

Rapport de Travaux Pratiques - 2024.09.22

**Internet des objets - Protocoles LoRaWAN pour le Positionnement sans GPS
par la Trilatération WiFi avec ESP8266 et modem LoRa E5**

Table des matières

| | |
|--|-----------|
| Introduction..... | 2 |
| 1. Calibrage de la mesure du RSSI en fonction de la distance..... | 3 |
| 1.1. Mise en place de points d'accès WiFi avec l'ESP8266..... | 3 |
| Figure 1.1.1 : Diagramme de séquence du déploiement de l'ESP en tant que point d'accès WiFi..... | 3 |
| Figure 1.1.2 : Visualisation du SSID depuis le gestionnaire WiFi d'un ordinateur..... | 4 |
| 1.2 Estimation de la distance à partir du RSSI..... | 4 |
| Figure 1.2.1 : Diagramme de séquence du calcul des distances en fonction des RSSI collectés..... | 5 |
| Figure 1.2.2 : Graphique linéaire de la mesure du RSSI en fonction de la distance pour différentes contraintes environnementales..... | 6 |
| Figure 1.2.3 : Graphique semi-logarithmique de la mesure du RSSI en fonction de la distance pour différentes contraintes environnementales..... | 7 |
| Figure 1.2.4 : Graphique semi-logarithmique comparant la relation entre le RSSI et la distance, avec les valeurs mesurées dans un environnement peu contraint..... | 8 |
| 2. Positionnement estimé et transfert des données par protocole LoRaWAN vers le serveur..... | 9 |
| Figure 2.1.1 : Diagramme de séquence du relevé et de l'envoi sur le réseau TTN des données collectées..... | 10 |
| 3. Visualisation et transfert des données sur le serveur..... | 16 |
| Annexe..... | 17 |
| A. Mise en place de l'environnement de travail..... | 17 |
| A.1 Installation de Node-RED par étapes..... | 17 |
| A.2 Configuration de Arduino IDE pour échanger avec l'ESP..... | 17 |
| A.3 Configuration du matériel TTN par étapes..... | 18 |
| B. Brouillon du Relevé du RSSI en dB en fonction de la distance à la source dans différents environnements..... | 21 |

Introduction

Dans un environnement où les objets connectés, notamment les objets connectés mobiles comme les robots convoyeurs en entrepôts ou encore un simple smartphone, occupent une place de plus en plus importante, en particulier dans les environnements de travail fortement automatisés, il devient essentiel de pouvoir localiser des capteurs, des personnes et des objets mobiles. Ce système est appelé IPS (système de géolocalisation en intérieur).

En effet, dans des zones intérieures telles que les centres commerciaux, les parkings, les usines, ou même certaines rues étroites bordées de bâtiments hauts, comme le Paris d'avant [les grands travaux d'élargissement des rues menés par le préfet Haussmann](#), ou encore dans des zones forestières denses comme à Saclay, les signaux haute fréquence, à courte longueur d'onde des GPS satellites, dont la portée dépend d'une ligne de vue dégagée, sont souvent trop faibles pour traverser les bâtiments aux murs épais en béton.

Ainsi, pour guider avec précision une flotte de cyclistes en livraison ou pour orienter une personne dans un centre commercial, une nouvelle technologie de positionnement en intérieur est nécessaire. Nous avons exploré une méthode de trilatération par WiFi Sniffing qui permet de détecter les BSSID (adresses MAC uniques de l'interface radio) et leurs niveaux RSSI (indicateurs de puissance du signal) pour trianguler une position avec des balises de positionnement.

Après avoir établi une relation entre la distance à un émetteur WiFi et la puissance du signal reçu, les données seront transmises via le protocole LoRaWAN à un serveur où elles seront croisées avec une base de données telle que [Geolocalisation API de Google](#) ou [WIGLE](#).

Pour une meilleure gestion, portativité et communication de notre projet, nous avons, sur les conseils de Thibault, mis en place un suivi avec [GitHub](#) et une remise en ligne via [Google Doc](#).

Aide rédactionnelle, au débogage, au code, soutien et remerciement : Daniel, Yulin, Maxime, Ayman, Victor, Quentin, Karlitou, Sekouba, Nicolas, ChatGPT, HARIAN Elyoth, DOUZE Yann, Benjamin, l'incroyablissime Sylvain VIATEUR.

Lien utile : https://fr.wikipedia.org/wiki/Système_de_positionnement_en_intérieur

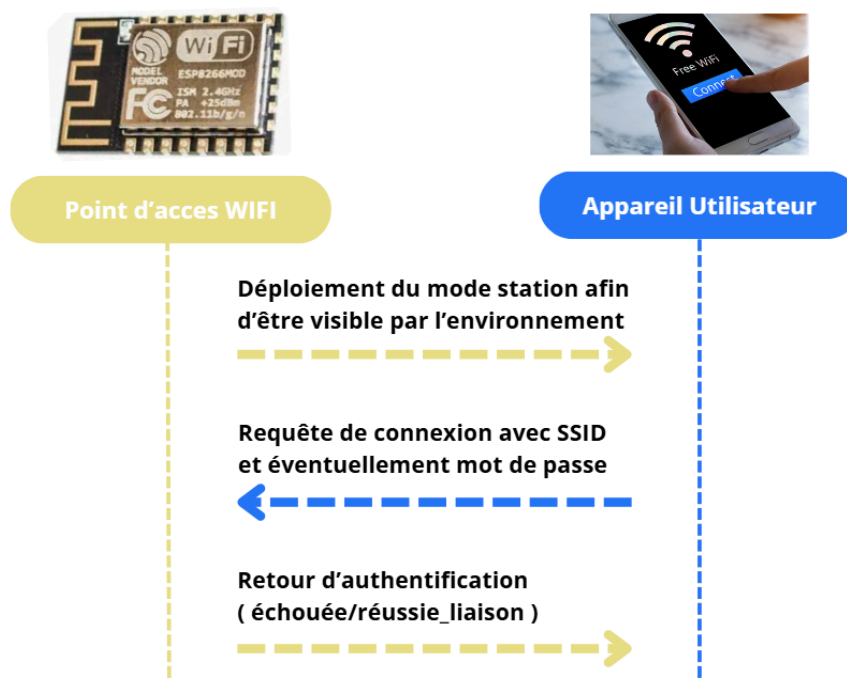
1. Calibrage de la mesure du RSSi en fonction de la distance

Dans cette première partie, nous chercherons à développer une méthode pour estimer la distance entre un appareil et une source émettant un signal WiFi. Cette estimation permettra, par la suite, de trianguler la position d'un nouvel appareil.

