

Institut National des Sciences Appliquées et de Technologie



---

# SMART GARDEN: Intégration de Capteurs et Acquisition de Données vers Azure IoT et Blynk

---

réalisé par :

Karboul Achraf  
Chefrid Malak  
Messaoud Alaa

Encadrant INSAT:  
Mme. Ben Hatira Fafa

30-05-2023

### Remerciement

C'est avec grand plaisir que nous réservons ces lignes en signe de gratitude et de reconnaissance à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce projet de fin d'année.

Nous tenons à remercier tout particulièrement Mme. **Ben Hatira** pour ses précieux conseils , son soutien permanent et sa confiance tout au long de ce travail.

Finalement, aucun mot ne peut exprimer, à votre égard, notre gratitude et notre entière reconnaissance.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Besoins et contraintes</b>	<b>1</b>
2.1	Etude de l'existant . . . . .	1
2.1.1	ezzayra . . . . .	1
2.1.2	iFarming . . . . .	1
2.1.3	Smart Farm . . . . .	1
2.2	Motivation . . . . .	1
<b>3</b>	<b>Développement technique</b>	<b>2</b>
3.1	Solutions Matérielles . . . . .	2
3.1.1	Capteur de température et d'humidité . . . . .	2
3.1.2	Capteur de lumière . . . . .	2
3.2	Capteur d'humidité du sol . . . . .	3
3.2.1	électrovanne 12V . . . . .	3
3.2.2	Micro-contrôleur . . . . .	3
3.2.3	Modules supplémentaires . . . . .	4
3.2.4	plaque électronique . . . . .	4
3.2.5	convertisseur AD . . . . .	4
3.3	Solutions Logicielles . . . . .	5
3.3.1	ARDUINO IDE . . . . .	5
3.3.2	BLYNK . . . . .	5
3.3.3	AZURE IOT hub . . . . .	5
3.3.4	FRITZING . . . . .	5
3.3.5	Exploitation des données . . . . .	5
<b>4</b>	<b>diagramme fonctionnel</b>	<b>5</b>
<b>5</b>	<b>démarche</b>	<b>7</b>
5.1	montage électronique . . . . .	7
5.2	configuration Blynk . . . . .	8
5.3	configuration AZURE FOR IOT . . . . .	15
5.3.1	compte étudiant azure . . . . .	15
5.3.2	création de iot central . . . . .	16
5.4	test et validation . . . . .	19
<b>6</b>	<b>Perspectives</b>	<b>23</b>
<b>7</b>	<b>Bibliographie</b>	<b>23</b>

## Liste des Figures

1	CAPTEUR DHT 11 . . . . .	2
2	CAPTEUR lumiere . . . . .	2
3	CAPTEUR sol . . . . .	3
4	électrovanne . . . . .	3
5	ESP 8266 . . . . .	4
6	plaque d'essais . . . . .	4
7	convertisseur . . . . .	5
8	diagramme . . . . .	7
9	schema electronique complet . . . . .	8
10	blynk step 1 . . . . .	9
11	blynk step 2 . . . . .	10
12	blynk step 3 . . . . .	11
13	blynk step 4 . . . . .	12
14	blynk step 5 . . . . .	13
15	blynk step 6 . . . . .	14
16	blynk step 7 . . . . .	15
17	azure for student . . . . .	16
18	workspace . . . . .	16
19	configuration . . . . .	17
20	application déployée . . . . .	17
21	interface personnalisée . . . . .	18
22	variables . . . . .	18
23	interface . . . . .	18
24	connexion . . . . .	19
25	connexion . . . . .	20
26	connexion . . . . .	20
27	connexion . . . . .	21
28	connexion . . . . .	22
29	connexion . . . . .	22

# 1 Introduction

Quand on dit "Smart Garden", on entend par ce terme que c'est un processus automatisé, inter connecté qui peut être commandé à distance. C'est le fait d'avoir un espace vert "intelligent".

## 2 Besoins et contraintes

Que ce soit une petit pot à l'intérieur de votre maison ou un grand jardin, nous sommes d'accord que le jardinage nécessite beaucoup d'effort, d'entretien et de tâches manuelles. Le projet "Smart Garden" vise à réduire considérablement la charge de travail. La solution qu'on doit développer doit satisfaire les contraintes budgétaires du projet mais aussi les échéances fixés. Notre projet traite le cas des plantes au sol, on ne va pas considérer la culture hydroponique pour le moment, en effet l'hydroponie bien qu'elle soit une manière plus raffiné de faire les choses, il y a peut de chance que vous trouverez quelqu'un qui a une culture hydroponique dans son jardin, or nous on travaille principalement sur les besoins domestiques.

### 2.1 Etude de l'existant

#### 2.1.1 ezzayra

C'est une solution ERP dans le domaine de l'agriculture et l'agritech. Elle permet de mieux gérer les business de l'agriculture en fournissant aux clients un tableau de bord permettant de visualiser, de suivre et de piloter les KPI. Permet la gestion en temps réel de vos plante, facilitant ainsi la facturation et le calcul des dépenses et des coûts de chaque produit.

#### 2.1.2 iFarming

Cette société a mis en oeuvre la solution "Phyt'Eau". En effet cette solution a pour but d'automatiser le processus d'irrigation des plantes et de L'algorithme doit prendre en considération un certain nombre de paramètre pour nous fournir le résultat le plus proche possible des besoins réels de notre culture. Cette solution a des économies d'eau de 40

#### 2.1.3 Smart Farm

L'entreprise "Smart Farm" a réussi à développer la solution "Smart Soft Pro" qui est une solution servant aussi à rendre le processus d'irrigation plus efficace en l'automatisant et en le rendant plus fiable. La récolte des données sert à estimer d'une façon précise les besoins de la culture en eau.

### 2.2 Motivation

Par le biais de ce projet, on vise à récolter les données du milieu de la plante pour essayer de faire des études et de prévoir les besoins de la plante en nutriment et en eau. Pour aussi pouvoir constater sur les contraintes et les causes d'une bonne croissance de notre culture. Il est vrai que sur le marché, il y a déjà des entreprises qui ont travaillé sur le "Smart Garden" mais leur travail n'est pas tout à fait similaire à ce qu'on désire faire. Notre but et d'essayer d'intégrer les solutions déjà proposées sur le marché en une seule solution compacte. De cette façon, le client aura juste a commander un seul produit au lieu de s'inquiéter pour plusieurs et de comment les faire fonctionner ensemble. On va faire un peu de tout, calculer les besoins de la plante, estimer ses besoins, fournir à l'utilisateur un tableau de bord là où il peut suivre les KPI, calculer les coûts de chaque produit...

### 3 Développement technique

#### 3.1 Solutions Matérielles

Pour cette partie, nous avons fait une petite recherche pour étudier les matériels existants dans le marché et pour juger ceux qui conviennent le plus à nos besoins et à notre utilisation. On doit s'assurer que le matériel choisi répond aux contraintes de ce projet, satisfait nos besoins mais aussi qu'on ne tue pas une mouche avec un canon. Notre étude a abouti aux choix suivants.

##### 3.1.1 Capteur de température et d'humidité

Il s'agit d'un capteur modèle dans le domaine de l'iot, grâce à ses cas d'utilisation divers (smart home, smart garden, météo).

La grande différence réside dans le type à choisir, en effet il existe plusieurs modèles de capteurs de cette famille, tels que DHT, présent sous plusieurs formes avec (11,21,21).

Pour notre cas on a choisi le DHT11 vu sa disponibilité dans le marché et sa précision qui satisfait notre cas de figure, en récupérant les données relatives à la température et à l'humidité.

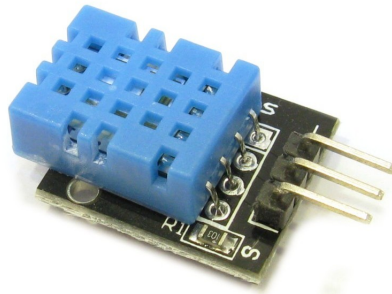


Figure 1: CAPTEUR DHT 11

##### 3.1.2 Capteur de lumière

Ce type de capteur permet de détecter la lumière ambiante. Son but principal est de savoir s'il s'agit d'un jour ensoleillé ou pas.

Pour en suite décider le besoin d'irrigation. Il utilise une technologie basique à l'aide d'une photo-résistance inversement proportionnelle à la luminosité. Sa valeur diminue lorsque la lumière est trop élevée.

