

VITINODE

GUIDE D'INSTALLATION



Sommaire

A- Préambule et informations générales.....	1
1- Les bornes LoRaWAN.....	1
2- Trajet du signal.....	1
B- Besoins logiciels.....	2
1- Visual Studio Code (VSCode).....	2
2- Platformio.....	2
3- Télécharger le programme du vitilab.....	3
C- Besoins matériels.....	3
1- Le module ESP32.....	3
2- Le capteur de température.....	3
3- Les résistances.....	4
4- Boîtier sur mesure du vitilab.....	4
5- Batterie et câble micro-usb.....	4
D- Câblage.....	5
1- Câblage des éléments sur le module ESP32.....	5
2- Illustration du câblage.....	5
E- The Things Network.....	6
1- Rejoindre The Things Network.....	6
2- Ajouter la passerelle (gateway).....	7
3- Ajouter un nœud (node).....	7
4- Payload formatter.....	8
5- Webhook.....	8
F- Programmation de l'esp32 ttgo lora.....	9
1- Paramétrage du fichier configuration.h.....	9
2- Test de compilation.....	10
3- Upload : programmation de l'esp32.....	10

End-Device

Connection LoRa

Connection IP

Standard System with a LoRaWAN

2- Trajet du signal

Les données sont envoyées dans l'air par le terminal d'application et les données sont reçues par la passerelle. Ces données sont envoyées à la passerelle par le trajet radio et pour être communiquées le terminal radio qui est relié à la passerelle par le réseau d'application.

Station radio (end device) --- LoRa ---> Gateway ---> Internet / The Things Network /
 les données sont envoyées dans l'air par le terminal d'application et les données sont reçues par la passerelle. Ces données sont envoyées à la passerelle par le trajet radio et pour être communiquées le terminal radio qui est relié à la passerelle par le réseau d'application.

1- L'ensemble des informations est fourni au point de départ du projet de votre système.

<https://github.com/vitilab/Arduino>

2- Source d'inspiration

A- Préambule et informations générales¹

1- Les bornes LoRaWAN²

LoRaWAN est l'acronyme de long-range-wide-area network (que l'on peut traduire par réseau étendu à longue portée).

LoRaWAN est un protocole de communication radio fondé sur la technologie LoRa. Dans le cadre de l'Internet des objets. Il permet de structurer un réseau étendu à basse consommation (LPWAN), intégrant des équipements terminaux à faible consommation électrique par l'intermédiaire de passerelles.

