

UNIVERSITE MOHAMMED PREMIER

ECOLE SUPERIEURE DE TECHNOLOGIE OUJDA

DEPARTEMENT GENIE INFORMATIQUE

FILIERE : GENIE INFORMATIQUE EMBARQUEE

Rapport de projet de fin d’étude s :

Contribution à la réalisation d’un prototype de système de détection de feux de forêt par capteurs.

Réalisé par :

o CHARMOUH Abdellah

o AMAR Mohammed

o AMRANI Hasnae

o AZZEDDINE Mohammed

Encadré par :

o M. Omar MOUSSAOUI

Soutenu le 20 Avril 2022 devant le jury composé de :

M. H. Ouariachi, M. M. Moussaoui, M. O. Moussaoui

Invitée : Mme. O. Mtafi

Année universitaire : 2021/2022

Dédicace :

• Nous dédions ce travail :

o A nos chers parents ,

o A nos chères sœurs et nos chers frères ,

o A nos familles ,

o Enfin à tous nos amis et collègues

Remerciements :

• Au terme de ce travail nous tenons à remercier :

o Dieu, le tout puissant de nous avoir donné la force, le courage et la

volonté pour réaliser ce travail,

o Notre encadreur Monsieur :

Omar MOUSSAOUI pour son soutien, sa disponibilité, l’effort et la

qualité d’encadrement dont il a dépensé pour nous et surtout ses

constructifs conseils qui ont alimenté notre réflexion tout au long du

réalisation du projet,

o Les membres de jury, de nous avoir fait l’honneur d’évaluer notre

travail.

o Tous les professeurs qui nous ont suivis durant notre formation,

ainsi que tout le corps professoral et administratif de l’Ecole

supérieure de technologie d’Oujda.

Résumé :

Le but de ce projet est celui d’étudier et implémenter un SDF -RCSF pour détecter les

déclenchements des incendies de forêts en temps réel, et non après des heures et des jours,

comme c’est le cas des méthodes actuelles basées sur l’imagerie thermique et la surveillance

par satellites, ou celles- ci traditionnelles basées sur les observations humaine (à titre d’exemple

les gardes forestiers …). Ce système doit détecter les déclanchements des feux et définir

l’intensité du danger selon les informations collectées . Ensuite, après la détection d’un ou

plusieurs événements (changements inhabituels) considérés dangereux (seuil élevé, détection

de flemmes, taux d’humidité …), l’information sera transmise via le réseau déployé tout en

assurant une longue durée de vie de ce réseau.

Les nœuds de capteurs utilisés sont capables de collecter des grandeurs physiques telles que

la température l’humidité, les gaz spécifique émis par les feux de forêt, adossés de la technologie de communication sans fil e t d’autres protocoles de communication afin

d’acheminer les informations et les données collectées vers le cloud où elles seront traitées et analysées.

Table des matières :

Dédicace**…………………………………………………………...………………………..02**

Remerciements**……………………………………………………………………………...03**

Résumé**……………………………………………………………………………………...04**

Table des matières**…………………………………………………………………………05**

Liste d’abréviation**…………………………………………………………………………07**

Liste de figures**…………………………………………………………………………….08**

Introduction générale**………………………………………………………………………10**

**CHAPITRE 1 : description, conception et modélisation du projet………………….....11**

1 Introduction**…………………………………………………………………………....11**

2 Le SDF-RCSF**................................................................................................................11**

2.1 La description du système à réaliser**……………………………………………...11**

3 Le prototype du SDF-RCSF**………………………………………………………….12**

3.1 Analyse des besoins du système **………………………………………………….12**

3.1.1 Exigences Fonctionnelles**………………………………………………….....12**

3.1.2 Vision de la solution proposée**……………………………………………….12**

**CHARITRE 2 : Matériels et plateforme utiliser pour la réalisation du SDF-RCSF ....13**

1. Introduction**……………………………………………………………………………13**

2. Les nœuds de ca pteurs (End-Devices)**………………………………………………..13**

2.1 La carte TTGO LoRa32 SX1276 OLED**………………………………………….13**

2.2 La programmation de la TTGO LoRa32 SX1276 OLED**…………………………15**

2.3 Les capteurs utilisés**……………………………………………………………….**16

2.3.1 Capteur d’humidité et de température DHT22 (AM2302)**…………………...17**

2.3.2 Le capteur du gaz CO2 (MG-811)**…………………………………………...18**

2.3.3 Le Capteur de gaz CO ME2**………………………………………………….19**

3. Description de la carte Raspberry pi 4 **………………………………………………..21**

**CHARITRE 3 : Réalisation du SDF-RCSF…………………………………………….23**

1. Installation de la carte ESP32 dans Arduino IDE (Windows, Mac OS X, Linux)**….23**

1.1 Installation du module complémentaire ESP32 dans Arduino IDE**……………….23**

1.2 Test de l’installation**……………………………………………………………….27**

1.3 Conclusion**…………………………………………………………………………31**

2. ESP32 avec capteur de température et d’humidité DHT11/DHT22 utilisant Arduino

IDE**………………………………………………………………………………………..31**

2.1 Brochage DHT**…………………………………………………………………….31**

2.2 Installation de bibliothèques**………………………………………………………32**

3. Connecter esp32 avec raspberry par Bluetooth**…………………………………….35**

3.1 Installez pybluez pour Python3, entrez la commande dans Terminal**……………...37**

3.2 L’affichage et l’envoie des informations des capteurs par bleutooth à la carte

Raspberry**…………………………………………………………………………………..37**

4. Connexion entre l’esp32 et la Raspberry Pi4**……………………………………...39**

5. Utilisation de la plate-forme Thingspeak**…………………………………………..44**

5.1 Description de thingspeak**………………………………………………………….44**

Conclusion générale**………………………………………………………………………..47**

Annexe : Brochage et exécution du SDF**…………………………………………………..48**

Bibliographie**……………………………………………………………………………….51**

***Liste d’abréviation :***

SDF : Système détecteur de feux

RCSF : Réseau de capteur sans fil

MQTT: MQ Telemetry Transport

IOT: internet of things

WIFI: Wireless Fidelity

DHT : Capteur d’humidité et température

***Liste de figures :***

Figure 1 : brochage de la carte ESP32 (TTGO LoRa32 SX1276)**……………………………………………....15**

Figure 2 : DHT22**………………………………………………………………………………………………..17**

Figure 3 : le capteur MG-811**…………………………………………………………………………………....19**

Figure 4 : le capteur ZE07\_CO ME2**………………………………………………………………………… ...20**

Figure 5 : Raspberry pi4**………………………………………………………………………………………....21**

Figure 6 : Schéma de RCSF**……………………………………………………………………………………..23**

Figure 7 : La 1ère étape de connexion d’esp32 avec IDE**………………………………………………………..23**

Figure 8 : La 2ème étape de connexion d’esp32 avec IDE**……………………………………………………….24**

Figure 9 : Remarque de la 2ème étape de connexion d’esp32 avec IDE**………………………………………....24**

Figure 10 : La 3ème étape de connexion d’esp32 avec IDE**……………………………………………………...25**

Figure 11 : La 4ème étape de connexion d’esp32 avec IDE**……………………………………………………...26**

Figure 12 : La dernière étape de connexion d’esp32 avec IDE**…………………………………………………26**

Figure 13 : La 1ère étape de test de connexion d’esp32 avec IDE**……………………………………………….27**

Figure 14 : La 2ème étape de test de connexion d’esp32 avec IDE**……………………………………………....28**

Figure 15 : La 3ème étape de test de connexion d’esp32 avec IDE**………………………………………………29**

Figure 16 : La 4ème étape de test de connexion d’esp32 avec IDE**……………………………………………....30**

Figure 17 : La dernière étape de test de connexion d’esp32 avec IDE**………………………………………….31**

Figure 18 : Câblages de DHT22**………………………………………………………………………………...32**

Figure 19 : La 1ère étape d’installation de la bibliothèque DHT**………………………………………………...33**

Figure 20 : La 2ème étape d’installation de la bibliothèque DHT**………………………………………………..33**

Figure 21 : La 3ème étape d’installation de la bibliothèque DHT**………………………………………………..34**

Figure 22 : Le code d’installation de DHT**……………………………………………………………………...34**

Figure 23 : Le code de fonctionnement de DHT**………………………………………………………………..35**

Figure 24 : La 1ère étape de connexion d’esp32 avec la Raspberry par Bluetooth**……………………………...35**

Figure 25 : La 2ème étape de connexion d’esp32 avec la Raspberry par Bluetooth**……………………………..36**

Figures 26 et 27 : Code source de la connexion d’esp32 avec la Raspberry par Bluetooth**………………….…36**

Figures 28, 29 et 30 : Code source d’installation de bibliothèque Pybluez**……………………………………..37**

Figure 31 : Installation des bibliothèques CO2, DHT et Bluetooth**……………………………………………..38**

Figures 32, 33, 34 et 35 : Code source de l’affichage et l’envoie des informations des capteurs par Bluetooth à la carte Raspberry**……………………………………………………………………………………………….38-39**

Figure 36 : Schéma résumé des étapes de l’affichage et l’envoie des informations des capteurs par Bluetooth à la carte Raspberry**…………………………………………………………………………………………………..39**

Figure 37 : Connexion de l’esp32 avec la Raspberry**……………………………………………………………40**

Figures 38,39 et 40 : Code source de la carte esp32 Sender**………………………………………………….40-41**

Figures 41, 42 et 43 : Code source de la carte esp32 Receiver**……………………………………………….41-42**

Figures 44 et 45 : Code source de l’adresse Mac d’esp32**……………………………………………………42-43**

Figures 46, 47 et 48 : Code source de mix receiver et Bluetooth**…………………………………………….43-44**

Figure 49 : Fonctionnement de Thingspeak**……………………………………………………………………...45**

Figures 50, 51, 52 et 53 : Code source de la connexion de Raspberry et Thingspeak**…………………………...46**

Figures 54, 55 et 56 : Câblage de matériels de SDF-RCSF**…………………………………………………..48-49**

Figure 57 : Exécution finale de projet**……………………………………………………………………………50**

Introduction générale :

Ce rapport est la synthèse de notre travail dans le cadre de projet de fin d’études sous l’encadrement de M. Omar MOUSSAOUI.

Notre projet consiste à réaliser un réseau de capteur sans fil qui s’intitule « Détection des feux de forêt à base d’un réseau de capteurs sans fil ou SDF-RCSF »

Ces dernières années, grâce aux récents développements réalisés dans le domaine de

l’informatique, la microélectronique e t les technologies de communication, le monde a

rapidement été couvert par des réseaux sans fil, ces réseaux permettent aux appareils

numériques de s'interconnecter et de transmettre les données entre eux, les avancés ont permis

aussi de produire des composants de petite taille appelés des capteurs qui sont désormais

omniprésents et collectent et transmettent d'énormes quantités d’informations. Ces données

peuvent être stockées et analysées à une date ultérieure, ou être analysées et exploitées

immédiatement (en temps réel). Les capteurs peuvent être installés dans les maisons, sur les

feux de circulation, dans les champs agricoles et mêmes sur nos corps, dans notre projet intitulé

système de détection des feux de forêts (noté : SDF à base de RCSF), ces petites unités vont être

installées dans les forêts sous forme des nœuds (dits nœuds de capteurs) . Ils vont, en

collaboration avec d’autres équipements, former des Réseaux de capteurs sans fil (noté RCSF) ,

les données issues des capteurs sont analysées et utilisées par des logiciels et des plateformes.

Le but de notre projet SDF-RCSF est de trouver une nouvelle méthode pour la lutte contre les

feux de forêt. Cette méthode consiste à mettre en place un réseau de capteurs capable de

recueillir et de traiter des informations environnementales (température, humidité, etc.)

provenant d’une zone cible couverte. Ce réseau, est déployé afin d’assurer la détection des feux

en temps réel et permettre ainsi une intervention rapide pour sauvegarder l’écosystème forestier.

A cet effet, ce rapport s’articul e autour de quatre chapitres :

❖ Le chapitre 1 : Description, conception et modélisation du projet.

❖ Le chapitre 2 : Matériels et plateformes utiliser pour la réalisation du SDF-RCSF

❖ Le chapitre 3 : Réalisation du SDF-RCSF.

❖ L’Annexe : Brochage et exécution du SDF-RCSF.

Le rapport se termine par une conclusion générale et les travaux en perspective.

CHAPITRE 1 : description,

conception et modélisation du projet .

.

1 Introduction :

Avant d’aller à l’étape de l’implémentation et la réalisation du projet, la phase de

la conception du système à réaliser est très importante afin de bien comprendre et cerner la

problématique du projet, dans ce contexte, pour bien concevoir un SDF en temps réel basé sur

les RCSF, il est nécessaire de bien choisir la solution technologique adéquate. Cette solution

est inspirée des solutions techniques existantes utilisées dans ce domaine et vise à répondre aux

besoins et aux exigences du cahier des charges fonctionnel.

2 Le SDF-RCSF :

Dans un système basé sur les RCSF, Les capteurs doivent être connectés au réseau pour que

les données collectées puissent être stockées et partagées. Nous aurons donc besoin de les

connecter à un microcontrôleur, généralement on utilise des cartes de développement à base de

microcontrôleur, l’ensemble constitue les nœuds du réseau, qui sont des unités d’acquisition,

de traitement, et de transmission de données vers des passerelles (appelées aussi stations de

base,) qui peuvent ; elles aussi être des cartes de développement puissantes, elles permettent la

récupération des données et l’interconnexion avec d’autres réseaux. Ils peuvent être à même de

prendre des décisions immédiates ou ils peuvent envoyer ces données au plateformes Cloud,

ces plateformes permettent l’interfaçage et la présentation des données , envoyées par les

passerelles, pour l’utilisateur, ou à un ordinateur plus puissant en vue d'une analyse. Cet

ordinateur peut se trouver dans le même réseau local que le contrôleur ou n'être accessible que

par l'intermédiaire d'une connexion Internet.

