

# LORA ET IOT : CONNECTIVITÉ LONGUE PORTÉE POUR UN MONDE INTELLIGENT



## Pourquoi LoRa et l'IoT

LoRa offre des communications longue portée et une faible consommation, un atout clé pour les capteurs IoT dans la gestion des ressources intelligentes.



Explorons les liens entre les signaux LoRa et les applications des capteurs IoT !

## Qu'est-ce qu'un signal LoRa ?

LoRa = modulation radio longue portée.  
Utilisé dans les réseaux IoT pour transmettre des données sur des kilomètres.  
Faible consommation d'énergie.

## Qu'est-ce qu'un capteur IoT ?

Capteurs connectés qui collectent et transmettent des données (température, humidité, etc.).  
Application : Villes intelligentes, agriculture, industries.

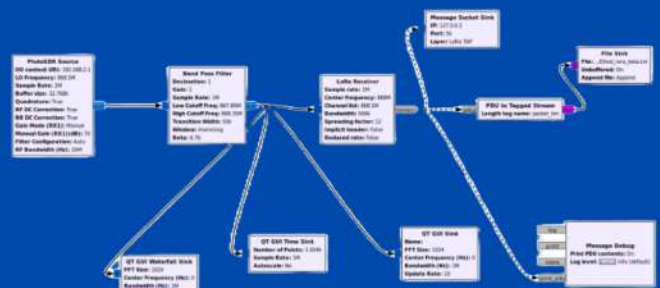
## OBJECTIF

Concevoir un système complet permettant d'intercepter, d'analyser et de visualiser des signaux LoRa via un site interactif, démontrant le potentiel de cette technologie pour les réseaux IoT.



Zones de couverture LoRaWAN (Orange) pour tester et analyser les signaux.

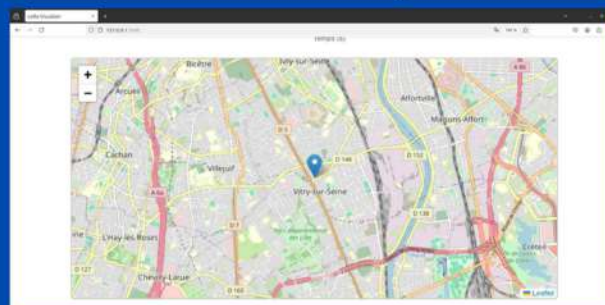
(Recherche d'une zone LoRa outdoor pour optimiser le positionnement.)



Configuration du schéma GNU Radio en respectant les normes européennes : fréquence centrale de **868 MHz**, bande passante de **500 kHz** et facteur d'étalement optimisé (**SF12**).  
Le schéma inclut des blocs pour capter, filtrer et visualiser les signaux LoRa en réduisant les interférences.

Réalisé par **JAVAID Omer** ainsi que **BELOUKA Djibril** dans le cadre de la **SAE 301**.

(Les données captées sont transmises vers un site interactif)



Site interactif développé pour géolocaliser les appareils LoRa détectés grâce à notre SDR. La carte, conçue avec **Leaflet**, est **entièrement interactive** : elle permet de **zoomer** et **naviguer**. En plus de la géolocalisation, les données des signaux LoRa interceptés sont affichées en **temps réel**.

Malgré une configuration conforme aux normes européennes et l'utilisation de filtres pour réduire les interférences, nous avons rencontré des limitations significatives.

Le bruit ambiant, les interférences provenant de dispositifs comme le Wi-Fi, le Bluetooth ou d'autres appareils IoT, ont affecté la qualité des signaux captés. De plus, l'émission sporadique des signaux LoRa et l'atténuation due aux obstacles environnementaux, comme les structures des bâtiments, ont rendu leur analyse difficile et partielle.

Ces contraintes combinées ont limité la possibilité d'une analyse claire des signaux LoRa.

LOGICIELS UTILISÉS



# *SAE 31 : METTRE EN OEUVRE UN SYSTEME DE TRANSMISSION*

---

## *Technologies LoRa : Analyse Pratique des Signaux via GNU Radio*

JAVAID Omer  
BELOUKA Djibril

Superviseurs :

- M. Mohamed-Aymen Labiod
- M. Christian Lafont
- M. Gérard Bourdon

RT2A1

*122 Rue Paul Armangot,  
Vitry-Sur-Seine 94440*

# Sommaire

- GLOSSAIRE (Pages 3-4)
- LISTE FIGURES/TABLEAUX (Page 5)
- CHAPITRE 1 : Introduction (Pages 6-7)
- CHAPITRE 2 : Présentation étendue du cahier des charges et argumentaire des choix (Pages 8-13)
- CHAPITRE 3 : Présentation des outils et du matériel utilisés (Pages 14-15)
- CHAPITRE 4 : Présentation du schéma final et des choix techniques (Pages 16-19)
- CHAPITRE 5 : Présentation de la partie expérimentale et des résultats obtenus (Pages 20-25)
- CHAPITRE 6 : Conclusion, retour d'expérience et principaux enseignements (Pages 26-29)
- ANNEXES (Pages 30-31)

## Glossaire :

- **LoRa (Long Range)** : Protocole de communication sans fil conçu pour transmettre des données sur de longues distances avec une faible consommation d'énergie. Utilisé dans les applications IoT (Internet of Things).
- **LoRaWAN (LoRa Wide Area Network)** : Couche réseau qui fonctionne au-dessus du protocole LoRa, permettant de gérer des réseaux IoT à grande échelle avec des fonctionnalités comme la sécurité, la gestion des nœuds, et les mécanismes d'accès au réseau.
- **SDR (Software Defined Radio)** : Technologie permettant de traiter des signaux radio à l'aide de logiciels au lieu de matériel spécialisé.
- **Bande passante** : Plage de fréquences qu'un signal ou un système peut transmettre efficacement.
- **Chirp** : Signal dont la fréquence varie linéairement ou exponentiellement dans le temps, caractéristique des transmissions LoRa.
- **Waterfall (Affichage en cascade)** : Représentation graphique qui montre l'évolution de la puissance spectrale d'un signal en fonction du temps et de la fréquence.
- **Atténuation** : Réduction de l'amplitude d'un signal lorsqu'il traverse un milieu ou une distance.
- **Interférences** : Perturbations dues à d'autres signaux ou sources de bruit qui peuvent dégrader la qualité d'un signal.
- **Démodulation** : Processus consistant à extraire les informations d'un signal modulé, comme dans le cas des trames LoRa.
- **Fréquence porteuse (LO - Local Oscillator)** : Fréquence utilisée pour transposer un signal entre la bande de base et la bande passante.
- **GNU Radio** : Plateforme logicielle utilisée pour développer des applications SDR, notamment la réception et le traitement de signaux radio.
- **PDU (Protocol Data Unit)** : Unité de données utilisée dans les protocoles de communication, transportée à travers la chaîne de traitement.
- **Spreading Factor (SF)** : Paramètre LoRa qui détermine le compromis entre la portée et le débit des communications.
- **Signal-to-Noise Ratio (SNR)** : Rapport entre la puissance du signal utile et celle du bruit, indiquant la qualité de la transmission.