1.1. Mise en place de points d'accès WiFi avec l'ESP8266

Après avoir configuré l'environnement de travail disponible dans l'annexe [A. Mise en place de l'environnement de travail](#), nous allons, avec nos copromotionnaires, déployer un réseau d'ESP en mode AP (point d'accès WiFi) dont les emplacements seront connus. Cette technologie, fondée sur les normes IEEE 802.11, permet la communication de manière non filaire entre différents appareils, facilitant l'accès à Internet et aux réseaux locaux sur des [distances modérées](#). Elle offre des vitesses de transmission de plus en plus élevées grâce à l'amélioration de la bande passante et à l'utilisation de hautes fréquences, bien que celles-ci entraînent un coût énergétique croissant avec l'évolution des générations WiFi demandant une puissance plus élevée pour maintenir un signal stable sur de longues distances.

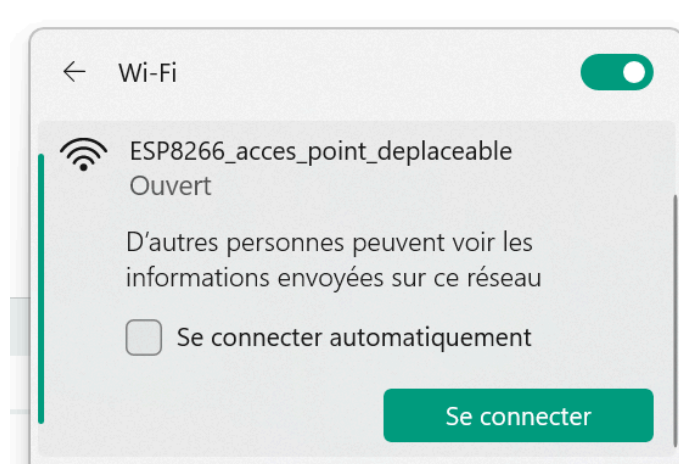
Figure 1.1.1 : Diagramme de séquence du déploiement de l'ESP en tant que point d'accès WiFi.



Nous avons choisi un débit de transmission légèrement plus lent que celui des affichages typiques, afin de pouvoir visualiser progressivement l'apparition des informations transmises. Cependant, en cas de besoin d'échanges plus lourds ou de communications en temps réel, il serait nécessaire d'augmenter le baudrate.

De plus, nous avons mis en place un système d'affichage des logs dans le terminal Serial Monitor, permettant d'afficher de manière récurrente l'adresse MAC de l'ESP et son SSID associé. Avec une légère modification, ces logs pourraient, dans une éventuelle future application, être affichés sur un mini module d'écran LCD, ce qui permettrait de retrouver l'adresse MAC d'un point d'accès sans avoir à passer par le Serial Monitor.

Figure 1.1.2 : Visualisation du SSID depuis le gestionnaire WiFi d'un ordinateur.



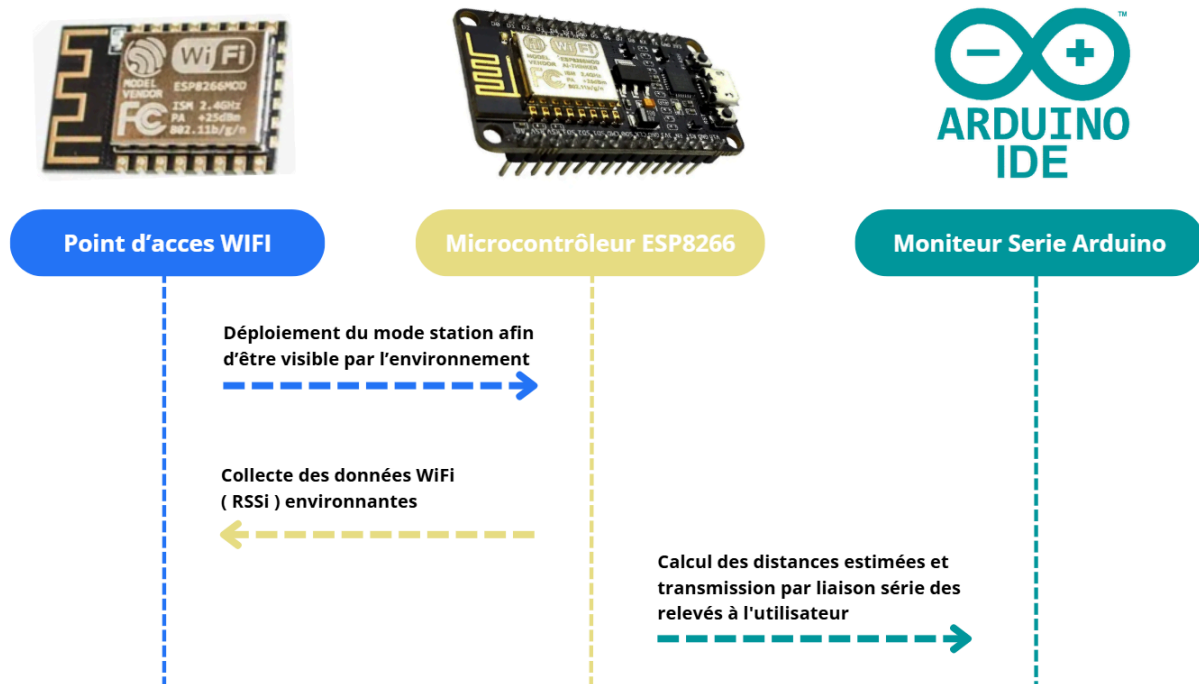
Nous constatons ainsi la bonne publication de notre réseau WiFi en mode point d'accès, ce qui conclut cette section.

1.2 Estimation de la distance à partir du RSSI

Dans cette partie, nous cherchons à établir la relation entre la puissance du signal reçu par notre ESP et sa distance par rapport aux appareils émettant un signal WiFi, tels que les ESP en mode point d'accès.

Le Wi-Fi est ici une technologie pertinente, car il permet non seulement de transmettre des données, mais aussi de fournir des informations de localisation à partir de l'intensité du signal reçu (RSSI) et des identifiants uniques des points d'accès WiFi (adresses MAC). En connaissant l'emplacement de chaque appareil connecté, un système de positionnement efficace et peu coûteux peut être mis en place. Bien que le WiFi soit énergivore, sa disponibilité dans les environnements urbains en fait une solution viable.

Figure 1.2.1 : Diagramme de séquence du calcul des distances en fonction des RSSI collectés.



Dans le cadre de ce projet, nous avons évalué la portée du signal en WiFi pour estimer les distances. Nous posons initialement l'hypothèse d'une relation linéaire entre la distance et le RSSI, selon la formule :

$$d = C_{P_{\text{linéique}}} \times RSSI + C^{ST0}$$

où d est la distance entre le récepteur et l'émetteur en *mètres*,

$C_{P_{\text{linéique}}}$ est le coefficient de puissance linéique du signal reçu en $m. dB^{-1}$,

$RSSI$ est la force du signal reçu en dB ,

C^{ST0} une constance correspondant à l'offset en *mètres*.