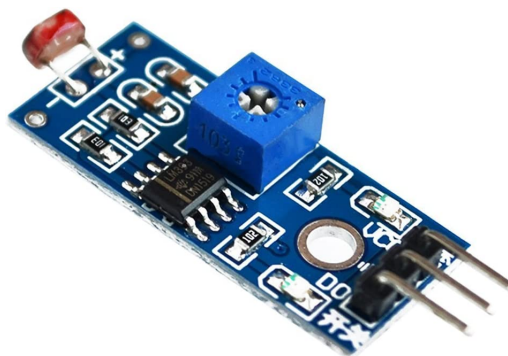


Figure 2: CAPTEUR lumière

### 3.2 Capteur d'humidité du sol

Pour mesurer l'humidité du sol qui contient les plantes ,et interpréter si elles ont besoin d'être irriguées, on injecte dans le sol un capteur d'humidité de sol permettant de mesurer son humidité. Il s'agit d'une sonde qui mesure le pourcentage de l'air dans le sol et interprète la valeur d'humidité avec une formule simple : humidité sol = 45 - pourcentage de l'air dans le sol

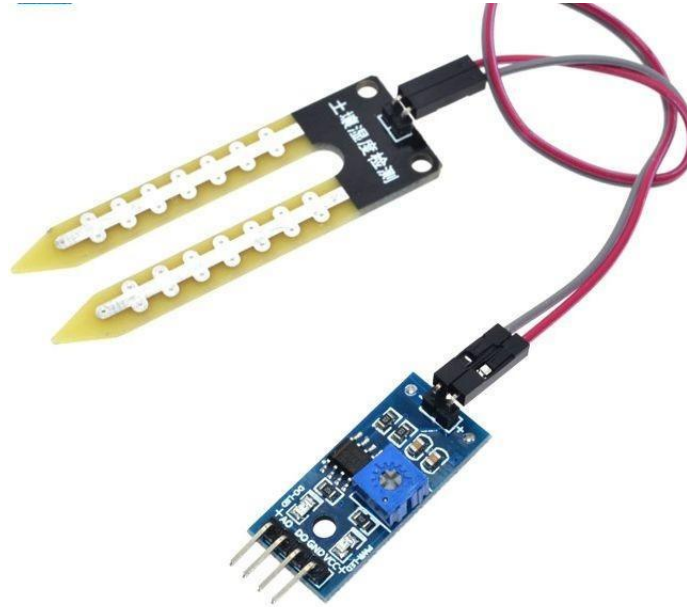


Figure 3: CAPTEUR sol

#### 3.2.1 électrovanne 12V

cet élément est indispensable dans notre partie actionneur . Elle représente l'élément majeur dans notre smart garden , qui sera piloté à distance par le Micro-contrôleur si les conditions d'irrigation sont satisfaites ou par forçage avec le smart phone de l'utilisateur.



Figure 4: électrovanne

#### 3.2.2 Micro-contrôleur

Avec une large variété de choix de Micro-contrôleur on devrait choisir un modèle qui offre une connectivité divers ,entre autre , WI-FI et Bluetooth pour permettre une connexion avec les serveurs et la carte . Et donc le choix est fixé sur un module ESP32 modèle 8266.

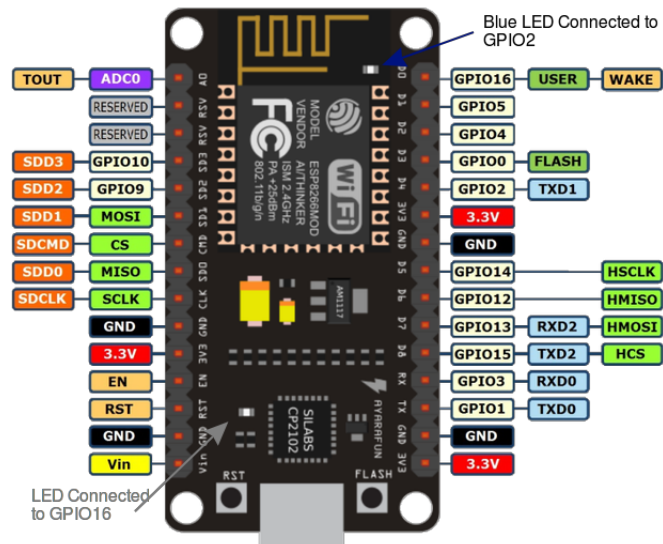


Figure 5: ESP 8266

### 3.2.3 Modules supplémentaires

### 3.2.4 plaque électronique

Elle joue un rôle primordiale dans la liaison des composants , elle se comporte comme une carte mère pour épargner un nombre considérable de câblage .

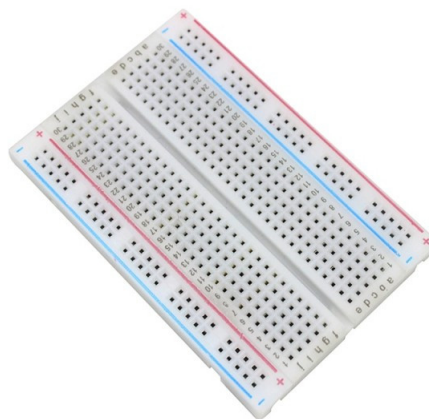


Figure 6: plaque d'essais

### 3.2.5 convertisseur AD

Un convertisseur Analogique digital , c est la pièce maîtresse de tous les capteurs en relation avec un phénomène physique. il comporte un potentiomètre pour définir un seuil à respecter pour passer du 0 à 1 ou l'inverse . il offre aussi la possibilité de lire directement le signal analogique tout en protégeant la carte à l'aide d'une résistance interne pour ne pas dépasser la limite de 5V en INPUT.



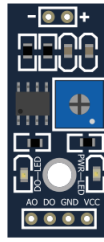


Figure 7: convertisseur

### 3.3 Solutions Logicielles

#### 3.3.1 ARDUINO IDE

L'éditeur Arduino nous permet d'inclure des bibliothèque externe et open source , il permet aussi de configurer une large variété de Micro-contrôleur y compris la famille ESP .

Il support aussi plusieurs bande passante qui nous permet de varier la fréquence .

pour notre cas de figure on a utilisé :

```
include ESP8266WiFi.h
include src/iotc/common/stringbuffer.h
include iotc.h
include DHT.h
include BlynkSimpleEsp8266.h
```

#### 3.3.2 BLYNK

Avec son protocole TCP/IP et sa fiabilité en terme de collectivité la plate-forme Blynk offre une interface très fluide offrant la possibilité de de monitoring et d'affichage en temps réel via un smart phone , mais son point fort est le contrôle à distance via les bouton virtuel et les slide buttons qui représente un potentiomètre virtuelle

#### 3.3.3 AZURE IOT hub

Et pour la fin on utilise aZURE comme une plate-forme cloud , qui offre une service de monotoring plus avancé que celle de blynk et qui reprsente son point fort .

il permet ainsi d'afficher ,suivre et stockr les données collecter dans une base de données avec un clé primaire chronologique et d'exploiter les données sous la forme souhaitée .

#### 3.3.4 FRITZING

Pour réaliser notre schéma électronique , on utilise le logiciel FRITZING avec sa licence gratuit qui offre la possibilité de ajouter des composant open source ou d'en créer un , puis lier tout les composants de notre système pour avoir un vue d'ensemble et complète .

#### 3.3.5 Exploitation des données

Ce site permet de récupérer les données relative au météo , voire température, humidité , vent .... l'utilité de cet outil de comparer les valeurs réel relevées par nos capteurs avec les valeurs du site pour avoir une idée sur la précision de notre système et decider de sa fiabilité.

## 4 diagramme fonctionnel

Dans cette partie on va détailler le code implémenté:

```

1  /* -----
   #include <MySQL_Connection.h>
3  #include <MySQL_Cursor.h>
   #include <DHT.h>
5  #include <ESP8266WiFi.h>
   #include <BlynkSimpleEsp8266.h>
7  #include "src/iotc/common/string_buffer.h"
   #include "src/iotc/iotc.h"
9  -----*/

```

Listing 1: déclaration des bibliothèques

```

1  /* -----
   #define DHTPIN D3
3  #define DHTTYPE DHT11
   #define SOILPIN D7
5  #define BLYNK_AUTH_TOKEN "sfW3L3vcIwVrxQH392bu89PziXJldM60 "
   #define RELAY_PIN D4
7  DHT dht(DHTPIN, DHTTYPE);
   -----*/

```

Listing 2: déclaration des variables

```

1  /* -----
2  intitialiser les variables de
   1/ la connexion azure
4  2/ la connexion BLYNK
   const char* ssid = "Airbox-343A";
6  const char* password = "EJJBHJmJ";
   -----*/

```

Listing 3: intitialiser les variables

```

1  /* -----
   1/ debuter la liaison avec le DHTPIN
3  2/ debuter la liaison avec SOILPIN
   3/ intialiser la valeur d'humidit
5  4/ activer la connexion Wifi et blynk
   5/ se connecter azure
7  -----*/

```

Listing 4: SETUP

```

1  /* -----
   r cup rer les donn es re ues par les capteurs:
3  1/ d'humidit
   2/ de temp rature
5  3/ de la lumiere embiante
   4/ d'humidit du SOL
7  -----*/