- **Modulation chirp spread spectrum (CSS)** : Technique utilisée par LoRa pour augmenter la portée et la robustesse des communications en étalant le spectre.
- **Table FFT** : Représentation graphique du signal dans le domaine fréquentiel, permettant d'identifier la présence et l'intensité des signaux à différentes fréquences.
- **Machine virtuelle (VM)** : Environnement informatique simulé permettant d'exécuter un système d'exploitation et des applications comme GNU Radio de manière isolée.



## Liste des tableaux :

- **Tableau 1 : Configurations des blocs GNU**

## Liste des figures :

- **Figure 1 : Photographie d'une ADALM Pluto**
- **Figure 2 : Spécifications LoRa régionales**
- **Figure 3 : Schéma initiale GNU Radio**
- **Figure 4 : Carte LORAWAN Orange**
- **Figure 5 : 32 Rue de France, 94440 Vitry-sur-Seine (Google MAPS)**
- **Figure 6 : Test de réception au 32 rue de France**
- **Figure 7 : Capture d'écran de ce qu'on a pu réceptionner**
- **Figure 8 : Schéma Final GNU RADIO**
- **Figure 9 : Localisation d'Orange Prisme (Google)**
- **Figure 10 : Test de réception en face d'Orange Prisme**
- **Figure 11 : WaterFall du signal intercepté**
- **Figure 12 : Court instant de la réception du signal**
- **Figure 13 : Signal LORA typique**
- **Figure 14 : Code Python pour transmettre les données en temps réel**
- **Figure 15 : Script JavaScript pour créer une carte interactive**
- **Figure 16 - Document PDF mentionné précédemment**

# *Chapitre 1 : Introduction*

Les systèmes de communication modernes jouent un rôle essentiel dans la connectivité des objets intelligents, en particulier dans le cadre de l'**Internet des Objets (IoT)**.

Parmi les technologies les plus répandues dans ce domaine, le protocole **LoRa (Long Range)** se distingue par sa capacité à assurer des communications à longue portée, tout en optimisant la consommation énergétique. Cette caractéristique le rend idéal pour des applications telles que la surveillance environnementale, la gestion de réseaux urbains intelligents ou encore les réseaux de capteurs industriels.

Dans le cadre de notre **SAE**, nous avons entrepris d'explorer et d'analyser cette technologie en utilisant un **système de radio logicielle (SDR - Software Defined Radio)**. La radio logicielle offre une flexibilité inégalée en permettant de capter, décoder et analyser des signaux radio grâce à des outils logiciels comme **GNU Radio**, plutôt qu'à des matériels spécialisés.

Le matériel central de notre projet, le **PlutoSDR (ADALM-Pluto)**, nous a permis de capter des signaux LoRa grâce à sa large plage de fréquences (de 70 MHz à 3 GHz) et ses capacités d'échantillonnage performantes. Accompagné d'antennes omnidirectionnelles, il constitue une solution idéale pour les expérimentations dans la bande européenne des 868 MHz, largement utilisée pour les réseaux IoT.



*Figure 1 - Photographie d'une ADALM Pluto*

L'objectif principal de ce projet est de concevoir un système complet permettant :

1. **L'interception des signaux LoRa.**
2. **Leur traitement et leur visualisation sur un site interactif.**
3. **L'analyse des données captées.**





***Chapitre 2 : Présentation  
étendue du cahier des  
charges et argumentaire des  
choix***

L'objectif principal de notre SAE était de concevoir un système capable d'intercepter, analyser et exploiter les signaux LoRa à l'aide d'une radio logicielle (SDR). Ce projet devait répondre aux besoins d'un scénario réel, comme la communication entre bâtiments ou la surveillance IoT.

Pour garantir une approche conforme et efficace, nous avons étudié les normes LoRa en Europe. Ces spécifications définissent les paramètres essentiels pour la transmission et la réception des signaux :

	Europe	North America	China	Korea	Japan	India
Frequency band	867-869MHz	902-928MHz	470-510MHz	920-925MHz	920-925MHz	865-867MHz
Channels	10	64 + 8 + 8	In definition by Technical Committee	In definition by Technical Committee	In definition by Technical Committee	In definition by Technical Committee
Channel BW Up	125/250kHz	125/500kHz				
Channel BW Dn	125kHz	500kHz				
TX Power Up	+14dBm	+20dBm typ (+30dBm allowed)				
TX Power Dn	+14dBm	+27dBm				
SF Up	7-12	7-10				
Data rate	250bps- 50kbps	980bps-21.9kbps				
Link Budget Up	155dB	154dB	In definition by Technical Committee	In definition by Technical Committee	In definition by Technical Committee	In definition by Technical Committee
Link Budget Dn	155dB	157dB				

Figure 2 - Spécifications LoRa régionales

Ces normes ont servi de cadre pour configurer le système en respectant les standards régionaux.

Pour débiter le projet, nous avons élaboré une configuration initiale sous GNU Radio. Ce premier schéma avait pour objectif de valider la compatibilité entre le

**Le schéma, illustré ci-dessous, repose sur des blocs essentiels :**

- **Source PlutoSDR** : Permet de capter les signaux LoRa sur la fréquence centrale de 868.1 MHz.
- **Bloc LoRa Receiver** : Décode les signaux captés en fonction des paramètres configurés (bande passante, facteur d'étalement).
- **Visualisation (QT GUI Sink)** : Offre une vue en temps réel des signaux dans les domaines temporel et fréquentiel.
- **Socket de transmission** : Envoie les données reçues pour une analyse plus approfondie.