Nous proposons de réaliser les mesures dans trois conditions distinctes pour observer l'effet des obstacles sur la puissance du signal. Ainsi, à plusieurs distances (de 0 m, avec les appareils collés l'un à l'autre, jusqu'à 10 m), nous mesurerons :

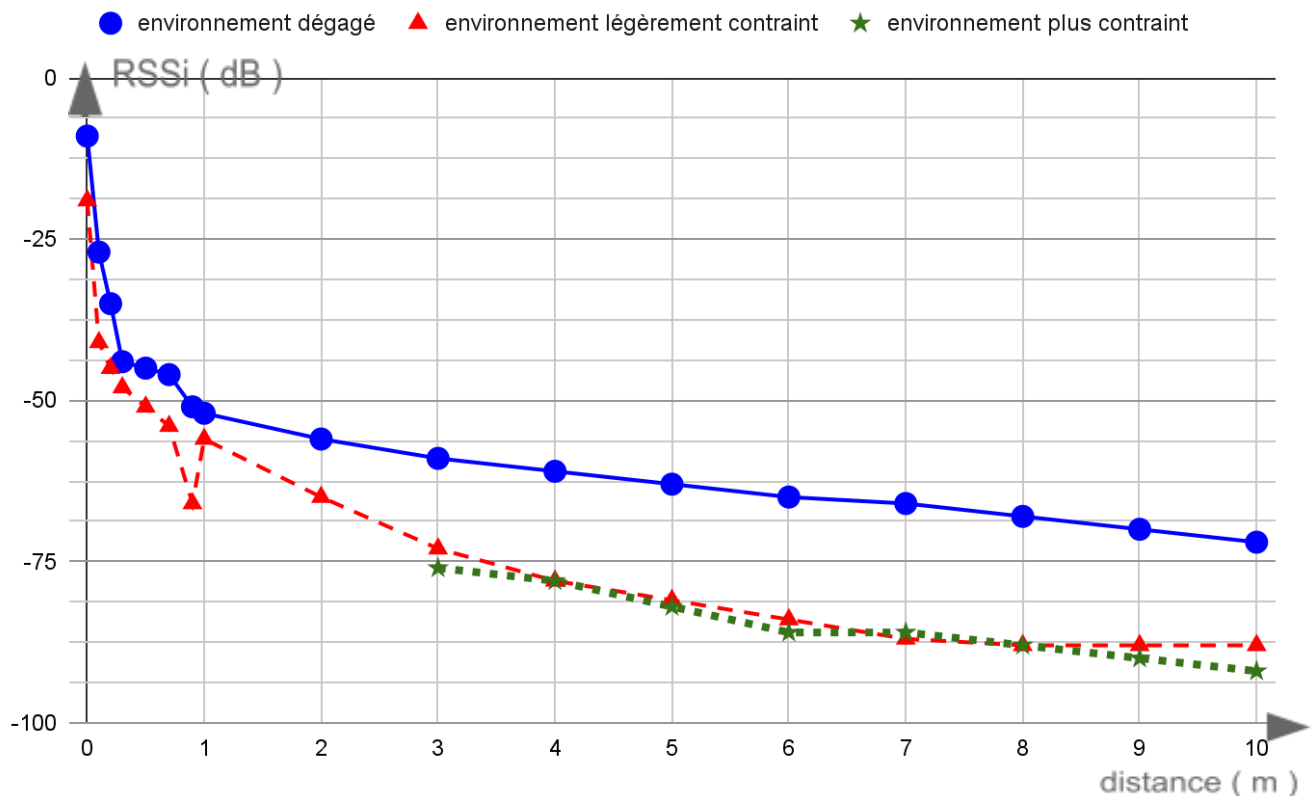
v) Dans un environnement dégagé, sans obstacle entre l'émetteur et le récepteur.

vv) Dans un environnement légèrement contraint (présence d'obstacles mineurs tels qu'un écran d'ordinateur, une table, une chaise, du matériel de bureau, ...).

vvv) Dans un environnement plus contraint, avec des obstacles significatifs, tels qu'un ou plusieurs murs voir un plancher.

Ces mesures nous permettront d'évaluer si l'impact des obstacles entre la source et le récepteur est suffisamment important pour être pris en compte ou s'il peut être négligé.

Figure 1.2.2 : Graphique linéaire de la mesure du RSSI en fonction de la distance pour différentes contraintes environnementales.



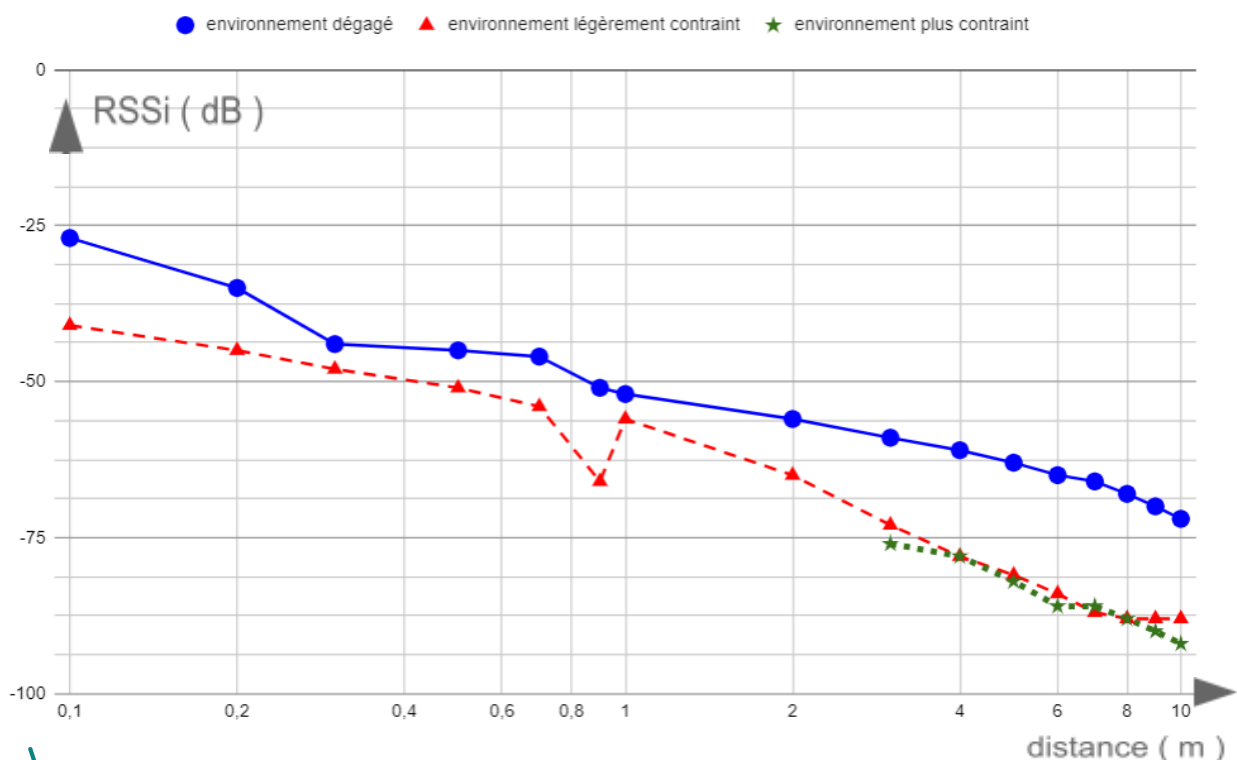
Graphique accessible aux daltoniens - 4% population française