2.1 La description du système à réaliser :

Afin de mettre en œuvre un système de détection des feux de forêt basé sur les réseaux de

capteurs sans fil, un ensemble des nœuds de capteurs

sont requis pour couvrir la zone de surveillance, ces capteurs doivent avoir la possibilité de

mesurer l’humidité, la température ambiante et le taux des gaz issus des feux et communiquer

sans fil ces mesures vers les passerelles qui vont eux même les transmettre aux serveurs à

14

travers un système de transport dédié développé dans la cadre de ce projet. Ces capteurs sont

dispersés dans la région observée et doivent respecter l’architecture sous -jacente de notre

système basée sur le clustering.

3 Le prototype du SDF-RCSF :

3.1 Analyse des besoins du système

3.1.1 Exigences Fonctionnelles

Ce sont les exigences liées au fonctionnement du système à développer. Le système de détection

des feux de forêts devra être capable d’établir les fonctionnalités suivantes :

• Transmettre des données via une technologies de communication .

• Implémenter de protocole MQTT pour la communication (broker, Publisher,

subscribers).

• Acquérir les mesures des grandeurs physiques issues des capteurs utilisés.

• Assurer la sécurité du système.

• Permettre la consultation des données.

• Permettre la visualisation des données sur un smart phone.

• Permettre la visualisation des données sur un site web.

• Pouvoir transmettre et visualiser les données sur une plateforme IOT.

• Détecter des feux et envoyer des alertes.

3.1.2 Vision de la solution proposée :

Notre solution prend en compte les valeurs transmises par les différents dispositifs situés

dans l'environnement cible la détection des feux de forêts, pour effectuer le suivi de données

(température, humidité, taux d es gaz carboniques, …) en temps réel d’une façon fiable et

adéquat. Le système à proposer est basée sur le paradigme des systèmes d’IoT. En outre, l’usage

des technologies et protocoles de communication sans fil (Bluetooth, wifi) .

La solution envisagée est celle des réseaux de capteurs sans fils. En effet, les cartes de

développement associées aux capteurs sont considérées comme des nœuds du réseau qui

permettent de me surer en permanence les paramètres et de lancer une alerte en cas d’incendie.

Chapitre 2 : Matériels et plateformes utiliser pour la réalisation du SDF-RCSF

1 Introduction :

Dans ce chapitre, nous allons présenter notre approche de détection des feux de forêt

en détaillant la méthodologie adoptée pour réaliser notre système de détection des feux

de forêt appelé SDF-RCSF ( Système de détection des feux de forêts à base de réseau

de capteurs sans fil ). Nous détaillons ensuite, les différentes parties qui constituent ce

système de point de vue équipements – matériel – et de point de vue logiciel.

2 Les nœuds de capteurs (End -Devices) :

Le rôle principal des nœuds de capteurs est celui de collecter les données et de les transmettre

par la suite aux Gateways (passerelles) via la technologie LoRa. Dans ce travail les nœuds sont

composés des cartes à microcontrôleurs TTGO LoRa32 SX1276 OLED et des capteurs,

l’ensemble est alimenté par des batteries puissantes.

2.1 La carte TTGO LoRa32 SX1276 OLED :

◆ Description de la carte :

La carte est un module complet et parfait pour les systèmes intelligents. Le module se compose

d'un Espressif ESP32, d'un Semtech SX1276 et d'un écran OLED. Il dispose des connectivité

Wi-Fi, Bluetooth et BLE, et LoRa pour cela il est livré avec une antenne. Le module OLED

SX1276 dispose de protocoles ADC, SPI, I2C et UART haute résolution pour la communication

d'informations. Il est embarqué avec 32 bits de processeur LX6 et une SRAM de 540 Ko. Le

module est fourni avec un connecteur de batterie intégré pour fournir l'alimentation et un

connecteur USB pour associer le module directement à l'ordinateur pour le codage.

22

◆ Principales caractéristiques de TTGO LoRa32 SX1276 OLED :

ESP32 :

• CPU : Xtensa 32-Bit LX6 Dual Core

• Horloge : de 80 à 240 MHz

• Mémoire ROM : 448 Ko

• Mémoire SRAM : 520 Ko

• Mémoire Flash : 32 Mo

• Interface : SD, UART (3 canaux), SPI (3 canaux), SDIO, I2C (2 canaux), I2S (2

canaux), LED IR PWM (2 canaux) et PWM (3 canaux) ;

• GPIO : Digital IO (36), ADC 12 bits (16 canaux), DAC 8 bits (2 canaux), capteur

capacitif (10 canaux) ; LNA ;

WiFi :

• 802.11 b/g/n : da 2,4 a 2,5 GHz

• Data TX : 150 Mbps @ 11n HT40., 72 Mbps @ 11n HT20, 54 Mbps @ 11g, 11 Mbps

@ 11b

• Puissance TX : 19 ,5dBm @ 11b, 16,5 dBm @ 11g, 15,5 dBm @ 11n

• Sensibilité : -98dBm

• Sécurité WiFi : WPA / WPA2 / WPA2-Enterprise / WPS ;

• Crypto WiFi : AES/RSA/ECC/SHA ;

Bluetooth :

• 4.2 BR / EDR e BLE (Bluetooth Low Energy)

• Écran OLED :

• Bluetooth 0,96” SSD1306

LoRa™ :

• Fréquence : 868 MHz

• Modulation : FSK, GFSK, MSK, GMSK, LoRa TM, OOK

• Sensibilité : -139dBm

Puissance : 3,3 V 7 V

◆ Brochage de TTGO LoRa32 SX1276 OLED :

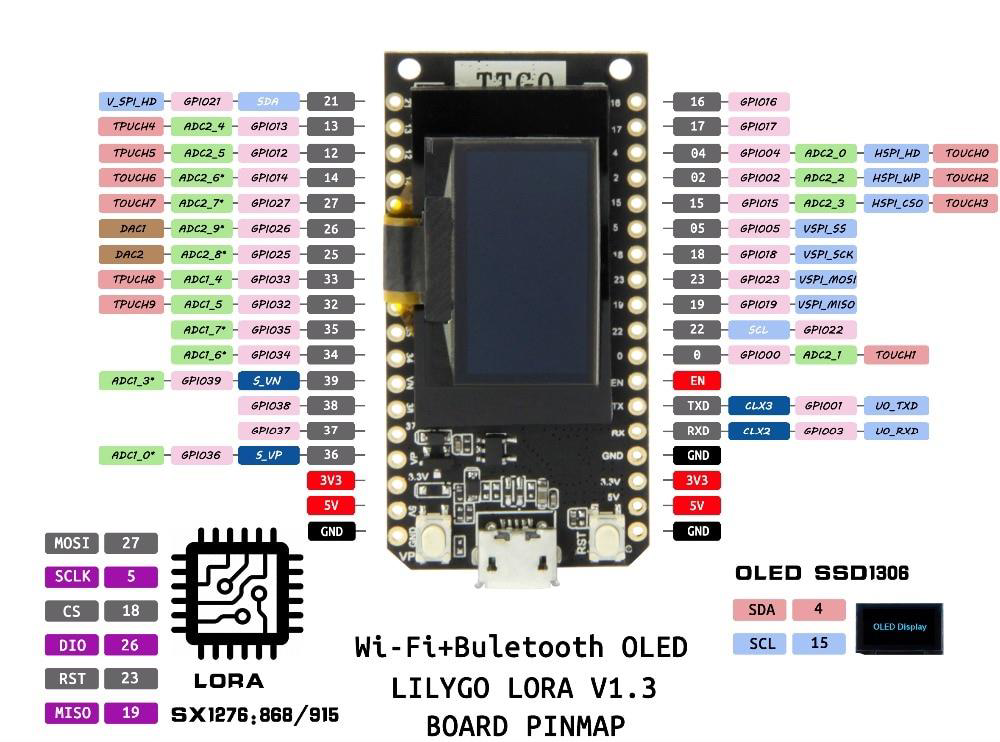


Figure 1: brochage de la carte TTGO LoRa32 SX1276 :

2.2 La programmation de la TTGO LoRa32 SX1276 OLED :

Pour programmer la TTGO LoRa32 SX1276 OLED nous avons utiliser l’environnement de

développement d’Arduino car cet IDE présente beaucoup d’avantages :

• Gratuit et open-source.

• Multiplateforme (Windows, mac et linux).

• Facilité d’installation et programmation claire et intuitive.

• Disponibilité de beaucoup d’exemples de programmes et de cours.

• Disponibilité d’une grande quantité de librairies qui permett ent de faciliter la

programmation.



◆ La description de l’Arduino IDE :

C’est est un environnement de développement intégré (noté : EDI) qui. Il fonctionne sur

divers systèmes d’exploitation (Windows, Mac Os, Gnu/Linux). Et qui

permet :

1. D’éditer un programme : des croquis (sketch en Anglais),

2. De compiler ce programme dans le langage « machine » de

l’Arduino,

3. De téléverser le programme dans la mémoire de l’Arduino,

4. De communiquer avec les cartes grâce au terminal. [6]

Grâce à Arduino IDE, nous sommes en mesure de communiquer et transférer des données

facilement au circuit imprimé. Après avoir installé le programme, l'interface du logiciel se

présente de manière simple et structurée. Contenant les traditionnelles barres d'outils et de

boutons, une fenêtre d'édition depuis laquelle nous allons rédiger le code de nos programmes.

Supportant plusieurs onglets, l'éditeur d'Arduino IDE nous permet de travailler sur plusieurs

programmations à la fois. Sous la fenêtre d'édition, c’est une zone de message qui nous indique

les actions en cours. Ensuite, une console de texte affiche les informations concernant les

résultats de compilation des programmes.

Enfin, après avoir rédigé nos programmes dans l'environnement dédié, nous devons vérifier et

compiler le code source, puis l'envoyer à notre carte dans le cas où aucune erreur de

programmation ne serait détectée.

2.3 Les capteurs utilisés :

Dans la réalisation de notre projet nous avons utilisé des capteurs qui sont capables de mesurer

les grandeurs relatives aux déclenchements des feux de forêts, les valeurs collectées par ces

capteurs se complètent :

• Le capteur de température va permettre de surveiller l’évolution de la température du

milieu, plus la température augmente plus le risque augmente.

• Le capteur d’humidité va faciliter la surveillance de l’évolution de l’humidité, plus cette

dernière diminue plus le risque augmente.



• Les capteurs des gaz vont permettre de surveiller le taux des gaz dans l’aire en ppm,

notamment le Co et le Co2 issus des feux, l’augmentation anormale du taux de ces gaz

indique la présence des feux.

2.3.1 Capteur d’humidité et de température DHT22 (AM2302) :

◆ Description du capteur :

Le DHT22 (AM2302) est un capteur assez connu dans la détermination d'humidité relative et

la température, il se compose d'un capteur d'humidité capacitif et d'une thermistance. En outre,

le capteur contient un CAN (ADC) pour convertir les valeurs analogiques d'humidité et de

température. Ce capteur numérique est basé sur un protocole qui utilise un fil / bus à collecteur

ouvert pour la commun ication, de sorte qu'une résistance de 5 à 10 kΩ doit être tirée vers

l'alimentation positive.

Le capteur de température et d'humidité DHT22 communique avec un microcontrôleur via un

port série. Le capteur est calibré et ne nécessite pas de composants supplémentaires pour

pouvoir être utilisé.

Remarque : le protocole de communication utilisé par DHT22 n'est pas le même que le bus 1-

Wire (1-Wire / One Wire) de Dallas Semiconducteur.

Figure 13 : le capteur DHT22



◆ Les caractéristiques du capteur :

● Tension d'alimentation de 3 à 6 V cc

● Courant de mesure : 1,5 mA

● Consommation au repos (courant de veille): 40 à 50 µA

● Plage de mesure de :

Température : -40 à +80 °C

Humidité : 0 à 100 % RH

● Précision de mesure de :

Température : ± 0,5 °C

Humidité : ± 2 % RH

● La période de mesure : deux secondes.

● Fréquence de mesur e : 0,5 Hz

● Résolution de 16 bits

2.3.2 Le capteur du gaz CO2 (MG-811) :

◆ Description du capteur :

Ce module est basé sur le capteur de gaz MG-811 permettant de détecter la présence de

dioxyde de carbone CO2. La tension de sortie du module diminue à mesure que la concentration

de CO2 augmente. Le module vient avec un potentiomètre permettant de régler le seuil ou la

sensibilité de tension. Il intègre un circuit chauffant fournissant au capteur la meilleure

température de fonctionnement. Ce capteur est doté d'un circuit de conditionnement intégré,

permettant d'amplifier le signal de sortie. Ce module est un peu spécial car il nécessite 6VDC

pour fonctionner correctement. Le MG811 est très sensible au CO2 et moins sensible à l'alcool

et au CO. Il peut être utilisé pour surveiller la qualité de l’air, des procédés de fermentation ou

pour des utilisations In-Door.

◆ Les caractéristiques et les avantages du capteur :

● Tension d’alimentation : 6 V cc (il peut fonctionner même avec 5 VDC)

● Capable de détecter des niveaux de concentration de CO2 dans l'air entre 350 et 10000

ppm.

● Le signal valide de sortie TTL est faible. (Avec sortie de signal analogique, signaux de

niveau TTL, sortie compensée en température).

● Sortie analogique + TTL

● Signal de sortie TTL est faible-efficace

27



● Sortie analogique 30 ~ 50 mV, (consistance plus élevée, tension plus élevée)

● Capteur durable et fiable avec une bonne sélectivité

● Sortie : analogique (et une sortie numérique).