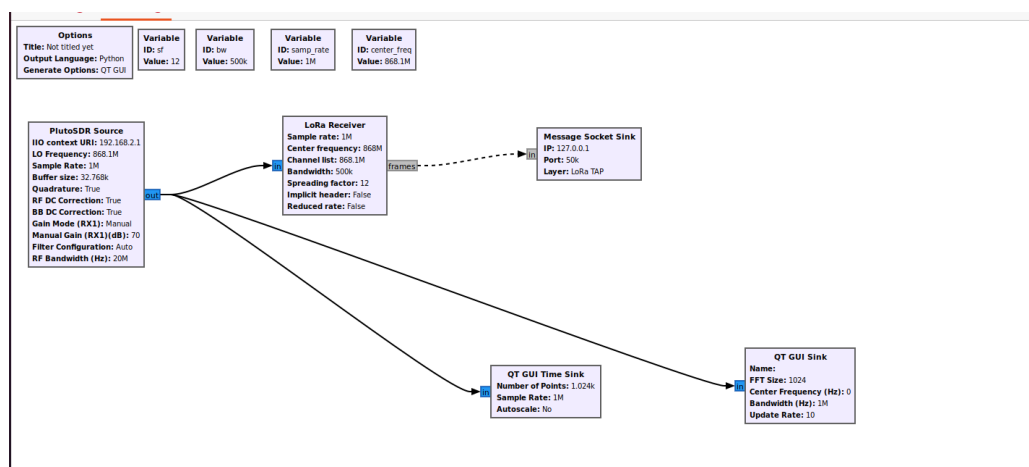


Figure 3 - Schéma initiale GNU Radio

Ce tableau présente les configurations initiales définies :

Paramètre	Valeur	Description
LO Frequency (Hz)	868.1e6	Fréquence locale pour centrer sur la bande LoRa
Sample Rate (Hz)	1e6	Taux d'échantillonnage pour capturer le signal avec précision
Gain RX1 (dB)	70	Gain appliqué au récepteur pour amplifier les signaux
Spreading Factor (SF)	12	Facteur d'étalement pour une meilleure robustesse
Bandwidth (Hz)	500e3	Largeur de bande typique pour LoRa
Center Frequency (Hz)	868e6	Fréquence centrale cible pour le signal
IP Address	127.0.0.1	Adresse locale pour la transmission des données
Port	50k	Port utilisé pour transmettre les données via le socket
Buffer Size (kB)	32.768	Taille du tampon pour le traitement des données
RF Bandwidth (Hz)	20e6	Largeur de bande RF totale gérée par le PlutoSDR
Quadrature	True	Activation des composantes en quadrature pour le traitement des signaux
BB DC Correction	True	Correction des composantes en courant continu (Baseband DC Correction)
RF DC Correction	True	Correction du courant continu RF
Manual Gain RX1 (dB)	70	Réglage manuel du gain pour le canal RX1
Channel List	868.1M	Fréquence des canaux à analyser dans la bande cible
Implicit Header	False	Utilisation ou non des en-têtes implicites pour le décodage LoRa
Reduced Rate	False	Indique si le mode débit réduit est activé

Tableau 1 - Configurations des blocs GNU

**Le projet a été réalisé dans un contexte contraignant, nécessitant de respecter :**

- **Un matériel limité** : Le PlutoSDR, accompagné d'antennes omnidirectionnelles, était le seul matériel autorisé.
- **Des ressources logicielles** : Tous les tests initiaux devaient être effectués via GNU Radio avant l'accès au matériel physique.

- **Un planning restreint** : Avec un total de 50 heures, incluant les tests et la rédaction des livrables.

Bien que le schéma initial conçu sur GNU Radio ait été fonctionnel, nous avons rapidement constaté que l'environnement de l'IUT ne permettait pas de capter efficacement les signaux LoRa.

Cela s'explique principalement par l'absence de capteurs IoT locaux émettant activement dans les bandes LoRa.

Pour mieux comprendre et explorer les capacités de réception de notre système, nous avons consulté la carte de couverture **LoRaWAN d'Orange**, qui indique les zones où ces signaux sont potentiellement accessibles.

Bien que cette carte se base sur le réseau LoRaWAN, il est important de noter que LoRaWAN est un protocole de réseau qui s'appuie sur la modulation LoRa. Par conséquent, les signaux captés dans ces zones correspondent au même concept et partagent les mêmes caractéristiques physiques que les signaux LoRa. Cela nous a donc permis d'utiliser cette carte comme référence pour orienter nos tests.

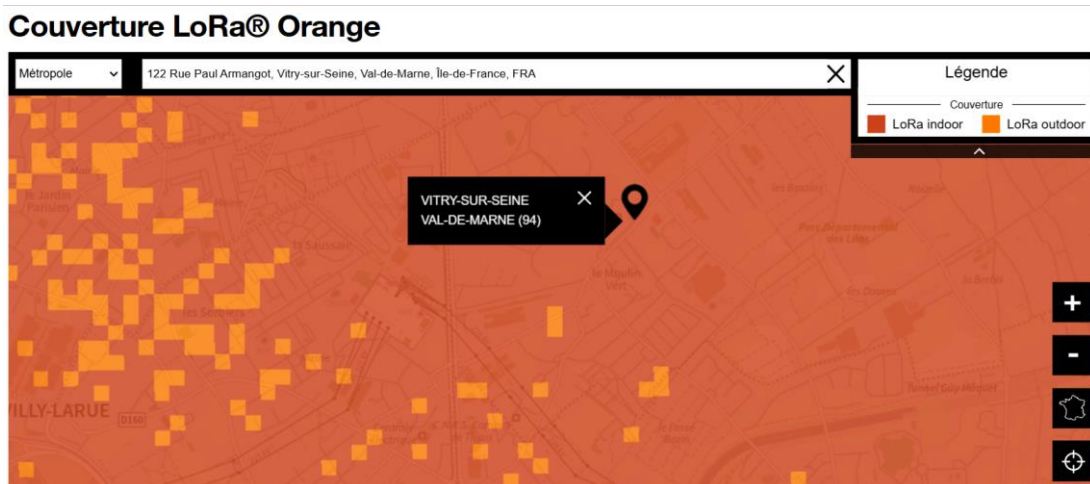


Figure 4 - Carte LORAWAN Orange : <https://www.orange-business.com/fr/reseau-iot>

La carte révèle que l'IUT se trouve dans une zone majoritairement rouge, indiquant une couverture LoRa indoor. Avec le matériel à notre disposition, notamment les antennes omnidirectionnelles du PlutoSDR, il serait difficile de capter des signaux LoRa indoor de manière fiable, car ces signaux sont souvent atténués par les structures des bâtiments.

Nous avons alors identifié la zone LoRa outdoor la plus proche, située au **32 Rue de France 94440 Vitry-sur-Seine**, et nous nous sommes rendus sur place pour effectuer des tests.