À la précision des relevés de mesure près, nous obtenons, malgré quelques anomalies (notamment un décalage à 1 m dans l'environnement dégagé, dû à une reprise des mesures à une distance plus proche) une tendance significative qui valide partiellement notre modèle. En effet, il semble acceptable d'utiliser une relation linéaire pour une distance allant de 1 à plusieurs mètres, en négligeant les très petites distances. Cela peut être suffisant pour des cas de localisation approximative, comme pour localiser un magasin dans un centre commercial, où une marge de 3 à 5 mètres dans la largeur des couloirs n'aura pas d'impact significatif. Cette approche peut également s'appliquer à un environnement comme une forêt dense, avec des points de connexion WiFi dispersés.

Ainsi, nous obtenons une loi relativement correcte pour les distances allant de 1 à 10 mètres, avec un coefficient de décroissance linéaire de -2 m. dB^{-1} et un offset raisonnable de -60 dB . Nous postulons qu'il est préférable de surestimer la distance plutôt que de la sous-estimer, afin de créer des rayons de distance autour des émetteurs qui se superposent, délimitant ainsi une zone probable de localisation.

De plus, nous constatons que les différences de contraintes environnementales (peu ou fortement contraint) n'ont pas d'impact significatif pour des distances de quelques mètres.

Figure 1.2.3 : Graphique semi-logarithmique de la mesure du RSSI en fonction de la distance pour différentes contraintes environnementales.



Graphique accessible aux daltoniens - 4% population française

Dans un second temps, pour une meilleure approximation de la relation entre le RSSI et la distance, nous observons que cette relation semble plutôt logarithmique. En représentant la distance sur une échelle semi-logarithmique, nous obtenons des droites plus marquées et cohérentes, indiquant une relation plus fidèle entre le RSSI et la distance. Une relation logarithmique permettrait de couvrir les courtes distances, ce qui est particulièrement pertinent pour les applications de localisation plus précise.

Nous utiliserons donc la courbe rouge-triangle dans un environnement légèrement contraint pour trouver une relation plus fidèle à la réalité, et telle que :

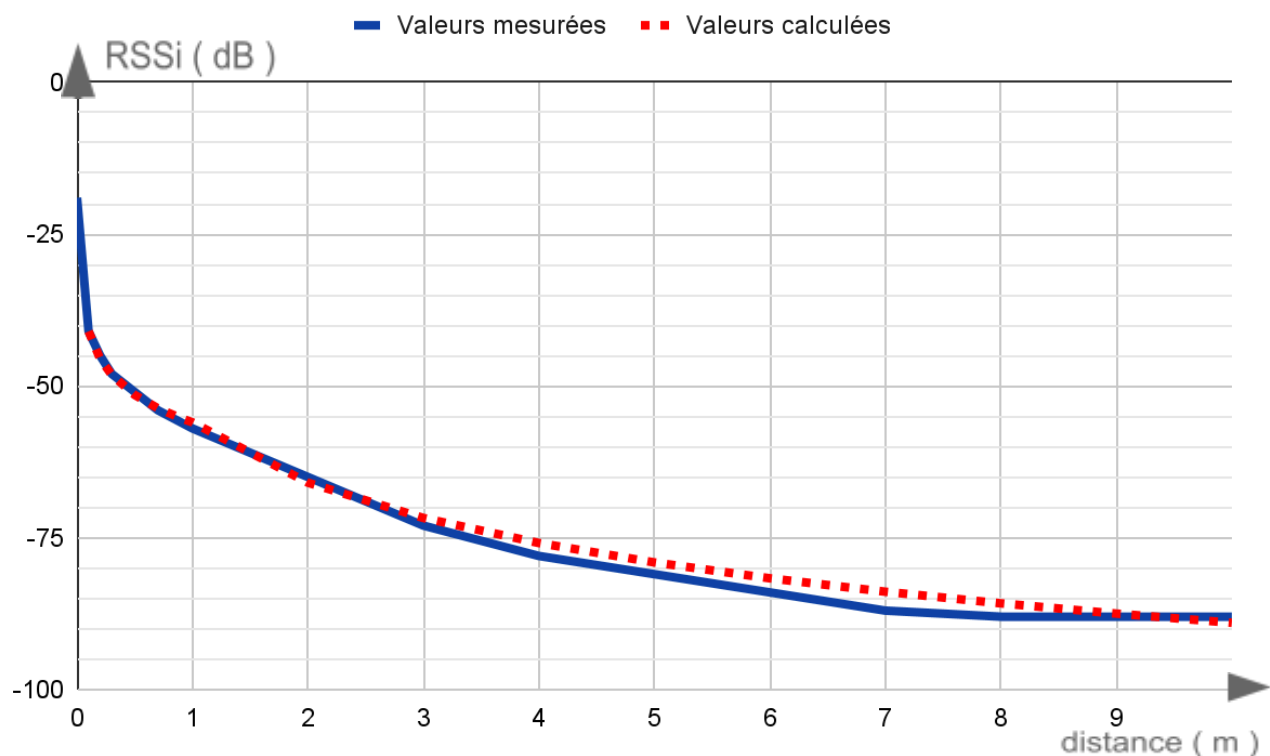
$$RSSi = (-41 - 15\log_{10}(10d)) \prod_{[0,1[} + (-56 - 33\log_{10}(d)) \prod_{[1,\infty[}$$

La fonction a deux principales parties :

a) Pour les distances de 0,1 m à 1 m avec - 41dB d'offset et une pente de -15dB par décade (où $\prod_{[0,1[}$ est une fonction porte valant 1 de 0 à 1 exclu et 0 ailleurs).

aa) Pour les distances de 1 m à 10 m avec - 56dB d'offset et une pente de -33dB par décade (où $\prod_{[1,\infty[}$ est une fonction porte valant 1 de 1 à l'infini et 0 ailleurs).

Figure 1.2.4 : Graphique semi-logarithmique comparant la relation entre le RSSi et la distance, avec les valeurs mesurées dans un environnement peu contraint.



Ainsi, nous obtenons la relation suivante :

$$\forall RSSi \in [- 56 ; + \infty[: d (RSSi) = 10^{-\left(\frac{RSSi + 56}{15}\right)}$$

$$\forall RSSi \in] - \infty ; - 56[: d (RSSi) = 10^{-\left(\frac{RSSi + 56}{33}\right)}$$

Après de nouvelles expérimentations, nous avons ajusté la première partie de la relation RSSI-distance en augmentant la pente pour les courtes distances. Ce choix repose sur l'observation que, dans des environnements dégagés, le signal est souvent de meilleure qualité à proximité immédiate des émetteurs, ce qui permet d'obtenir des relevés de RSSI plus précis et cohérents.

