

Notice de mise en place du projet

Nous avons deux codes distincts sur le git :	2
Etape 1 : Installer vs code :	2
Etape 2 : ajout de l'extension	2
Etape 3 : Modification des liens	3
Etape 4 : Clear le build	3
Etape 5 : Branchements	4
Etape 6 : Lancer, compiler et flasher le projet	7
Etape 7 : Lecture des données sur NRFConnect	8

Nous avons deux codes distincts sur le git :

PFE_BLE_ALLSensorGit qui lui contient le programme qui envoie toutes les données de tous les capteurs en BLE.

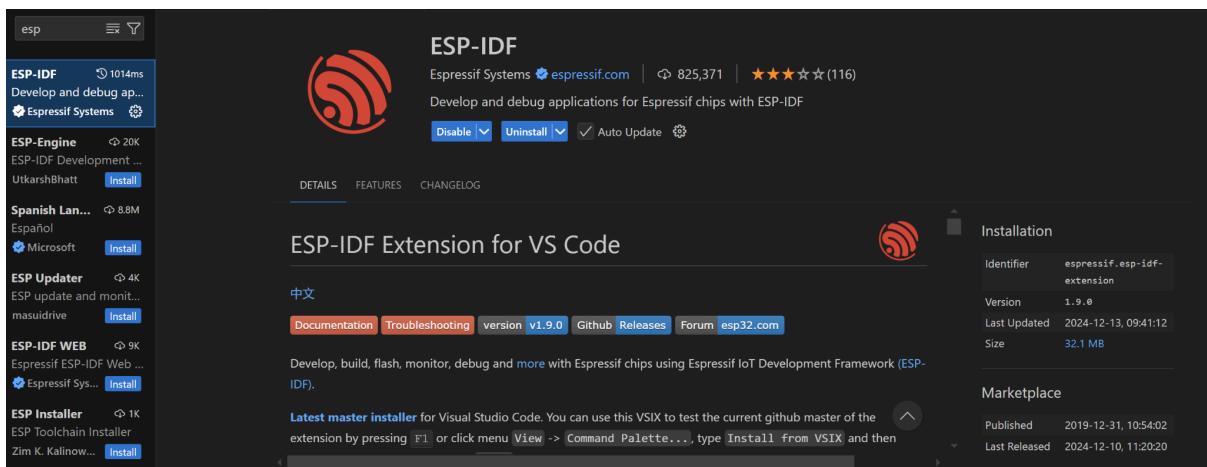
Comp_APP_Code : qui est compatible avec la version laissée de l'application et qui quant à lui envoie des données aléatoire et du capteur cardiaque.

Etape 1 : Installer vs code :

<https://code.visualstudio.com/>

Etape 2 : ajout de l'extension

Aller dans extension :

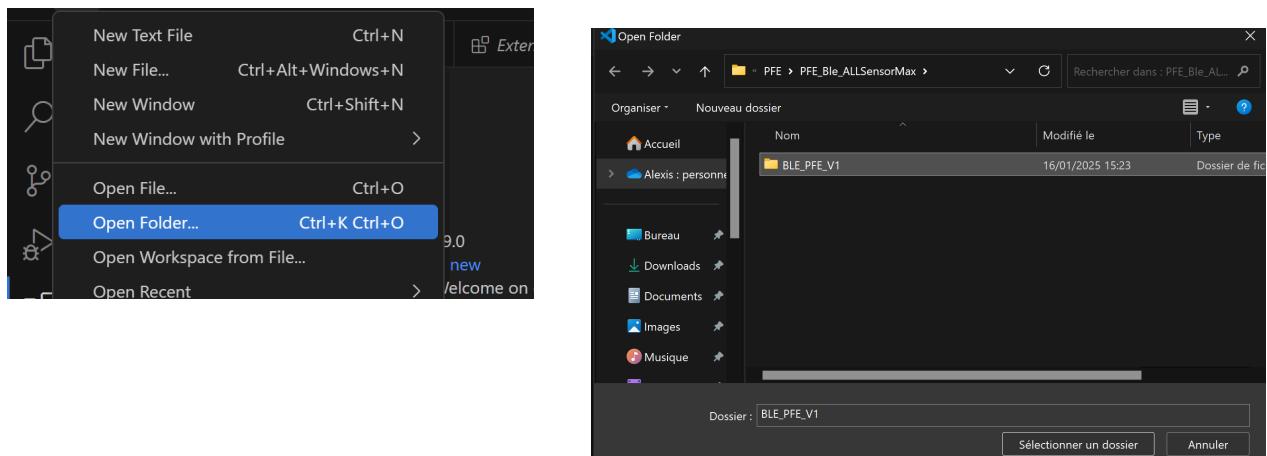


Rechercher ESP-IDF, télécharger la version 5.3.1 pour être sûr d'être compatible au projet.

