 **Université Abdelmalek Essaadi**

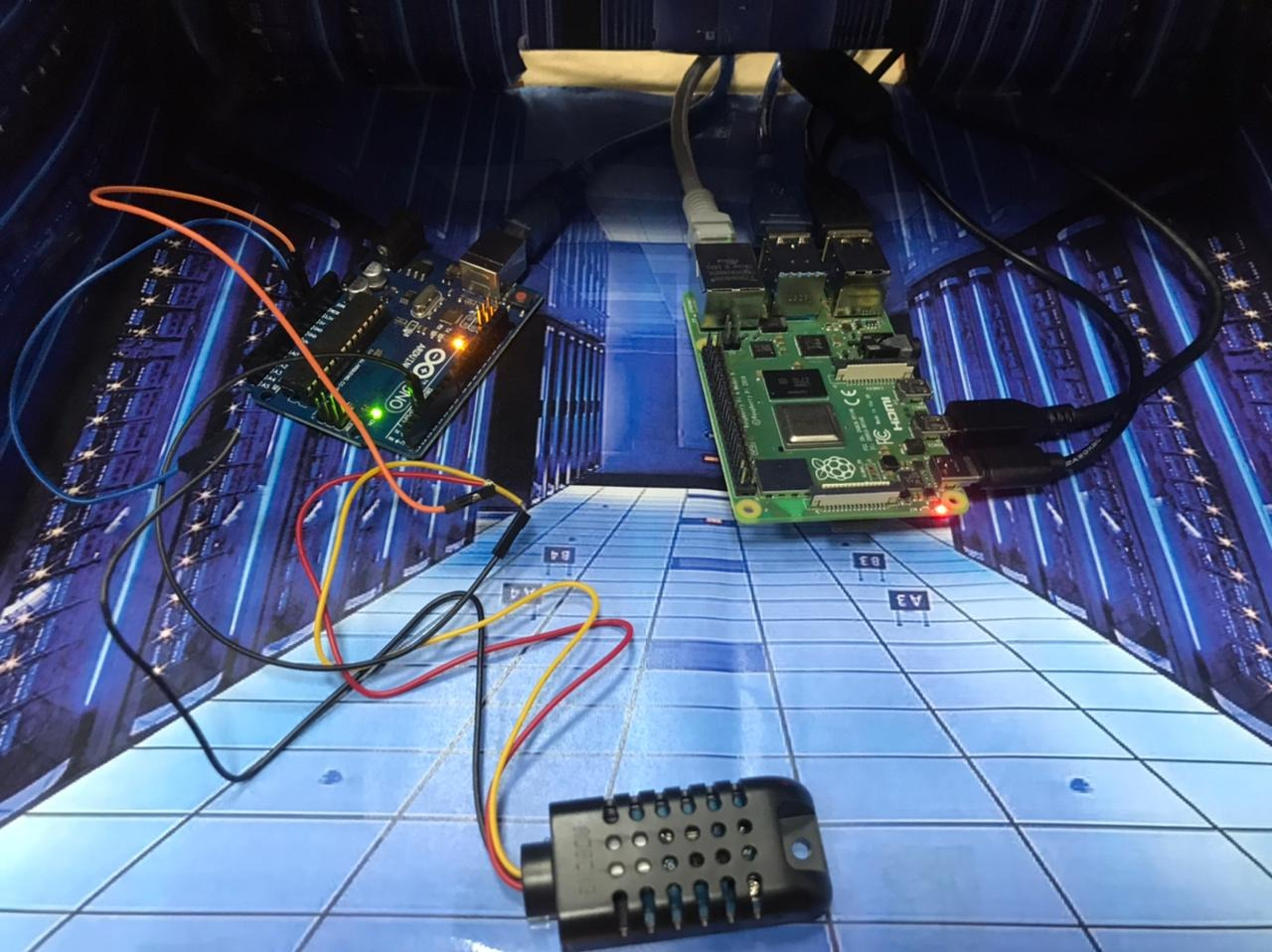
**Faculté des sciences et techniques Tanger**

**Logiciels & Systèmes Intelligents**

**Rapport de projet de fin de module**

**Systèmes embarqués & Internet des objets**.

Système de supervision de température et d’humidité

****

**Encadré par : Réalisé par groupe 2.4 :**

**Anouar Abdelhakim BOUDHIR Yassine EL KASMY**

**Ahmed MELLOUK**

**Nora EL YOUSSEFI**

**Lamiae MOUDDENE**

Remerciements

Avant de commencer la présentation de ce travail, nous profitons de l'occasion pour remercier ALLAH pour la volonté, la force et la santé qu’il nous a donné afin de réaliser ce travail. Au terme de ce projet, nous tenons À remercier toute personne ayant contribué, de près ou de loin, À l'aboutissement de ce travail, en particulier : Nos respectueuses gratitudes À notre cher professeur Monsieur BOUDHIR pour nous avoir donné le goût de l'Apprentissage et d'avoir partagé avec nous sa passion pour l'enseignement. Nous avons grandement apprécié son implication et sa qualité d’enseignement dont nous avons bénéficié tout au long du semestre. C’est grâce à lui que nous avons acquis de nouvelles compétences indispensables pour la réalisation de ce projet. Merci à nos camarades de classe pour avoir partagé avec nous leurs connaissances.

Contents

[Introduction 4](#_Toc96722317)

[Objectif 5](#_Toc96722318)

[Cas d’utilisations : 5](#_Toc96722319)

[La température idéale de la salle 6](#_Toc96722320)

[L'humidité idéale de la salle 6](#_Toc96722321)

[Composant électroniques utilisés : 7](#_Toc96722322)

[Technologies utilisés : 10](#_Toc96722323)

[schéma : 12](#_Toc96722324)

[Code source: 15](#_Toc96722325)

[Collectes des informations par les capteurs liés au arduino: 15](#_Toc96722326)

[Upload Des données captées à une plateforme cloud: 16](#_Toc96722327)

[Application web: 18](#_Toc96722328)

[L’architecture du projet: 20](#_Toc96722329)

[app.component.html: 21](#_Toc96722330)

[app.component.ts: 23](#_Toc96722331)

[app.module.ts: 28](#_Toc96722332)

[gauge.component.html: 28](#_Toc96722333)

[gauge.component.ts: 30](#_Toc96722334)

[Conclusion: 31](#_Toc96722335)

# Introduction

Dans toutes les disciplines commerciales, industrielles et scientifiques, le réglage parfait de la température et humidité a été toujours un des facteurs crucial pour le succès d’un produit, experiment et projets.

C’est pour cela que, dans le cadre de nos études, et afin d'améliorer nos connaissance dans le domaine de l’IOT, on a viser de réaliser un système de supervision de température et d’humidité qui sera localisé dans un “Server Room”.