Nous avons également affiné notre formule d'estimation de distance de manière empirique, en nous concentrant sur la plage de 0,1 à 1 mètre sans obstacle. Cette portion de la courbe a montré une meilleure stabilité et précision. En revanche, les expérimentations en extérieur, bien que plus limitées, ont révélé une forte variabilité des résultats, probablement due à une dégradation du signal dans ces conditions. Les mesures extérieures doivent donc être interprétées avec précaution, car elles peuvent entraîner une imprécision accrue dans l'estimation des distances.

2. Positionnement estimé et transfert des données par protocole LoRaWAN vers le serveur

Dans cette deuxième partie, nous allons proposer un modèle pour approcher les coordonnées gps d'un récepteur par les relevés reçu en première partie et de les envoyer pour leurs exploitation graphique.

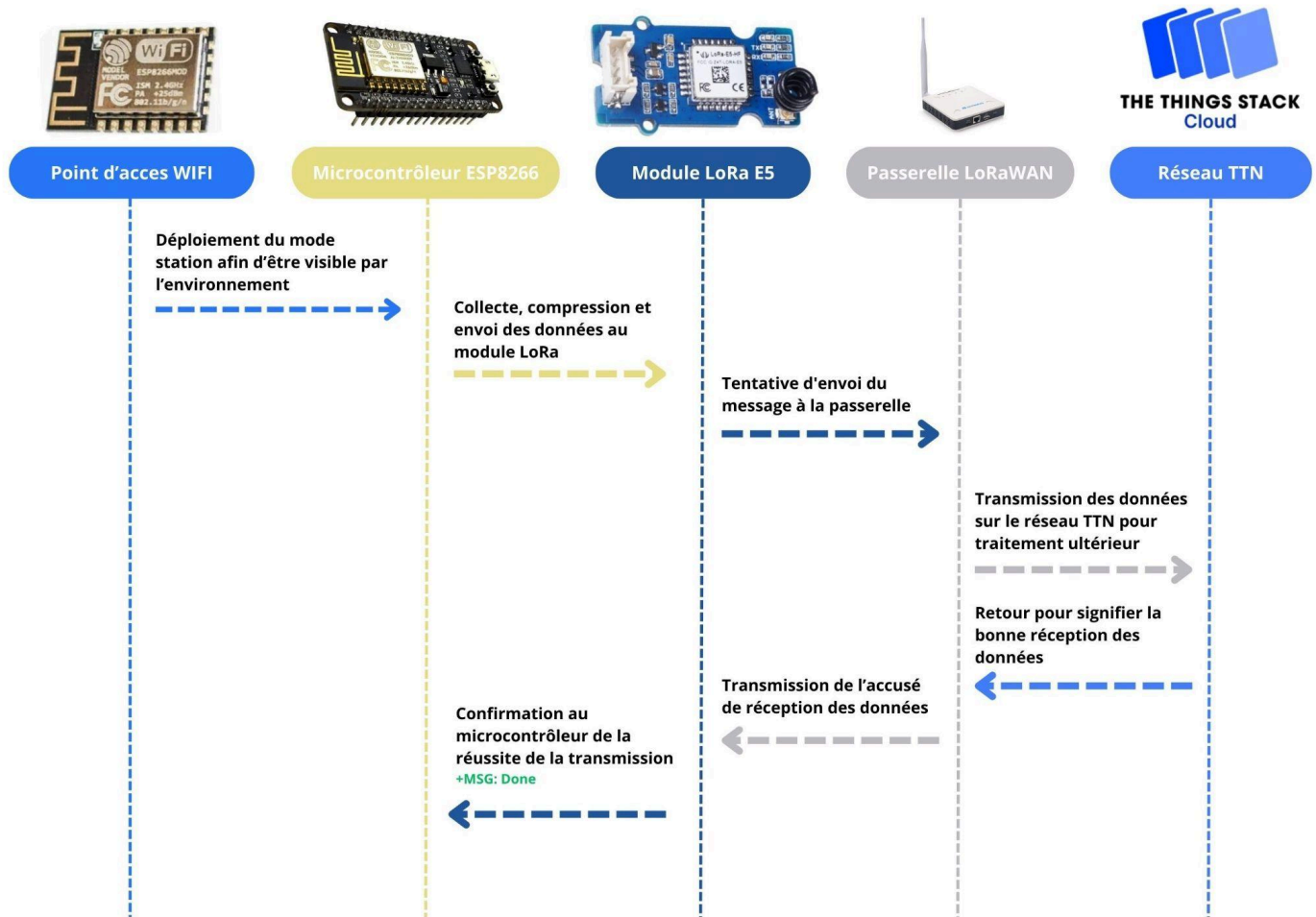
2.1. Modèle de trilatération

Pour estimer la position GPS basée sur les données WiFi et les relevés de RSSI, nous proposons ici un modèle de trilatération. Ce modèle repose sur l'idée que les valeurs les plus faibles de RSSI (signal fort) indiquent une proximité relative avec l'émetteur, car en intérieur ou en milieu urbain avec des obstacles, seuls des récepteurs proches peuvent capter un signal suffisamment fort.

Les mesures de RSSI pour un même point et réseau peuvent fluctuer dans le temps, en fonction des légers changements d'environnement, comme le passage d'une personne ou l'orientation du récepteur. Par conséquent, nous attribuons un poids plus élevé aux RSSI les plus faibles, car ils sont davantage représentatifs de courtes distances dans ces contextes. En revanche, des RSSI élevés ne garantissent pas systématiquement une grande distance, car des interférences peuvent fausser les mesures pour les points plus éloignés.

Idéalement, il aurait été préférable de disposer de relevés de RSSI projetés sur un plan en 2D pour minimiser les perturbations dans l'estimation des positions.

Figure 2.1.1 : Diagramme de séquence du relevé et de l'envoi sur le réseau TTN des données collectées.