```

Listing 5: LOOP

```

1  /* -----
   exploitation de donn es prises par les capteurs:
3  1/ calculer le stress temp
   2/ calculer le point de ros e
5  3/ calculer l' vapotranspiration

7  double stressIndex = (1,8 x temperature + 32) - ( 0,55 - 0,0055 x humidity) x (1,8 x
   temperature - 26);
   double dewPoint = temperature - ((100 - humidity) / 5);

```

```

9 // double etp = coeffTemp * temperature + coeffHumidite * humidite + coeffVent * vent +
  coeffRayonnement * rayonnement;
  -----*/

```

Listing 6: data processing

```

1 /* -----
   Envoyer les donn es      Blynk
3 Envoyer les donn es      Azure
  -----*/

```

Listing 7: sending data

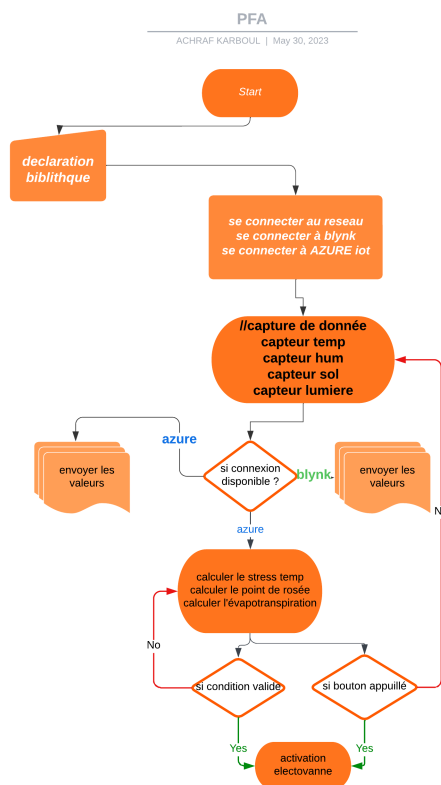


Figure 8: diagramme

## 5 démarche

### 5.1 montage électronique

On doit d'abord relier les convertisseur avec les capteurs :

Photo-résistance + convertisseur AD

Sonde d'humidité + convertisseur AD

Relier les les deux convertisseur , le DHT11 , le relais de commande avec VCC et GND .

Relier la deuxième partie du relais avec l'electrovanne et les 12V .

Relier tout les composants au micro-contrôleur dans leur broche respective.

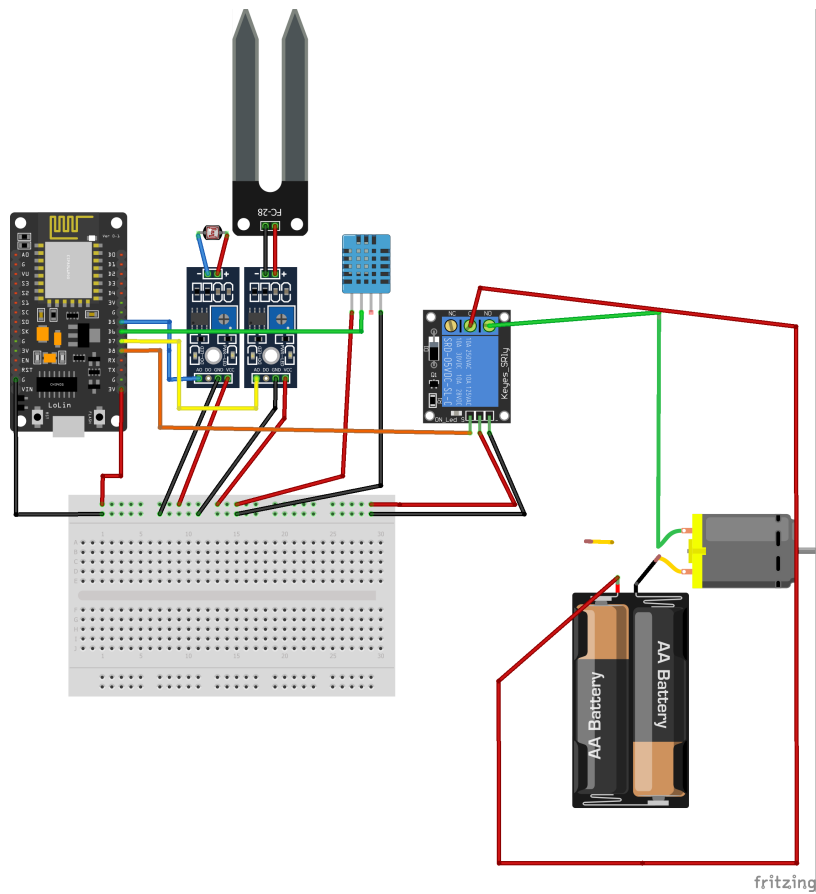


Figure 9: schema electronique complet

## 5.2 configuration Blynk

On commence par télécharger et installer Blynk dans sa dernière version sur android , et se connecter via un compte google .

Après vient la tout première étape de la configuration : on choisit notre méthode de connexion qui sera la méthode rapide dans notre cas :

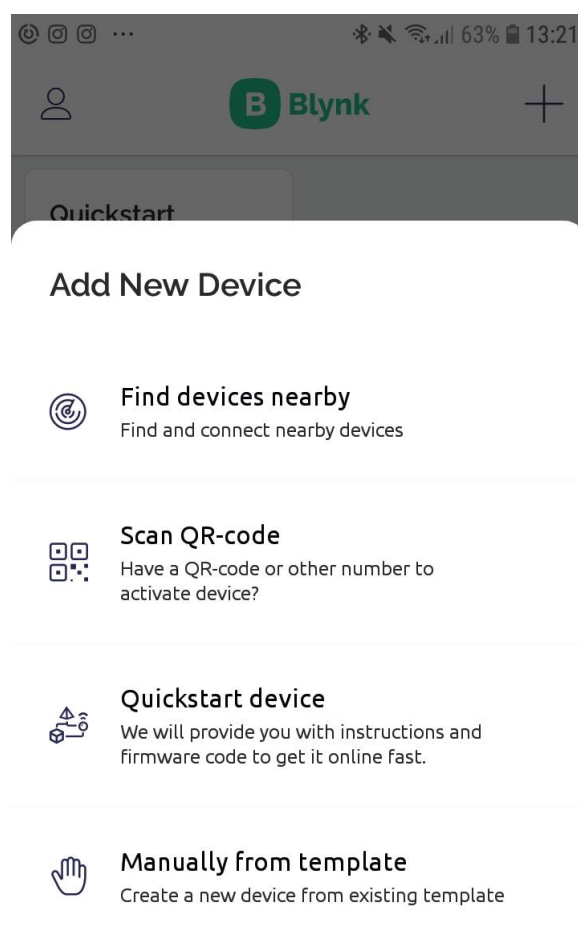


Figure 10: blynk step 1

Suivre les instruction afficher et cliquer sur "lets start"

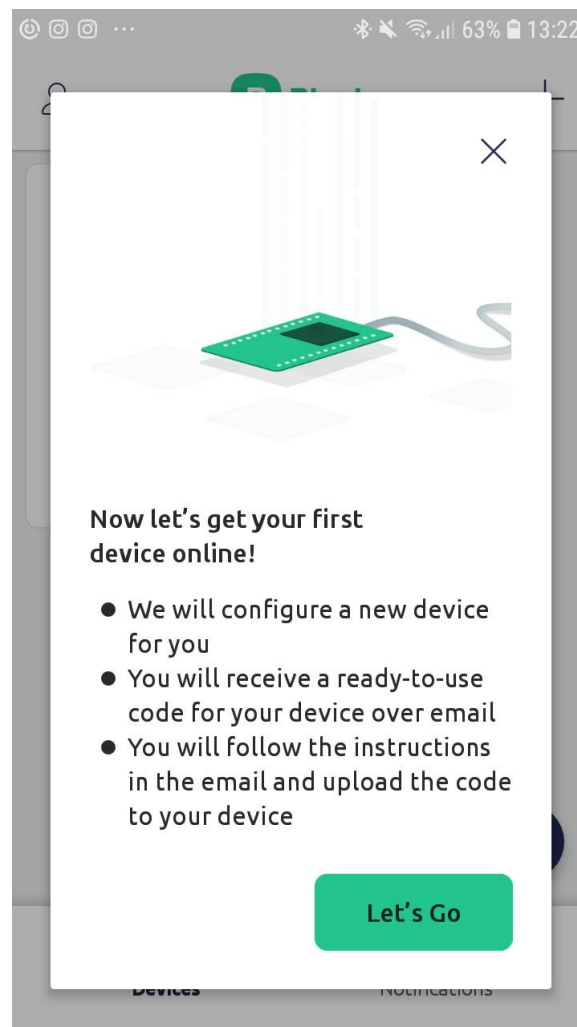


Figure 11: blynk step 2

Choisir ensuite la carte esp32 model 8266 et la méthode de connexion qui sera le wifi pour notre cas:

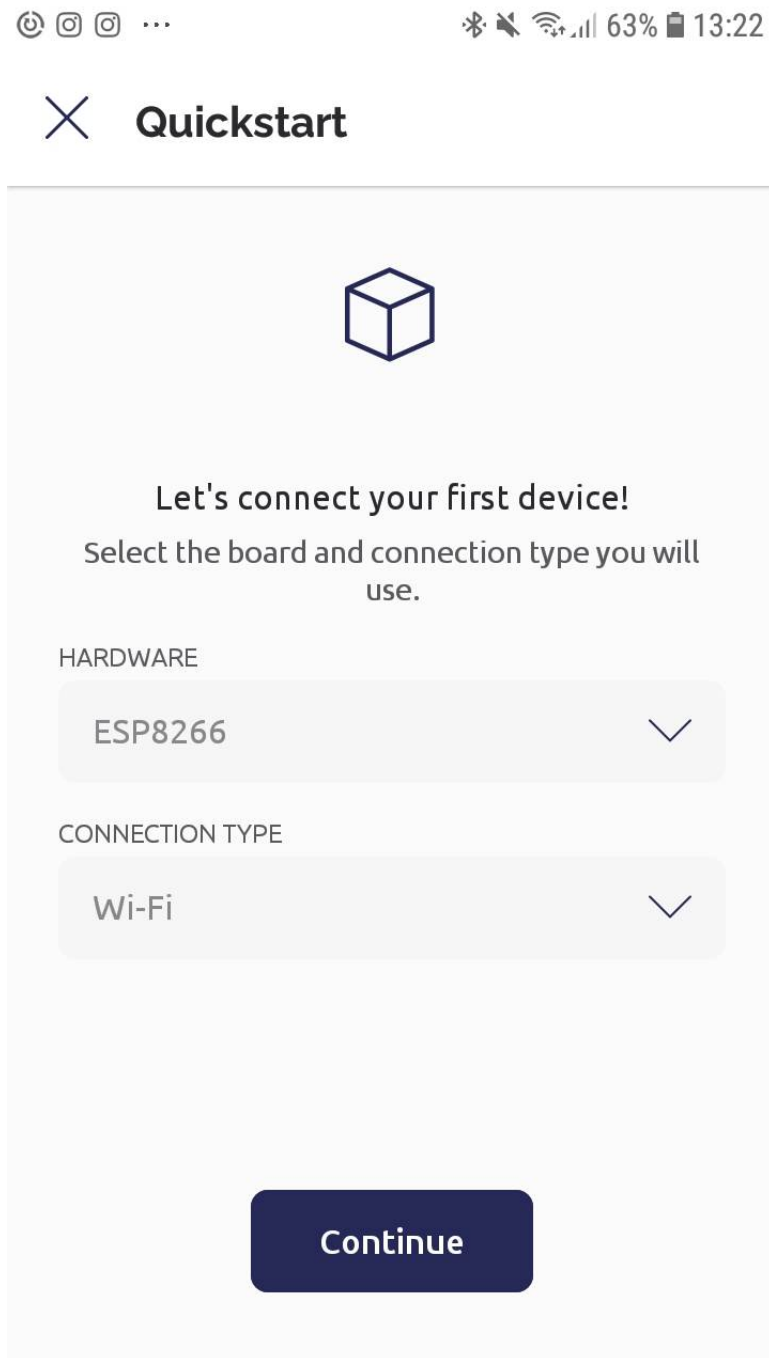


Figure 12: blynk step 3

Entrer dans notre espace de travail et aller tout à droite en haut pour sélectionner les WIDGET nécessaire à notre projet :

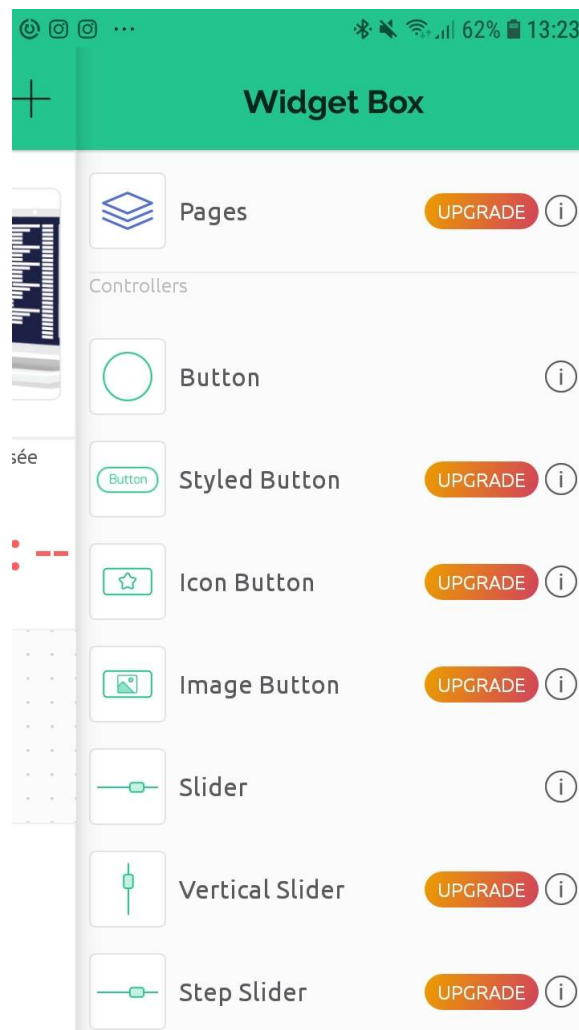


Figure 13: blynk step 4

On choisit par exemple "labeled value" , qui représente un affichage pour une valeur passer par le micro-contrôleur Et on va dans la suite configurer la valeur d'humidité:

On définit le type de donnée qui est double .

On définit le PIN virtuelle "V1" qui représente la liaison entre l'esp32 et BLYNK .



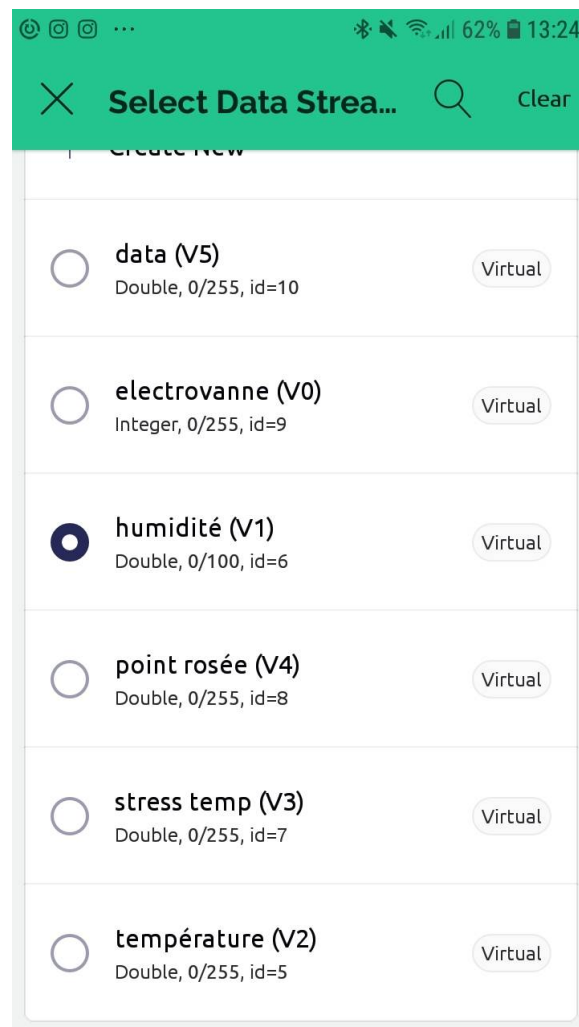


Figure 14: blynk step 5

On passe ensuite au design de notre indicateur y compris la couleur, police et même la taille. Et on introduit le libelle de notre variable

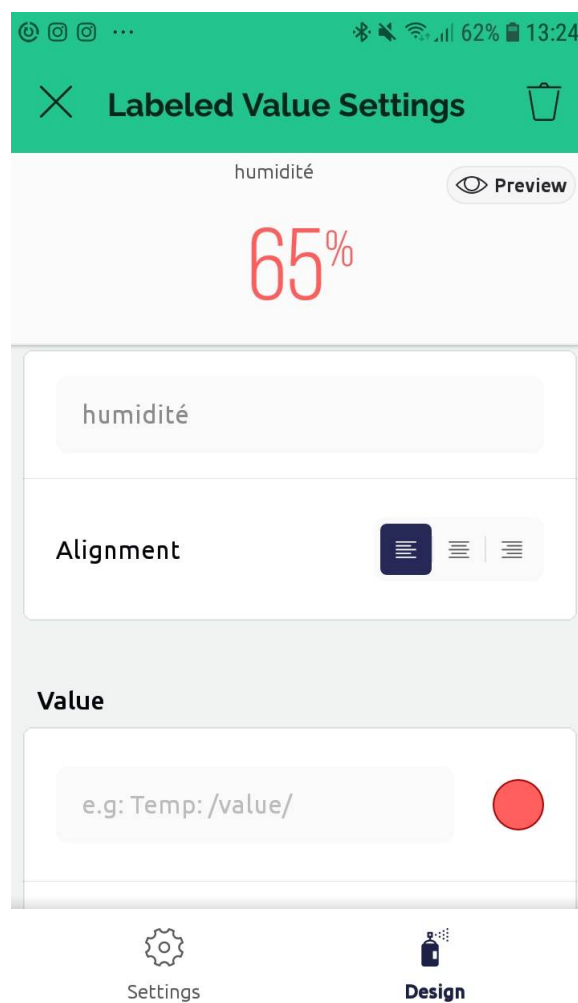


Figure 15: blynk step 6

Finalement on passe autre variable et on les organise sur l'écran d'accueil :