Figure 5 - 32 Rue de France, 94440 Vitry-sur-Seine (Google MAPS)

Sur place, nous avons réussi à intercepter des signaux, ce qui laisse penser qu'ils pourraient correspondre à des communications LoRa.



Figure 6 - Test de réception au 32 rue de France

Cependant, ces signaux étaient fortement bruités et manquaient de clarté.

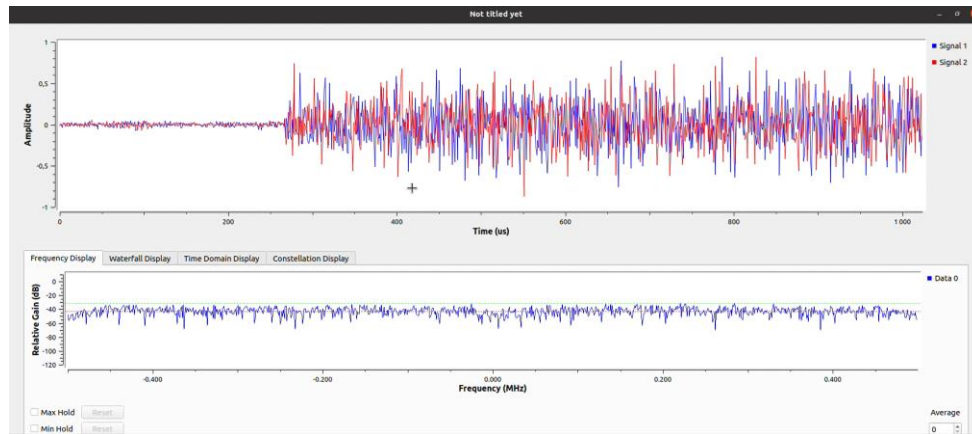


Figure 7 - Capture d'écran de ce qu'on a pu réceptionner

Bien qu'il soit possible que ces signaux soient effectivement du LoRa, il n'y avait aucune garantie de leur origine ni de leur conformité avec les caractéristiques attendues. Ce doute persiste en raison des limitations suivantes :

- **Bruit ambiant élevé** : La zone testée pouvait contenir de multiples sources d'interférences radio.
- **Puissance des émetteurs locaux** : Les capteurs IoT éventuels émettant dans cette zone ne disposaient peut-être pas de la portée ou de la puissance nécessaires pour être clairement captés.
- **Limites matérielles** : Le PlutoSDR, bien que performant, reste dépendant des conditions environnementales et de la qualité des signaux à analyser

Malgré ces incertitudes, le fait d'avoir intercepté des signaux spécifiques dans cette zone confirme que notre configuration matérielle est opérationnelle. Ces observations renforcent l'idée que des ajustements supplémentaires, tels qu'une meilleure isolation contre le bruit ou un affinement des paramètres de réception, pourraient améliorer les performances du système.



## **Chapitre 3 : Présentation des outils et du matériel utilisés durant notre travail**

Pour mener à bien notre projet, nous avons utilisé une gamme de matériels et logiciels adaptés à la réception et à l'analyse de signaux LoRa. Voici une synthèse des outils mobilisés.

### 3.1. Matériel

- **PlutoSDR (ADALM-Pluto)** : Un appareil de radio logicielle compact et polyvalent, doté d'une plage de fréquences de 70 MHz à 3,8 GHz et d'une largeur de bande RF de 20 MHz.
- **Antenne omnidirectionnelle** : Fournie avec le PlutoSDR, elle permet une couverture optimale pour la réception des signaux de différentes directions.
- **Câble USB** : Utilisé pour connecter le PlutoSDR au PC et garantir un transfert de données stable.
- **VM GNU Radio de l'IUT** : Une machine virtuelle préconfigurée avec GNU Radio, essentielle pour le développement et l'exécution de nos schémas.
- **Ordinateur personnel** : Employé pour exécuter la VM GNU Radio et développer un site interactif en HTML, CSS et JavaScript, destiné à visualiser les données capturées.

### 3.2. Logiciels

- **GNU Radio** : La plateforme principale utilisée pour concevoir et exécuter nos schémas SDR.
- **Python** : Utilisé pour le traitement des données issues des signaux capturés.
- **Wireshark** : Employé pour inspecter les trames capturées et mieux comprendre les structures des données reçues.
- **HTML, CSS, JavaScript** : Déployés pour développer un site interactif destiné à visualiser et manipuler les données capturées.

### 3.3. Ressources complémentaires

- **Liens GitHub utiles** :
  - Installation des paquets LoRa pour GNU Radio : <https://github.com/jkadbear/gr-lora>
  - Utilisation de Leaflet pour la création d'une carte interactive : <https://github.com/Leaflet/Leaflet>
- **Autres ressources** :
  - Documentation officielle GNU Radio : <https://wiki.gnuradio.org/>

En résumé, l'ensemble des outils mobilisés a permis de répondre aux exigences du projet tout en fournissant un cadre robuste pour l'analyse et l'expérimentation des signaux LoRa.

## **Chapitre 4 : Présentation du schéma final et des choix techniques**

Pour concevoir notre schéma final dans GNU Radio, nous avons ajusté et enrichi notre configuration initiale. Ces modifications, basées sur les limites observées lors des premiers essais, avaient pour objectif de rendre le système plus performant et de lui ajouter des fonctionnalités essentielles.

Nous avons cherché à maximiser la réception, le traitement et l'interprétation des signaux LoRa tout en offrant des outils supplémentaires pour l'analyse.