Pour ce modèle, nous utilisons une moyenne pondérée des distances calculées, en affectant un poids plus élevé aux distances courtes (RSSI fort) et un poids plus faible aux distances longues (RSSI faible). Ce système permet d'obtenir une estimation plus stable des coordonnées GPS en intégrant la variabilité des RSSI dans le calcul de distance. Ainsi nous obtenons comme 1er modèle :

\forall point émetteur p_n , de coordonnées (x_n, y_n, z_n) où x_n la latitude, y_n la longitude et z_n l'altitude, et de distance estimée d_n où n le nombre de points en compte.
Les coordonnées $(x_{\text{recepteur}}, y_{\text{recepteur}})$ sont alors :

$$x_{recepteur} = \frac{\sum_{n=1}^n \left(x_n \times \frac{1}{d_n} \right)}{\sum_{n=1}^n \left(\frac{1}{d_n} \right)} \quad \text{et} \quad y_{recepteur} = \frac{\sum_{n=1}^n \left(y_n \times \frac{1}{d_n} \right)}{\sum_{n=1}^n \left(\frac{1}{d_n} \right)}$$

Ce modèle admet ces limites dans le cas où la coordonnée d'altitude z , n'est pas prise en compte au vu de l'affichage sur une carte en 2D. Ce qui peut causer des erreurs d'estimations pour les points en altitude élevée ou basse par rapport au récepteur, compensé par la moins bonne prise en compte des fortes distances, souvent le cas de ces relevés étant à d'autres étages pour les relevés en intérieur, dont le signal WiFi doit traverser une ou plusieurs couches de plancher plus ou moins épais et affaiblissant le signal.

Bien que le modèle où le coefficient de pondération est fractionnaire du type

$$\frac{f(d_n)}{g(d_n)} = \frac{1}{(d_n)^1} \text{ où ici } f(d_n) = 1 \text{ et } g(d_n) = (d_n)^1 \text{ est très pertinent au vu}$$

de sa forte atténuation des longues distances au vu des courtes distances (lorsque $g(d_n)$, fonction de la distance la distance croît). Par ailleurs ce modèle a l'avantage de conserver la pondération bien qu'atténuée des longues distances et non de les supprimer totalement où la pertinence se relève où il n'y aurait que de hautes RRSi et donc supposément que de longues distances.

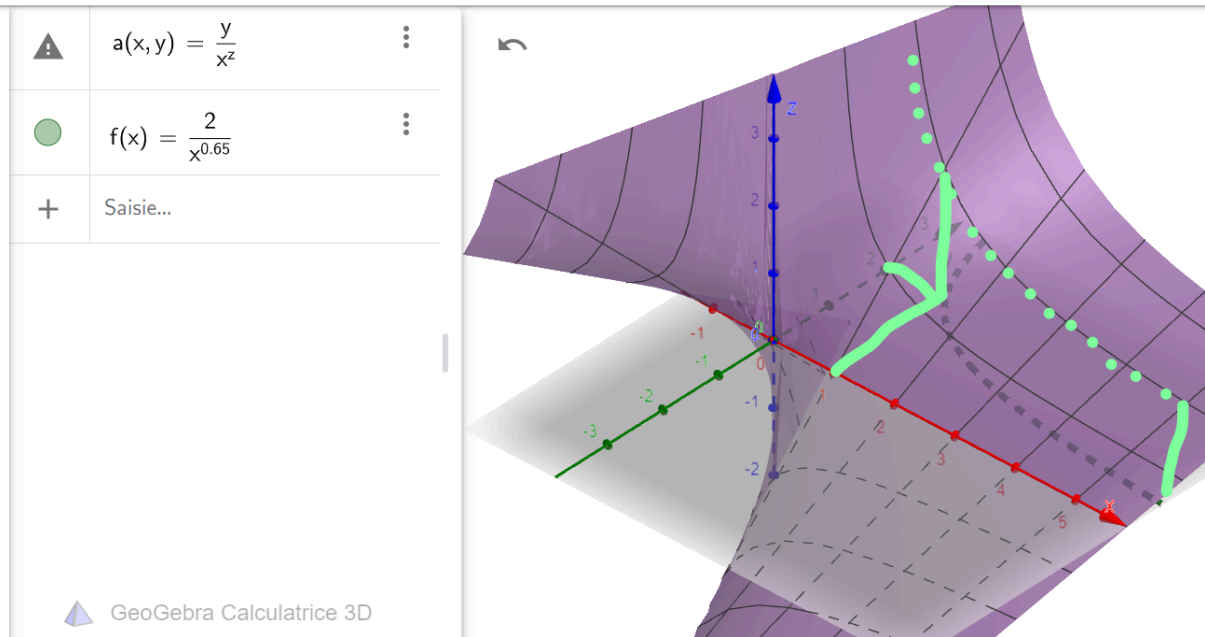
Une plus forte pondération veut être donnée aux distances proches de 1. Nous allons alors chercher un modèle similaire avec une fonction $f(d_n) = Cst$ où

$$Cst \in \mathbb{R} \text{ et } f(d_n) = (d_n)^v \text{ où } v \in \mathbb{R}$$

Support utilisé :

GeoGebra

DISTRIBUTION



Après études des différents cas disponibles, nous choisirons le modèle suivant : $\frac{f(d_n)}{g(d_n)} = \frac{2}{(d_n)^{0,65}}$ où ici $f(d_n) = 2$ et $g(d_n) = (d_n)^{0,65}$

Celui ci à pour avantage une pente plus douce pour conserver les distances proches de 1m par défaut tout en donnant un poids plus modéré au distance proche de 1m par excès.

Ainsi l'équation finalement retenue est :

$$x_{\text{recepteur}} = \frac{\sum_{n=1}^n \left(x_n \times \frac{2}{(d_n)^{0,65}} \right)}{\sum_{n=1}^n \left(\frac{2}{(d_n)^{0,65}} \right)} \quad \text{et} \quad y_{\text{recepteur}} = \frac{\sum_{n=1}^n \left(y_n \times \frac{2}{(d_n)^{0,65}} \right)}{\sum_{n=1}^n \left(\frac{2}{(d_n)^{0,65}} \right)}$$

LoRaWAN (support de communication) : Il est utilisé uniquement pour la transmission des données (= adresses MAC des points d'accès et leurs RSSI) vers serveur central

Applications > Transmission de données ESP82... > End devices > grove-lora-e5-stm32wle5jc > **Live data**