Ce système sera capable de détecter les valeurs de la température et d’humidité en temps réel et envoyer des notifications au manager du serveur room au cas où les valeurs captées dépassent la limite assurant des conditions favorables pour le fonctionnement des serveurs.

L’objectif de ce rapport est de clarifier le fonctionnement de ce système et son architecture, les composants électroniques utilisés et ses cas d’utilisations.

# Objectif

L’objectif principale de ce projet est d’utiliser le cloud Firebase pour collecter les valeurs de l'humidité et de température

dans un “server room” à l'aide des capteur de température et d’humidité puis les données détectées par les capteurs sont contrôlées et collectées par l'Arduino UNO avec microcontrôleur Raspberry Pi 4, cette dernière est le cœur de ce système c'est comme un serveur des données et d'application.

A propos les résultats sont affichés et analysés textuellement et graphiquement, Elle affichera non seulement les données actuelles, mais également les valeurs passées sous forme de graphiques via une plateforme web responsive.

# Cas d’utilisations :

* **Spa et piscine**
* **le domaine d’agriculture**
* **Domaine industriel**
* **Laboratoire chimique**
* **Salle des serveurs**

Dans ce projet ,on va prendre comme cadre d'étude” la salle des serveurs” ,un exemple plus proche de notre parcours universitaire .

Alors l'un des critères qu'il est important de contrôler pour assurer la sécurité des data centers est la gestion de la température ambiante et de l'humidité dans la salle de serveurs ou les locaux informatiques.

Pour éviter que les serveurs surchauffent, ils doivent être situés dans un local où la température et l'humidité peuvent être maîtrisée, été comme hiver.

Pour éloigner d'éventuelles pertes de données importantes et des dysfonctionnements, les serveurs informatiques doivent être situés dans un local où la température est maîtrisée.

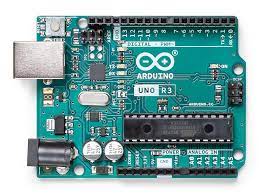
La température idéale de la salle

Afin d'établir au mieux la température ambiante en différents points de la salle, il est généralement procédé à l'installation de sondes thermiques fonctionnant en temps réel. Il est ensuite nécessaire de s'intéresser aux températures habituellement pratiquées et conseillées. La température communément admise pour une salle de serveurs oscille entre 18 et 27°C, avec une sorte de consensus autour de 23°C. En fait, cela dépend de la qualité et de la récence du matériel utilisé (plus il est ancien, moins il supporte les températures élevées).

L'humidité idéale de la salle

Comme souvent, les dangers de l'humidité sont beaucoup moins connus : en effet, une humidité relative basse d'environ 35% est dangereuse car elle est source de décharge électrostatique. Ces décharges passent généralement inaperçues car elles sont peu apparentes, mais elles peuvent provoquer des dégâts bien souvent non couverts par les garanties du fabricant, comme par exemple, la décomposition du plastique des équipements entraînant un vieillissement accéléré du matériel. Inversement, une humidité trop élevée s'accompagne de condensation, et comme on peut s'en douter, la présence d'eau dans les périphériques est néfaste (rouille et dégradation de la propreté). En définitive, le taux d'humidité idéale d'une salle serveur se situe aux environs des 50% avec une tolérance entre 40 et 60%.

# Composant électroniques utilisés :



arduino UNO

L’Arduino Uno est un microcontrôleur programmable qui permet de contrôler des éléments mécaniques : systèmes, lumières, moteurs, etc.



Raspberry pi 4

Le Raspberry Pi 4 est un nano-ordinateur qui se présente sous la forme d’une carte de processeur. Il peut se brancher directement sur un écran ou un moniteur.

très utiles pour créer des serveurs, serveur web domestique, gérer une imprimante 3D.



capteur de température et d’humidité

**Capteur DHT21** (maintenant appelé AM2301 bien que DHT21 soit toujours largement utilisé) intégré dans un boîtier pré-câblé pour détecter la température et le taux d'humidité, sur un circuit pour Arduino ou Raspberry.. Il offre un rang de détection entre -40°C et +80°C et un range en humidité situé entre 0 et 99,99%.

L'interfaçage série du capteur AM2301/DHT21 utilise seulement 3 câbles, [(-), (+) et output (signal)] rend l'utilisation de ce module facile.

Le câble noir est le négatif, le rouge est le positif et le jaune le signal.

**informations supplémentaire:**

Tension d'alimentation : 3.3 à 5,5V DC

Connectivité : signal digital Bus (3 broches)

Précision : Humidité + -3%, Température + - 0,5 ℃

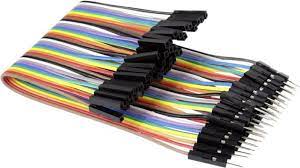
Résolution : Humidité 0.1%, Température 0,1 ℃

Dimensions : 59,2 mm x 27 mm x 13,5 mm

Ecran LCD



cable Ethernet



câbles de liaison



cable HDMI

# Technologies utilisés :

Arduino (IDE) facilite l'écriture de code et son téléchargement sur la carte. Ce logiciel peut être utilisé avec n'importe quelle carte Arduino.



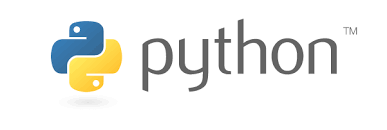
Arduino IDE

Firebase est une plate-forme d'applications mobiles avec des bibliothèques clientes intégrées et unifiées dans divers langages de programmation. Les différentes fonctionnalités backend-as-a-service (BaaS) de Firebase vous aident à développer des applications de haute qualité, à développer votre base d'utilisateurs et à gagner plus d'argent.



Firebase cloud

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions.