● Les conditions de détection idéale pour le capteur de gaz CO2 MG-811 est de 25 ° C ±

2 ° C à 65% ± 5% d'humidité.

● Admet un circuit de conditionnement pour amplifier le signal de sortie.

● Admet un circuit chauffant intégré.

● Caractérisé par une Haute sensibilité et temps de réponse rapide

Figure 3 : le capteur MG-811

2.3.3 Le Capteur de gaz ZE07-CO ME2 :

◆ Description du capteur :

ZE07-CO est un module de détection de monoxyde de carbone électrochimique à usage général et de miniaturisation. Il utilise le principe électrochimique pour détecter le CO dans l'air, ce qui confère au module une sélectivité et une stabilité élevées. Le capteur de température intégré peut effectuer une compensation de température ; et il a une sortie numérique et une sortie de tension analogique. C'est une combinaison d'un principe de détection électrochimique mature et d'une conception de circuit sophistiquée.

hd

28



Figure 4 : le capteur ME2 CO

◆ Les caractéristiques et les avantages du capteur :

haha

● Tension d’alimentation : 5 Vcc

● Courant de fonctionnement : 150 mA

● Sortie analogique/numérique

● Température de service : -20 à 50 °C

● Capable de détecter des niveaux de concentration de CO2 dans l'air entre 20 et 2000

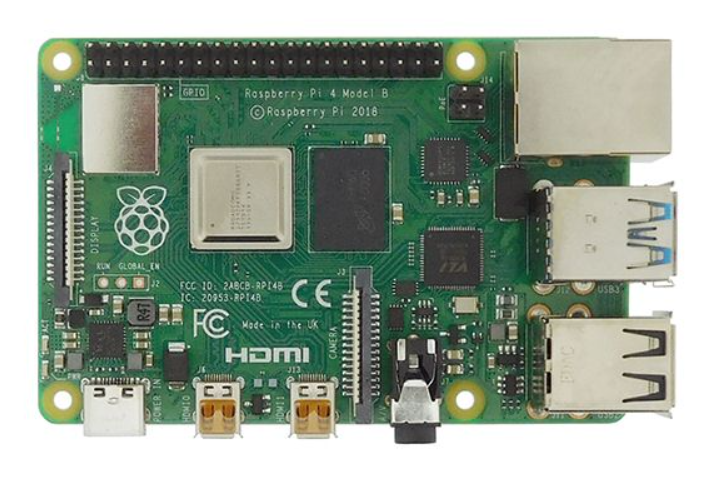
ppm.

● Haute sensibilité et temps de réponse rapide et la sensibilité

● La sensibilité est réglable par potentiomètre intégré.

● Temps de réponse rapide et haute sensibilité Longue durée de vie et bonne stabilité

3 Description de la carte Raspberry Pi 4 :

Le **Raspberry Pi 4** est un nano-ordinateur pouvant se connecter à un moniteur, à un ensemble clavier/souris et disposant d'interfaces Wi-Fi, Bluetooth et Ethernet.  
  
Il fonctionne depuis une carte micro-SD et fonctionne avec un système d'exploitation basé sur Linux ou Windows 10 IoT. Il est fourni sans boîtier, alimentation, clavier, écran et souris dans le but de diminuer le coût et de favoriser l'utilisation de matériel de récupération.  


Cette nouvelle version de la carte Raspberry Pi 4 est basée sur:

* 1 x processeur ARM Cortex-A72 64 bits quatre coeurs à **1,5 GHz**
* **4 GB de mémoire RAM** (existe également en version 1 et 2 GB de RAM)
* 1 x interface Wi-Fi
* 1 x interface Bluetooth
* 2 x ports USB 2.0
* 2 x ports USB 3.0
* 1 x port Ethernet Gigabit
* 2 x ports micro-HDMI pour deux écrans jusqu'à 4k (3840 x 2160 pixels) à 60 FPS
* 1 x port micro-SD
* 1 x connecteur GPIO avec 40 broches d'E/S

Les interfaces **Ethernet et Bluetooth ont été améliorées**par rapport à la version Pi 3 B+ et supportent maintenant l'**Ethernet Gigabit**ainsi que le **Bluetooth 5.0**.  
  
Cette carte est basée sur un processeur ARM et permet l'exécution du système d'exploitation GNU/Linux/Windows 10 IoT et des logiciels compatibles.  
Le Raspberry Pi peut effectuer des tâches d’un PC de bureau (feuilles de calcul, traitement de texte, jeux). Il peut également diffuser des vidéos en haute définition grâce à son circuit Broadcom VideoCore VI (permet le décodage des flux vidéo 4k H.265 et 1080p H.264). Cette nouvelle version supporte également le 4k en natif via ses sorties micro-HDMI.

La Raspberry Pi 4 B nécessite une [carte SD munie d'un OS](https://www.gotronic.fr/art-carte-microsd-16-gb-raspberry-pi-os-34444.htm), une [alimentation](https://www.gotronic.fr/art-alimentation-usb-type-c-alrpi4-b-30732.htm), un [clavier USB](https://www.gotronic.fr/cat-claviers-et-souris-1794.htm), une [souris USB](https://www.gotronic.fr/cat-claviers-et-souris-1794.htm), un [boîtier](https://www.gotronic.fr/cat-boitiers-pour-raspberry-pi-4b-1832.htm)et des câbles (non inclus). Pour préparer une carte SD bootable, il faut disposer d'un PC avec lecteur de carte SD.

**◆ Principales caractéristiques de la carte Raspberry Pi 4 :**

* Alimentation à prévoir: 5 Vcc/maxi 3 A\* via prise USB Type C (\* intensité maxi si toutes les fonctions sont utilisées)
* CPU: ARM Cortex-A72 quatre coeurs 1,5 GHz
* Wi-Fi: Dual-band 2,4 et 5 GHz, 802.11b/g/n/ac (Broadcom BCM43438)
* Bluetooth 5 compatible BLE (Broadcom BCM43438)
* Mémoire: 4 GB LPDDR4
* Circuit vidéo: VideoCore VI à 500 MHz
* 2 ports USB 2.0
* 2 ports USB 3.0
* Port Ethernet Gigabit: RJ45
* Bus: SPI, I2C, série
* Support pour cartes micro-SD
* Sorties audios :  
  - 2 x micro-HDMI avec gestion du 5.1  
  - Jack 3,5 mm en stéréo (partagé avec vidéo)
* Sorties vidéos :  
  - 2 x micro-HDMI (4K @ 60 fps maxi)  
  - Jack 3,5 mm (partagé avec audio)
* Interface CSI pour caméra
* Interface DSI pour écran
* Dimensions : 88 x 58 x 17 mm
* Poids : 46 g

Chapitre 3 : Réalisation du SDF-RCSF.

* + Avant de commencer notre chapitre ce schéma et une explication facile du SDF-RCSF et sa description :

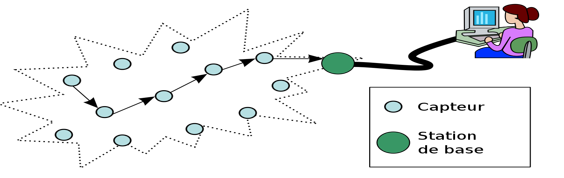


Figure 6 : Schéma de RCSF.

1. Installation de la carte ESP32 dans Arduino IDE (Windows, Mac OS X, Linux)

Il existe un module complémentaire pour l’IDE Arduino qui vous permet de programmer l’ESP32 en utilisant l’IDE Arduino et son langage de programmation. Avant de commencer cette procédure d’installation, assurez-vous que la dernière version de l’IDE Arduino est installée sur votre ordinateur. Si vous ne le faites pas, désinstallez-le et réinstallez-le. Sinon, cela risque de ne pas fonctionner.

1.1 Installation du module complémentaire ESP32 dans Arduino IDE

1) Pour installer la carte ESP32 dans votre IDE Arduino, suivez ces instructions suivantes :

Dans votre IDE Arduino, accédez à Fichier> Préférences

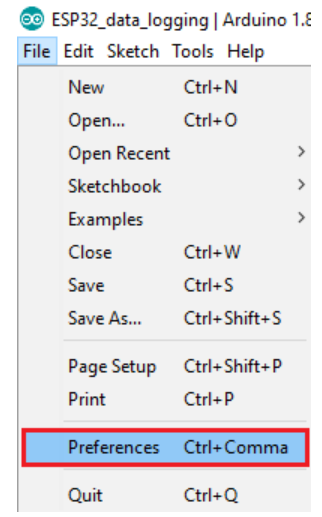


Figure 7 : La 1ère étape de connexion d’esp32 avec l’IDE

2)Entrez https://dl.espressif.com/dl/package\_esp32\_index.json dans le champ « URL supplémentaires du gestionnaire de carte », comme illustré dans la figure ci-dessous. Ensuite, cliquez sur le bouton « OK »:

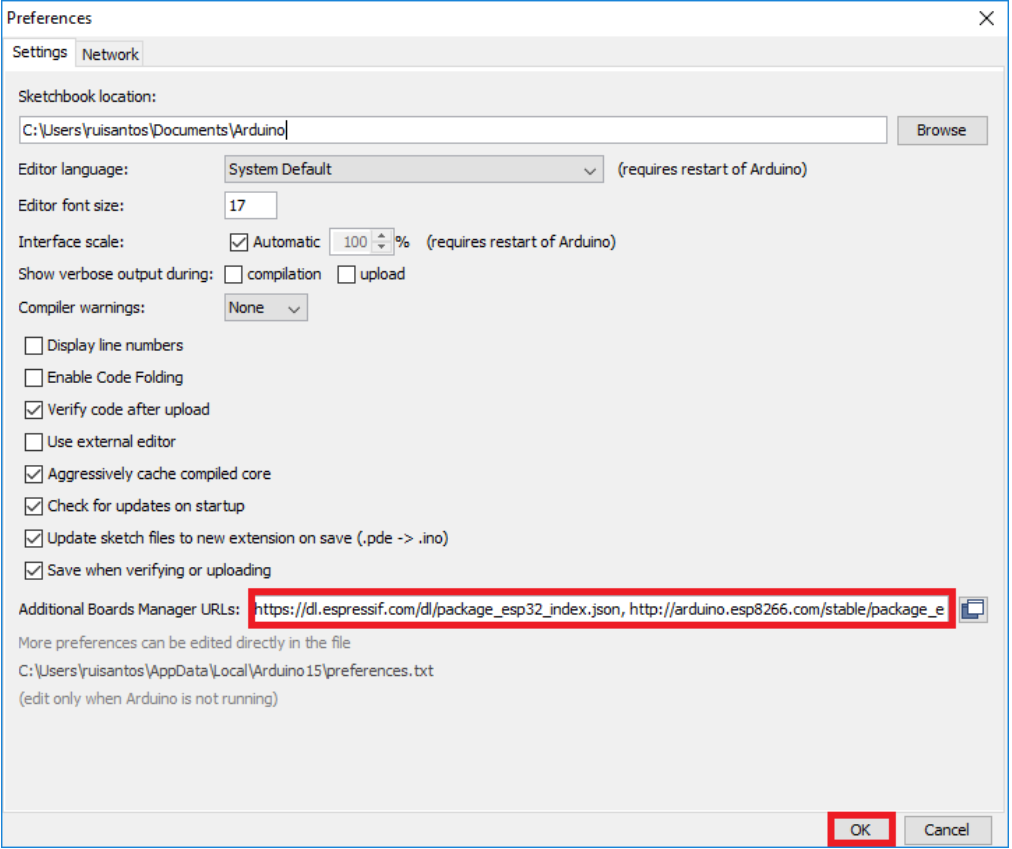


Figure 8 : 2ème étape de connexion d’esp32 avec l’IDE

Remarque : si vous avez déjà l’URL des cartes ESP8266, vous pouvez séparer les URL par une virgule comme suit :

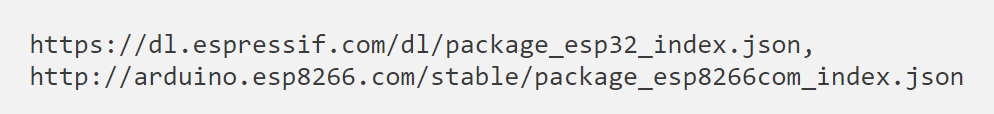


Figure 9 : Remarque de la 2ème étape de connexion d’esp32 avec l’IDE

3) Ouvrez le Gestionnaire de tableaux. Accédez à Outils > conseil d’administration > gestionnaire de conseils d’administration...