- humidité en cercle de niveau
- température en labeled value
- humidité sol en labeled value
- temp de stress en labeled value
- indicateur sur l'activation d'électrovanne en LED
- point de rosé en labeled value

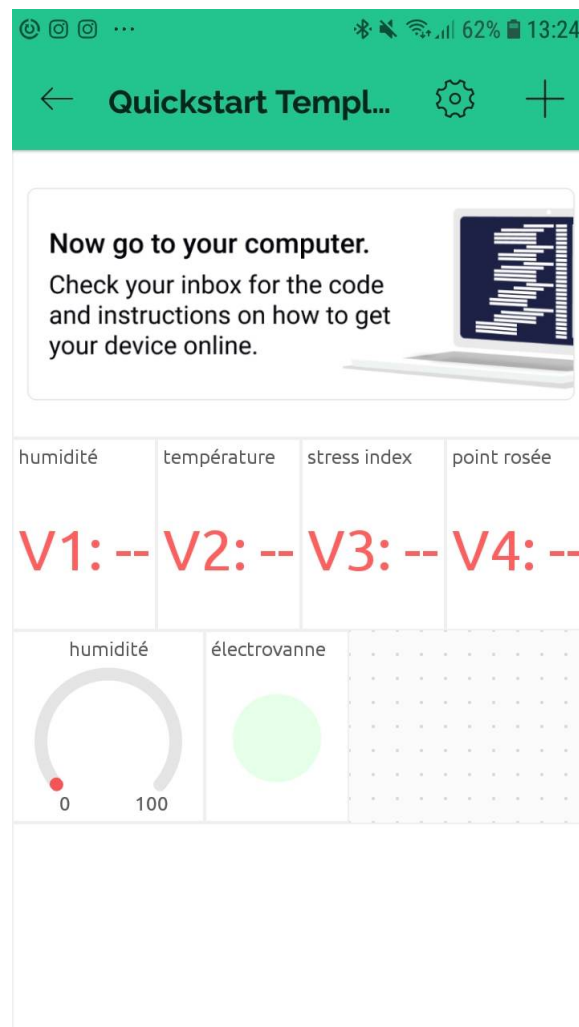


Figure 16: blynk step 7

### 5.3 configuration AZURE FOR IOT

Azure IoT Central est une plate forme d'application IoT qui simplifie la création des solutions IoT. il fournit une surface d'expérience utilisateur et API prête à l'emploi conçue pour connecter, gérer et exploiter des flottes d'appareils à grande échelle.

Dans la suite nous présenterons les étapes à suivre pour configurer les affichage et le stockage de donnée de notre solutions et on suppose que la partie matériel est déjà monté :

#### 5.3.1 compte étudiant azure

En tant que étudiant, et par mail institutionnel , Microsoft nous accorde un crédit de 100 dollar renouvelable chaque année pour pouvoir tester les services AZURE .

On commence tout d'abord par accéder à "azure for student"

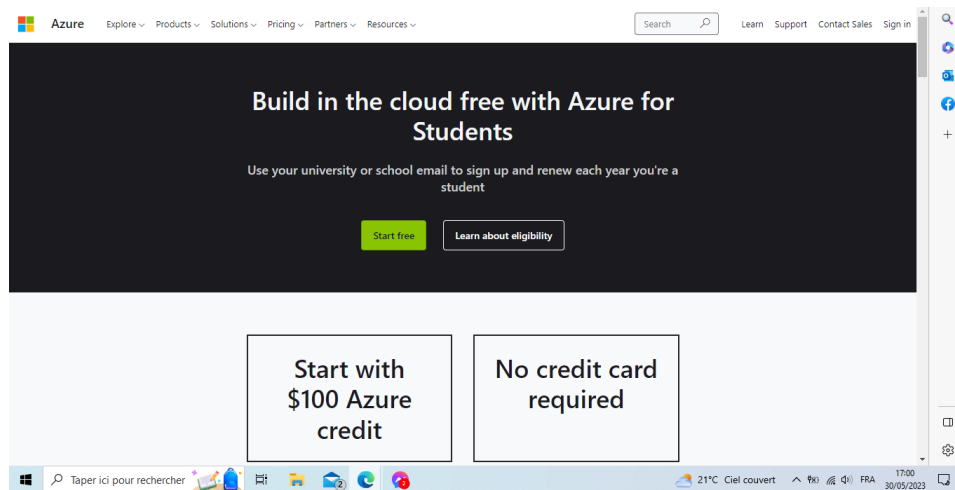


Figure 17: azure for student

Ensuite on pourra se connecter pour recevoir notre crédit et arriver à l'écran d'accueil et créer notre environnement de travail

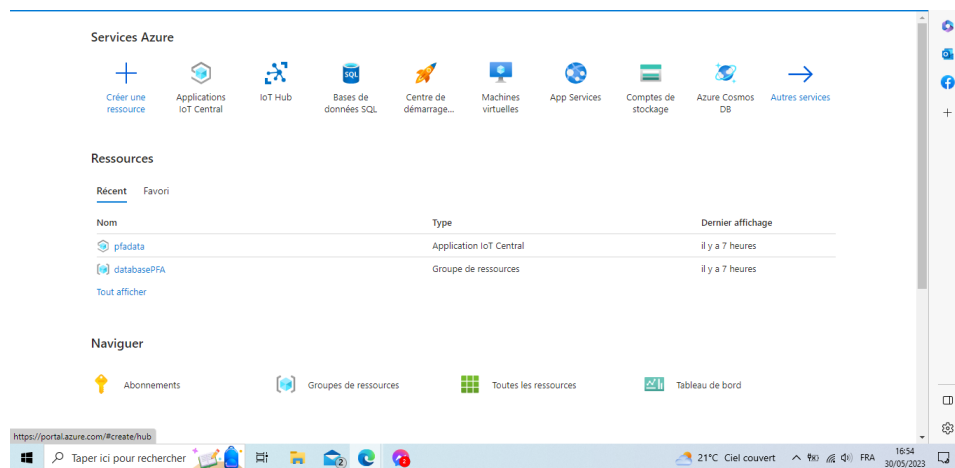


Figure 18: workspace

### 5.3.2 création de iot central

Dans la barre de recherche, accédez à IoT Central Applications . Cliquez ensuite sur **Application centrale IoT** . Vous serez redirigé vers la page suivante :

Dans la partie abonnement, on selectionne notre abonnement etudiant . Sous Groupe de ressources,on chosisi le groupe de ressource **dataBasePFA** .

Dans la partie des détails de l'instance on définit le nom de notre HUB : PFADATA .

ainsi notre serveur sera : <https://pfadata.azureiotcentral.com>

Dans la partie modèle, on sélectionne un **modèle personnalisé** et une région en fonction de votre emplacement géographique le plus proche. enfin **Review + Create**

Accueil > Applications IoT Central >

## Application IoT Central

Select the subscription to manage the deployed IoT Central resource and costs. Use resource groups like folders to organize and manage all your resources.

Abonnement \*

Groupe de ressources \*  [Créer nouveau](#)

**Instance details**

Nom de la ressource \*

URL d'application \*  azureiotcentral.com

Modèle \*

Région \*

Plan tarifaire \*

[Vérifier + créer](#) [Précédent](#) [Suivant : Tags](#) [Envoyer des commentaires](#)

Figure 19: configuration

Une fois le déploiement s'effectue correctement on pourra ouvrir notre ressource :

Accueil > Applications IoT Central >

## Applications IoT C...

Ministère de l'Enseignement Supérieur et de la Re...

+ Créer Gérer la vue ...

Rechercher

Filter un champ...