Python

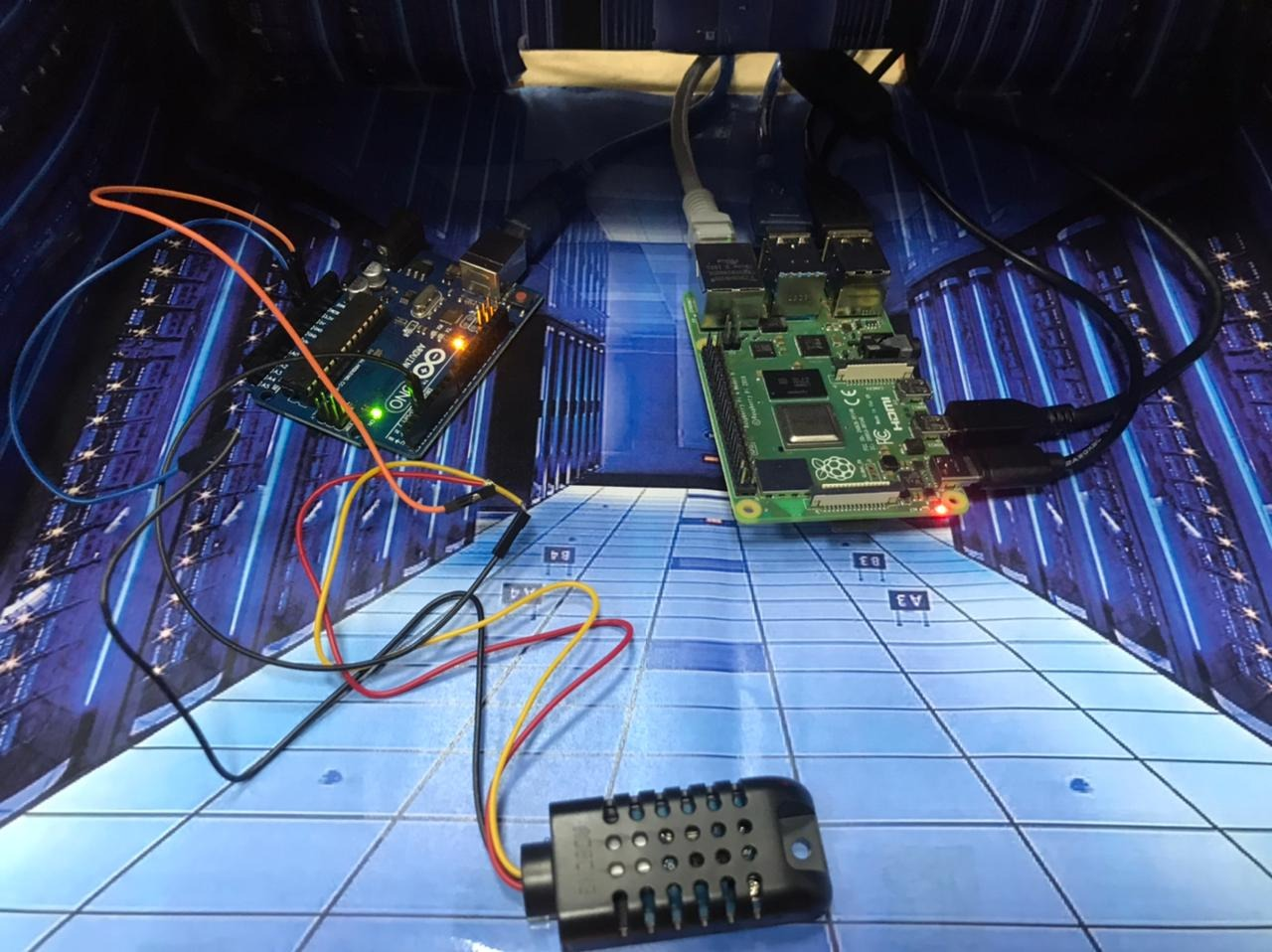
Arduino possède sa propre langage de programmation qui est similaire au javascript, et contient deux fonctions de plus, setup() ou on initialise les pins et loop() ou le code s'exécute d’une façon itérative.



Langage arduino

# schéma :

Voiçi notre montage qui se compose d’un capteur de température et d’humidité qui envoie les données vers l’arduino qui est responsable de stocker ces données dans le cloud firebase et puis les récupérer et les afficher dans la plateforme web

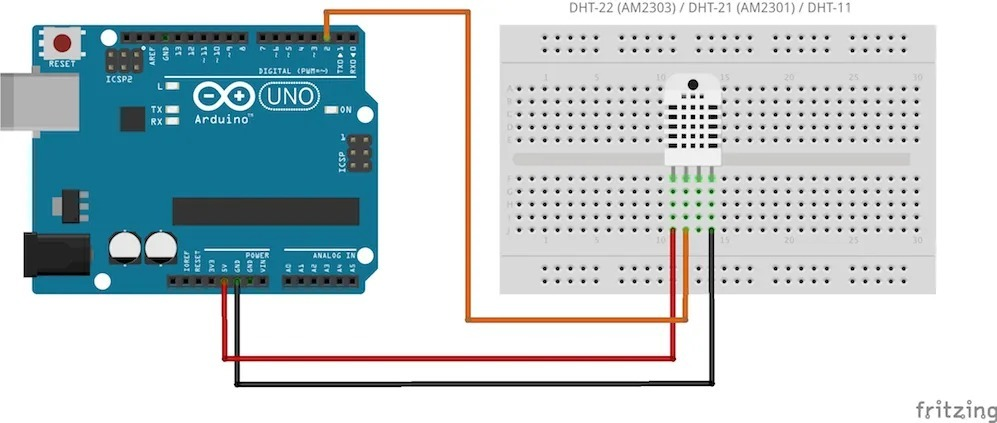
****

* les 3 câbles du capteur sont liés avec les pins d’arduino comme suit:

cable negatif (noir) : GND

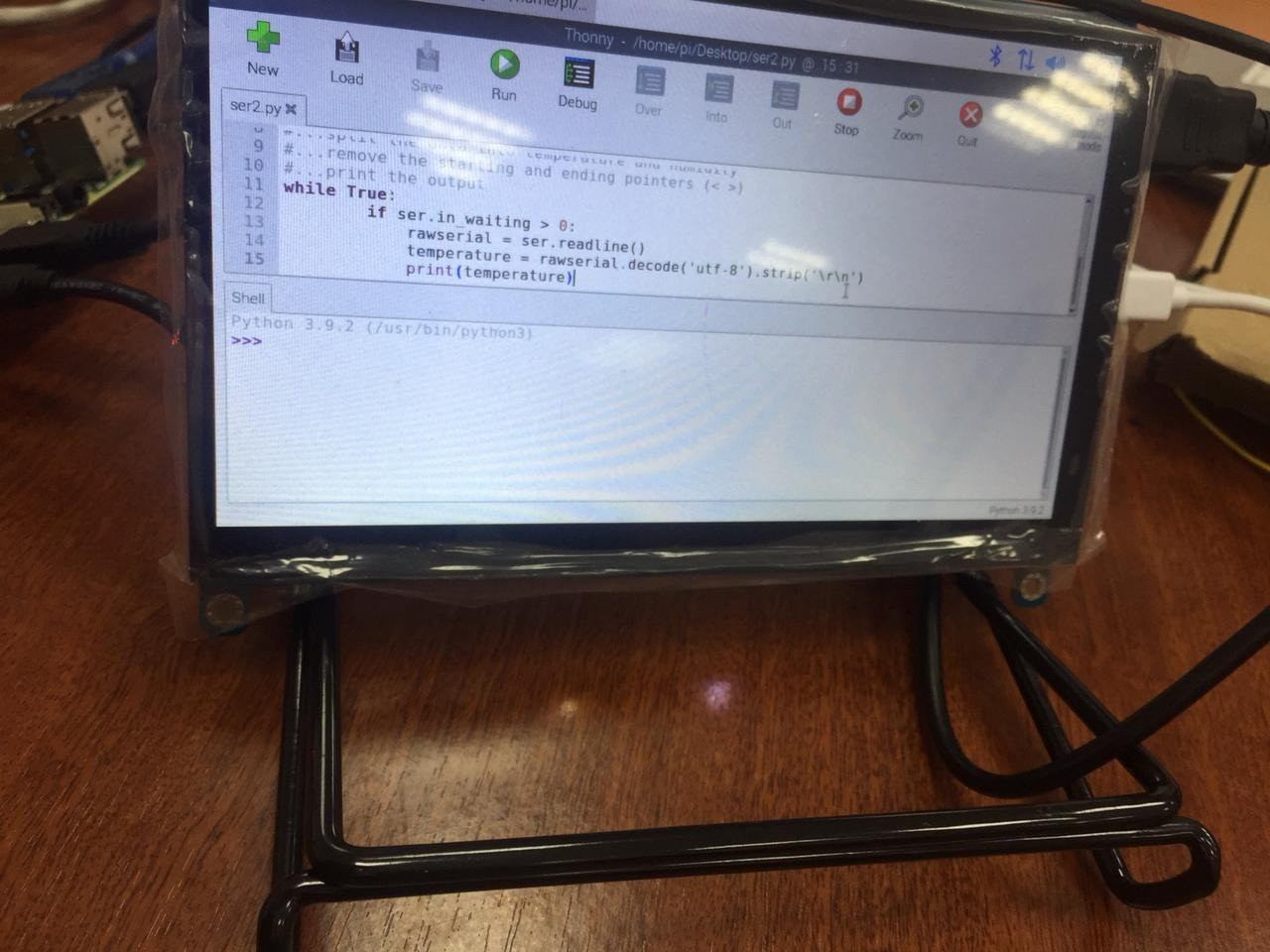
cable positif(rouge) : 5V

cable output (jaune) : 2V



* le raspberry Pi est lié avec:

**l’arduino** via le port USB pour garantir une communication série entre eux



**les périphérique d’entré/sortie** comme le clavier,la souris et l'écran LCD, afin d'exécuter le scripte python qui permet d'accueillir les données à partir du capteur et les envoyés vers le cloud



**cable Ethernet** pour garantir une connexion internet nécessaire pour le stockage des données dans le cloud

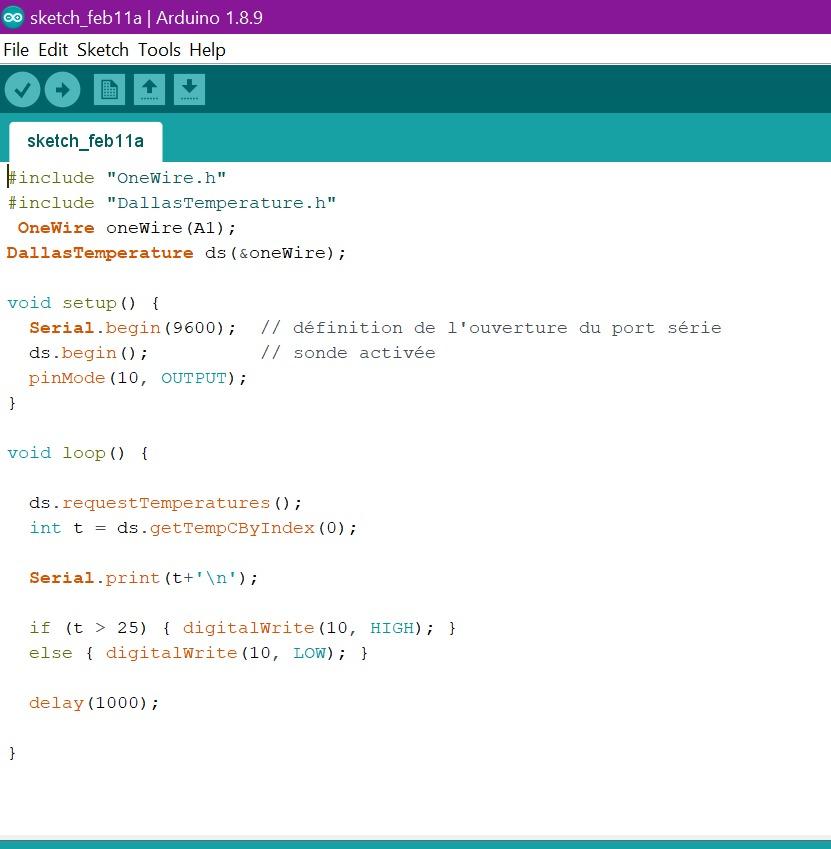
# Code source:

Dans cette partie on se focalise sur le côté fonctionnement de notre système et les différents code et script qui sert à les réaliser, les étapes par lesquelles passe notre information pour qu’elles soient transmises du capteur à l' application web pour la visualisation.

Ces étapes Sont:

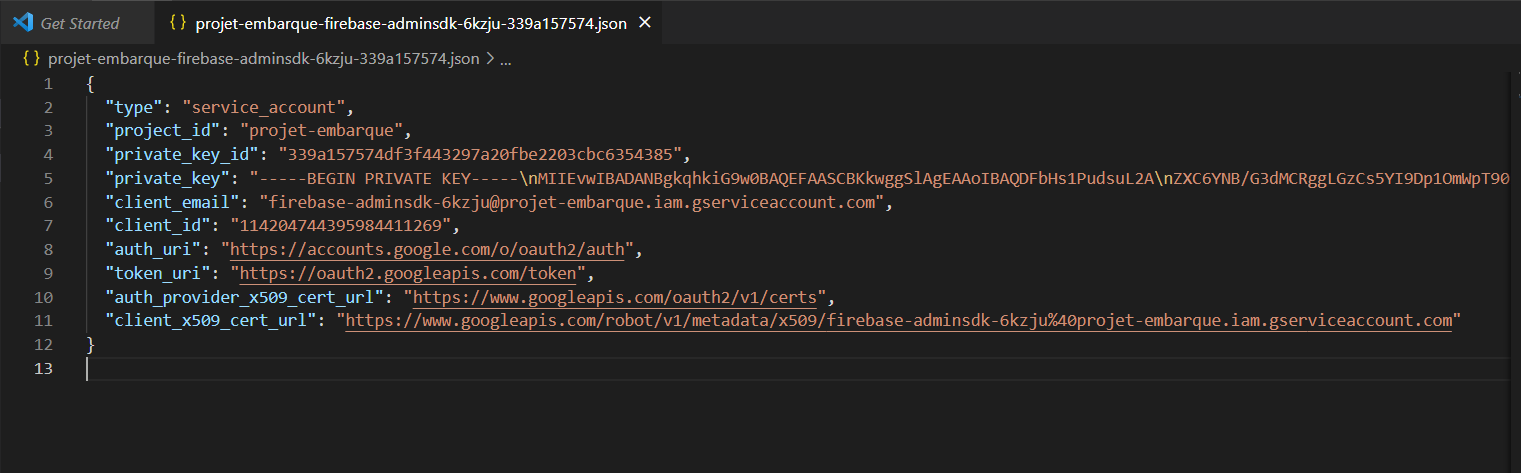
Collectes des informations par les capteurs liés au arduino:

Afin d’utiliser le capteur de température et humidité on doit importer les librairies “OneWire” et “DallasTemperature” qui nous les fonctions capables de capter les valeurs.