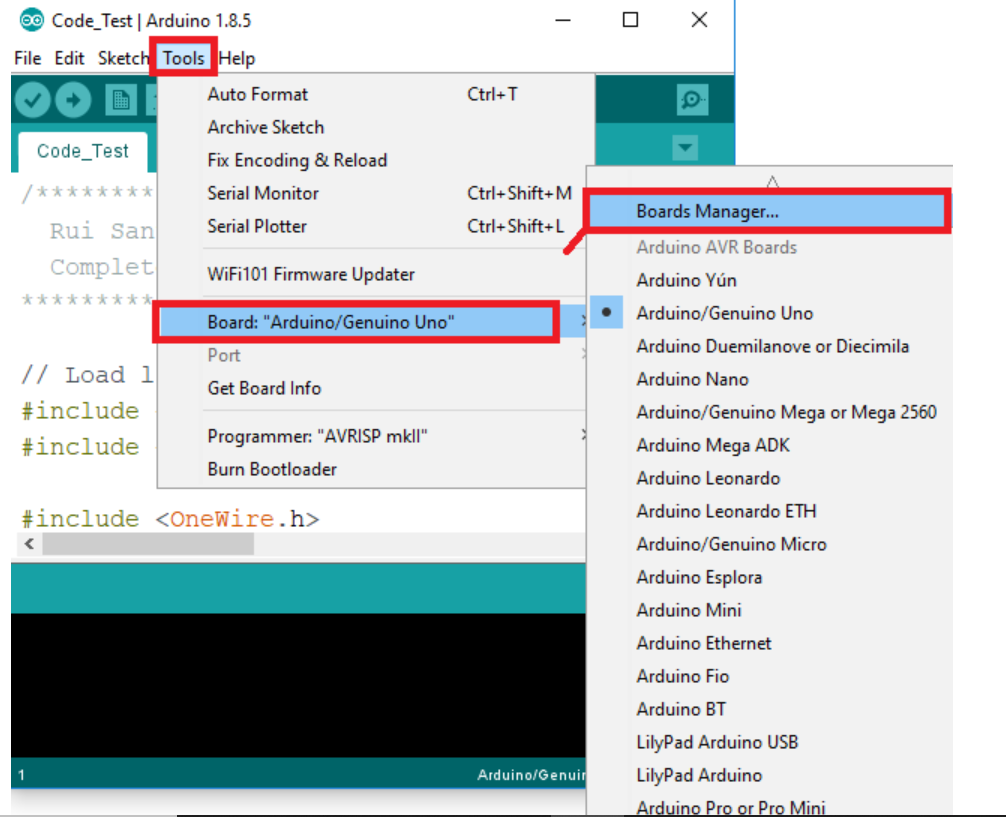


Figure 10 : La 3ème étape de connexion d’esp32 avec l’IDE

4) Recherchez ESP32 et appuyez sur le bouton d’installation pour le « ESP32 by Espressif Systems »:

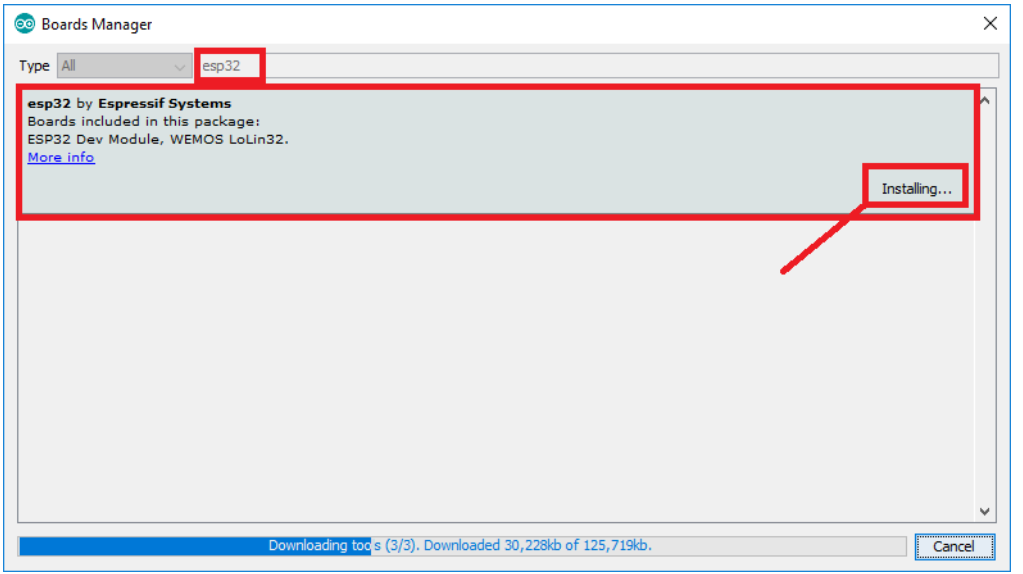


Figure 11 : La 4ème étape de connexion d’esp32 avec l’IDE

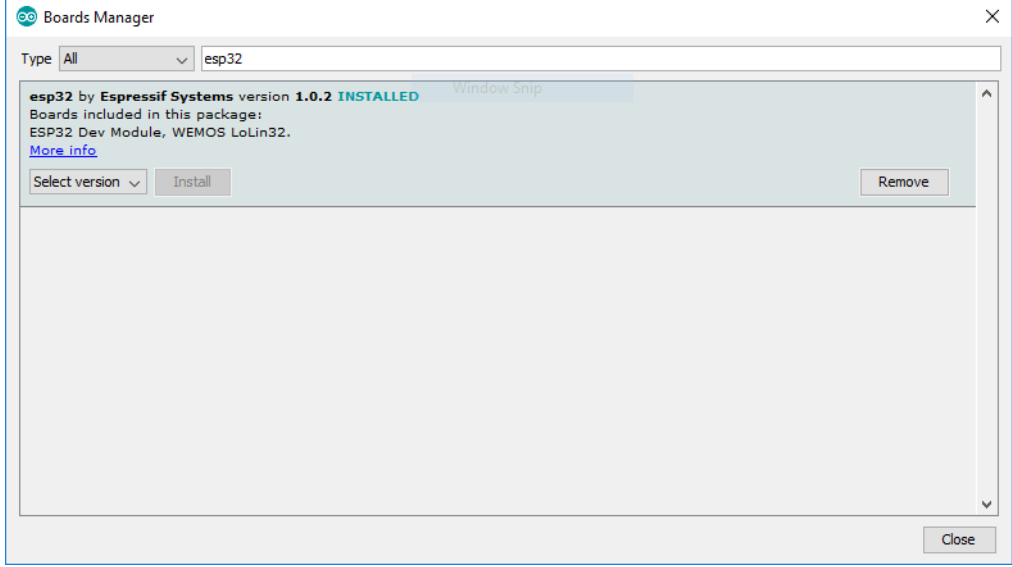
5) Voilà. Il devrait être installé après quelques secondes.

Figure 12 : La dernière étape de connexion d’esp32 avec l’IDE

1.2 Test de l’installation

Branchez la carte ESP32 à votre ordinateur. Une fois votre IDE Arduino ouvert, procédez comme suit :

1. Sélectionnez votre carte dans le menu Outils > (dans mon cas, il s’agit du DOIT ESP32 DEVKIT V1)

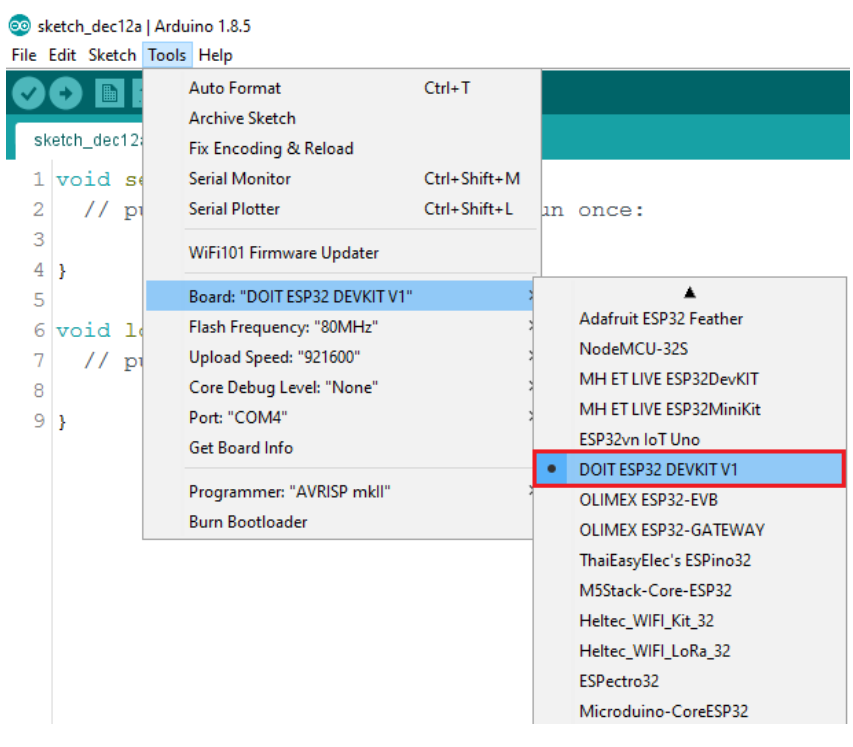


Figure 13 : La 1ère étape de test de connexion d’esp32 avec l’IDE

1. Sélectionnez le port (si vous ne voyez pas le port COM dans votre IDE Arduino, vous devez installer les pilotes CP210x USB vers UART Bridge VCP):

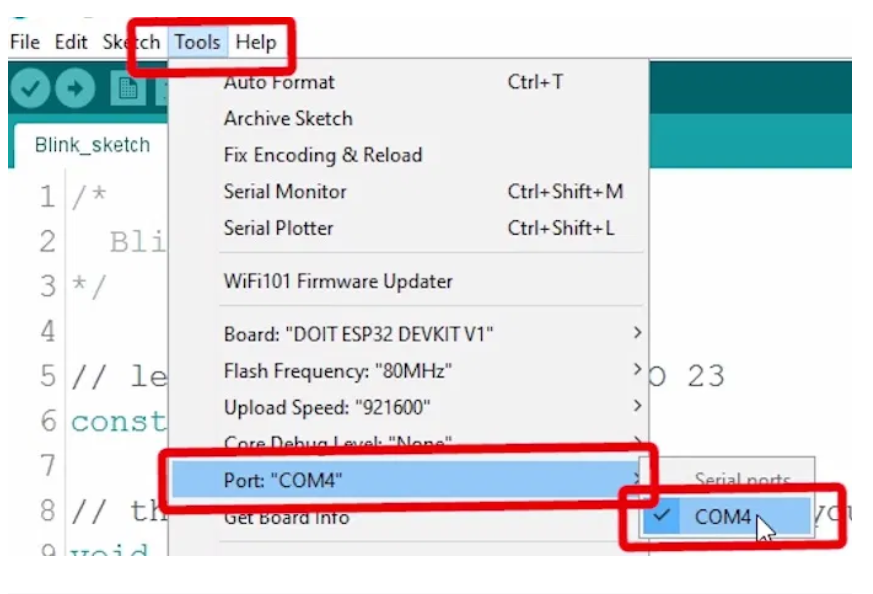


Figure 14 : La 2ème étape de test de connexion d’esp32 avec l’IDE

1. Ouvrez l’exemple s uivant sous Fichier > Exemples > WiFi (ESP32) > WiFiScan

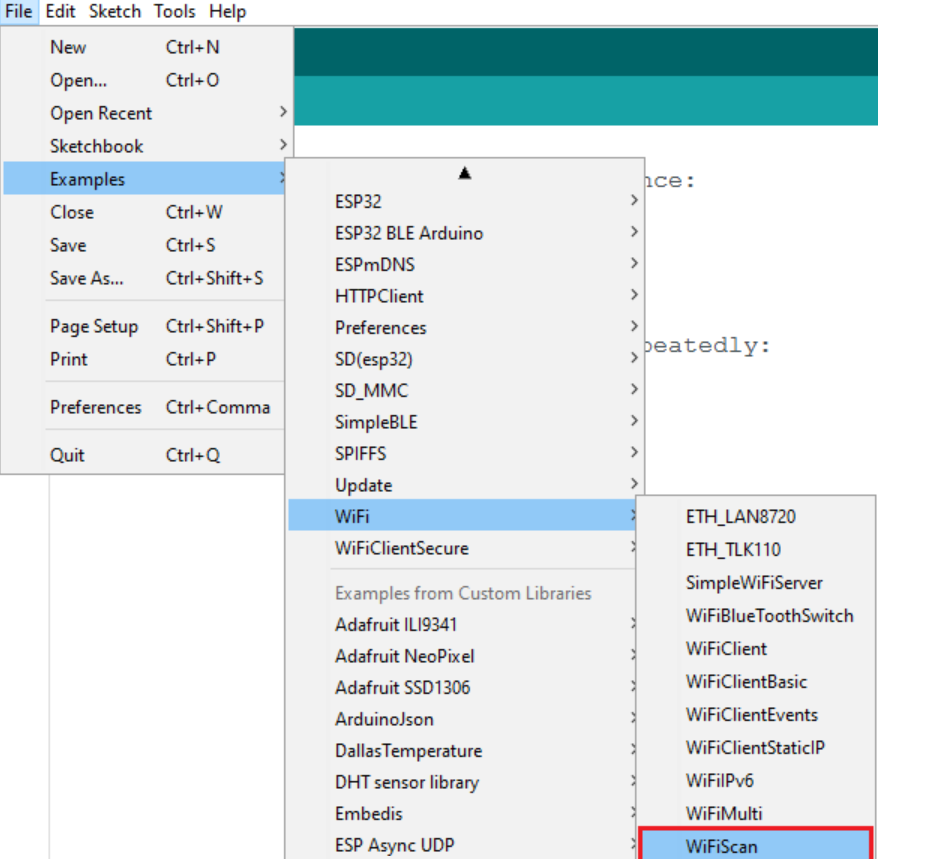


Figure 15 : La 3ème étape de test de connexion d’esp32 avec l’IDE

1. Un nouveau croquis s’ouvre dans votre IDE Arduino :

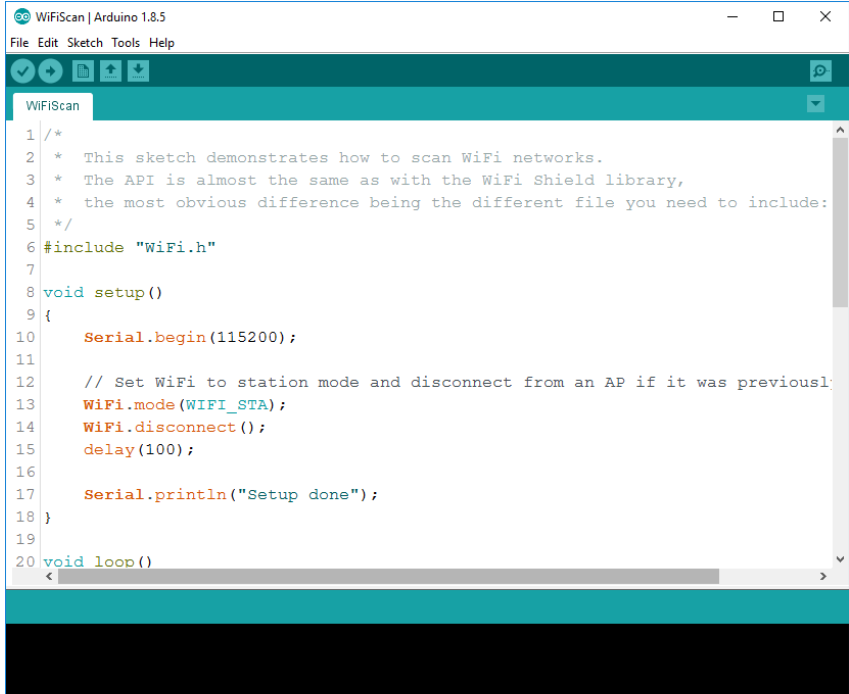


Figure 16 : La 4ème étape de test de connexion d’esp32 avec l’IDE

1. Appuyez sur le bouton Upload dans l’IDE Arduino. Attendez quelques secondes pendant que le code se compile et se télécharge sur votre tableau.
2. Si tout s’est passé comme prévu, vous devriez voir un message « Terminé le téléchargement ».
3. Ouvrez le moniteur série Arduino IDE à un débit en bauds de 115200:
4. Appuyez sur le bouton ESP32 à bord Activer et vous devriez voir les réseaux disponibles près de votre ESP32:

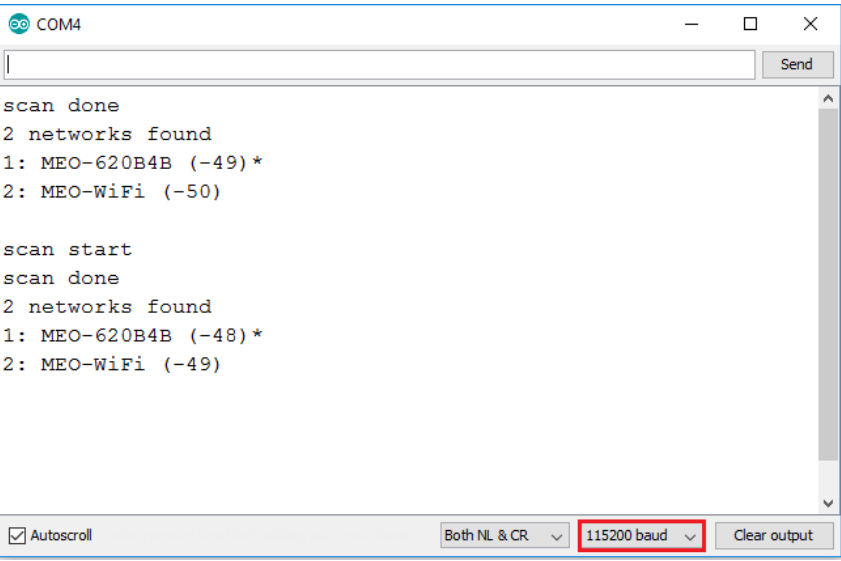


Figure 17 : la dernière étape de test de connexion d’esp32 avec l’IDE

1.3 Conclusion :

Ceci est un guide rapide qui illustre comment préparer votre IDE Arduino pour l’ESP32 sur un ordinateur Windows, Mac OS X ou Linux. Si vous rencontrez des problèmes au cours de la procédure d’installation, consultez le "guide de dépannage ESP32."Maintenant, vous pouvez commencer à construire vos propres projets IoT avec l’ESP32 !

1. ESP32 avec capteur de température et d’humidité DHT11/DHT22 utilisant Arduino IDE

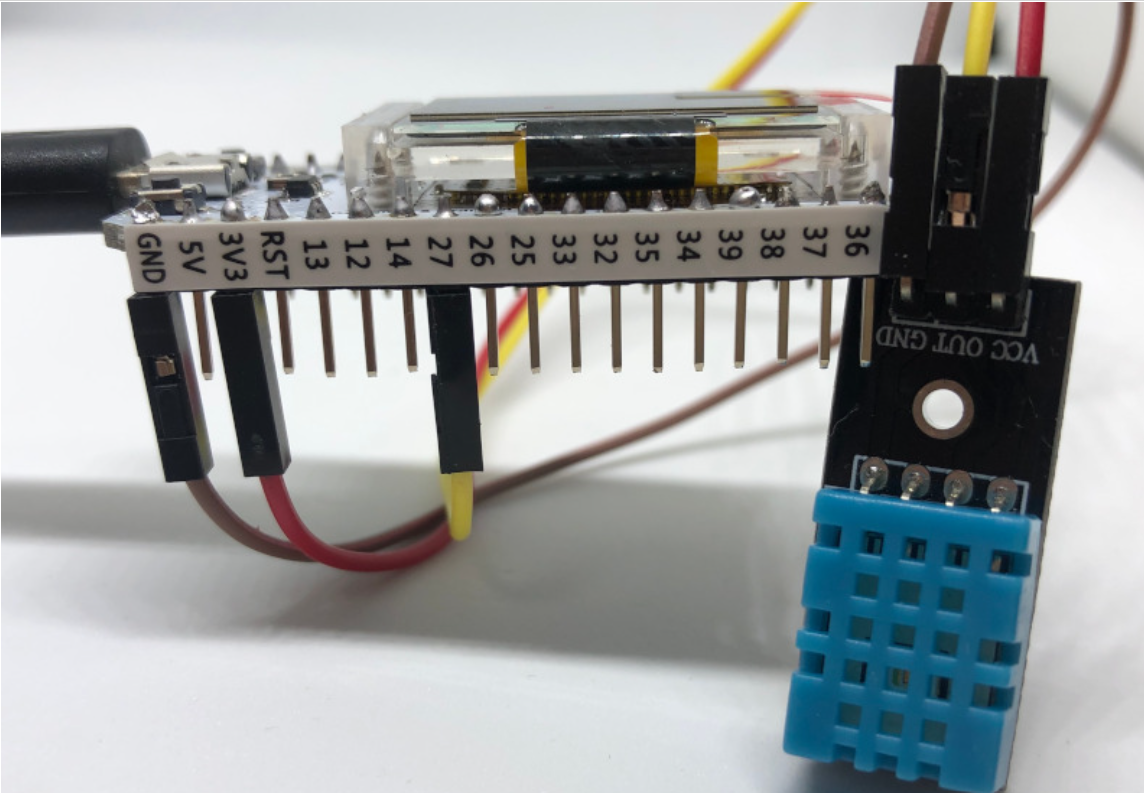
-Au début on a travaillé avec le capteur dht 22 et on a essayé d’envoyer les informations par Bluetooth

2.1 Brochage DHT

Lorsque vous regardez votre capteur DHT11 / DHT 22, vous pouvez voir trois broches étiquetées +/VCC, out/s/data et -/GND. Connectez-le :

* +/VCC broche à une broche 3v3 sur votre carte broche de sortie/s/données vers GPIO27 sur votre carte.
* /GND broche à une broche GND sur votre carte.

Par exemple, si vous avez un capteur DHT11, voici à quoi peut ressembler votre connexion :

Figure 18 : Câblage de DHT22

* 11 Connectez votre capteur DHT11 / DHT 22 à votre carte de développement WiFi Kit 32.

2.2 Installation de bibliothèques

- Pour lire à partir du capteur DHT, nous utiliserons la bibliothèque DHT d’Adafruit. Pour utiliser cette bibliothèque, vous devez également installer la bibliothèque Adafruit Unified Sensor. Suivez les étapes suivantes pour installer ces bibliothèques.

- Ouvrez votre IDE Arduino et accédez à Esquisse > Inclure la bibliothèque > Gérer les bibliothèques. Le gestionnaire de bibliothèque doit s’ouvrir.

- Recherchez « DHT » dans la zone de recherche et installez la bibliothèque DHT d’Adafruit

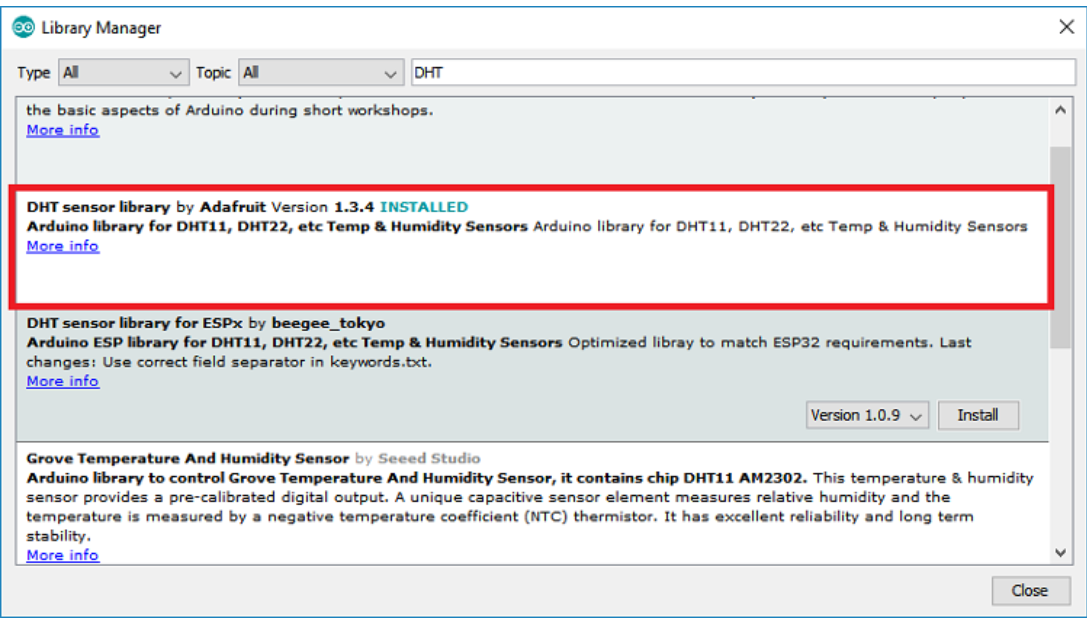
- 

Figure 19 : 1ère étape d’installation de la bibliothèque DHT

- Après avoir installé la bibliothèque DHT à partir d’Adafruit, tapez « Adafruit Unified Sensor » dans la zone de recherche. Faites défiler tout le chemin vers le bas pour trouver la bibliothèque et l’installer.

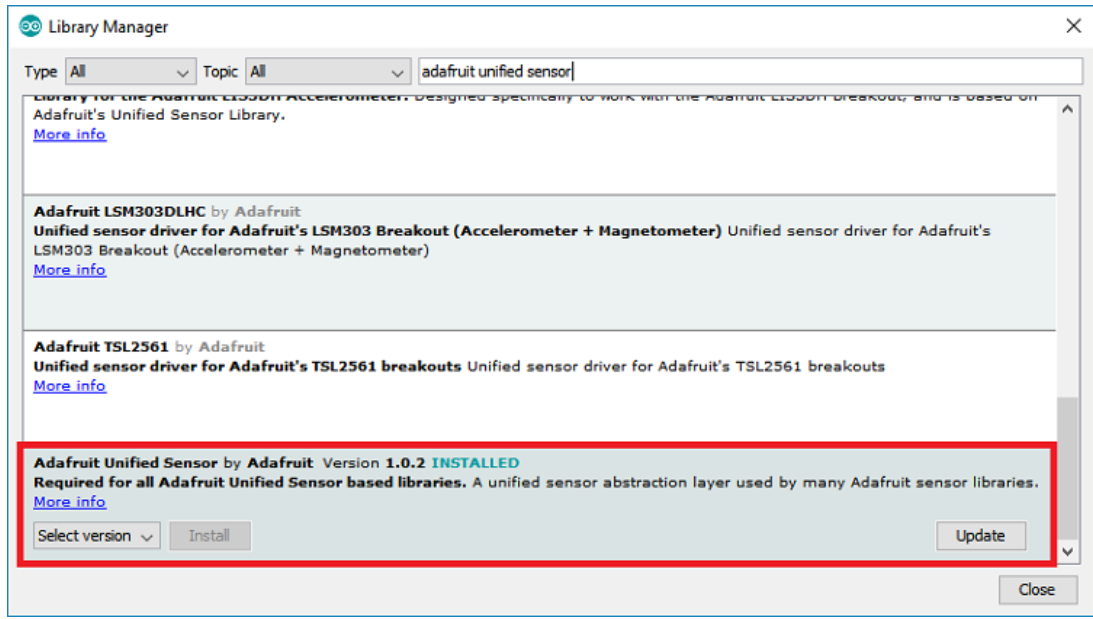


Figure 20 : La 2ème étape d’installation de la bibliothèque DHT

* Installez la bibliothèque de capteurs DHT pour ESPx pour lire la température et l’humidité de votre capteur DHT11 / DHT22
* Pour ce faire, allez d’abord dans Outils -> Gérer les bibliothèques.... Après cela, la fenêtre Gestionnaire de bibliothèque apparaîtra. Recherchez DHT ESPx et installez la bibliothèque de capteurs DHT pour ESPx par beegee\_tokyo :

