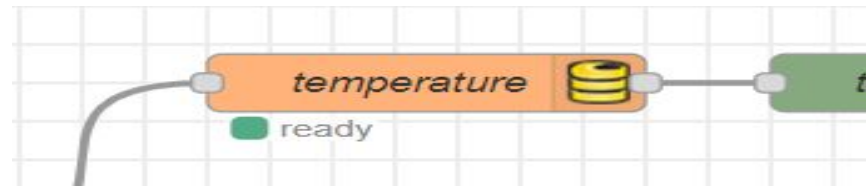
16/01/2024 14:28:27  noeud: Pression
/Pression : msg.payload : number
1006.1

16/01/2024 14:28:28  noeud: temperature
/temperature : msg.payload : number
25.9

16/01/2024 14:28:29  noeud: Humidite
/Humidite : msg.payload : number
57.8

16/01/2024 14:28:30  noeud: Pression
/Pression : msg.payload : number
1006
```


5. L'intégration d'une base de données pour stocker de manière archivée les valeurs.



Modifier le noeud firebase modify

Supprimer Annuler Terminer

Propriétés

Firebase

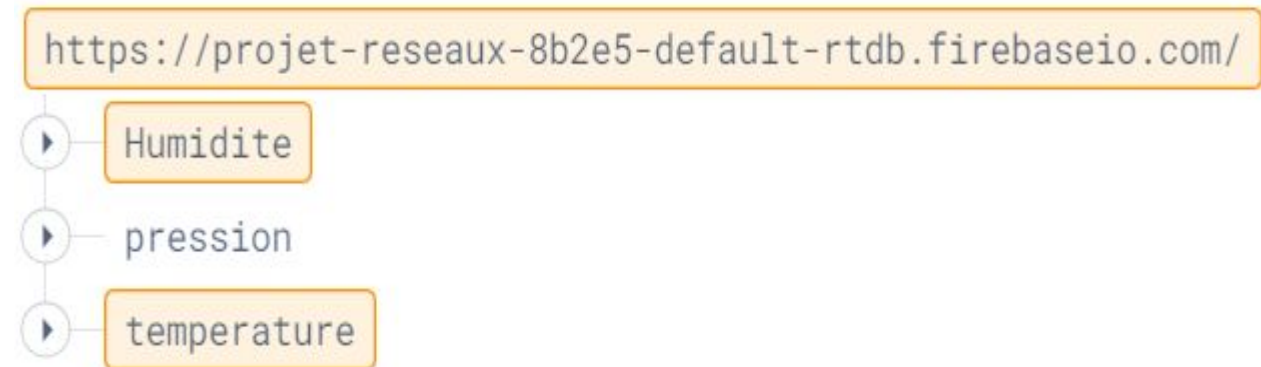
Child Path

>_ Method

Value

Name

<https://projet-reseaux-8b2e5-default-rtdb.firebaseio.com>



https://projet-reseaux-8b2e5-default-

▼ Humidite

-NoHTtieN1RS6gbXd2Ka: 57.8
-NoHTuLgvIJyFLlgzZxJ: 57.9
-NoHTuvYkH31nd1pAoUV: 57.8
-NoHTvZF0R4Zi7b1Pp2b: 57.8
-NoHTwChPesmF4vbQWl6: 57.7
-NoHTw1lbvRVgNnzYBBE: 57.7
-NoHTxNrjCxa3yWJW4am: 57.8
-NoHTy3TfYWSUjPzymK0: 57.8
-NoHTyeZTQLKwEBZMisy: 57.8

https://projet-reseaux-8b2e5-default

▶ Humidite

▼ pression

-NoHTtqA4HfHq0ywUh1i: 1006.1
-NoHTuShBmUr1r6TcLjI: 1006.1
-NoHTv6UFupUoJvGoTD1: 1006
-NoHTviTCNYa_ayFGAsV: 1006.1
-NoHTwJisF6FjX0AECt1: 1006
-NoHTwxMQ5pxr2F5VsiR: 1006
-NoHTxawZnoYPGu4wU9a: 1006