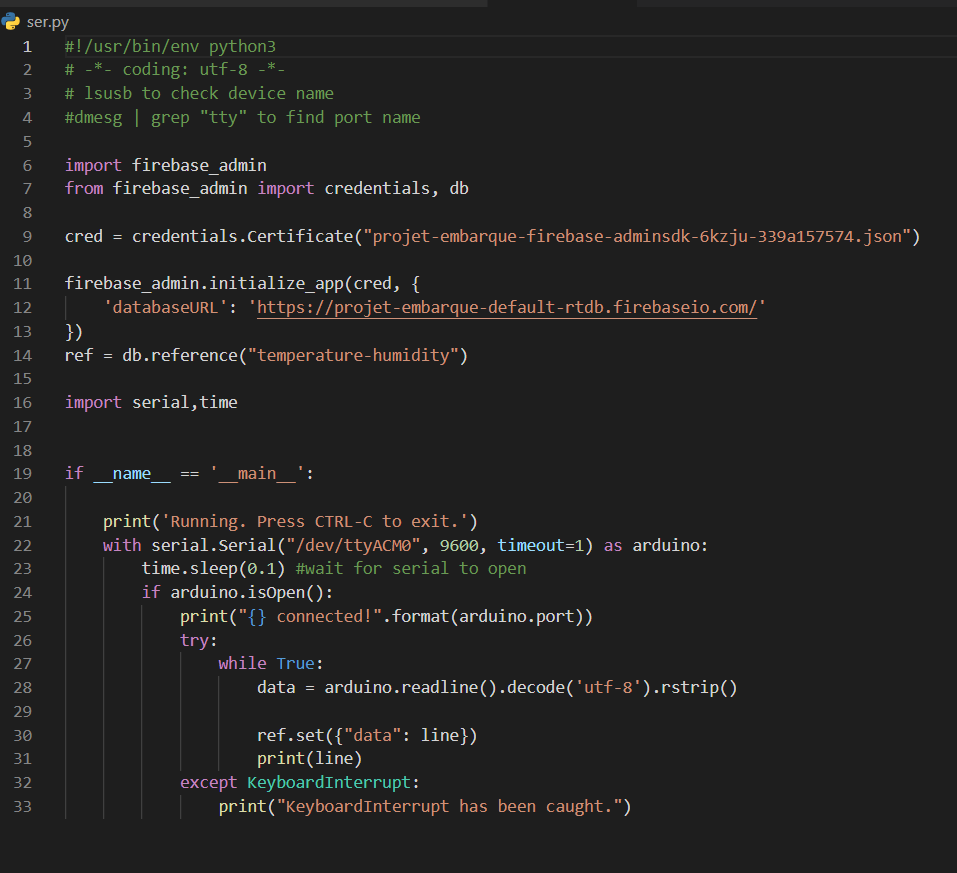


Upload Des données captées à une plateforme cloud:

Une fois que les données sont captées on les lit à l'aide d’un script python qui nous permet de les envoyer à la plateforme cloud de firebase qui offre l'utilité de real time database, qui nous offre la possibilité de créer un système en temps réel.

****

Configuration de firebase



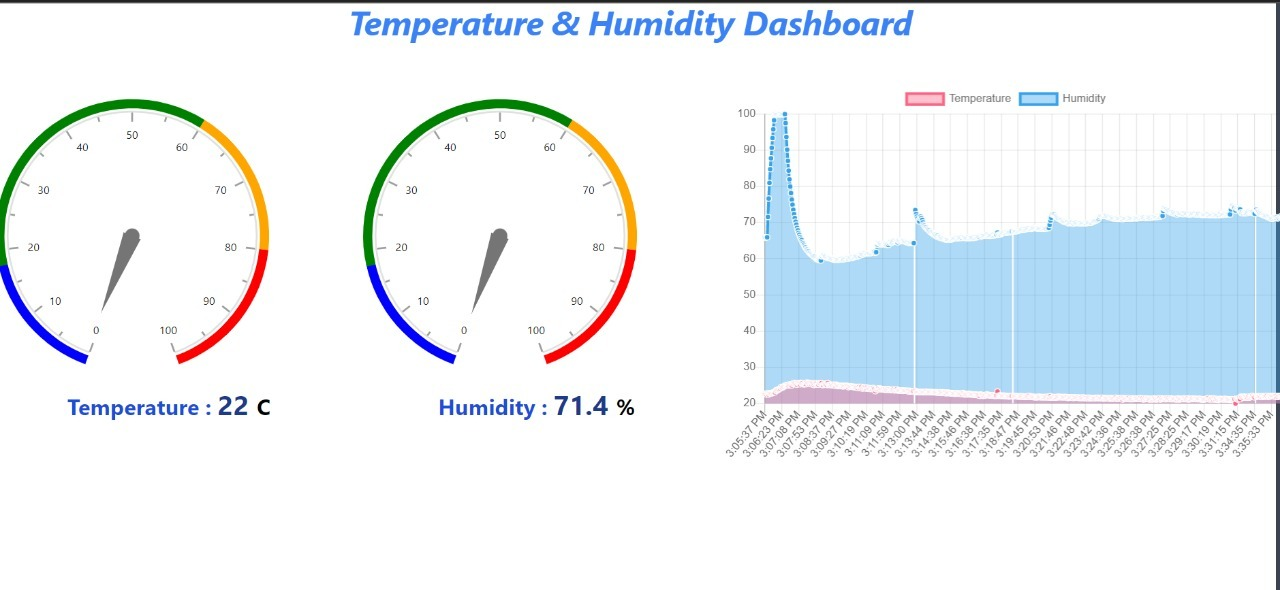
ser.py - script python qui envoie les données à la plateform cloud

# Application web:

Pour l’application web, l’objectif a été de créer une plateforme de visualisation des données qui seront récupérées à partir du Cloud Firebase.



**Angular** est une plate-forme et un framework permettant de créer des applications clientes d'une seule page à l'aide de HTML et de TypeScript. Angular est écrit en TypeScript. Il implémente les fonctionnalités de base et facultatives sous la forme d'un ensemble de bibliothèques TypeScript que vous importez dans vos applications.

****

Notre application représente deux jauges ,une pour la température et l'autre pour l'humidité. Avec chaque changement de ces derniers, les jauges se mettent à jour .