Télécharger le dossier sur GitHub nommé : **PFE_BLE_ALLSensorGit** ou **Com_APP_Code** selon votre choix.

Décompresser ce dossier.

Ouvrir le dossier se situant dans le dossier précédent nommé **BLE_APPV1** dans VSCode.



Etape 3 : Modification des liens

Dans le fichier `c_cpp_properties.json`, modifier les liens afin de correspondre à votre environnement. Sinon la compilation ne va pas se faire.

```

{
    "configurations": [
        {
            "name": "Win32",
            "includePath": [
                "${workspaceFolder}/**",
                "C:/Users/havar/esp/v5.3.1/esp-idf/**",
                "C:/Users/havar/esp/v5.3.1/esp-idf/components/nvs_flash/include",
                "C:/Users/havar/esp/v5.3.1/esp-idf/components/freertos/FreeRTOS-Kernel-SMP/portable/linux/include/freertos",
                "C:/Users/havar/esp/v5.3.1/esp-idf/components/soc/esp32/include",
                "C:/Users/havar/esp/v5.3.1/esp-idf/components/bt/host/nimble/esp-hci/include",
                "${workspaceFolder}/components/esp32-max30100/include/max30100",
                "C:/Users/havar/esp/v5.3.1/esp-idf/components/linux/linux_include",
                "C:/Users/havar/esp/v5.3.1/esp-idf/components/linux/include"
            ],
            "defines": [
                "_DEBUG",
                "UNICODE",
                "_UNICODE"
            ],
            "compilerPath": "C:\WinGW\bin\gcc.exe",
            "cStandard": "c11",
            "cppStandard": "gnu++14",
            "intelliSenseMode": "windows-gcc-x86"
        }
    ]
}

```

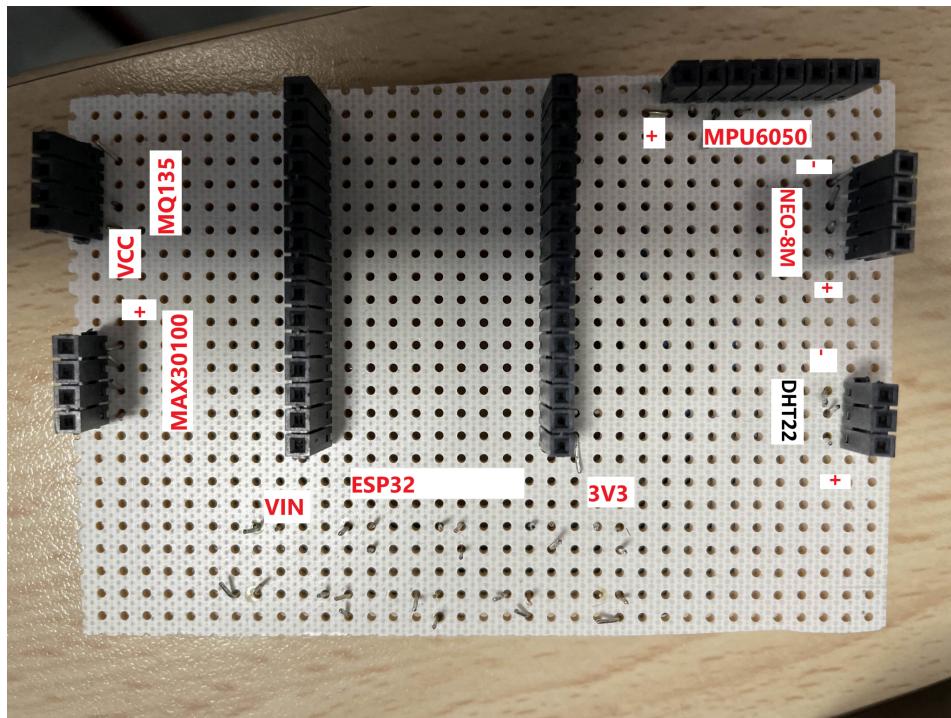
Etape 4 : Clear le build

Il faut penser à **clear le build** présent (l'icône corbeille).