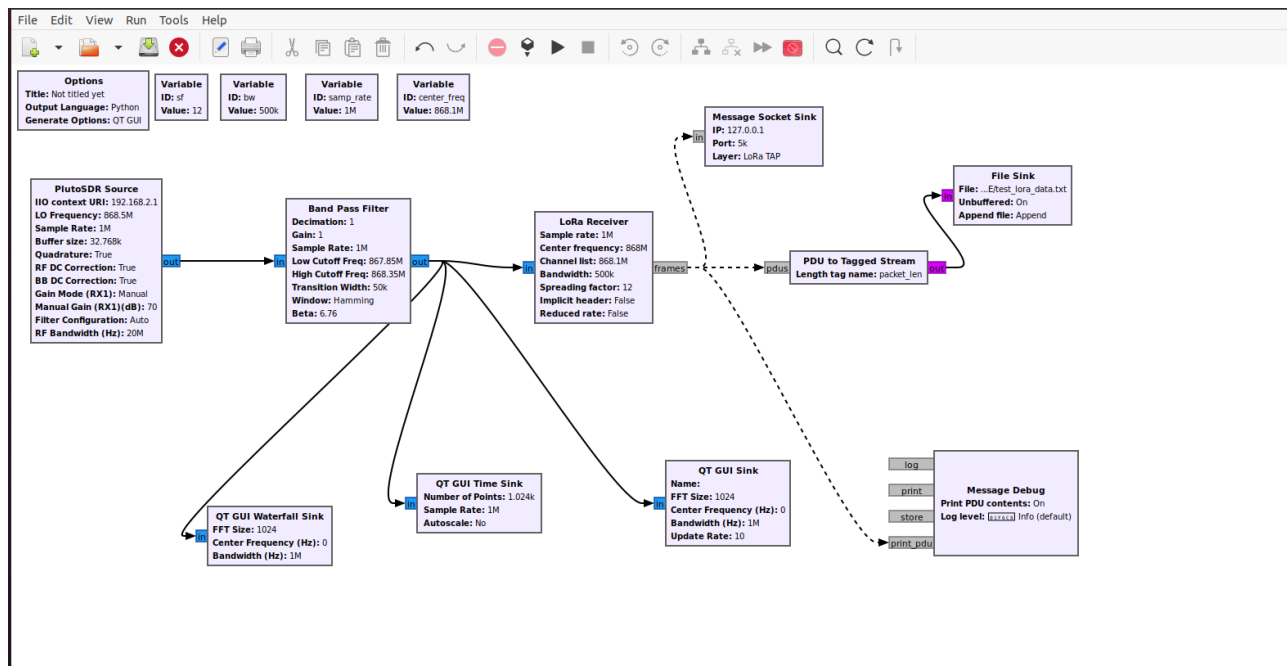


Figure 8 - Schéma Final GNU RADIO

## Description et justification des blocs adoptés

### 1. PlutoSDR Source (bloc inchangé)

- **Rôle :** Captation des signaux RF dans la bande de fréquences définie.
- **Paramètres :**
  - **Fréquence LO :** Après avoir testé plusieurs fréquences LO (868,3 MHz et 868,5 MHz), nous avons conclu que **868,1 MHz** offrait les résultats les plus convaincants
  - **Bande passante RF :** 500 KHz.
  - **Gain :** 70 dB pour amplifier les signaux faibles tout en minimisant le bruit.
- **Pourquoi ?** Ce bloc constitue la base matérielle du système, captant les signaux LoRa dans la bande cible pour les transmettre à la chaîne de traitement. Le réglage précis des paramètres assure une bonne sensibilité tout en limitant les interférences.

### 2. Band Pass Filter (nouveau bloc)

- **Rôle** : Filtrer les fréquences hors bande pour isoler précisément les signaux LoRa.
- **Paramètres** :
  - **Plage de fréquences** : 867,85 kHz à 868,35 kHz.
  - **Transition Width** : 50 kHz (pour une coupure nette).
  - **Fenêtre** : Hamming.
- **Pourquoi ?** Dans le schéma initial, les signaux reçus incluaient beaucoup de bruit provenant des bandes adjacentes. Ce filtre a été ajouté pour cibler uniquement la bande LoRa, garantissant une réception plus propre et précise.

### 3. LoRa Receiver (bloc déjà présent)

- **Rôle** : Décodage des signaux LoRa en trames exploitables.
- **Paramètres** :
  - **Fréquence centrale** : 868 MHz.
  - **Facteur d'étalement (SF)** : 12.
  - **Largeur de bande** : 500 kHz.
- **Pourquoi ?** Ce bloc est essentiel pour interpréter les signaux captés. Les paramètres choisis respectent les normes européennes et garantissent une robustesse accrue face aux interférences.

### 4. QT GUI Time Sink (bloc inchangé)

- **Rôle** : Visualisation des signaux dans le domaine temporel.
- **Pourquoi ?** Permet de suivre en direct l'évolution des signaux reçus pour vérifier leur activité et leur stabilité.

### 5. QT GUI Waterfall Sink (nouveau bloc)

- **Rôle** : Fournir une vue en cascade des signaux dans le domaine fréquentiel.
- **Pourquoi ?** Ce bloc complète le Time Sink en offrant une visualisation spectrale, essentielle pour identifier les signaux LoRa dans la bande de fréquences ciblée.

### 6. Message Socket Sink (bloc déjà présent)

- **Rôle** : Transmettre les trames décodées vers une application externe via un réseau local.
- **Pourquoi ?** Ce bloc facilite l'analyse avancée des données, notamment pour des applications telles que la géolocalisation ou le traitement statistique en temps réel.

### 7. File Sink (nouveau bloc)

- **Rôle** : Enregistrement des données captées dans un fichier.
- **Pourquoi ?** L'ajout de ce bloc permet de conserver une trace des signaux interceptés pour une analyse postérieure. Les données archivées peuvent être utilisées pour tester des algorithmes ou valider les résultats.

### 8. PDU to Tagged Stream (nouveau bloc)

- **Rôle :** Convertir les PDU en flux de données exploitables par d'autres blocs.
- **Pourquoi ?** Ce bloc est indispensable pour rendre les trames LoRa compatibles avec le File Sink et d'autres blocs. Il agit comme un pont entre les données brutes décodées et leur traitement final.

### 9. Message Debug (nouveau bloc)

- **Rôle :** Affichage en temps réel des trames décodées.
- **Pourquoi ?** Ce bloc sert principalement au débogage. Il permet de vérifier rapidement si des données valides sont interceptées sans devoir les enregistrer.

### Comparaison avec le premier schéma

Dans le schéma initial, les blocs étaient limités aux éléments fondamentaux pour capter et décoder un signal LoRa.

Si cette configuration suffisait pour des tests préliminaires, elle manquait de robustesse et de flexibilité pour des analyses approfondies.

Voici les principaux ajouts et leur impact :

1. **Amélioration de la qualité des signaux :**
  - Grâce au Band Pass Filter, les interférences hors bande ont été largement réduites, offrant une réception plus précise.
2. **Enrichissement des capacités d'analyse :**
  - L'ajout des blocs File Sink, PDU to Tagged Stream et Message Debug permet d'archiver et de visualiser les données sous différents formats, augmentant la modularité du système.
3. **Visualisation complète :**
  - Le Waterfall Sink a ajouté une perspective spectrale, essentielle pour valider visuellement la présence des signaux LoRa dans la bande ciblée.
4. **Transmission des données :**
  - Le Message Socket Sink facilite l'interfaçage avec des outils externes, rendant le système plus dynamique et évolutif.