Nom ↑

pfadata

**pfadata** Application IoT Central

Supprimer

Vue d'ensemble

Étiquettes

Paramètres

Supervision

Métriques

Paramètres de diagnostic

Classeurs

Automatisation

Tâches (préversion)

Exporter le modèle

Bases

Groupe de ressources ([déplacer](#))

Emplacement

Abonnement ([déplacer](#))

ID d'abonnement

Status

Étiquettes ([modifier](#))

Paramètres de diagnostic

Classeurs

Automatisation

Tâches (préversion)

Exporter le modèle

URL de l'application IoT Central

<https://pfadata.azureiotcentral.com>

Vue JSON

Page 1 sur 1

Figure 20: application déployée

Nous devons créer un modèle ou un plan pour notre appareil, le modèle sera structuré en fonction du type de données que nous recevrons de notre esp32 . Ainsi, ce que nous devons d'abord configurer est le modèle d'appareil.

À partir de la page ci-dessus, vous obtiendrez une LIEN d'application centrale IoT. Cliquez sur l'URL et vous serez redirigé vers la page suivante maintenant.

Sur le côté gauche du tableau de bord, on choisit sur Modèle d'appareil. ensuite sur Créer un modèle d'appareil.

De nombreux modèles sont prêts à l'emploi créés par le réseaux . Mais nous devons créer notre propre modèle. Pour ce faire, on choisit Périphérique IoT. ensuite sur Suivant : Personnaliser.

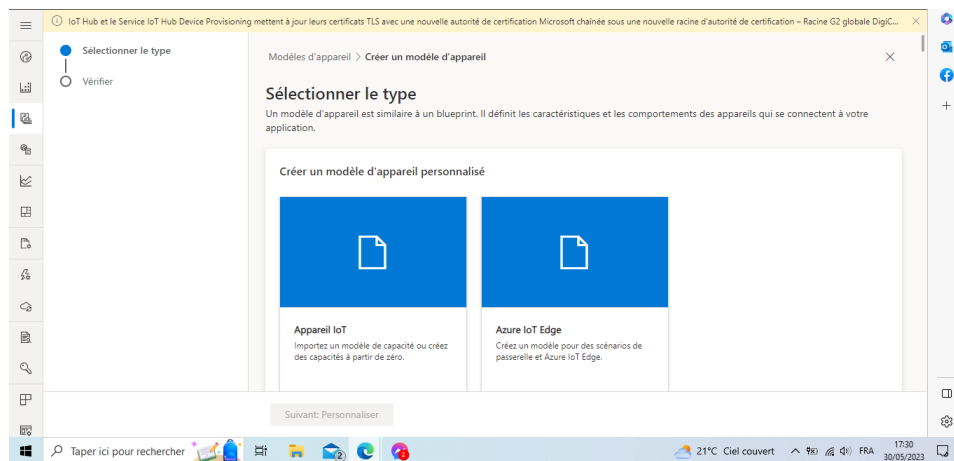


Figure 21: interface personnalisée

Une fois l'interface est créée, on doit ajouter toutes les variables qui seront affichées sur notre interface, leurs type, et mode d'affichage.

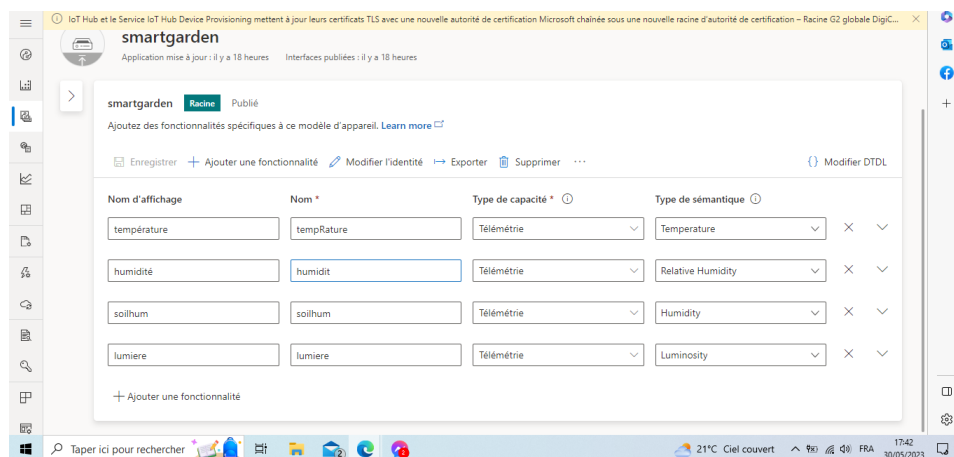


Figure 22: variables

Nous sélectionnons les vues générales par défaut. ensuite sur Générer des vues de tableau de bord par défaut. Dans la partie vues, et on doit changer le valeur en "Dernière valeur connue". Après cela, "Enregistrer" et publier le modèle.

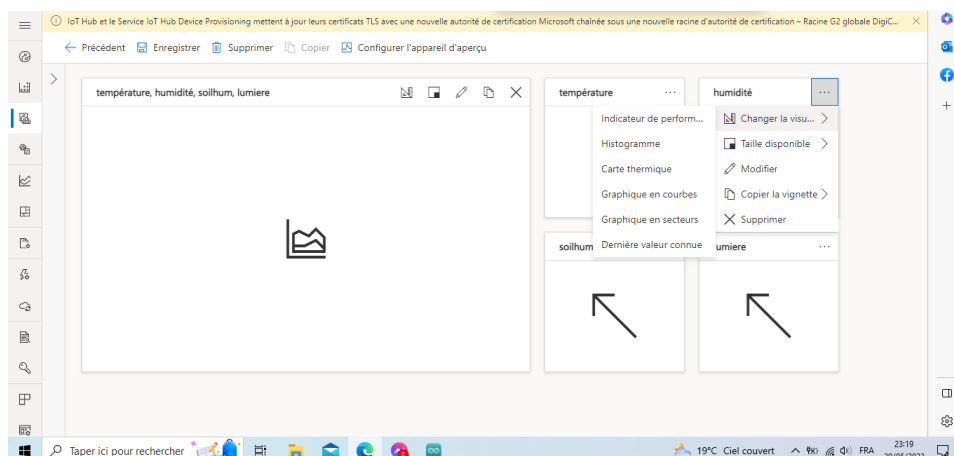


Figure 23: interface

Comme dernière étape nous cliquons sur Appareils puis on choisit Ajouter un appareil. nous donnons le nom de l'appareil et l'ID de l'appareil. Dans la partie Modèle d'appareil, nous sélectionnons le modèle d'appareil que

nous avons créé ci-dessus. Ensuite, on choisi Créer.

Nous cliquons sur le nom de l'appareil et nous verrons de nombreux onglets comme à propos, aperçu, données brutes et alliés mappés. La visualisation des données prend à l'onglet vue d'ensemble.

En haut à gauche, nous verrons une option pour se connecter. nous cliquons sur Connecter.

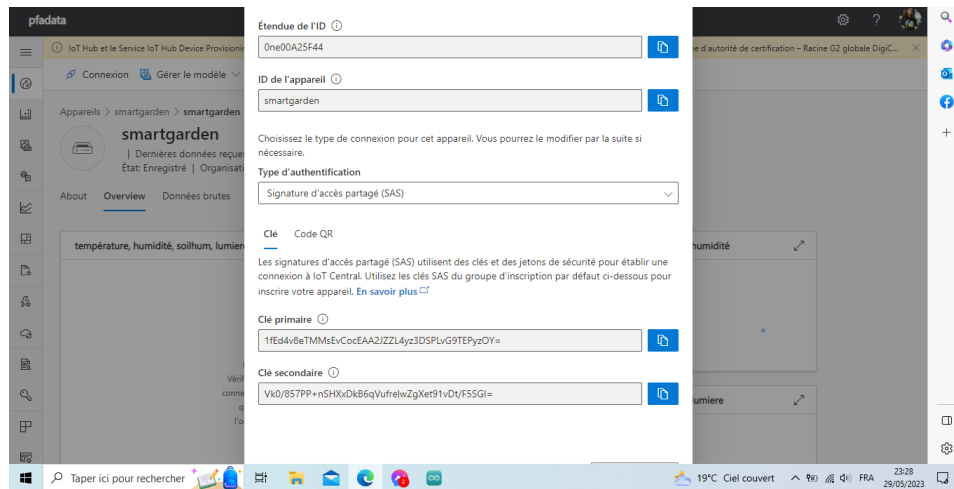


Figure 24: connexion

Ici, nous présenterons les informations dont nous avons besoin pour connecter votre NodeMCU ESP8266 à Azure IoT Central, telles que DeviceID, Name, Primary Key et Public Key.