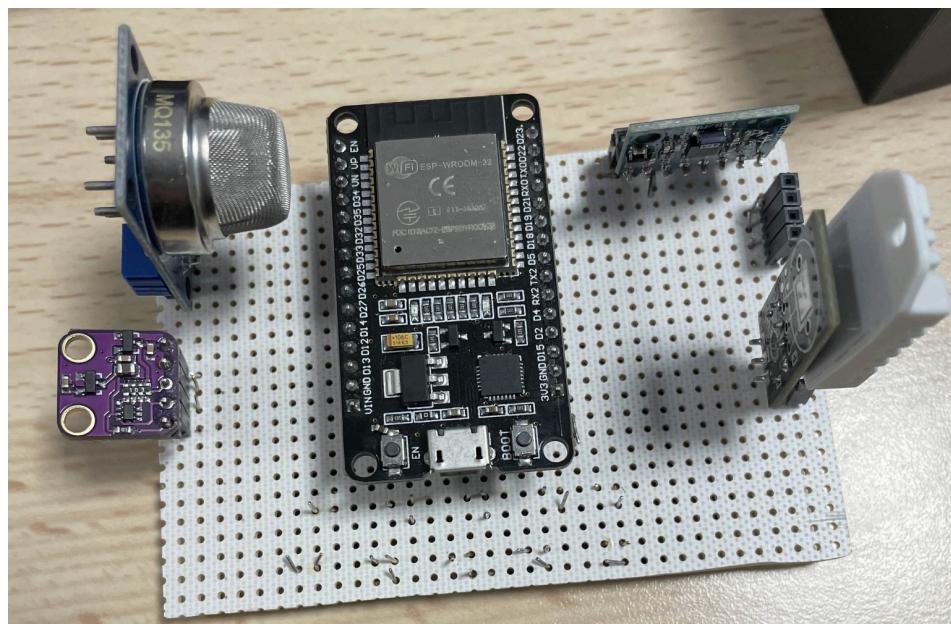


Etape 5 : Branchements

Une fois le build clear, branchez l'ESP en respectant les branchements adéquat selon ce schéma :



Cela nous donne la maquette suivante :



Ports associés à la devkit :

EN - 1	D23 - 16
VP - 2	D22 - 17
VN - 3	Tx0 - 18
D34 - 4	Rx0 - 19
D35 - 5	D21 - 20
D32 - 6	D19 - 21
D33 - 7	D18 - 22
D25 - 8	D5 - 23
D26 - 9	Tx2 - 24
D27 - 10	Rx2 - 25
D14 - 11	D4 - 26
D12 - 12	D2 - 27
D13 - 13	D15 - 28
GND - 14	GND - 29
Vin - 15	3v3 - 30

Liste composants et leurs ports :

- Max30100 :
 - Vin → 3v3 (30)
 - Gnd → GND (29)
 - Scl → D25 (8)
 - Sda → D26 (9)
- MPU6050 :
 - Vin → 3v3 (30)
 - Gnd → GND (29)
 - Scl → D22 (17)
 - Sda → D21 (20)
- MQ135 :
 - Vin → 3v3 (30)
 - Gnd → GND (29)
 - A0 → VP (2)
- DHT 22 :
 - Vin → 3v3 (30)
 - Gnd → GND (29)
 - Out → D4 (26)
- Neo-8M :
 - Vin → 3v3 (30)
 - Gnd → GND (29)
 - Rx → Tx2 (24)
 - Tx → Rx2 (25)

Etape 6 : Lancer, compiler et flasher le projet

Choisir le bon port de communication ici je suis sur le port COM3 :