### Conclusion

Le schéma final illustre une transition réfléchie et méthodique depuis la version initiale. Chaque bloc ajouté ou modifié répond à un besoin clairement identifié, qu'il s'agisse de réduire les interférences, de fournir des outils d'analyse ou de faciliter la transmission des données. Ce schéma constitue une plateforme solide pour capter, décoder et analyser des signaux LoRa, tout en restant flexible pour des améliorations futures.



# **Chapitre 5 : Présentation de la partie expérimentale et des résultats obtenus**

Après plusieurs tests, nous avons décidé de nous déplacer dans une zone avec une forte densité supposée de signaux LoRa, en utilisant les outils et méthodologies développés au cours des phases précédentes. Ce chapitre détaille nos observations, nos hypothèses et les résultats obtenus.

### Déplacement vers une zone mieux couverte

D'après la carte de couverture LoRaWAN d'Orange, la plus forte activité de signaux IoT dans notre zone était supposée à **132 Av. de Stalingrad, 94800 Villejuif**, ce qui correspond à la localisation d'Orange Prisme.



Figure 9 - Localisation d'Orange Prisme (Google)

Cette localisation a été choisie pour maximiser nos chances de capter des signaux LoRa authentiques, compte tenu des limitations rencontrées à l'IUT.



Figure 10 - Test de réception en face d'Orange Prisme

Une fois sur place, nous avons installé notre matériel et configuré le schéma GNU Radio en fonction des spécifications LoRa européennes :

- **Fréquence LO** : 868 100 000 Hz

- Bande passante ciblée : 500 000 Hz
- Gain : 70 dB

## Observations et analyses

### Résultats temporels et fréquentiels

Lors des tests, nous avons obtenu les résultats suivants :

#### 1. Waterfall Display :

- Le waterfall montre une activité rougeâtre qui pourrait indiquer la présence d'un signal LoRa ou d'autres signaux RF proches.
- Cependant, cette activité n'est pas structurée de manière claire, ce qui rend difficile la confirmation d'un signal LoRa authentique.
- Une forte composante de bruit est également visible, probablement causée par des interférences RF dans la zone (Wi-Fi, Bluetooth, ou autres émetteurs IoT).

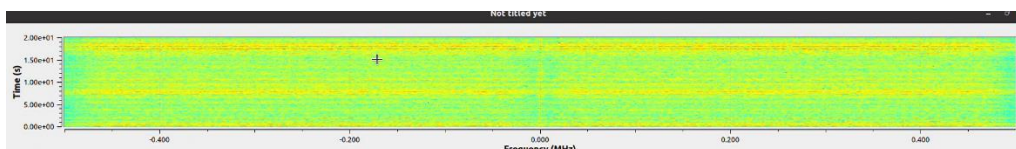


Figure 11 - WaterFall du signal intercepté

#### 2. Graphique FFT :

- Un pic visible autour de la fréquence centrale cible (868 MHz) a été observé. Ce pic pourrait correspondre à un signal LoRa, mais son analyse est limitée par le bruit ambiant et les possibles interférences.

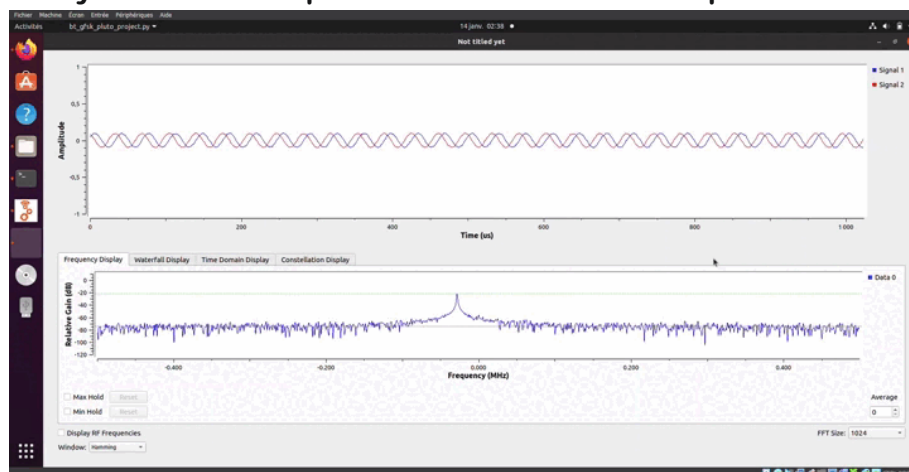


Figure 12 - GIF du Court instant de la réception du signal, (Pour consulter l'animation complète, veuillez accéder au fichier GIF via ce lien : <https://ibb.co/k9Gkbbm>).

#### 3. Domaine temporel (Time Sink) :

- Les signaux captés affichent des formes sinusoïdales, mais ces résultats ne correspondent pas parfaitement aux chirps caractéristiques des signaux LoRa (upchirp et downchirp). Cela soulève des doutes sur l'origine exacte du signal capté.

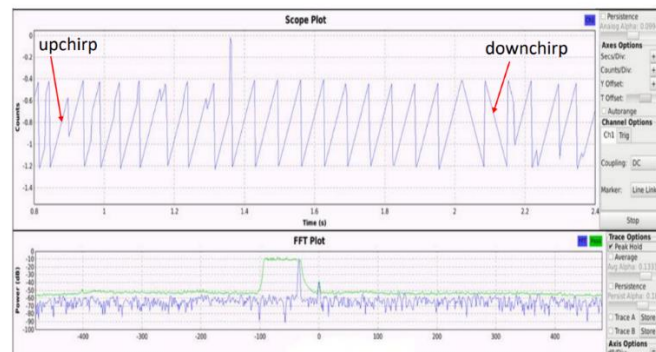


Figure 13 - Signal LORA typique

## Essais avec le site interactif

En parallèle, un outil interactif basé sur Python et Flask a été développé pour visualiser les trames reçues et géolocaliser leur origine grâce à une carte générée avec Leaflet.js. Le code Python, qui semblait fonctionnel, lançait un serveur local pour afficher les données captées, cependant, aucune trame n'a pu être affichée sur la page web, malgré le bon fonctionnement apparent du code et de la configuration du serveur.

```

1 from flask import Flask, render_template, jsonify
2 import socket
3 import threading
4
5 app = Flask(__name__)
6
7 # Stockage temporaire des données captées
8 data_store = []
9
10 # Fonction pour écouter les trames UDP
11 def udp_listener():
12     udp_ip = "127.0.0.1"
13     udp_port = 5000 # Nouveau port
14
15     sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
16     sock.bind((udp_ip, udp_port))
17     print(f"Écoute des trames sur {udp_ip}:{udp_port}")
18     while True:
19         try:
20             data, addr = sock.recvfrom(1024)
21             decoded_data = data.decode("utf-8")
22             data_store.append({'data': decoded_data, 'source': addr})
23             print(f"Reçu : {decoded_data} de {addr}")
24         except Exception as e:
25             print(f"Erreur UDP : {e}")
26
27 # Lancer l'écoute des trames dans un thread
28 thread = threading.Thread(target=udp_listener, daemon=True)
29 thread.start()
30
31 @app.route('/')
32 def index():
33     return render_template('index.html')
34
35 @app.route('/data', methods=['GET'])
36 def get_data():
37     return jsonify(data_store)
38
39 if __name__ == '__main__':
40     app.run(debug=True)