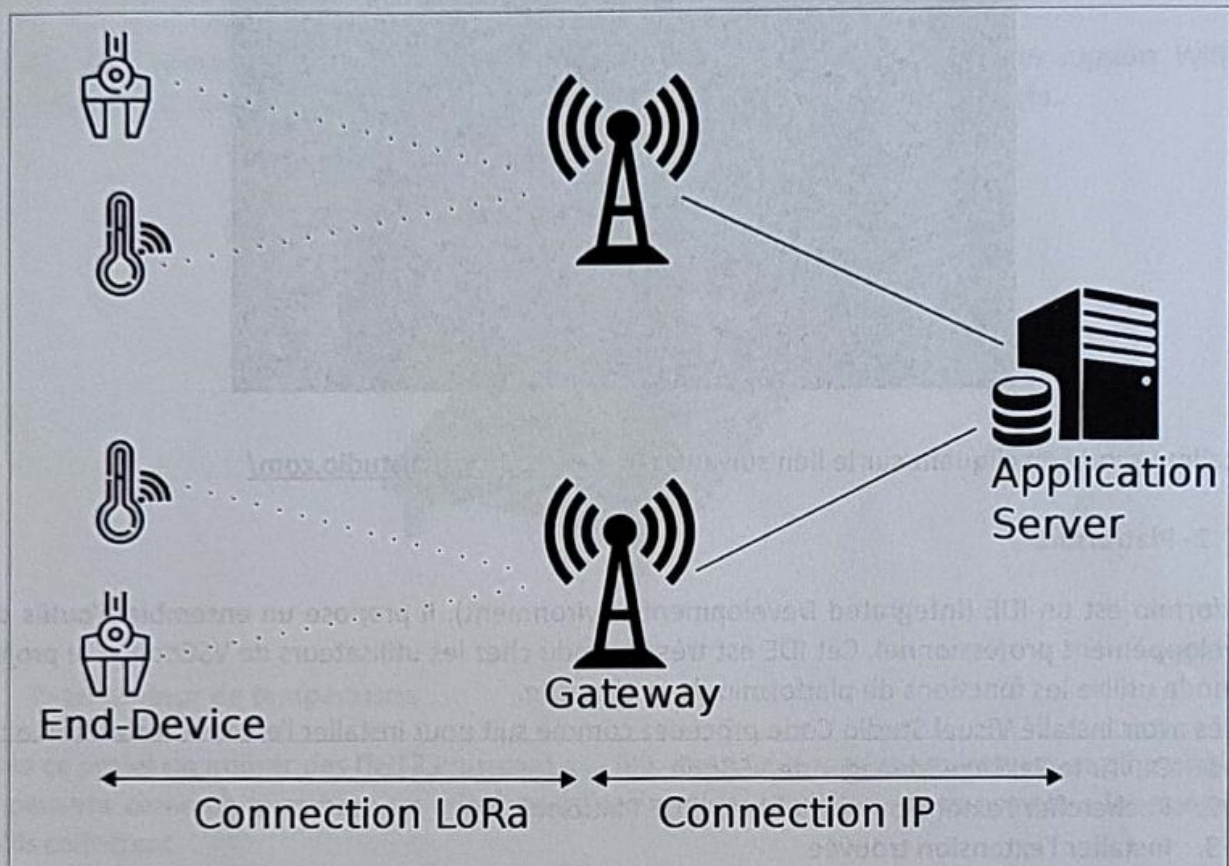


Schéma illustrant une architecture LoRaWAN

2- Trajet du signal

Les différentes étapes concernant l'échange de données entre le serveur d'application et les objets connectés sont récapitulées ci-après. Ces étapes permettent d'avoir une idée plus précise sur le trajet emprunté pour faire communiquer le terminal (quel que soit l'objet connecté) avec le serveur d'application :

Station météo (end device) -----LoRA-----> GATEWAY -----Wifi-----> Internet | TheThingsNetwork |
iot.protechebdo.fr Station météo (end device) -----LoRA-----| | Station météo (end device) -----
LoRA-----| | Station météo (end device) -----LoRA-----|> GATEWAY -----Wifi-----> Internet | Station
météo (end device) -----LoRA-----|

1 L'ensemble des informations est issu du dépôt git du projet dont voici l'adresse :

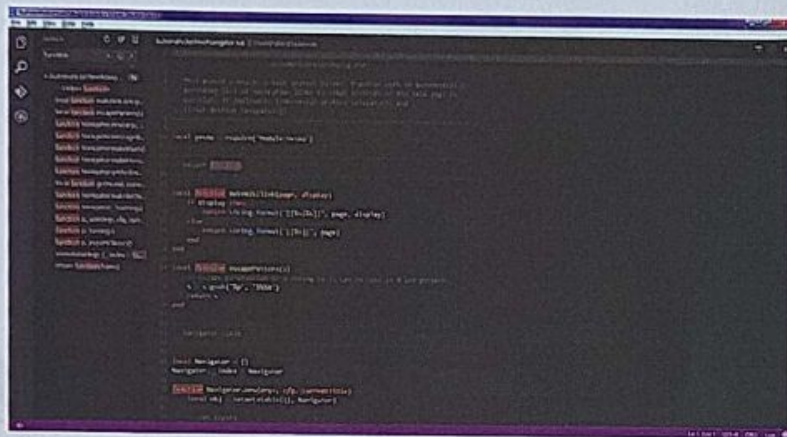
<https://github.com/acetayls/Vitinode>

2 Source : [Wikipedia](#)

B- Besoins logiciels

1- Visual Studio Code (VSCode)

Pour saisir les informations spécifiques à vos besoins, vous aurez besoin de modifier le code du programme que vous pourrez télécharger sur le dépôt git dont l'adresse est fournie page 1. Visual Studio Code est un éditeur de code développé par Microsoft. Il s'agit d'un logiciel gratuit qui installable sur Windows mais aussi Linux ou MacOS.



