

Radio FM, ADS-B et IMSI-catcher

Objectifs : Comprendre et manipuler la radio logicielle.

Ce TP est à rendre avec les consignes de rendu suivantes :

- Chaque TP fait l'objet d'un rendu par binôme ou monôme.
- Le TP doit être rendu sous la forme d'un seul fichier pdf, plus les GRCs.
- L'archive est à rendre dans les 10 jours après le dernier TP.

► Exercice 1 : Récepteur FM

- Brancher le SDR USRP et charger son firmware en utilisant la commande suivante : `uhd_usrp_probe`
 - Quel est le type de la carte (Detected Device) ?
 - Quelle est la bande de fréquence supportée par la carte ?
- Placer une antenne sur la sortie TX/RX.
- Lancer le flow graph **Récepteur-FM** (inclus dans l'archive fournie avec le TP) en utilisant l'outil GNURadio Companion (GRC).
- Exécuter le flow graph, et trouver une chaîne radio FM de bonne qualité.
 - Quelle est la fréquence de cette chaîne ?
 - Quelle est la largeur d'un canal radio FM ?
- Augmenter le gain `rx_gain` :
 - Commenter l'impact de ce paramètre sur la qualité du signal reçu test. Expliquer.
- Examiner le flow graph :
 - Expliquer brièvement le rôle de chaque bloc de flow graph.
 - Quelle est la fréquence d'échantillonnage utilisée par le récepteur USRP ?
 - Expliquer la relation entre cette fréquence et la bande passante d'un canal radio FM.
 - Quelle est la fréquence d'échantillonnage utilisée par la carte son ?
 - Expliquer comment l'interpolation et la décimation concilient les deux fréquences d'échantillonnage.
 - Donner la formule utilisée par le bloc *Rational sampler*.

► Exercice 2 : Émetteur radio FM

- Lancer le flow graph **Émetteur-FM** (inclus dans l'archive fournie avec le TP) en utilisant l'outil GNURadio Companion (GRC).
- Télécharger un fichier audio de format WAV.
 - Quelle est sa fréquence d'échantillonnage ?
- Si sa fréquence d'échantillonnage est proche de 48KHz , importer le dans le flow graph via bloc **Wav File Source**.
- Exécuter le flow graph :
 - Quelle est la fréquence porteuse utilisée par l'émetteur ?
- Pour éviter les interférences, chaque groupe doit changer sa fréquence porteuse après concertation.

6. Utiliser l'application Radio FM d'un smartphone pour écouter la fréquence de l'émetteur.
 - (a) Commenter le résultat.
 - (b) Quelle est la portée radio de l'émetteur ?
7. Désactiver le bloc **WAV File Source**, et activer le bloc **Wav File Source** pour permettre l'utilisation d'un microphone.
8. Activer le micro sur votre machine en utilisant l'outil **alsamixer**, et vérifier l'enregistrement d'un fichier son par les commandes *arecord/aplay*.
9. Examiner le fonctionnement via l'émetteur FM.

► Exercice 3 : Générateur de Bruit

1. Lancer via un terminal la commande *uhd_signgen_gui* avec comme option une fréquence d'une chaîne radio FM de bonne qualité. Cette commande initialise un générateur de signal développé par Ettus Research.
2. Sélectionner un bruit Gaussien comme signal et faire varier le paramètre **TX Gain**.
 - (a) Qu'est-ce qu'un bruit Gaussien ?
 - (b) Commenter l'impact sur la chaîne radio cible.
3. Visualiser les différents réseaux WiFi détectés par votre smartphones en utilisant l'application **WiFi Analyzer**.
 - (a) Attaquer l'une des fréquences porteuses par bruit et commenter le résultat ?
4. Réaliser votre propre générateur de bruit en développant un flow graph qui fait varier le gain d'une antenne et la fréquence échantillonnage.
 - (a) Expliquer son fonctionnement et tester le résultat sur les fréquences radio FM et WiFi. **PS : il faut joindre le fichier dans le rapport.**

► Exercice 4 : Projet 1 : Systeme ADS-B avions

L'Automatic dependent surveillance-broadcast (ADS-B) est un nouveau système de surveillance coopératif pour le contrôle du trafic aérien et d'autres applications connexes. Un avion équipé de l'ADS-B détermine sa position par un système de positionnement par satellite (GNSS), et envoie périodiquement cette position et d'autres informations aux stations sols et aux autres appareils voisins équipés de l'ADS-B.

L'**objectif de ce projet** est d'intercepter les signaux (positions, vitesse, direction, ...) envoyés par les avions aux stations sol en temps réel et lire le résultat par **Google Earth**.

Avant de commencer le projet, installer les dépendances suivantes :

- apt-get install sqlite3 libsqlite3-dev
- apt-get install python-numpy python-scipy python-matplotlib ipython ipython-notebook python-pandas python-sympy python-nose python-zmq
- apt-get install libzmq-dev
- apt-get install pyqt4-dev-tools
- apt-get install gr-osmosdr

► Exercice 5 : Projet 2 : IMSI-catcher

L'IMSI-catcher est un appareil qui permet de créer une fausse antenne-relais, en s'intercalant entre le réseau de l'opérateur de téléphonie et l'appareil surveillé. Une fois déployé, il force les dispositifs situés à proximité à passer par lui plutôt que par la véritable antenne. Il reçoit alors les communications de ces téléphones, avant de les transmettre à l'antenne de l'opérateur, de manière invisible. C'est ce qu'on appelle la classique attaque de *l'homme du milieu*

- Monter et configurer un opérateur téléphonique en utilisant l'outil OpenBTS.
- Récupérer l'IMSI des téléphones connectés.