grove-lora-e5-stm32wle5jc
ID: grove-lora-e5-stm32wle5jc

Last activity 13 seconds ago • 55 up / 15 (Nwk) down

Device overview **Live data** Messaging Location Payload formatters Settings

TIME TYPE DATA PREVIEW

Verbose stream ☒ Export as JSON Pause Clear

| | | | |
|----------|-------------------------------------|---|--|
| 17:58:05 | Forward uplink data message | DevAddr: 26 08 61 96 Payload: { nb_Appareils: [], nb_Reseaux: 14, numero_envoi: 2, total_envoi: 2 } | 22 0E 08 47 32 5C 68 61 AF AC 1F 09 08 A6 AE B7 58 D9 D5 41 58 98 A8 |
| 17:58:05 | Successfully processed data message | DevAddr: 26 08 61 96 | |
| 17:49:54 | Fail to send webhook | Request | |
| 17:49:53 | Forward uplink data message | DevAddr: 26 08 61 96 Payload: { nb_Appareils: [], nb_Reseaux: 14, numero_envoi: 1, total_envoi: 2 } | 12 0E EE FA BC A8 8A 54 C9 58 97 BD 9C B2 02 A7 58 97 BD 9C B2 81 A8 |
| 17:49:53 | Successfully processed data message | DevAddr: 26 08 61 96 | |
| 17:49:48 | Fail to send webhook | Request | |
| 17:49:38 | Forward uplink data message | DevAddr: 26 08 61 96 Payload: { nb_Appareils: [], nb_Reseaux: 17, numero_envoi: 3, total_envoi: 3 } | 33 11 58 97 BD CD 92 62 B7 58 97 BD CD 92 61 B4 58 97 BD CD 92 63 B4 |
| 17:49:38 | Successfully processed data message | DevAddr: 26 08 61 96 | |
| 17:49:27 | Fail to send webhook | Request | |
| 17:49:27 | Forward uplink data message | DevAddr: 26 08 61 96 Payload: { nb_Appareils: [], nb_Reseaux: 17, numero_envoi: 2, total_envoi: 3 } | 23 11 D8 47 32 5C 68 61 B1 AC 1F 09 08 A6 AE B6 58 97 BD 9C B2 83 A8 |
| 17:49:27 | Successfully processed data message | DevAddr: 26 08 61 96 | |
| 17:49:16 | Fail to send webhook | Request | |
| 17:49:14 | Forward uplink data message | DevAddr: 26 08 61 96 Payload: { nb_Appareils: [], nb_Reseaux: 17, numero_envoi: 1, total_envoi: 3 } | 13 11 EE FA BC A8 8A 54 C7 86 F3 EB 58 B5 15 A4 58 D9 D5 41 58 98 A4 |
| 17:49:14 | Successfully processed data message | DevAddr: 26 08 61 96 | |
| 17:49:08 | Fail to send webhook | Request | |

Applications > Transmission de données ESP82... > End devices > grove-lora-e5-stm32wle5jc > **Live data**

grove-lora-e5-stm32wle5jc
ID: grove-lora-e5-stm32wle5jc

Last activity 6 seconds ago • 71 up / 15 (Nwk) down

Device overview **Live data** Messaging Location Payload formatters Settings

TIME TYPE DATA PREVIEW

Verbose stream ☒ Export as JSON Pause Clear

| | | | |
|----------|--------------------------------|---|-------------|
| 17:49:54 | Fail to send webhook | Request | |
| 17:49:53 | Forward uplink data message | DevAddr: 26 08 61 96 Payload: { nb_Appareils: [], nb_Reseaux: 14, numero_envoi: 1, total_envoi: 2 } | 12 0E EE FA |
| 17:49:53 | Successfully processed data... | DevAddr: 26 08 61 96 | |
| 17:49:48 | Fail to send webhook | Request | |
| 17:49:38 | Forward uplink data message | DevAddr: 26 08 61 96 Payload: { nb_Appareils: [], nb_Reseaux: 17, numero_envoi: 3, total_envoi: 3 } | 33 11 58 97 |
| 17:49:38 | Successfully processed data... | DevAddr: 26 08 61 96 | |
| 17:49:27 | Fail to send webhook | Request | |
| 17:49:27 | Forward uplink data message | DevAddr: 26 08 61 96 Payload: { nb_Appareils: [], nb_Reseaux: 17, numero_envoi: 2, total_envoi: 3 } | 23 11 D8 47 |
| 17:49:27 | Successfully processed data... | DevAddr: 26 08 61 96 | |
| 17:49:16 | Fail to send webhook | Request | |
| 17:49:14 | Forward uplink data message | DevAddr: 26 08 61 96 Payload: { nb_Appareils: [], nb_Reseaux: 17, numero_envoi: 1, total_envoi: 3 } | 13 11 EE FA |
| 17:49:14 | Successfully processed data... | DevAddr: 26 08 61 96 | |

Bien que 152, on peut économiser un quartet de donne celui transmettant le nb_total de message qui seront envoyés, déduissable depuis le nb_d'appareil (modulo 7) pour faire un multiplexeur pour pouvoir avoir différents formats de payload, (les 4 premiers bits serviraient donc à indiquer 16 différents types de formatage de la donnée pour leur décompression adéquate sous différents formats)

3. Visualisation et transfert des données sur le serveur

Enfin pour visualiser de façon dynamique la position nous avons effectué à l'aide de la source suivante : https://youtu.be/9H6GFXatOCY?si=2tuFL5xH_2H9Yp83 , repris, complété et adapté le transfert des données décompactées depuis le réseaux TTN à notre serveur Node-RED dont la mise en place est disponible à l'annexe [A.1 Installation de Node-RED par étapes](#).

Figure 4.2 : Schéma des flux du serveur Node-RED pour la montée des données en MQTT.

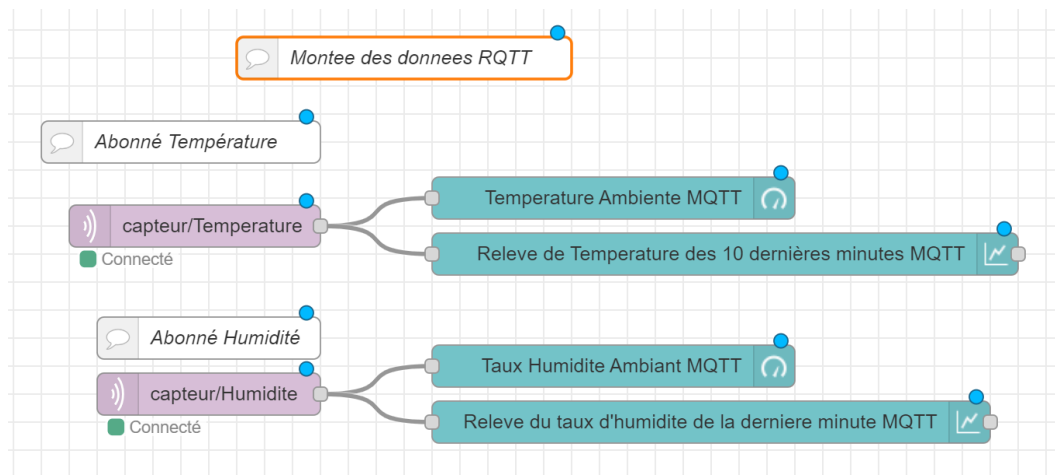
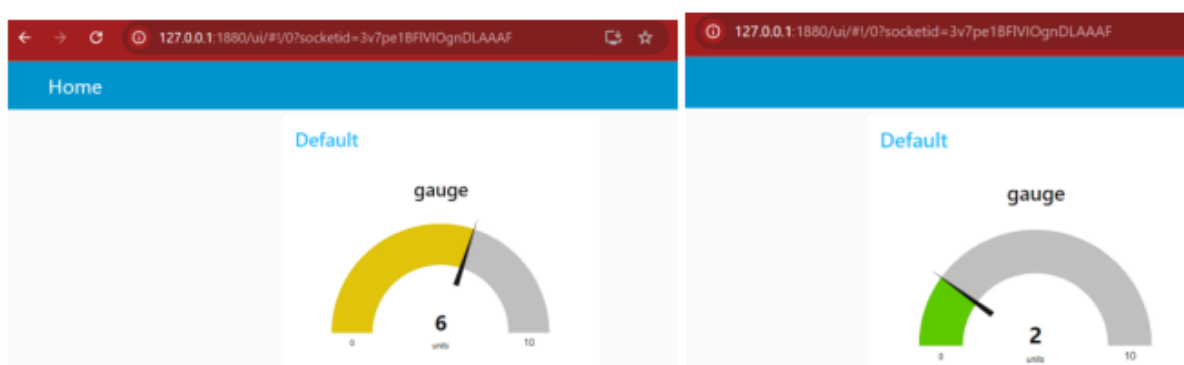


Figure 4.3 : Interface Utilisateur visualisant montée des données en MQTT.