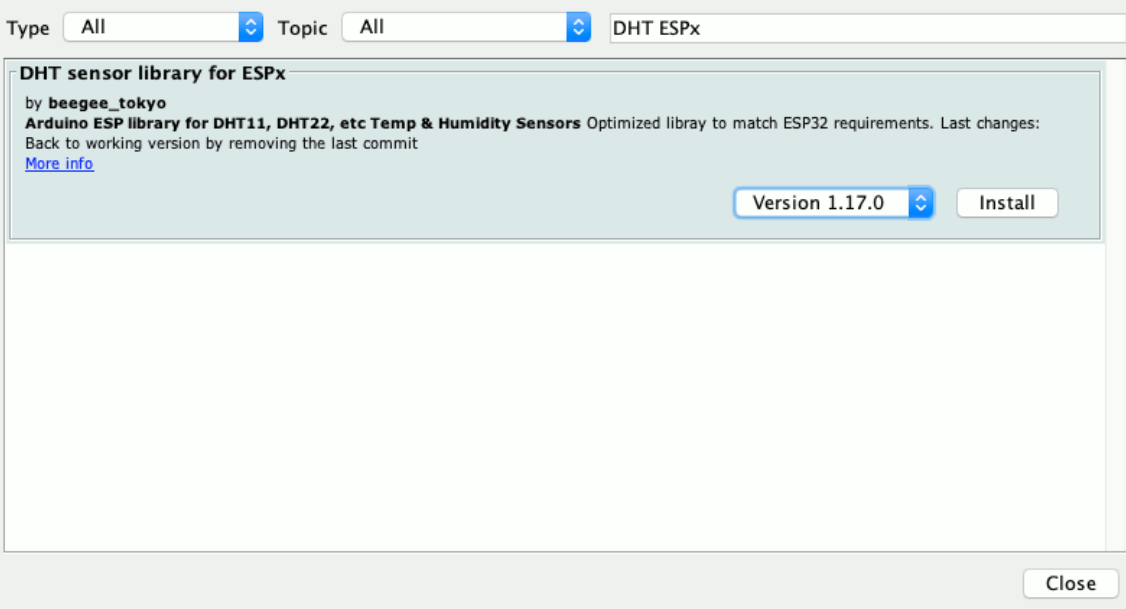


Figure 21 : La 3ème étape d’installation de la bibliothèque DHT

* Installez la carte Heltec ESP32 et la bibliothèque Heltec ESP32
* Finalement, écrivez l’Arduino Sketch pour lire la température et l’humidité du capteur DHT et affichez-le sur Heltec OLED.



Figure 22 : Le code d’installation de la bibliothèque DHT



Figure 23 : le code de fonctionnement de DHT

2.3 Connecter esp32 avec raspberry par Bluetooth.

- On doit d’abord avoir l’adresse mac de notre esp32 en suivant les étapes suivantes:

1) Pour que les exemples ESP32 apparaissent dans la liste des exemples, vous devez d’abord choisir la carte d’ESP32. C’est ESP32 Wrover Module dans le cas.

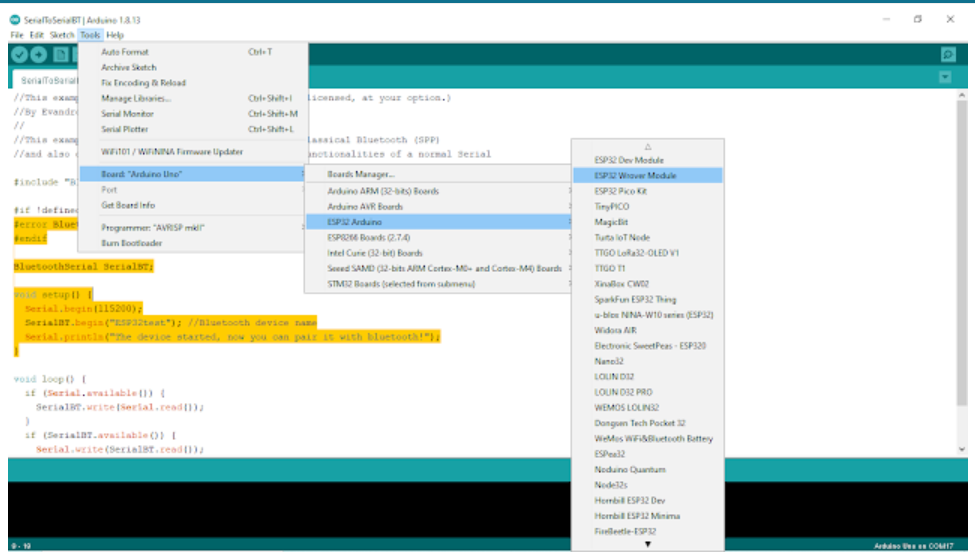


Figure 24 : la 1ère étape de connexion d’esp32 avec la Raspberry par Bluetooth

2) L’exemple disponible dans Arduino IDE MENU > File > Examples > Bluetooth Serial (sous Exemples pour wrover Module) > SerialToSerialBT.

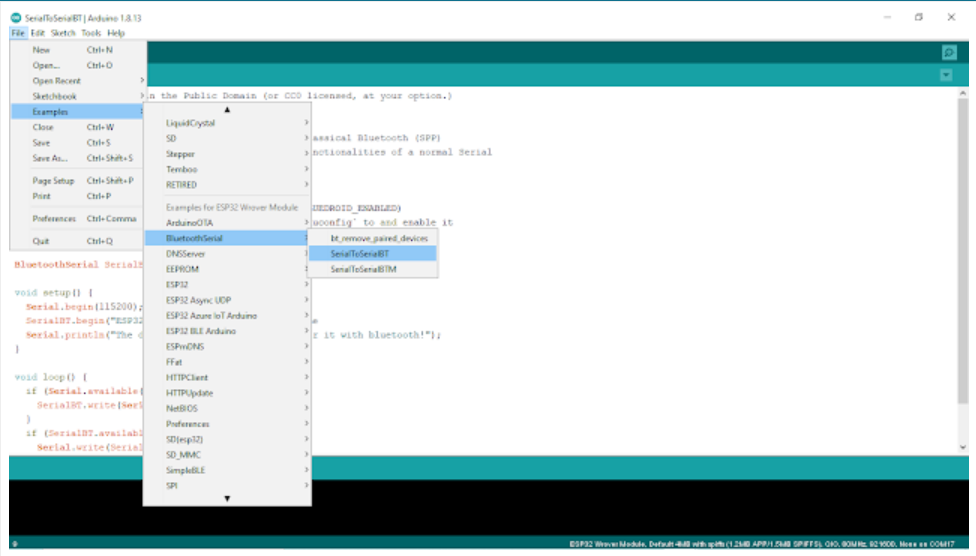


Figure 25 : La 2ème étape de connexion d’esp32 avec la Raspberry par Bluetooth

* Je fais une petite modification pour afficher son adresse MAC Bluetooth.
* Cela est le code qui nous affichera l’adresse mac de esp32)
* Après ça on connecte notre Raspberry avec la carte esp32 par Bluetooth. On suit les étapes suivantes.
* Les étapes mentionnées par la suite seront pour la Raspeberry.





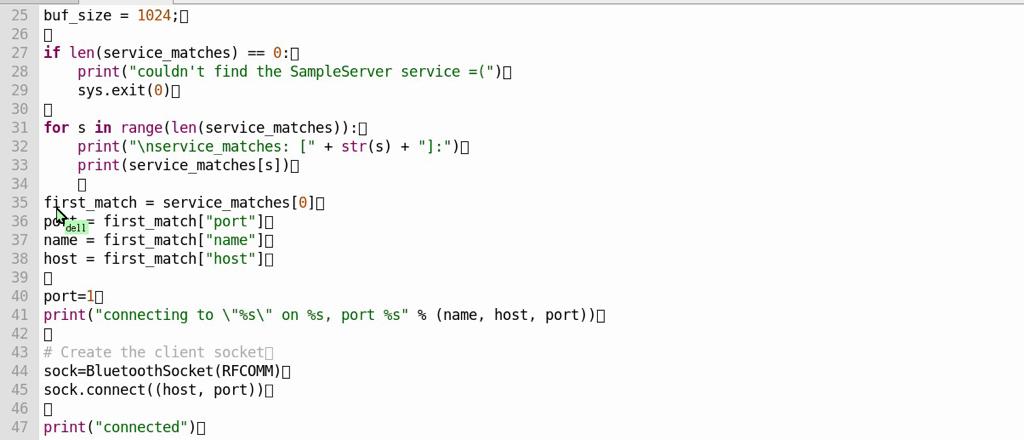
Figures 26 et 27 : Code source de connexion d’esp32 avec la Raspberry par Bluetooth

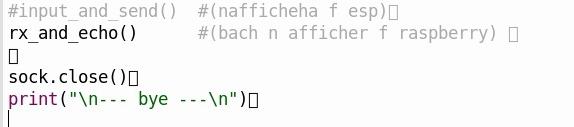
* + 1. Installez pybluez pour Python3, entrez la commande dans Terminal:

1. Sudo pip3 installer pybluez

* Le code qu’on écrit dans l’éditeur python de la Raspberry







Figures 28, 29 et 30 : Code source d’installation de bibliothèque Pybluez

1. L’affichage et l’envoie des informations des capteurs par bleutooth à la carte Raspberry :

* Pour envoyer et afficher les informations des capteurs par Bluetooth a la carte raspberry, on a télécharger la bibliothèque de co2 et dht et de Bluetooth.

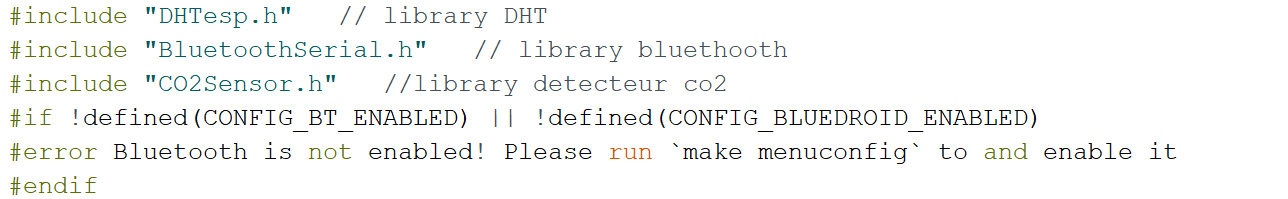


Figure 31 : Installation des bibliothèques CO2, DHT et Bluetooth

* Et on a écrit le code suivant :









Figures 32, 33, 34 et 35 : Code source de l’affichage et l’envoie des informations des capteurs par Bluetooth à la carte Raspberry

* Le but de ce programme est d’envoyer les valeurs et les données par Bluetooth à la Raspberry. Mais la méthode d’envoie utilisé était juste un test pour voir l’efficacité du code écrit des capteurs et pour les tester donc la méthode prévu à utiliser et la technologie de Lora/GPS\_HAT.

-Et voici un schéma qui résume cette étape.



Figure 36 : Schéma résumé des étapes de l’affichage et l’envoie des informations des capteurs par Bluetooth à la carte Raspberry

1. Connexion entre l’esp32 et la Raspberry Pi4 :

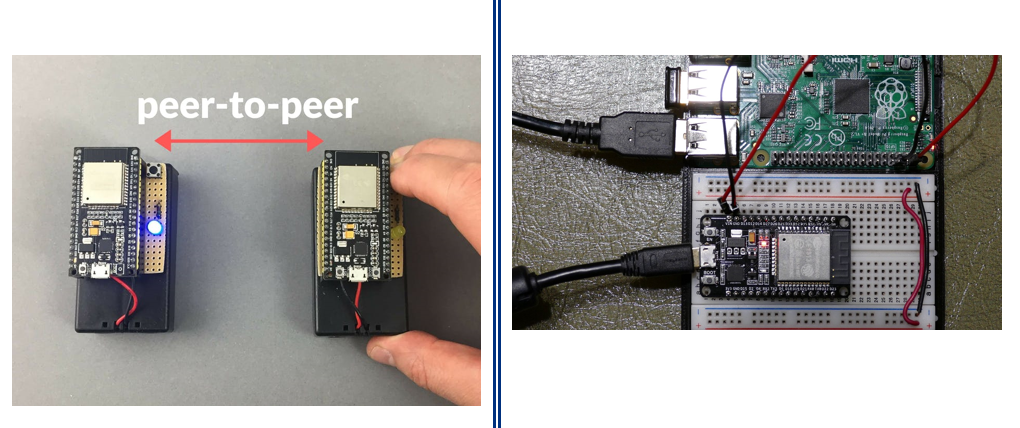
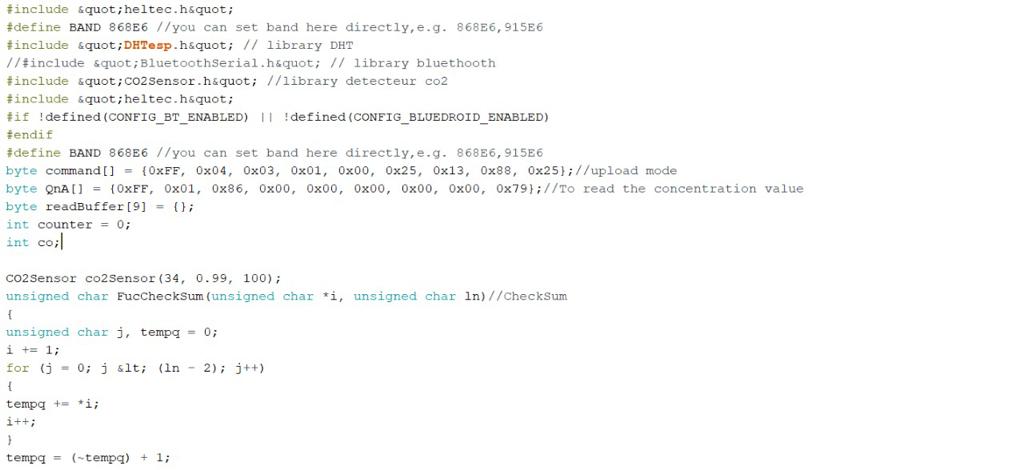