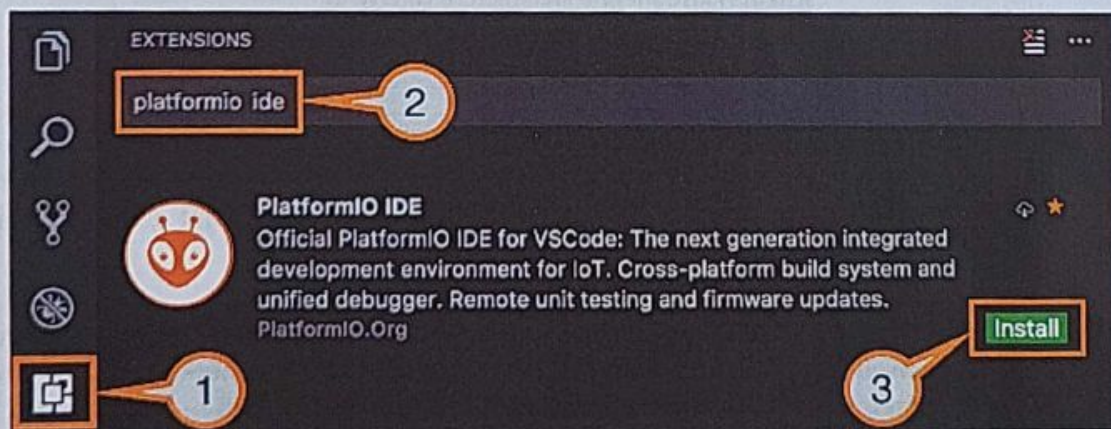
Installez vscode en cliquant sur le lien suivant : <https://code.visualstudio.com/>

2- Platformio

Platformio est un IDE (Integrated Development Environment). Il propose un ensemble d'outils de développement professionnel. Cet IDE est très répandu chez les utilisateurs de VSCode et le projet vitinode utilise les fonctions de platformio dans son code.

Après avoir installé Visual Studio Code procédez comme suit pour installer l'extension platformio :

1. Ouvrir le Package Manager de VSCode
2. Rechercher l'extension officielle de l'IDE Platformio
3. Installer l'extension trouvée



L'image ci-dessus illustre les 3 étapes d'installation de platformio

3- Télécharger le programme du vitilab

Pour que votre objet connecté puisse fonctionner vous aurez besoin du programme développé pour le vitilab.

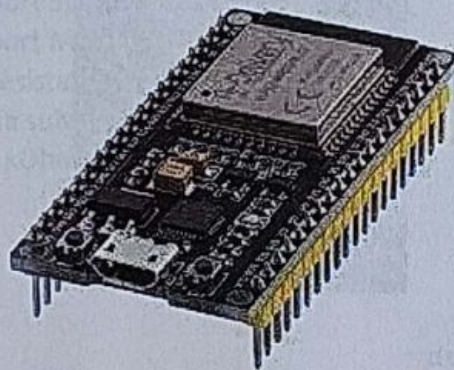
Vous trouverez ce dernier dans le dépôt git du projet ou bien en cliquant sur le lien ci-dessous :