Annexe

A. Mise en place de l'environnement de travail

A.1 Installation de Node-RED par étapes

Pour explorer la communication bidirectionnelle de manière plus visuellement agréable, avec le protocole HTTP REST (envoie requête serveur) MQTT (intermédiaire d'un serveur Broker gérant la réception et renvoie des sujets) et nous aurons besoin d'un serveur ici Node-RED installable tel que :

1) Installation de Node.js depuis :

<https://nodejs.org/dist/v20.17.0/node-v20.17.0-x64.msi>

En cochant l'installation

automatique des outils.

2) Vérification de l'installation depuis un terminal de commande : `node -v; npm`

`-v`

3) Installation de Node-RED depuis le terminal de commande : `npm install -g --unsafe-perm node-red`

4) Lancement depuis le terminal de commande avec : `node-red`

5) Ouverture du lien dans un navigateur sans fermer le terminal de commande :

<http://127.0.0.1:1880/>

6) Installation du Dashbord : `Gérer la palette-> installer -> DashBordNodeRed`

A.2 Configuration de Arduino IDE pour échanger avec l'ESP

1) Installation de Arduino IDE depuis :

<https://downloads.arduino.cc/arduino-ide/arduino-ide-windows.exe>

2) Dans Arduino IDE, indiquer le module à utiliser depuis `File->Preferences...->` et indiquer dans « Additional boards manager URLs : » l'url depuis recherche internet « esp8266 url for arduino ide » :

https://raw.githubusercontent.com/esp8266/Arduino/master/packages/esp8266/package_index.json / https://dl.espressif.com/dl/package_esp32_index.json,
http://arduino.esp8266.com/stable/package_esp8266com_index.json

sur le site

<https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/> ou celui de l'ESP8266

3) Installation de la carte ESP depuis **Tools->Board->Boards Manager...** : « **Arduino Nano ESP32** » et dans notre cas **ESP8266**

4) Après avoir relié le capteur avec le microcontrôleur, **Tools->Board : Node MCU 1.0 (ESP-12E Module) -> esp8266 -> Node MCU 1.0 (ESP-12E Module)**

5) Installation librairie LoRa :

a)

<https://www.gotronic.fr/art-module-lora-e5-grove-113020091-33673.htm>

b) Fiche technique :

https://wiki.seeedstudio.com/Grove_LoRa_E5_New_Version/

c) Téléchargement de la librairie au format zip

Seed LoRa librairie » -> code -> download zip

d) Inclusion de la librairie téléchargée : **Sketch -> Include Library -> Add .ZIP Library... -> LoRa-E5-main.zip**

5) Installation du pilote (si le port n'est pas détecté) depuis

<https://randomnerdtutorials.com/getting-started-with-esp8266-wifi-transceiver-review/> ->

<https://randomnerdtutorials.com/install-esp32-esp8266-usb-drivers-cp210x-windows/> ->

<https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads> ->

https://www.silabs.com/documents/public/software/CP210x_Windows_Drivers.zip

6) Connecter le Port : **Tools -> Port...-> COM3**

A.3 Configuration du matériel TTN par étapes

Pour garantir une communication correcte entre nos données et notre serveur, il est essentiel de configurer le réseau TTN (The Things Network).

1) Création d'une application depuis le réseau communautaire The Things Network :

<https://eu1.cloud.thethings.network/console/applications>

2) Ajout des modules utilisés : **End devices -> Top end devices -> Register end device**

<https://eu1.cloud.thethings.network/console/applications/echange-lorae5-esp8266/devices/add>

3) Selection du **type du module** tel que :

End device type

Input method ⓘ

- ☒ Select the end device in the LoRaWAN Device Repository
- ☐ Enter end device specifics manually

| | | | | |
|---|--|----------------------------------|----------------------------------|---|
| End device brand ⓘ * | Model ⓘ * | Hardware Ver. ⓘ * | Firmware Ver. ⓘ * | Profile (Region) * |
| <input type="text" value="Seeed Technolog..."/> | <input type="text" value="LoRaWAN Dev Kit"/> | <input type="text" value="1.0"/> | <input type="text" value="1.0"/> | <input type="text" value="EU_863_870"/> |

4) Choix de la **bande de fréquence** utilisée ici standard européen :

Frequency plan ⓘ *

Europe 863-870 MHz (SF9 for RX2 - recommended)

5) Generation d'identifiants réseau aléatoire pour pouvoir se connecter à l'application :

Provisioning information

JoinEUI ⓘ *

A0 A0 A0 A0 A0 A0 A0 A0

Reset

This end device can be registered on the network

DevEUI ⓘ *

70 B3 D5 7E D0 06 59

Generate

1/50 used

AppKey ⓘ *

B3 C2 F5 B9 D8 7A 0B 02 B2 D8 74

Generate

End device ID ⓘ *

grove-lora-e5-stm32wle5jc

B. Brouillon du Relevé du RSSI en dB en fonction de la distance à la source dans différents environnements

| Distance | ESP 327 mur | ESP 324 obstacles | Google Tél environnement à vide : | ESP319 Inconnu |
|----------|-------------|----------------------------|-----------------------------------|----------------|
| 0m | / | -38 qq cm -19 dessus | -31 à qq cm -9 dessus | / |
| 1m | / | -56 | -49 à -52 | / |
| 2m | / | -65 | -56 | / |
| 3m | -76 | -73 | -59 | / |
| 4m | -78 | -78 | -61 | / |
| 5m | -82 | -81 | -63 | / |
| 6m | -86 | -84 | -65 | -92 |
| 7m | -86 à 92 | -87 | -66 | -90 |
| 8m | -86 à 92 | -88 | -68 | -90 |
| 9m | -86 à 92 | -87 à 88 | -70 | -91 |
| 10m | -86 à 92 | -87 à 88 | -72 à -64 | / |
| 0,1m | / | -41 | -27 | / |
| 0,2 | / | -45 | -26 à 35 | / |
| 0,3 | / | -48 | -44 | / |
| 0,5 | / | -51 | -45 | / |
| 0,7 | / | -54 | -46 | / |
| 0,9 | / | -66 | -51 | / |