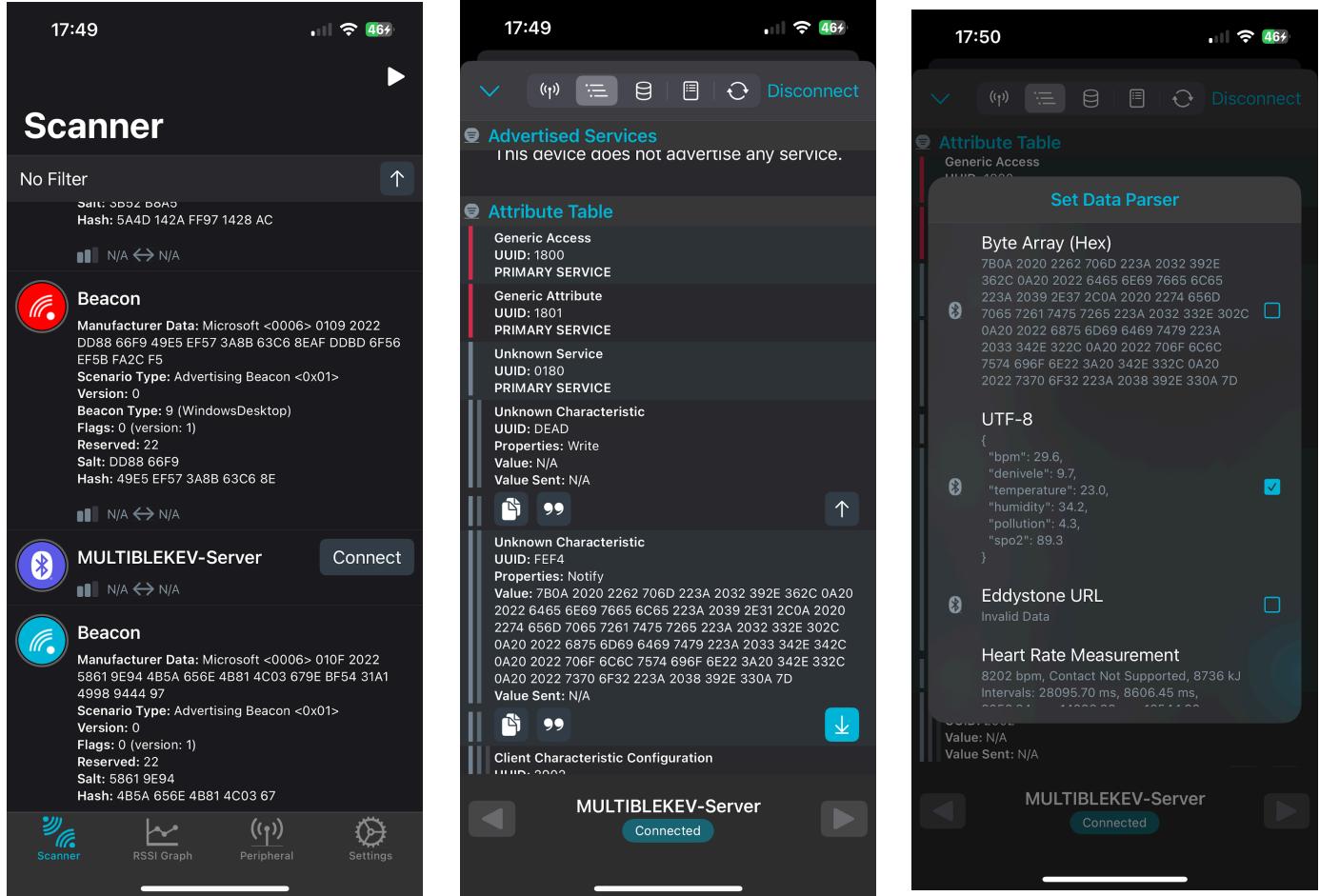


Puis construire, compiler et flasher le programme en appuyant l'icône flamme.

Les données devraient arriver d'une part dans la console.

Etape 7 : Lecture des données sur NRFConnect

Pour vérifier le bon fonctionnement du BLE, une application nommée NRFConnect permet de se connecter à l'ESP32 et de vérifier plus simplement à l'UUID FEF4 (2222 sur la version finale, ici c'était notre version expérimentale) les données arrivant au format JSON.



Étape 1 : Trouver l'ESP, il sera nommé BLE-Random (ici nous c'était MULTIBLEKEV-Server).

Étape 2 : Se connecter

Étape 3: aller dans le second onglet appuyer sur le flèche en bleu vers le bas pour lire ce qui se passe à l'adresse FEF4(2222 sur la version finale).

Étape 4 : appuyer sur les deux "" afin de changer le mode de lecture des datas en UTF-8

Les données devraient dorénavant s'afficher en temps réel sur votre smartphone.