<https://github.com/acetayls/Vitinode/archive/refs/heads/main.zip>

C- Besoins matériels

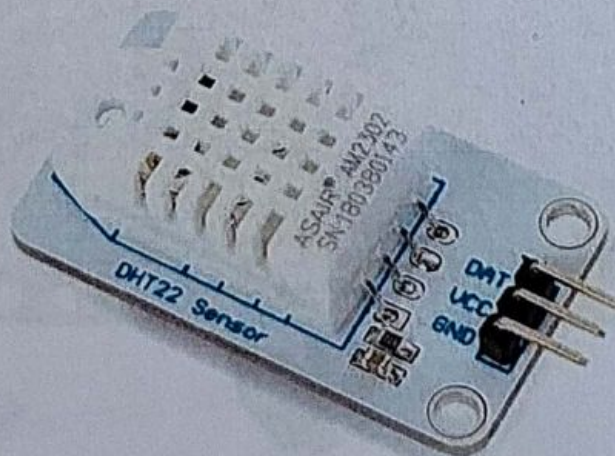
1- Le module ESP32

Il s'agit d'un microcontrôleur (comme arduino ou raspberry) qui dispose d'un support Wifi et Bluetooth. C'est pourquoi il est très utilisé dans le domaine de l'internet des objets.



2- Le capteur de température

Dans ce projet on utilise des DHT22 qui sont à la fois des capteurs de températures et d'humidité. Ils peuvent communiquer avec un microcontrôleur comme l'ESP32 pour transmettre les données qu'ils collectent.



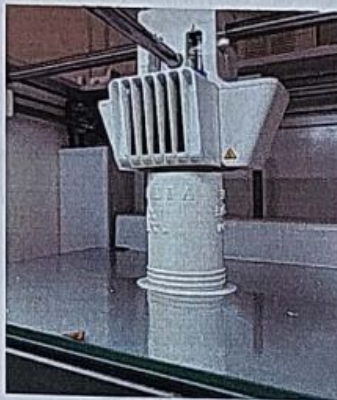
3- Les résistances

Pour ce projet, vous aurez besoin de 2 résistances de 3,3kOhms (niveau de batterie)



4- Boîtier sur mesure du vitilab

Le vitilab imprime grâce à son imprimante 3D un boîtier permettant d'accueillir l'ensemble des éléments matériels nécessaire au fonctionnement de l'objet connecté.



5- Batterie et câble micro-usb

Pour fonctionner, l'objet connecté aura besoin d'une batterie, par exemple une batterie Lipo de 2500mAh et 3,7V



Pour téléverser le programme ou recharger la batterie, un câble micro-usb sera nécessaire



D- Câblage

1- Câblage des éléments sur le module ESP32

Dans cette étape, il faudra souder les différentes broches du DHT22 et les fils de la batterie sur les GPIO (ports d'entrée-Sortie) du microcontrôleur ESP32. Il faudra donc avoir un fer à souder et un peu d'étain pour réaliser cette opération.