```

Figure 14 - Code Python pour transmettre les données en temps réel

Également le script JavaScript qui était censé géolocaliser précisément tout les signaux Lora que nous aurions du intercepter :

```
// Initialiser le graphique
const ctx = document.getElementById('signalChart').getContext('2d');
const signalChart = new Chart(ctx, {
  type: 'line',
  data: {
    labels: [],
    datasets: [{
      label: 'Amplitude du Signal',
      data: [],
      borderColor: 'rgba(75, 192, 192, 1)',
      borderWidth: 2,
    }]
  },
  options: {
    responsive: true,
    scales: {
      x: { title: { display: true, text: 'Temps (s)' } },
      y: { title: { display: true, text: 'Amplitude' } }
    }
  }
});

// Mettre à jour les données
async function fetchData() {
  const response = await fetch('/data');
  const data = await response.json();

  // Mettre à jour le graphique
  const timestamps = data.map((entry, index) => index);
  const values = data.map(entry => parseFloat(entry.data) || 0);

  signalChart.data.labels = timestamps;
  signalChart.data.datasets[0].data = values;
  signalChart.update();
}

setInterval(fetchData, 1000); // Récupérer les données toutes les secondes
});
```

Figure 15 - Script JavaScript, pour créer une carte interactive avec la géolocalisation des signaux lora en temps réel

## Tests et ajustements

Pour confirmer nos résultats, plusieurs configurations ont été testées :

- **Fréquence LO** : Nous avons testé plusieurs fréquences LO autour de 868 MHz (868 000 000 Hz, 868 100 000 Hz, 868 300 000 Hz), sans obtenir de changements significatifs dans les signaux captés.
- **Gain** : Différents niveaux de gain ont été essayés (50 dB, 60 dB et 70 dB). Le gain de 70 dB semblait optimal pour capter des signaux sans saturer le récepteur.
- **Filtre passe-bande** : Configuré avec une coupure basse de 867,850MHz et une coupure haute de 868,350MHz, il a été ajusté pour correspondre aux spécifications européennes. Cependant, même avec ces réglages, les résultats obtenus restent ambigus.

## Hypothèses sur les difficultés rencontrées

### 1. Bruit ambiant et interférences :

- La zone testée est dense en dispositifs RF, tels que Wi-Fi et Bluetooth, ce qui pourrait provoquer une saturation du spectre et masquer les signaux LoRa.

### 2. Atténuation des signaux :

- Les signaux LoRa indoor émis depuis les bâtiments environnants pourraient être fortement atténués avant d'atteindre notre récepteur.

### 3. Mauvaise configuration possible :

- Bien que nous ayons suivi les recommandations pour la bande passante et la fréquence, des paramètres pourraient encore nécessiter des ajustements pour améliorer la détection.

#### 4. Émission sporadique de signaux LoRa :

- Les signaux LoRa sont souvent émis de manière intermittente. Il est possible que nous ayons capté un signal, mais qu'il ait été trop bref ou masqué par le bruit pour être correctement analysé.

Également lors de l'exécution de GNU Radio sur la machine virtuelle (VM), des caractères "g" apparaissaient de manière répétée dans le terminal (CMD). Bien que la cause exacte de ce comportement ne puisse être confirmée, il est possible que l'utilisation de la VM ait introduit certaines limitations, notamment au niveau des ressources allouées (CPU, mémoire), ce qui aurait pu perturber le traitement des signaux en temps réel.

#### Comparaison avec des signaux LoRa typiques

En théorie, un signal LoRa est caractérisé par des chirps bien définis (upchirp et downchirp).

En revanche, nos résultats montrent des signaux plus proches de formes sinusoïdales, avec beaucoup de bruit, ce qui les rend difficilement exploitables.

#### Conclusion

Les efforts déployés montrent qu'il est possible d'intercepter un signal dans la bande LoRa, mais que les conditions matérielles et environnementales jouent un rôle critique dans la qualité et la précision des données captées. Les limites rencontrées soulignent la complexité des transmissions sans fil en environnement urbain et l'importance d'une analyse approfondie des paramètres.



## Chapitre 6 : Conclusion et retour d'expérience

## Une expérience riche en apprentissages techniques et théoriques

La réalisation de cette SAÉ nous a offert une opportunité unique de mettre en pratique presque l'intégralité des notions abordées dans le cours R3.05. En développant un système SDR (Software Defined Radio) dédié à la réception et au traitement des signaux LoRa, nous avons pu approfondir notre compréhension des chaînes de transmission numériques tout en confrontant nos connaissances théoriques à la réalité expérimentale. Cette expérience a démontré l'exigence de précision extrême dans chaque étape, qu'il s'agisse des paramètres matériels, des configurations logicielles ou de l'analyse des résultats.

## Bilan des notions abordées

### 1. La chaîne de transmission numérique : une application complète :

- **Modulation et démodulation** : Nous avons pu illustrer concrètement le rôle de la modulation en bande transposée, où un signal en bande de base est transmis autour d'une fréquence porteuse (868 100 000 Hz dans notre cas). La démodulation, réalisée via le bloc LoRa Receiver, a mis en lumière la complexité de la synchronisation et de l'extraction des données dans un environnement perturbé.
- **Codage et correction d'erreurs** : Bien que nous n'ayons pas implémenté directement le codage canal, nous avons étudié les défis liés au mapping et à la robustesse des données face aux erreurs induites par le bruit.

### 2. Les perturbations et leur impact :

- Le bruit, sous ses diverses formes (blanc, impulsionnel), ainsi que les interférences dues aux appareils Bluetooth, Wi-Fi et autres sources proches, ont illustré à quel point les signaux faibles comme ceux de LoRa nécessitent une précision et une optimisation sans faille. Ces perturbations soulignent l'importance de la qualité du signal (rapport signal/bruit) et de l'atténuation dans la chaîne.