Figure 37 : Connexion de l’esp32 avec la Raspberry

* On a utilisé deux esp32 on a considéré une comme **Sender** et l’autre comme **receiver** les informations s'envoient entre les 2 esp32 par Lora dans la **Sender** on a pris l’exemple disponible dans arduino ide, >File> Exemples> Lora> Lora Sender et on l’a mixé le code d’exemple avec le code des capteurs DHT, CO et CO2 ; et on l’a televersé et voilà le programme de la carte Sender :







Figures 38,39 et 40 : Code source de la carte esp32 Sender

* Dans l'autre carte qui est considéré comme une carte qui reçoit les informations de la **Sender** on a pris exemple disponible dans arduino ide, >File> Exemples> Lora> Lora Receiver ; et on a fait quelque modification on l’a televersé et voilà son programme :







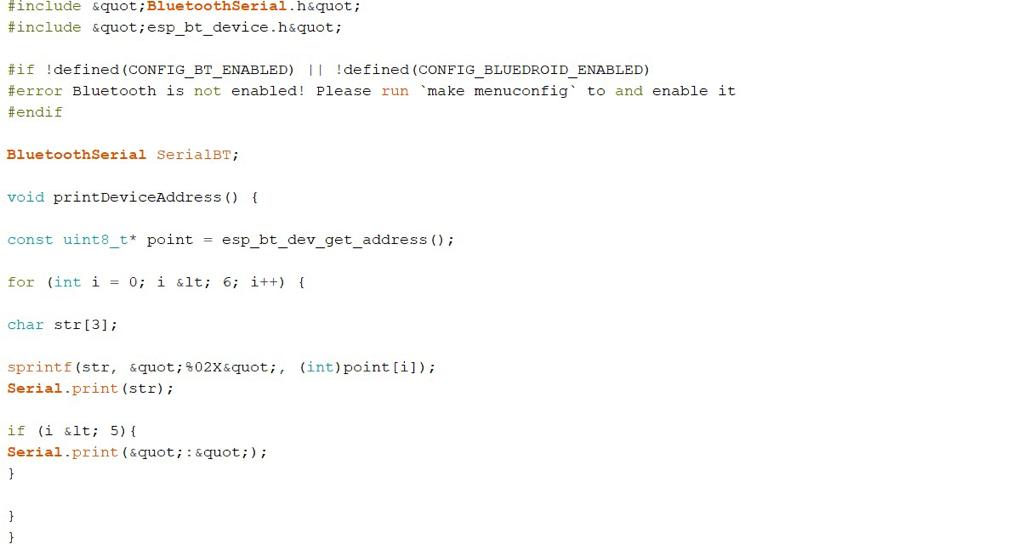
Figures 41, 42 et 43 : Code source de la carte esp32 Sender

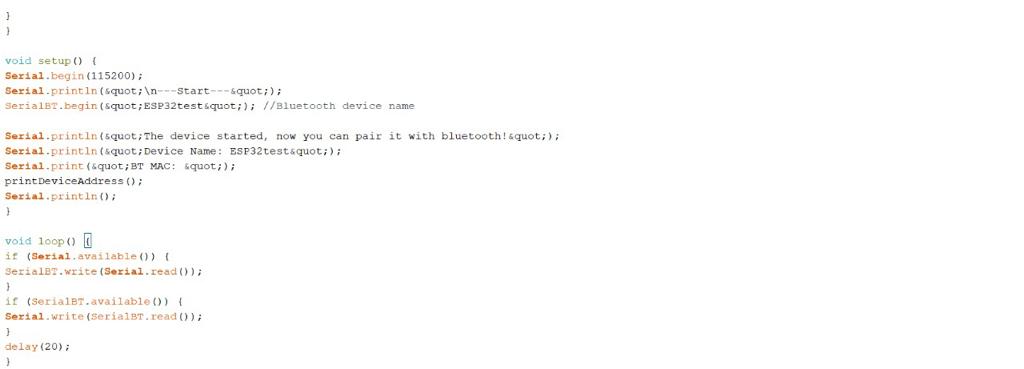
* Après ces 2 étapes on a remarqué que l'information de la première carte qui est considérée comme **Sender** s'envoie à la carte receiver et s'affiche dans le moniteur série de cette carte receiver après la communication entre ces 2 carte a été effectué avec succès on doit ajouter le code source de chaque capteur ; Dans l’exemple de **Sender :**

1. On commencer par le capteur DHT, on a appelé sa Library on a déclaré la température et l'humidité et on a écrit le code source de ce capteur et on a versé on a remarqué que les informations de ce capteur s'envoient a l’autre carte receiver et elle les reçoit dans son moniteur série (de la 2eme carte receiver).
2. On a passé à l’autre capteur CO2 (on a suit les mêmes étapes de capteur DHT).
3. Capteur CO aussi les mêmes étapes, à l'arrivée de ces étapes on a réussi à envoyer les informations de capteur à partir de carte Sender et les recevoir dans la carte receiver via **Lora**.

* Maintenant notre objectif et d’envoyer les informations reçu à la carte receiver à une carte Raspberry a l’aide de Bluetooth et on va suivre les étapes suivantes:

1. On doit connecter la carte receiver avec Bluetooth et mixer le code source de la reciver avec le code source de Bluetooth pour connecter esp32 avec raspberry par Bluetooth; L’exemple disponible dans Arduino IDE MENU > File > Examples > Bluetooth Serial (sous Exemples pour wrover Module) > SerialToSerialBT.
2. On doit avoir l’adresse mac de notre esp32 en écrivant ce code :





Figures 44 et 45 : Code source de l’adresse Mac d’esp 32

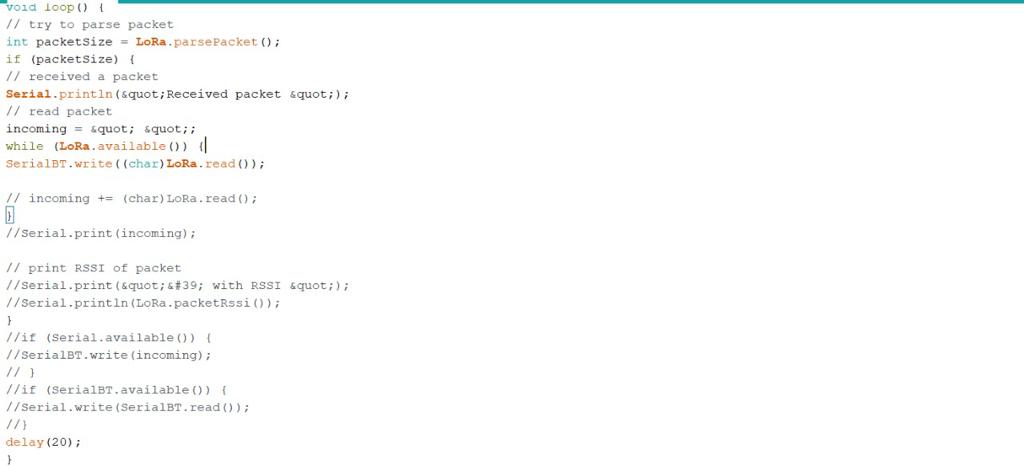
1. Après ça on connecte notre Raspberry avec la carte esp32 par Bluetooth.
2. on écrit ce code dans l’éditeur python de raspberry:

* Maintenant après faire ça on voit que lorsqu’ on écrit quelque chose dans le serial de l esp32 il s’affiche dans serial de raspberry donc on est dans la bonne voie.

1. Maintenant on va mixer le code de carte esp 32 receiver et le code qui est dans l’exemple de Bluetooth:







Figures 46, 47 et 48 : Code source de mix de receiver et Bluetooth

1. Après faire tout ça on vérifie que la carte esp32 est connecté avec la carte raspberry par Bluetooth et on exécute les programmes on remarque que les informations des capteurs s'affiche dans la carte raspberry donc on a réussi a effectué cette tâche.
2. Utilisation de la plate-forme Thingspeak :
   1. Description de thingspeak ***:***

* ThingSpeak est l’un des maillons de l’internet des objets. C’est une Plate-forme de collecte et de stockage de données avec traitements mathématiques. Ses trois fonctions principales sont :

1. Récupérer et stocker les données des objets ;
2. Analyser et visualiser les données avec MATLAB.
3. Déclencher des actions lorsque certaines conditions sont remplies.

* L'élément principal de ThingSpeak est le "Channel". Un Channel enregistre les données envoyées vers ThingSpeak et comprend les éléments suivants :

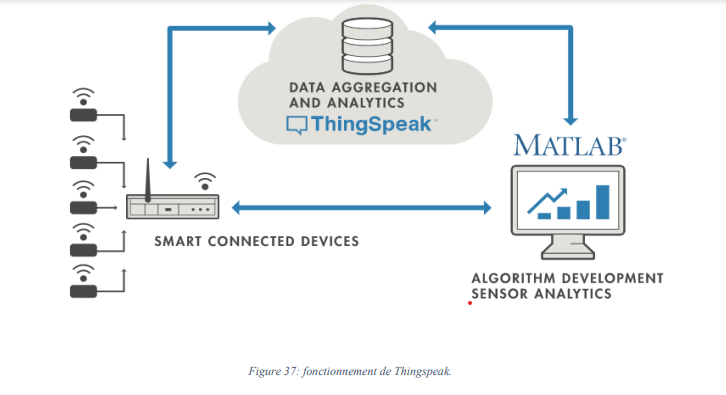
1. 8 champs pour stocker n'importe quel type de données ;
2. 3 champs de localisation, latitude, longitude et élévation ;
3. D’autres champs Metadata. 

Figure 49 : Fonctionnement de thingspeak

1. L’usage de Thingspeak

Les avantages apportés par le MQTT font de lui un excellent protocole de communication dans le monde IOT (l’Internet des Objets) Dans notre projet ;

La carte Raspberry Pi va utiliser le Protocol de communication MQTT afin de transmettre les données collectées vers la plate-forme ThingSpeak. Un système de clés sera présent pour assurer la sécurité des données, qui vont être ensuite stockées dans une base relationnelle (fichier csv). Les données par la suite seront visualisées dans des graphes sur des champs, l’ensemble Des champs est représenté sur les "Channels". Thingspeak permet de créer directement les graphes sans le besoin d’un autre logiciel ou Framework, Dans notre cas, nous cherchions une librairie utilisable dans la programmation en langage python sous Raspbian. Ainsi qu’un broker fonctionnant sur la carte Raspberry pi.

1. Le broker : Le broker de référence sous Linux s’appelle Mosquito, développé sous licence Eclipse Public License 1.0 par la fondation Apache.
2. Le client : Pour se linker avec le programme en python sous Raspbian, nous avons opté pour la bibliothèque Paho.mqtt.p, pour la clarté de sa documentation La réalisation de l’application.
3. La création d’un compte sur Thingspeak :

ӿEtape 1 : La création du compte :

-Apres avoir choisi ThingSpeak comme plateforme de visualisation des données collectées, nous avons créé un compte.

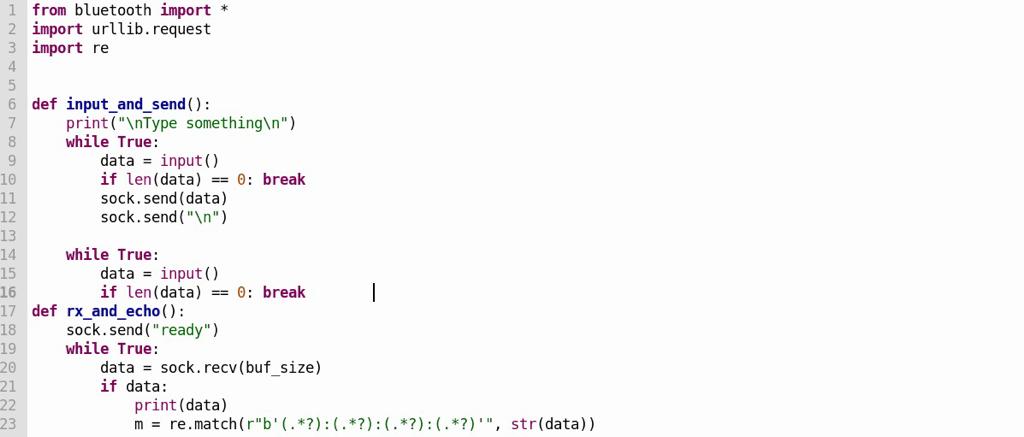
ӿEtape 2 : La création des Channels :

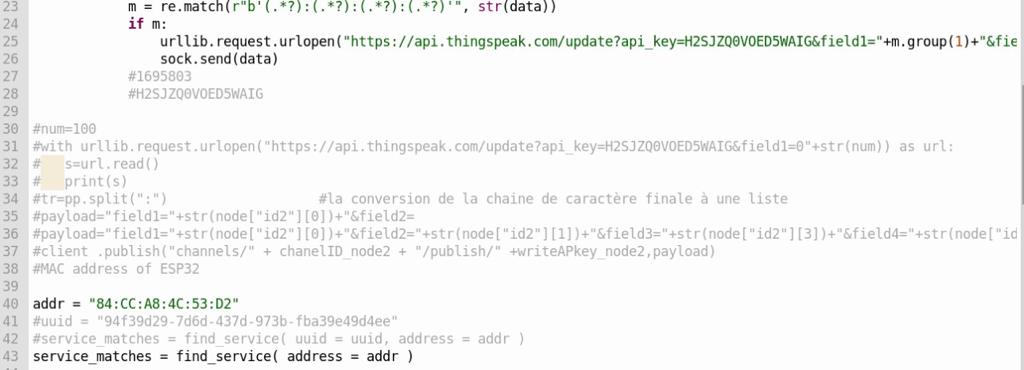
-Après la création un compte sur la plate-forme, nous avons commencé par la création les Channels ou chaines, chaque chaine correspond à une zone ou nœud

ӿEtape 3 : la création des Fields :

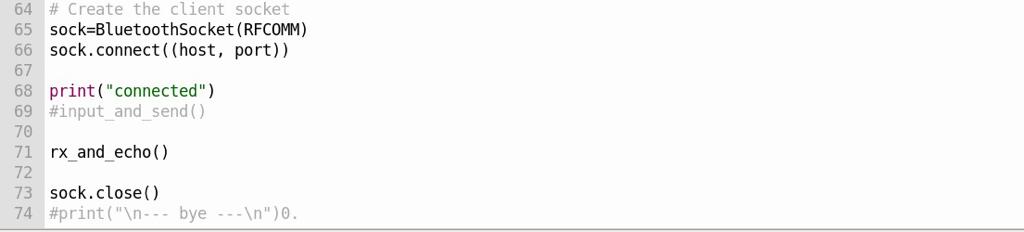
-Dans cette étape nous avons créé 4 champs ou "Fields" dans chaque Channel, chaque champ va permettre de stocker et visualiser les mesures météorologiques (température, humidité, CO, CO2) collectées par chaque nœud sous forme de graphes faciles à manipuler et exploiter.