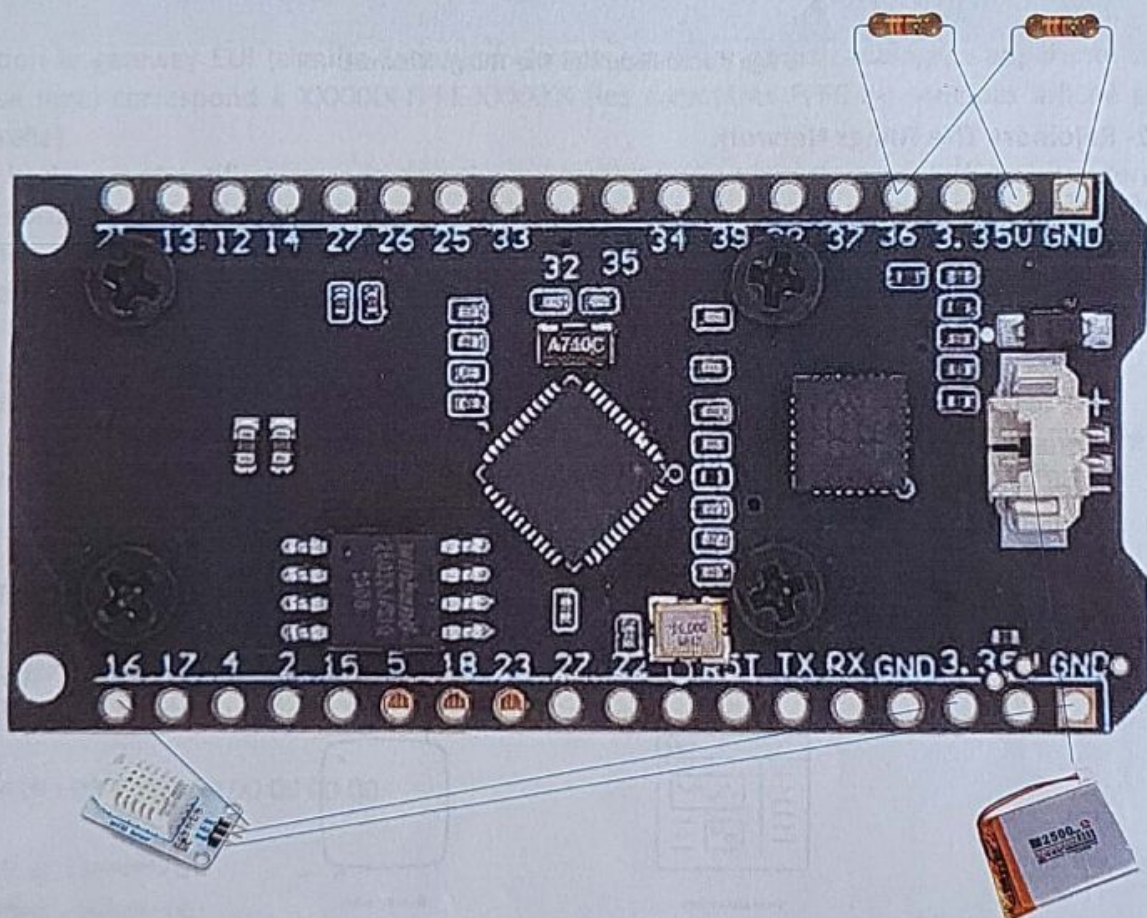
Toutefois, il est possible également d'utiliser une "breadboard" (Carte prototype) pour éviter d'avoir à faire les soudures et ainsi pouvoir faire des tests.

Cependant cette solution ne peut être retenue si on veut mettre le matériel dans le boîtier fourni par le vitilab.

Quoiqu'il en soit, les différentes broches et fils doivent être reliés aux ports (GPIO) du microcontrôleur suivant :

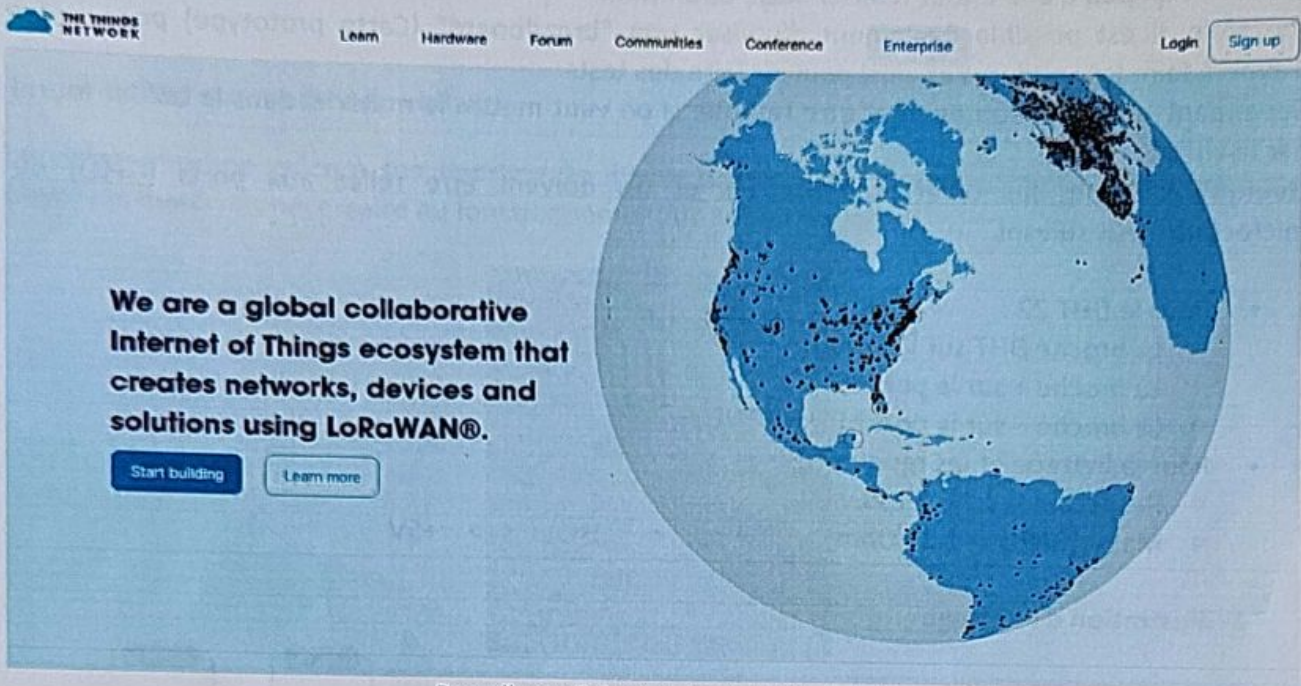
- Pour le DHT 22 :
 - La broche DHT sur le port 16
 - La broche + sur le port 3,3V
 - La broche - sur le port Masse (GND)
- Pour la batterie et les résistances :
 - Respecter le schéma suivant :
 - Masse(GND) <- 3.3 kOhm -> PIN 36 <- 3.3kOhms -> +5V