### 3. Le filtrage et la largeur de bande :

- L'ajout d'un filtre passe-bande (867 850 Hz - 868 350 Hz) a permis de réduire les interférences hors bande, bien que la précision requise dans la configuration de ces paramètres ait montré ses limites. Une erreur minime dans la largeur de bande peut rendre un signal LoRa inexploitable.

### 4. Visualisation et exploitation des données :

- Les blocs QT GUI ont fourni des visualisations utiles, mais ont également souligné l'écart entre la théorie et la pratique. Si nos signaux présentaient des caractéristiques sinusoidales proches de LoRa, les résultats n'étaient pas conformes à ce que nous attendions en théorie (chirps up/down). Cela a mis en évidence la difficulté d'obtenir des signaux clairs et exploitables dans un environnement non contrôlé.

## Retour d'expérience

Notre expérience a révélé à quel point la réception et le traitement des signaux LoRa nécessitent une précision et une synchronisation exceptionnelles. Malgré des ajustements successifs des paramètres (fréquence LO, gains, et filtrage), nous avons rencontré des difficultés majeures :

- L'atténuation des signaux LoRa en milieu urbain, accentuée par la densité d'interférences.
- La configuration de certains blocs qui aurait pu être optimisée pour mieux répondre aux spécificités des signaux LoRa.
- La nature éphémère et mobile des émetteurs LoRa, qui complexifie la capture des données en conditions réelles.

L'absence de trames visibles sur notre site interactif a également été une déception. Cela pourrait être dû à un signal trop faible, une mauvaise synchronisation ou encore une durée de réception trop courte pour capturer un signal exploitable. Cette difficulté nous a permis de mieux comprendre les exigences des couches physiques et applicatives dans un système de télécommunication.

### Enseignements et perspectives

- **Synthèse des apprentissages techniques** : Cette SAÉ nous a permis d'explorer de manière approfondie les éléments clés d'une chaîne de transmission numérique : de la modulation à la démodulation, en passant par le filtrage, l'analyse spectrale et la visualisation. Ces expériences pratiques ont solidifié nos acquis théoriques tout en nous confrontant à des défis concrets.
- **Maîtrise des outils** : GNU Radio, associé au matériel SDR PlutoSDR, nous a offert une plateforme flexible mais exigeante. Cette combinaison nous a permis d'expérimenter avec des signaux réels tout en identifiant les limitations et les opportunités qu'offrent ces outils.
- **Limites et pistes d'amélioration** : La réception des signaux LoRa a mis en évidence l'importance d'une configuration précise et d'un environnement optimisé. Pour de futures expérimentations, il serait pertinent d'approfondir l'analyse des diagrammes de l'œil, d'implémenter des mécanismes avancés de correction d'erreurs et de perfectionner les algorithmes de décodage. Par ailleurs, réaliser ces tests dans un environnement maîtrisé avec des émetteurs LoRa spécifiques permettrait de réduire considérablement les incertitudes liées aux interférences et au bruit ambiant.

En conclusion, cette SAÉ a représenté une véritable immersion dans le domaine des transmissions numériques, mettant en pratique quasiment toutes les notions vues en R3.05. Malgré les obstacles rencontrés, cette expérience nous a permis de développer une compréhension profonde des systèmes de télécommunication modernes et des compétences essentielles pour des projets plus complexes à l'avenir.

Nous souhaitons exprimer nos sincères remerciements aux enseignants **Christian Lafont, Mohamed Aymen Labiod, et Gérard Bourdon** pour leurs enseignements riches et précieux en transmissions, tant dans le cadre du module **R305** que pour leur encadrement tout au long de cette **SAÉ**. Leur expertise et leur disponibilité ont été des atouts majeurs dans la compréhension et l'application des concepts abordés.

# Annexes

## 1. Ressources YouTube

Ces vidéos ont été consultées pour approfondir la compréhension de GNU Radio, des signaux LoRa et des aspects techniques nécessaires à notre projet :

- **Présentation générale de GNU Radio :**  
[https://www.youtube.com/watch?v=vBES6Baxc0M&ab\\_channel=GNURadio](https://www.youtube.com/watch?v=vBES6Baxc0M&ab_channel=GNURadio)  
Cette vidéo présente une introduction complète à GNU Radio, son interface graphique, et ses applications.
- **Utilisation d'un SDR pour l'analyse de signaux :**  
[https://www.youtube.com/watch?v=c5aKZ1wrZNs&ab\\_channel=JeremyClark](https://www.youtube.com/watch?v=c5aKZ1wrZNs&ab_channel=JeremyClark)  
Cette vidéo explique comment utiliser un SDR pour capter et analyser différents types de signaux.
- **Introduction à LoRa et son fonctionnement :**  
<https://www.youtube.com/watch?v=4CzZI7yqxlQ>

## 2. Ressources GitHub

Ce dépôt GitHub a servi de références importantes pour la mise en œuvre des modules GNU Radio spécifiques aux signaux LoRa :

- gr-lora par Matt Knight (Module LoRa pour GNU Radio) :  
<https://github.com/MattKN/gr-lora>

## 3. Articles et sites web

- **Présentation générale de LoRa et LoRaWAN :**  
<https://www.hacf.fr/lora-presentation/>  
Cet article explique en détail les bases de LoRa et ses avantages pour les communications IoT.

## 4. Ressource académique

Un document intitulé *"Modeling and simulation of radio signals in confined environments for indoor tracking using a software-defined radio"* par **ZEAITER Hussein**, chercheur au laboratoire FEMTO-ST, Université de Franche-Comté (Montbéliard, France), a été une ressource clé.

Bien que ce PDF ne puisse être inclus directement, il a fourni des informations précieuses sur :

- Les techniques de modulation LoRa (chirp spread spectrum - CSS).
- Les paramètres LoRa comme le Spreading Factor et la bande passante.
- L'utilisation de radios logicielles comme le PlutoSDR pour l'analyse des signaux LoRa.



## Modeling and simulation of radio signals in confined environments for indoor tracking using a software defined radio

ZEAITER Hussein

Ph.D Researcher

Laboratoire FEMTO-ST

UMR CNRS

Université de Franche-Comté

Montbéliard France

*Figure 16 - Document PDF mentionné précédemment*

### **5. Documentation technique**

- Documentation officielle de GNU Radio :  
<https://wiki.gnuradio.org/>
- Carte de couverture LoRaWAN d'Orange :  
<https://www.orange-business.com/fr/reseau-iot>