Quatre couleurs entourées par la jauge:

Le bleu pour une valeur inférieure à l'intervalle idéal (comme exemple la salle de serveurs : température entre 18 C et 27 C et humidité entre 40% et 60%)

Le vert pour une valeur appartient à l'intervalle idéal

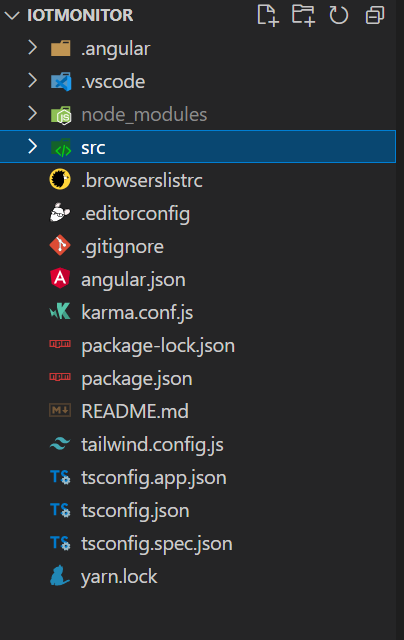
L’orange pour une valeur supérieure à l’intervalle idéal, c’est un signe d'avertissement.

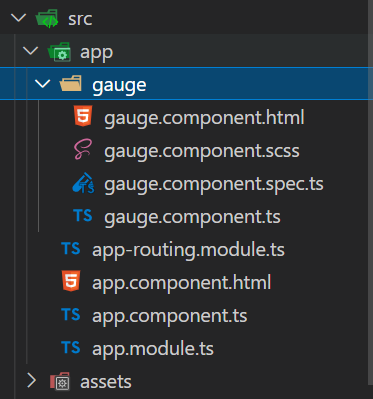
le rouge pour une valeur trop élevée et représente un danger pour les serveurs, il faut refroidir la salle pour ne pas les perdre.

Sur la même page, un graphe affiche les changements des valeurs de la température( représentée avec une couleur rouge) et de l'humidité ( représentée avec une couleur bleue) par rapport au temps.

**le code source de l’application angular :**

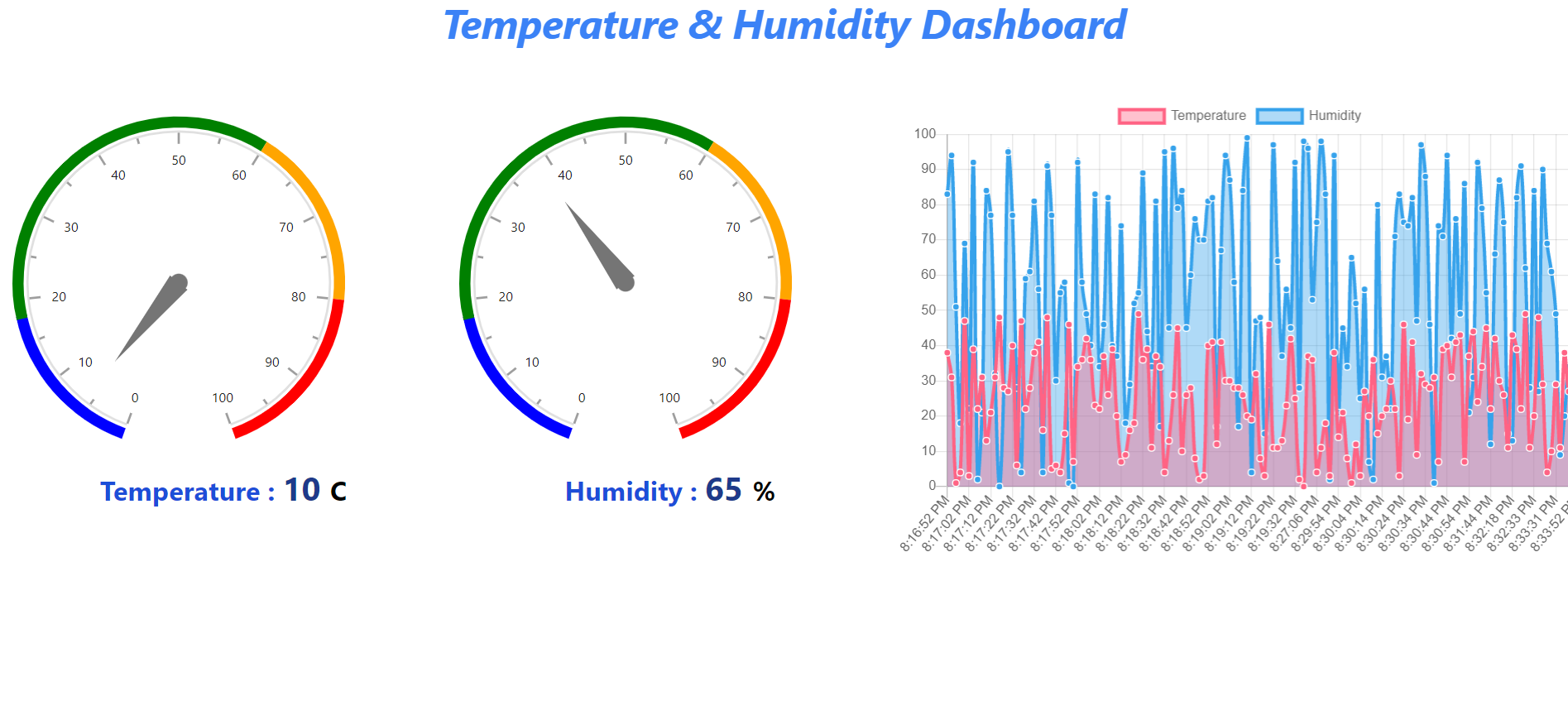
L’architecture du projet:

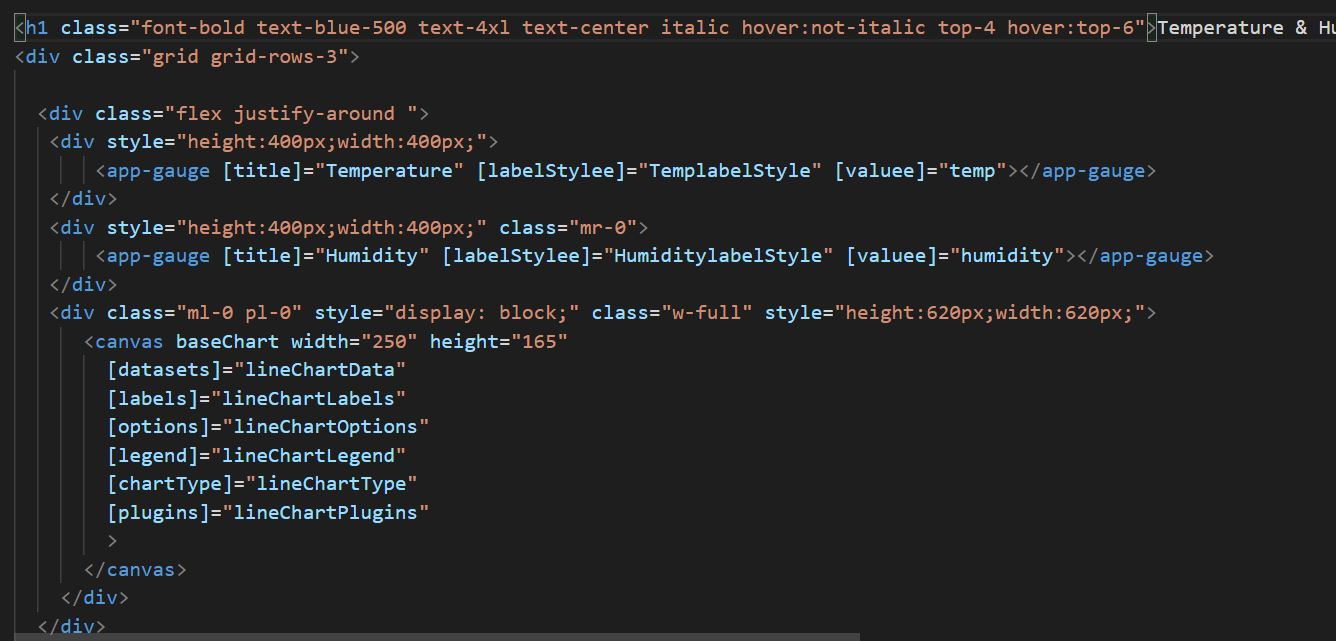
****

****

app.component.html:

le code html et tailwindcss pour créer l’interface suivante:

****

****

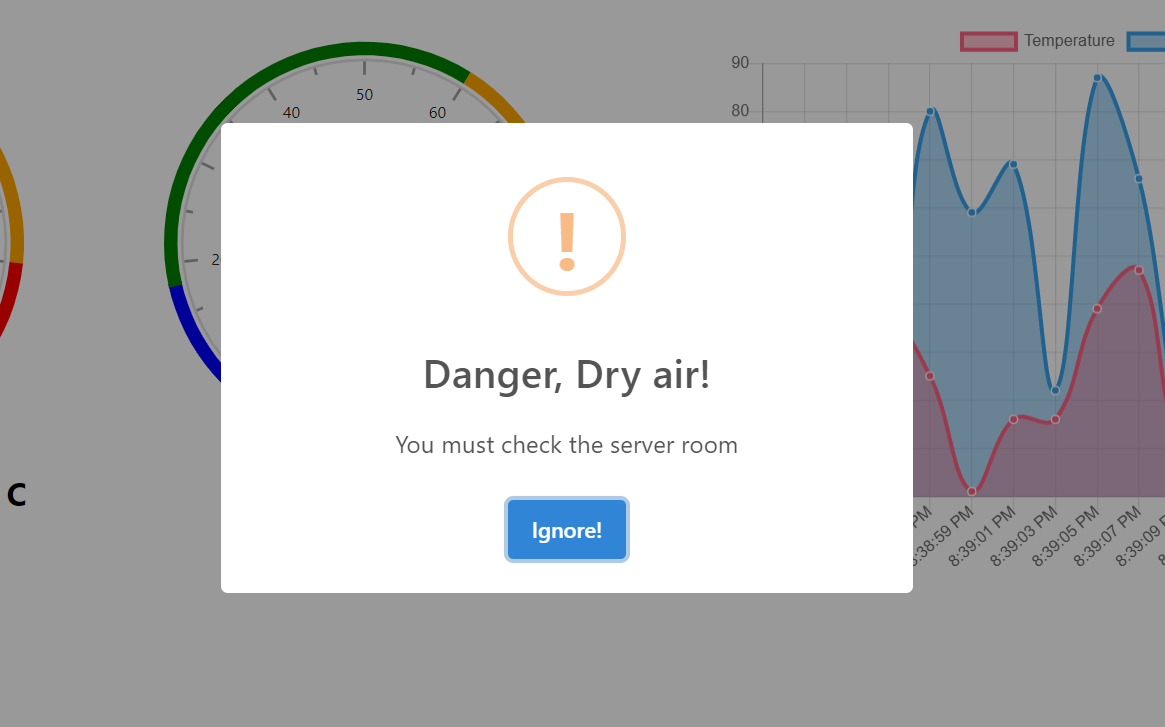
app.component.ts:

****

**sweetalert 2 :**

Ici on a intégré sweet alert 2, c’est une librairie JavaScript qui vous permettra d'avoir des boîtes d'alerte JavaScript sous forme de modal, avec un style beaucoup plus sympa que le style d'un simple "alerte" JavaScript.

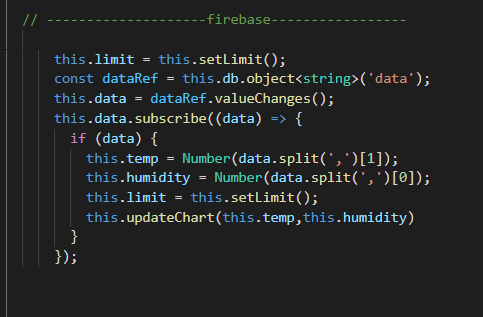
L’avantage d'utiliser ce genre de librairie est d'avoir aussi la même apparence sur les différents navigateurs, ce qui n'est pas le cas avec la fonction alert() de javascript.