2- Illustration du câblage



E- The Things Network

The things network est un réseau basé sur la technologie LoRaWAN. Il s'agit d'un projet communautaire et open-source. C'est une communauté internationale qui regroupe plus de 40000 contributeurs dans environ 90 pays dans lesquels sont déployés plus de 4000 passerelles. Il s'agit donc d'un réseau international et ouvert pour l'Internet des Objets (IoT en anglais).



Page d'accueil du site The Things Network

1- Rejoindre The Things Network

Pour faire fonctionner votre objet connecté, vous devrez créer un compte sur le site de la communauté. Procédez comme suit :

- Se rendre sur <https://www.thethingsnetwork.org/>
- Créer un compte à l'aide du bouton "Sign up"
- Activer le compte après validation de l'adresse mail
- Une fois connecté, se rendre sur cliquer sur son compte (en haut à droite) et choisir l'item "Console"
- Choisir la zone Europe1 pour afficher l'image ci-dessous
- Rejoindre l'organisation vitilab-chambragri



2- Ajouter la passerelle (gateway)

Il conviendra d'ajouter une passerelle afin que vos objets connectés puissent envoyer leurs données. Vous devrez alors compléter le formulaire suivant :

THE THINGS STACK Enterprise Overview Applications Gateways Organizations admin

Register gateway

Register your gateway to enable data traffic between nearby end devices and the network.
[Learn more in our Gateway Guide](#)

Gateway EUI

⚠ We detected that your gateway is likely a The Things Indoor Gateway that uses gateway claiming. Use the WiFi password printed on the TTIG as claim authentication code below.

Claim authentication code

Gateway ID

Frequency plan

Attention le gateway EUI (signifie Extended Unique Identifier et par analogie s'apparente à une adresse mac) correspond à XXXXXX-FFFE-XXXXXX (les caractères FFFE ne sont pas affichés sur la passerelle).

Aussi, le claim authentication code est celui qui correspond au mot de passe wifi de la passerelle. Enfin dans la partie Frequency plan, vous devez choisir Europe 863-870 Mhz (SF9 for RX2 – recommended)

Pour plus d'information, vous pouvez consulter [le guide suivant](#).