-NoHTyBfhMwHXLq8xQFs: 1006.1

https://projet-reseaux-8b2e5-defaul

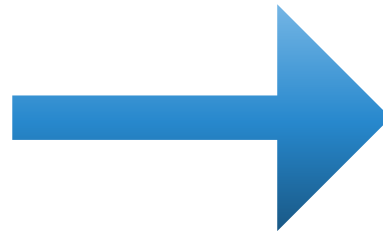
▶ Humidite

▶ pression

▼ temperature

-NoHTYEc5_R39oaqbHZk: 25.6
-NoHTYrX40ohJ55WYBLC: 25.6
-NoHTZVBuhgS6ydydMzc: 25.6
-NoHT_9LkIxoVVRXE87R: 25.6
-NoHT_i1AAkefLpjsYL0: 25.6
-NoHTaPKZ2H6v9NWCXfE: 25.6
-NoHTaxN9FsPkuL_ndTp: 25.6

6. Réalisation d'un Dashboard complet personnalisé permettant un suivi en temps réel de la température, l'humidité et la pression (Modification du template du Dashboard en utilisant le node « Edit Template »):



Propriétés

Template type: Widget in group

Group: [Weather] Group

Size: 7 x 8

Class: Optional CSS class name(s) for widget

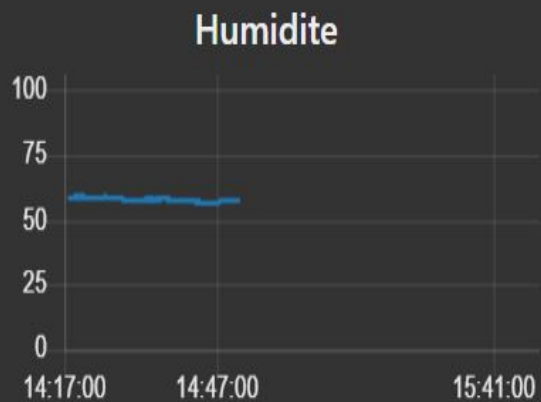
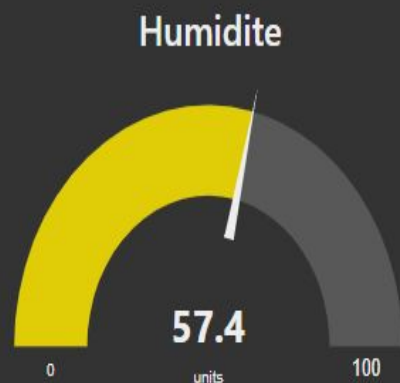
Nom: Nom

Template

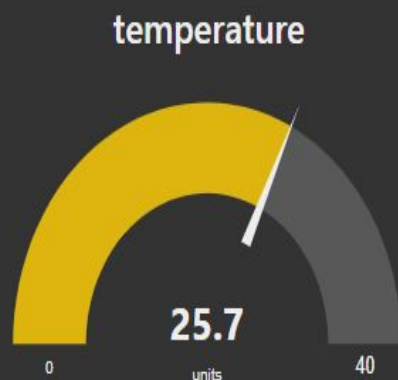
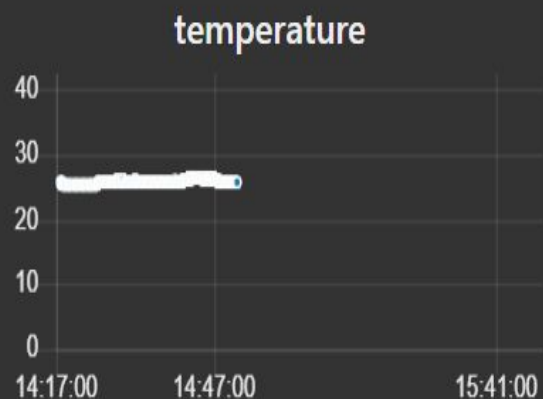
```
1 <div>
2   <h2>Travail réalisé par :</h2></p>
4     Nadine BOUZAIDA</p>
5     Wissal ABID</p>
6     Maissa BOUZIRI</p>
7 </div>
8
```

Weather

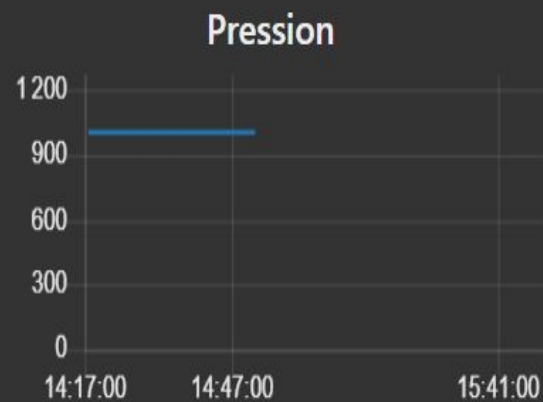
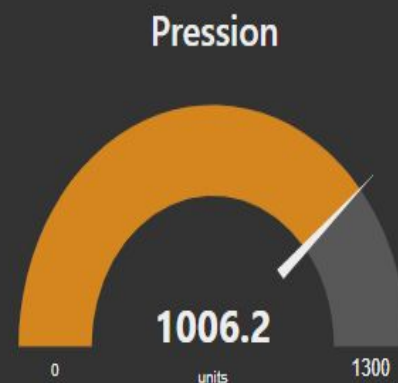
Humidite



temperature



Pression



Travail réalisé par :

Nadine BOUZAIDA

Wissal ABID

Maissa BOUZIRI