## 5.4 test et validation

Pour notre test , il suffit de vérifier que le code compile .

puis on passe à t'élèverez le code via Arduinio iDE avec la configuration suivante :

```
uploadspeed = 12500
flash size = 4mo
CPU FRQ= 80Mhz
node MCU 1.0
COM3
```

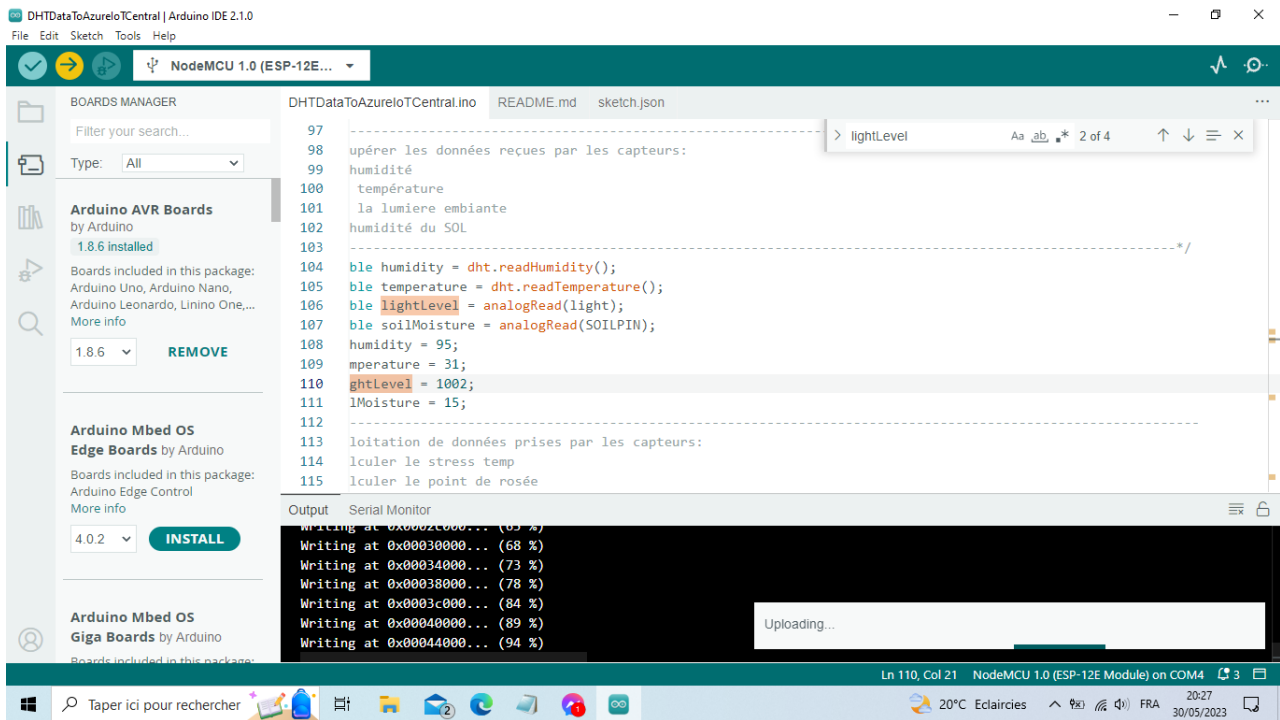


Figure 25: connexion

Juste après l'attente de 100 , on s'assure que le code est bien fonctionnel via le terminal sous un fréquence de lecture de 12500 .

on peut suivre aussi les étapes de connexion et les erreurs causés par la latence de WIFI qui tourne en boucle jusqu'à l'obtention d'un connexion internet fiable pour permettre l'envoi de donnée .

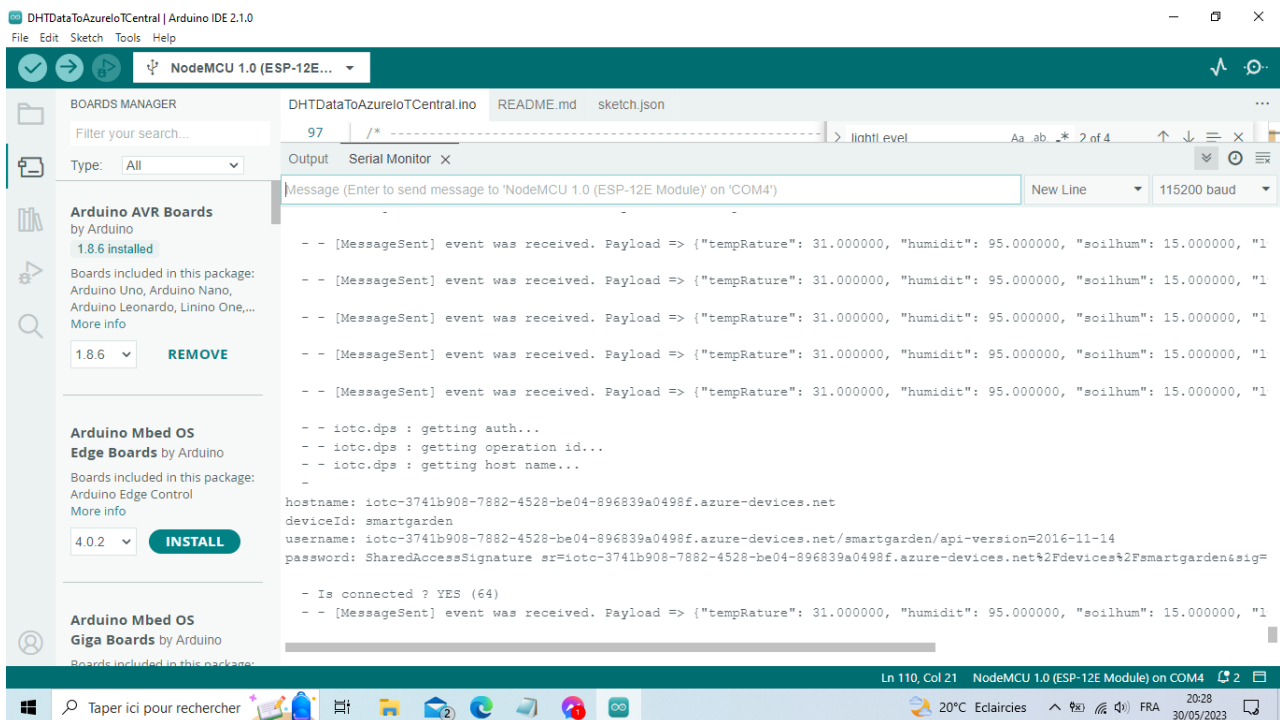


Figure 26: connexion

Finalement on peut suivre l'avancement sur notre application Blynk qui affiche les variable récupérées et calculées et l'état d'electrovanne :



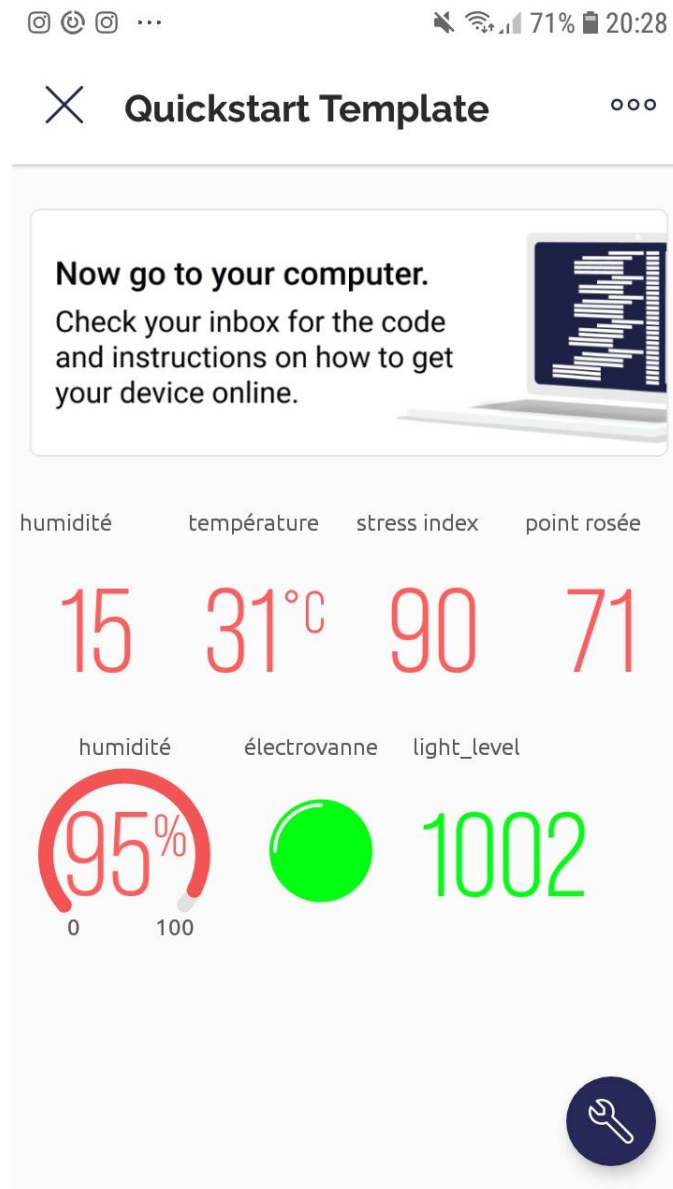


Figure 27: connexion

En meme temps on peut traquer la variation de nos indicateur sur AZURE IOT sous forme de valeur et de courbe de variation en fonction de temps à une période d'échantillonnage de 10sec