3- Ajouter un nœud (node)

Vous pouvez alors choisir l'application vini-meteo (elle doit exister si vous avez bien rejoint l'organisation vitilab-chambragri). Sinon vous pouvez en créer une avec le code ci-dessous :

+ Add end device

[X] Enter end device specifics manually

Frequency plan : Europe 863-870 MHz (SF9 for RX2 - recommended)

LoRaWAN version : LoRaWAN Specification 1.0.2

Regional Parameters version : RP001 Regional Parameters 1.0.2

JoinEUI : 00 00 00 00 00 00 00 00

DevEUI : Generate

AppKey : Generate

End device Id : nom facile à identifier

4- Payload formatter

Pour communiquer avec le serveur <https://iot.protechebdo.fr/devices>, les data des stations doivent être mise en forme. On utilise un uplink payload formatter dans ttn. Il est programmé par défaut dans l'application TTN. Voici le code :

```
function decodeUplink(input) {  
  var data = {};  
  var warnings = [];  
  
  if (input.fPort == 10) {  
    if (input.bytes[0] == 1) {  
      data.sonde = input.bytes[1] + input.bytes[2]/100;  
      data.battery = input.bytes[3] + input.bytes[4]/100;  
    }  
    else if (input.bytes[0] == 2) {  
      //data.counter = (input.bytes[0] << 8) + input.bytes[1];  
      data.temperature = input.bytes[3] + input.bytes[4]/100;  
      data.humidity = input.bytes[1] + input.bytes[2]/100;  
      data.battery = input.bytes[5] + input.bytes[6]/100;  
    }  
  }  
  else {  
    warnings.push("Unsupported fPort");  
  }  
  return {  
    data: data,  
    warnings: warnings  
  };  
}
```

5- Webhook

Un webhook est une fonction de rappel basée sur le protocole HTTP. Il permet à 2 API (Interface de programmation d'application) de communiquer entre elles. Ainsi, dans notre cas, le webhook permet la communication entre TTN et le site protechebdo

F- Programmation de l'esp32 ttgo lora

Dans cette partie, on détaillera les éléments du code à modifier afin que l'objet connecté puisse remplir son rôle et transmettre des données.

1- Paramétrage du fichier configuration.h

Avant de modifier ce fichier, vous devez réaliser une sauvegarde en dupliquant le fichier configuration.h en configuration_backup.h

Une fois cette étape réalisée, il faut renseigner les clé qui doivent correspondre à ce qui se trouve sur TTN concernant l'objet connecté. Voici en noir (gras), les éléments de la partie du code à modifier :

```
const char* devEui = "70B3D57ED0054974"; // Change to TTN Device EUI
const char* appEui = "0000000000000000"; // Change to TTN Application EUI
const char* appKey = "ED52F1573C19890A23FE023A273A7D3B"; // Change to TTN App Key
```

Le code doit alors avoir la forme suivante :

```
/*
 *
 * CHANGER LES VALEURS SUIVANTE selon le configuration !!!!
 *
 */

// identifiant du node TTN
const char* devEui = "70B3D57ED0054974"; // Change to TTN Device EUI
const char* appEui = "0000000000000000"; // Change to TTN Application EUI
const char* appKey = "ED52F1573C19890A23FE023A273A7D3B"; // Change to TTN App Key

// selection station meteo (2) ou vitibox (1)
#define VITI_TYPE 1 //si vitibox : 1 --> sonde DS18B20 sinon 2 --> sonde DHT22

// Temps entre les mesures (en secondes)
const unsigned long secs_between_send = 60*30;

// Coefficient de la batterie (en fonction des valeurs des resistances, mesure sur le terrain)
const double batt_coeff = 0.001074481;

/*
 *
 * FIN des valeurs à modifier !
 *
 */
```


2- Test de compilation

Pour vérifier que le programme s'exécutera correctement, il convient d'effectuer un test de compilation.

Pour ce faire, il suffit de cliquer sur build (dans VS Code, symbole valider dans la barre du bas).

Si le build réussi, cela signifie que le programme s'exécutera sans erreur.

Si besoin vous pouvez supprimer les fichiers suivants :

- .pio/libdeps/ttgo-lora-v32v1/TTTN_esp32/src/TTN_BLE_ESP32.cpp
- .pio/libdeps/ttgo-lora-v32v1/TTTN_esp32/src/TTN_BLE_ESP32.h

3- Upload : programmation de l'esp32

Une fois que vous êtes sûr que le programme s'exécutera correctement, vous devez brancher le module esp32 à l'aide du câble micro usb.

Il suffit alors de cliquer sur la flèche – upload (barre du bas) pour téléverser le programme modifié dans le module esp32.