-D’ailleurs l’allure des graphes va faciliter l’analyse de ces données, ce qui va garantir un suivie en temps réel des données relatives à la détection des Fux de foret.









Figures 50, 51, 52 et 53 : Code source de la connexion de la Raspberry et Thingspeak

**Conclusion générale :**

Dans ce travail, nous avons abordé un sujet d’actualité et d’intérêt sociétal qui traite la problématique des feux de forêt. Les incendies génèrent des dégâts néfastes et contribuent aux dégagements des gaz à effet de serre classés comme l’un des acteurs indispensables au réchauffement climatique de l’univers.

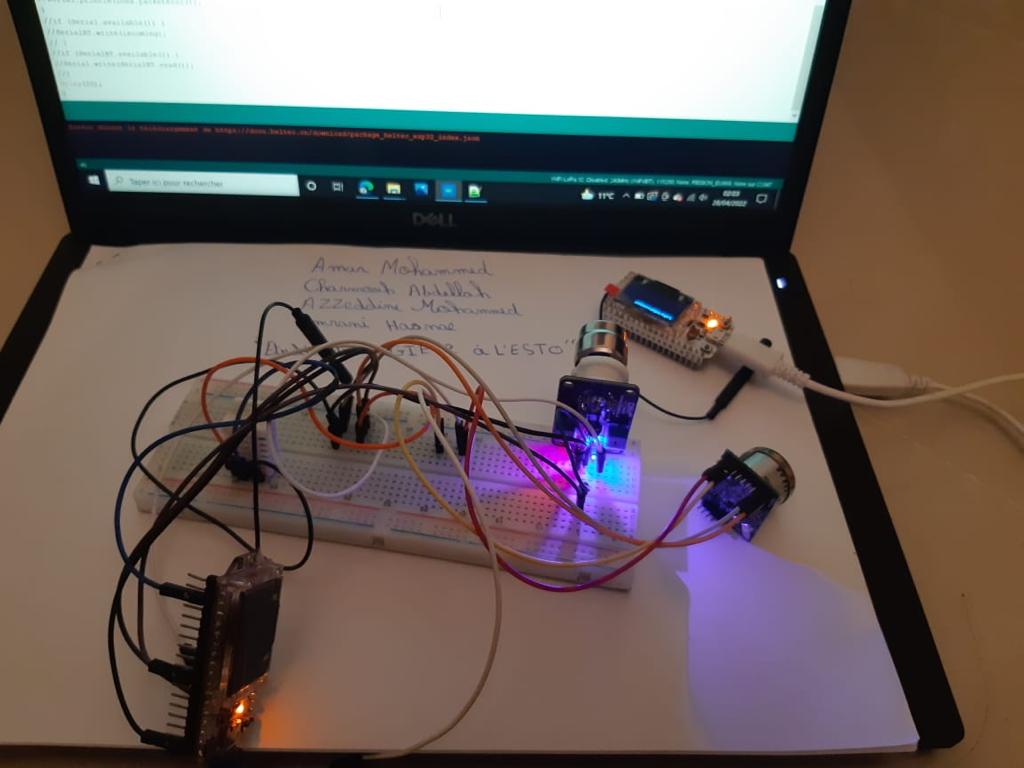
Notre projet est une implémentation d’un système pour la détection des feux de forêt à base des réseaux de capteurs sans fil appelé SDF-RCSF :

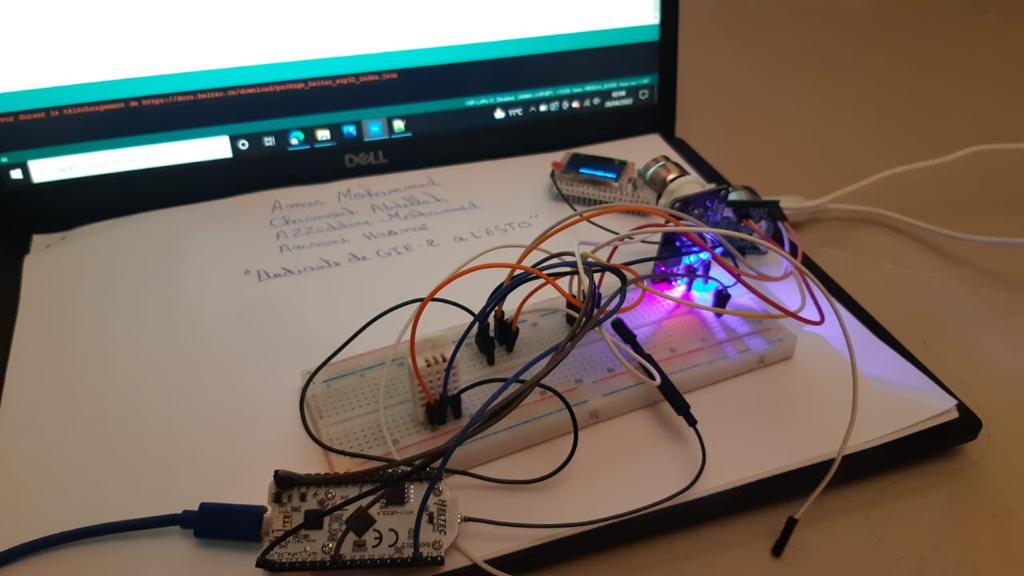
La méthodologie suivie pour la mise en œuvre de ce système s’articule autour des phases suivantes : phase de conception indépendamment des choix techniques. Phase de réalisation et phase de test.

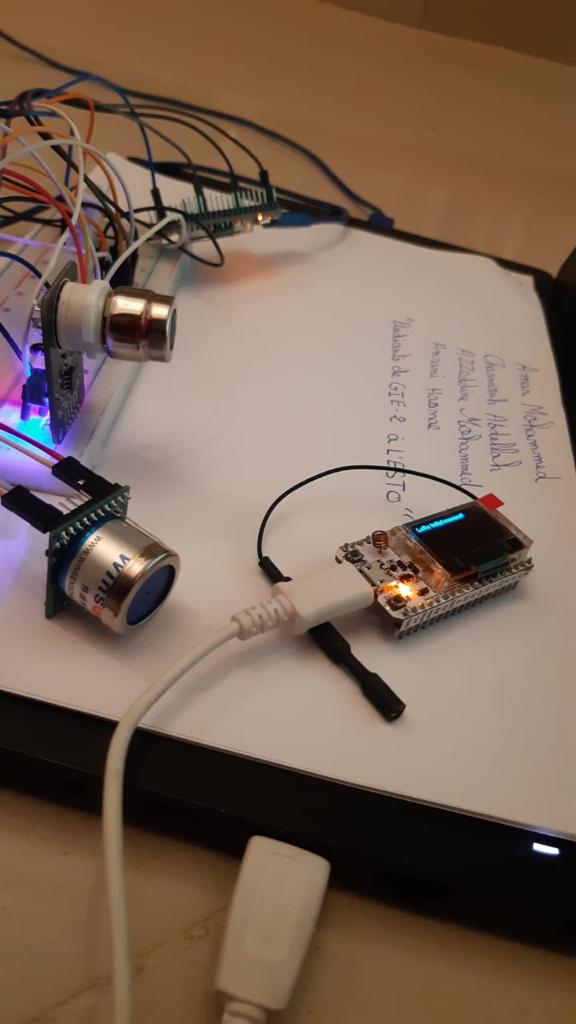
Le prototype que nous avons réalisé est capable de détecter les déclenchements des feux, ainsi qu’il permet le suivi des valeurs en temps réel.

Annexe : Brochage et exécution du SDF.

1. Brochage du SDF.







Figures 54, 55 et 56 : Brochage de matériels de SDF-RCSF

1. Exécution finale.

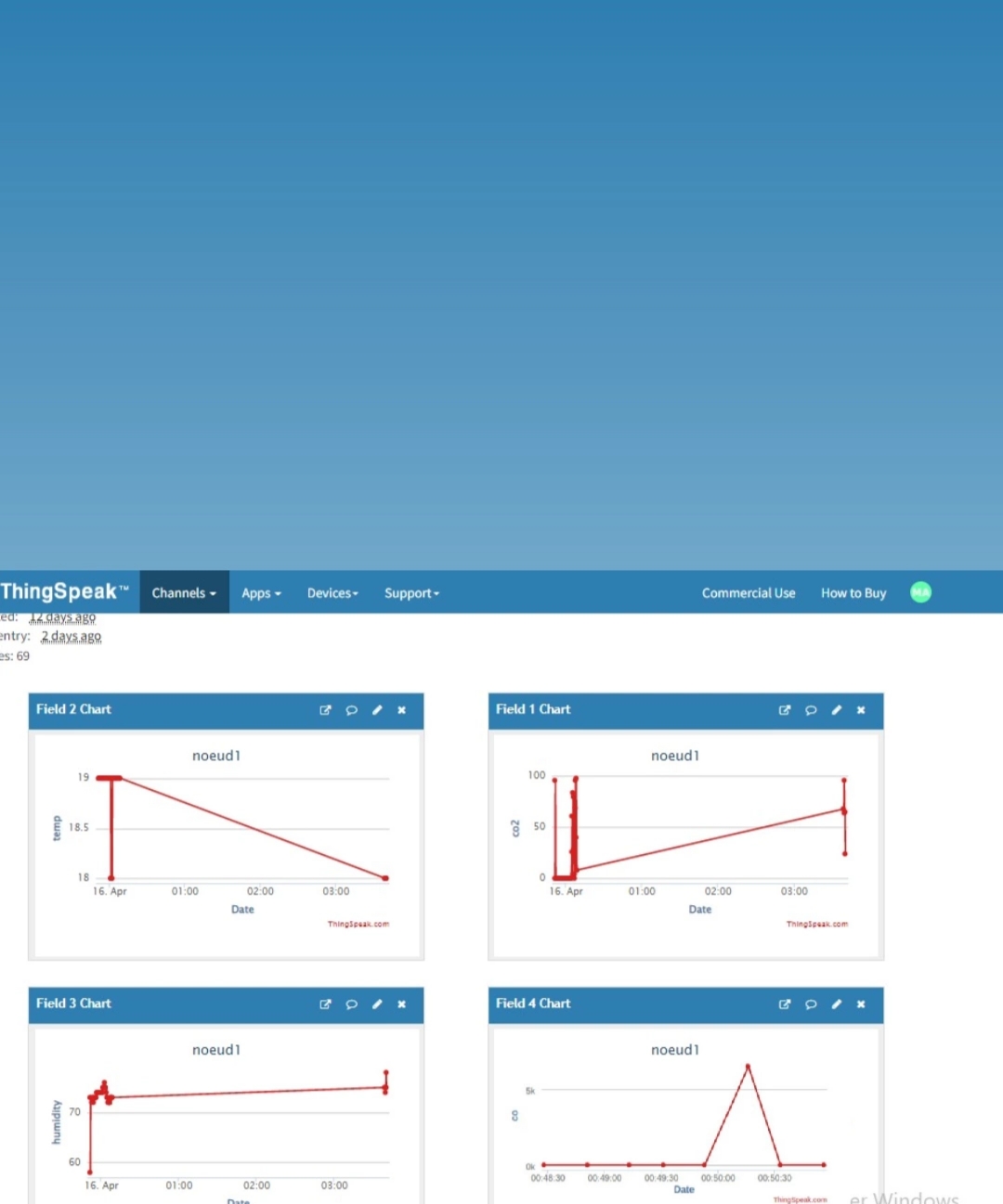


Figure 57 : Exécution finale de projet

Bibliographie :

Learn ESP32, ESP8266, Arduino IDE :

<https://randomnerdtutorials.com/>

Description de la carte raspberry Pi 4 :

<https://www.gotronic.fr/art-carte-raspberry-pi-4-b-4-gb-30209.htm#description>

Phyton exemple on Raspberry Pi 4 :

<https://helloraspberrypi.blogspot.com/>

ESP32 exemple :

<https://arduino-er.blogspot.com/2020/>

How to show the temperature and humidity on a heltec wifi kit 32 attached to a dht 11, dht22 sensor :

<https://www.techcoil.com/blog/how-to-show-the-temperature-and-humidity-of-your-room-on-a-heltec-wifi-kit-32-attached-to-a-dht11-dht22-sensor/>