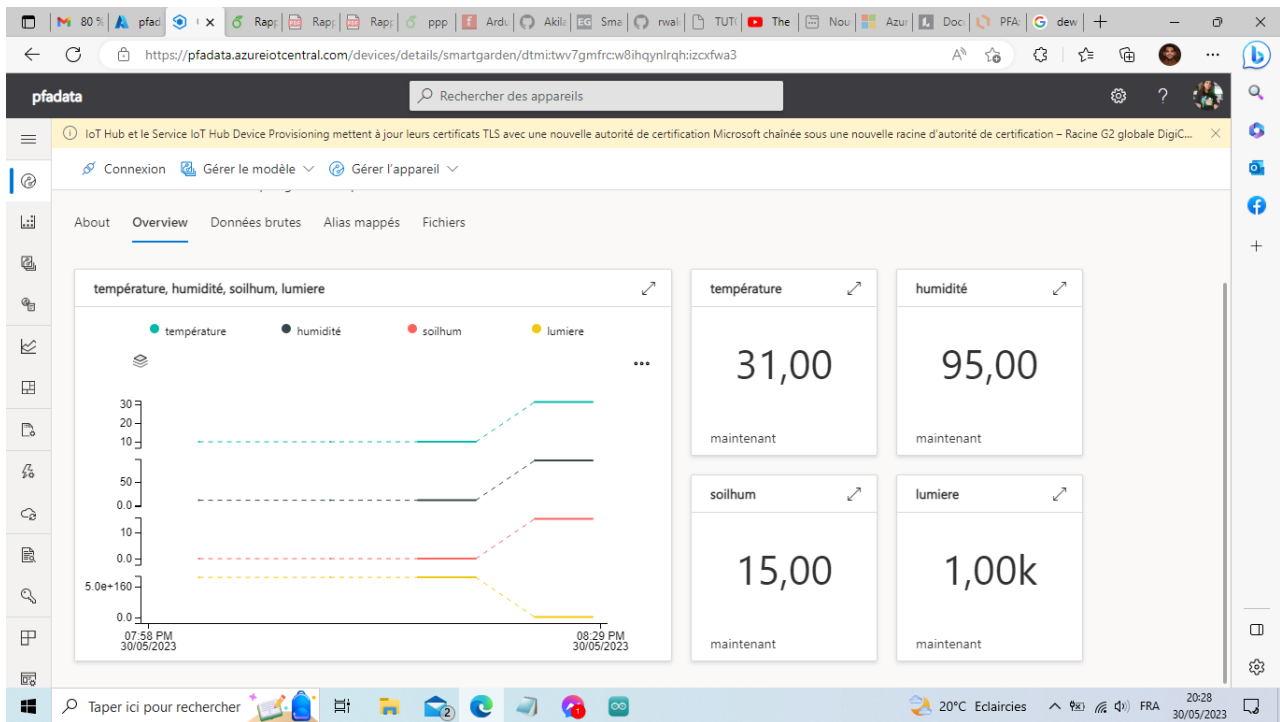


Figure 28: connexion

Et comme dernière validation on consulte la base de données résultante de notre travail qui ajoute les lignes de données avec une clé primaire chronologique (date+heure) avec toutes les variables mentionnées

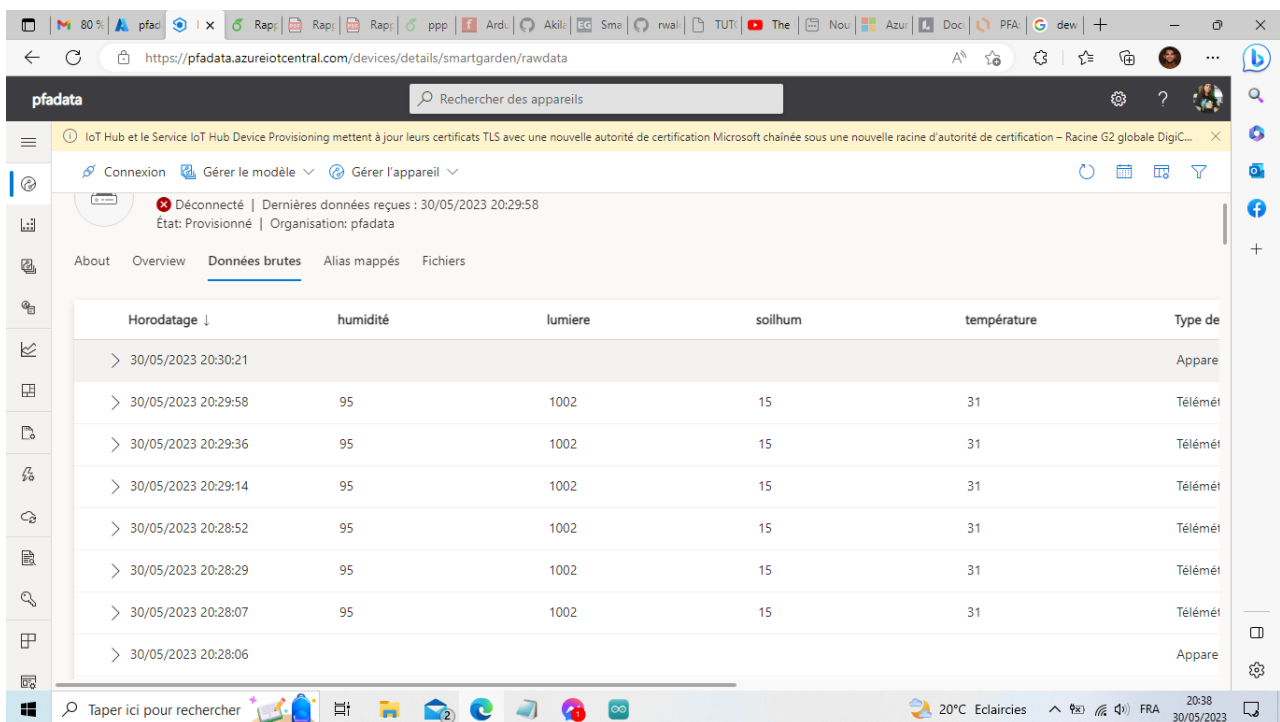


Figure 29: connexion

## 6 Perspectives

Durant la période de ce projet l'objectif était d'apprendre des nouvelle technologie et forger nos connaissance au niveau technologique que ce soit software ou hardware .

On a pu finalement réaliser cet objectif , en utilisant deux plate-forme IOT (blynk et AZURE) de manière très fluide grâce au documentation disponible sur par chacun des fournisseur de service . Il est important de mentionner que l'éditeur Arduino joue aussi un rôle primordial.il offre la possibilité de télécharger une variété de bibliothèque pour chaque domaine , il est possible aussi de créer sa propre bibliothèque. Comme extension à notre projet il est possible de traiter les données collecter , pour améliorer la précision de notre choix arrosage. Pour ce faire ,on peut utiliser des algorithme de machine learning sur des micro-contrôleur, c est ce qu'on appelle par le TinyML ou le machine learning miniature , adapté au CPU de faible consommation et à utilisation limité . La plateforme TENSORFLOW offre la possibilité de faire ce genre de process en se connectant via une un ESP32 avance ou arduino nano .

## 7 Bibliographie

<https://fritzing.org/>

<https://www.tutos.eu/1296>

<https://azure.microsoft.com/en-us/free/students/>

<https://learn.microsoft.com/en-us/azure/iot-central/>

<https://lucid.app/documents/templates?folder>

<https://towardsdatascience.com/tiny-machine-learning-the-next-ai-revolution-495c26463868>

<https://www.codeproject.com/Questions/1275871/How-do-I-send-arduino-sensor-data-to-SQL-server>

<https://how2electronics.com/getting-started-with-azure-iot-central-using-esp8266/>

<https://github.com/adesolasamuel/NodeMCUtoAzureIoTcentral>

<https://randomnerdtutorials.com/how-to-install-esp8266-board-arduino-ide/>

<https://blynk.io/getting-started>

[https://docs.blynk.io/en/getting-started/supported-boards?\\_gl=1\\*1ev38s\\*\\_g\\*\\_MjA4MzYyOTgyNC4xNjgzMjc3NDA3\\*\\_g\\*\\_aE376ZQ635Y\\*\\_MTY4NTQ3OTYwNy41LjAuMTY4NTQ3OTYwNy4wLjAuMA..](https://docs.blynk.io/en/getting-started/supported-boards?_gl=1*1ev38s*_g*_MjA4MzYyOTgyNC4xNjgzMjc3NDA3*_g*_aE376ZQ635Y*_MTY4NTQ3OTYwNy41LjAuMTY4NTQ3OTYwNy4wLjAuMA..)

<https://docs.blynk.io/en/getting-started/how-to-display-any-sensor-data-in-blynk-app>