****

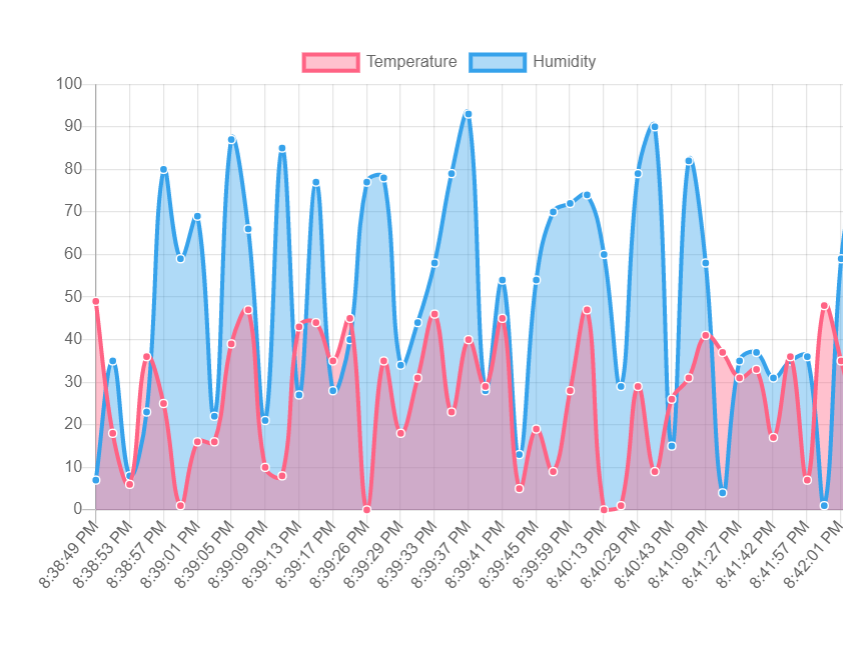
**Firebase:**

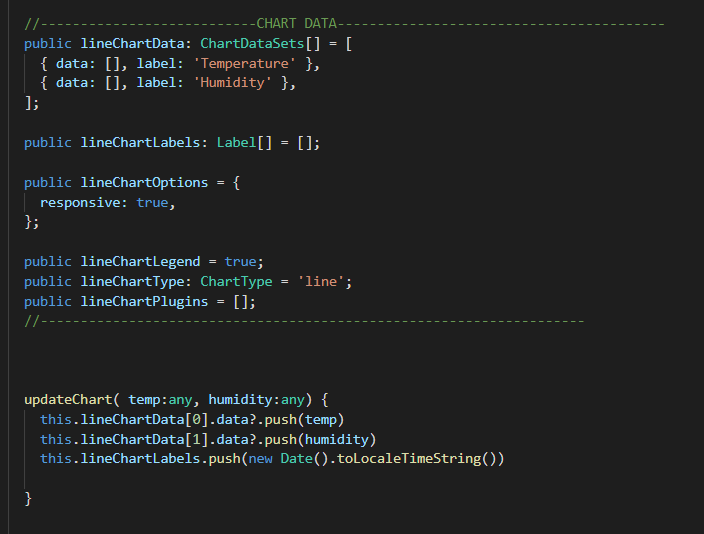
dans ce code on a intégrer firebase cloud qui facilite le messaging entre notre application et le server

****

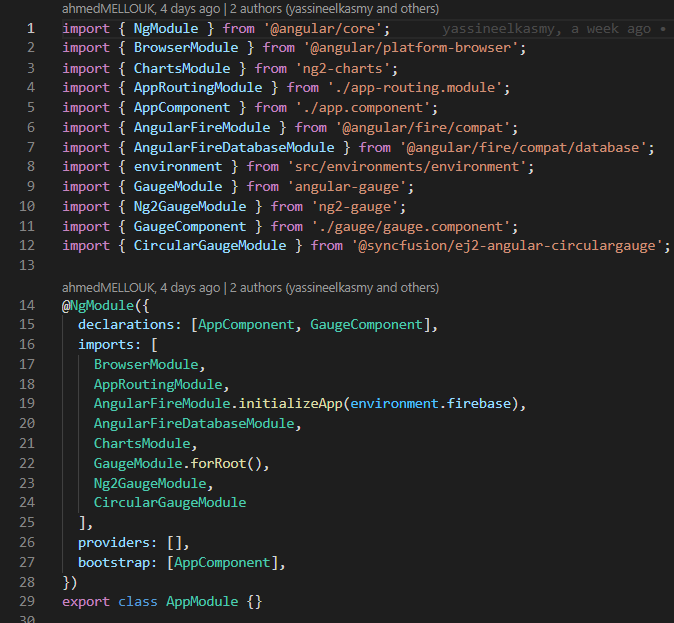
**chart js :**

l'intégration de la courbe qui représente la variation de la température et l'humidité dans la salle des serveurs en temps réel



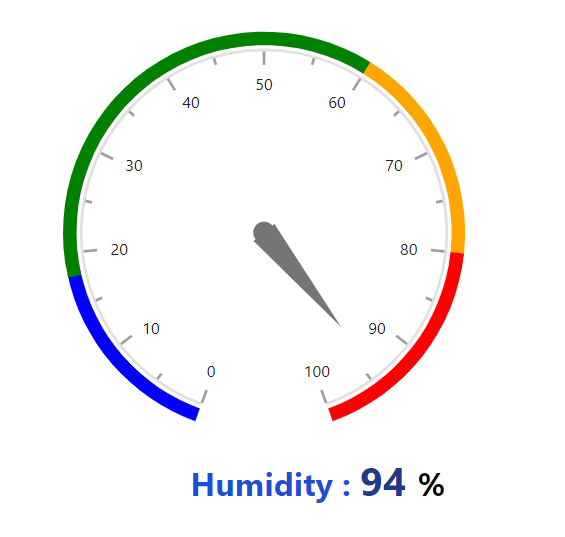
****

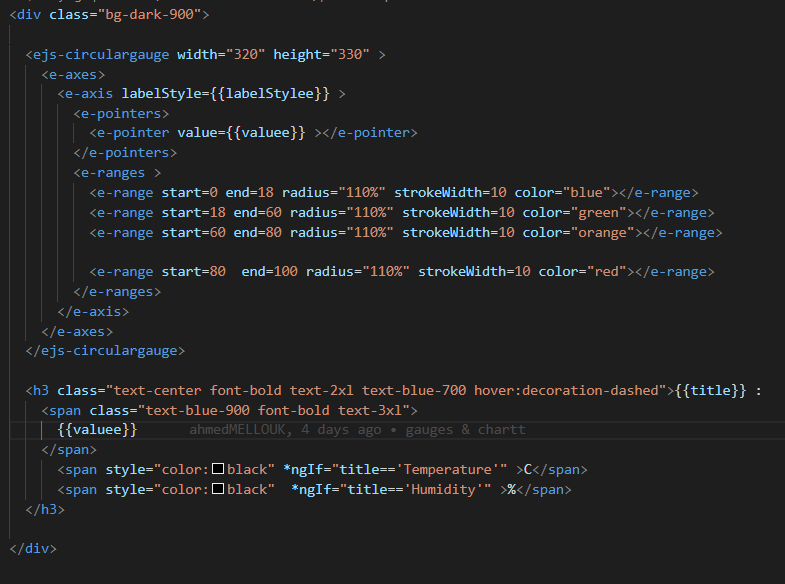
app.module.ts:



gauge.component.html:

le code html et tailwind css qui nous a créé les gauge suivant:

****

****

gauge.component.ts:

****

**les intervalles :**

****

# Conclusion:

Durant le travail sur ce projet on a eu une précieuse chance de pratiquer nos connaissances acquis sur le domaine de système embarqué et l’IOT .

De plus, on a pu élargir nos connaissances sur ce domaine à travers nos recherches et les problèmes qu’on a rencontrés durant l’exécution du travail, car les connaissances solides se constituent à travers les projets et l’effort fournis afin de surmonter les obstacles que ça soit techniques ou logistiques.