Spécifications Technique : Interface de Communication XBee & RTK

1. Introduction

Cette interface graphique est une application en Python utilisant Tkinter. Elle facilite la communication entre deux modules XBee tout en communiquant avec un serveur RTK, permettant l'affichage et l'envoi de données tout en relayant les corrections RTK.

2. Objectifs

- Fournir une interface graphique configurable permettant d'envoyer et de recevoir des données en XBee.
- Assurer la connexion et la gestion des corrections RTK via un serveur NTRIP.
- Automatiser l'affichage et la mise à jour des données reçues.

3. Fonctionnalités

3.1. Connexion XBee

- Configuration du port série pour la communication avec XBee.
- Envoi de données sous forme de paires clé-valeur.
- Réception et affichage des données en temps réel.

3.2. Connexion au Serveur RTK

- Initialisation de la connexion au serveur NTRIP.
- Récupération des corrections RTK.
- Transmission des corrections RTK vers le module XBee.

3.3. Interface Graphique

- Chargement de la configuration depuis un fichier YAML.
- Génération dynamique des champs d'entrée et de sortie.
- Mise à jour automatique des valeurs affichées.

3.3.1 Configuration YAML

Structure générale

- **Config** : Racine de la configuration, contenant le titre et les paramètres.
  - **Title** : Nom ou identifiant de la configuration.
  - **Containers** : Liste d'éléments contenant des sous-paramètres.

C'est une liste de groupes (ou modules), chacun ayant un **nom** et des **entrées/sorties** spécifiques.

- **Name** : Identifie chaque module.
- **outputs** : Définit les valeurs de sortie des modules (peut inclure des paramètres comme des coefficients ou des commandes).
- **inputs** : Définit les valeurs d'entrée attendues (elles peuvent être initialement définies à null).

Exemple :

```
Config:
  Title: config_interface

  Containers:
    - Name: correcteur
      outputs:
        KP: 0.1
        KI: 0.1
        KD: 0.1

    - Name: control
      outputs:
        CAP: 0

    - Name: received_value
      inputs:
        cap: null
        orientation: null
        servo: null
        current: null
        target: null
```

4. Installation

4.1. Prérequis

- Python 3
- Modules Python requis

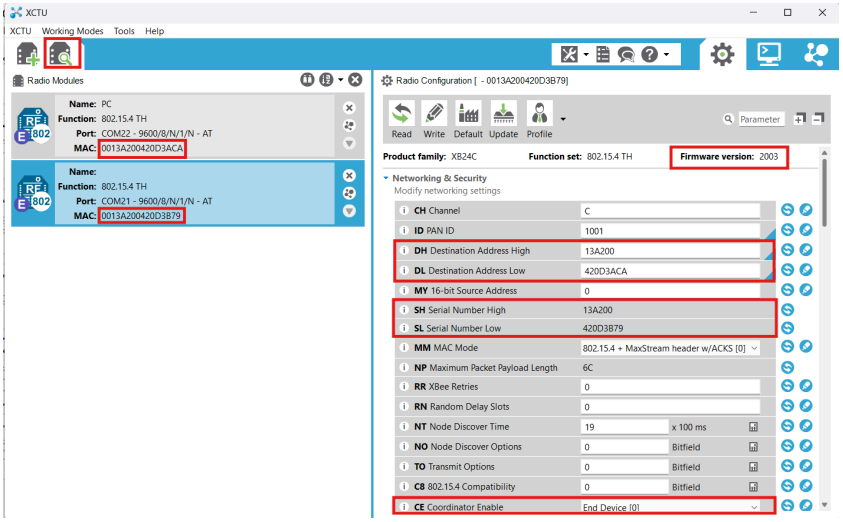
```
pip install pyserial pyyaml pyrtcm
```

4.2 Configuration antenne XBee

Lien pour télécharger le logiciel de configuration des XBee.

- Mettre un module XBee sur un adaptateur et le brancher directement au PC.
- Sur XCTU, appuyer sur **Ctrl+Shift+D**, sélectionner les ports COM et conserver les paramètres par défaut.

Ressource pour la configuration : modifier les valeurs de destination pour les faire correspondre entre les deux XBee.



Code de test à mettre sur une Raspberry Pi Pico avec un module XBee :

- Compiler et tester en vérifiant la réception des données envoyées sur le port série de l'autre XBee via un autre port COM.

4.3 Configuration RTK

Pour le RTk, il n'y as rien a configurer, il faut juste que le pc qui lance l'affichage soit connecté à un réseau qui ne bloque pas l'accès au serveur de correction (exemple : eduroam)

5. Utilisation

Lancer le programme avec les paramètres suivants :

```
python3 interface.py port=/dev/ttyUSB0 baud_rate=9600 path=interface_config/interface.yaml
```

5.1. Arguments Disponibles

Argument	Description	Valeur par défaut
`port`	Port série du module XBee	`/dev/ttyUSB0`
`baud_rate`	Baud rate de la communication XBee	`9600`
`path`	Fichier YAML de configuration	`interface_config/interface.yaml`

6. Contraintes et Limitations

- Compatible uniquement avec les modules **XBee** supportant la communication série.
- La connexion au serveur **NTRIP** nécessite un accès Internet.
- La configuration YAML doit respecter la structure prédéfinie.