****

**I**nstitut **S**upérieur d’**I**nformatique

**M**odélisation et de leurs **A**pplications

1, Rue de la Chebarde

TSA 60125

CS 60026

63178 Aubière Cedex

France

*Rapport de Projet*

*3ème année d’élève ingénieur*

*Filière 5 – Réseaux et Sécurité Informatique*

2015/2016

Création d’une radio pirate à l’aide de la radio logicielle



Présenté par : HOANG Jimmy

JOUGLA Benoit

Tuteur de projet : TILMANT Christophe

Enseignant Référent : CHEMINAT Michel

# Remerciements

Nous remercions Mr Christophe Tilmant, pour sa disponibilité et de son aide durant ce projet, ainsi que toutes les personnes qui nous ont aidé.

[Figure 1. Radio analogique 2](#_Toc444674484)

# Résumé

Nous sommes en permanence encerclés par les radios fréquences. Aussi bien la radio FM, les services météos, les téléphones portables ou plus récemment la démultiplication des objets connectés, Bon nombre des équipements utilisent les radios fréquences pour s’échanger des informations. La Radio Logicielle ou encore Software Defined Radio (SDR) est une manière de manipuler ces ondes à l’aide de logiciels adaptés. L’objectif de ce projet est donc de représenté la filière Réseaux et Sécurité Informatique lors des journées portes ouverte de l’ISIMA en montrant les façons et les menaces qui pèsent autour des ondes radios

*Given the infinite number of applications that can be achieve with the software radio we decided to focus on the development of a pirate radio. With this, we can broadcast the music and messages through ISIMA. We used a free software development kit that provides signal processing tools to implement SDR and signal processing systems: GNU Radio, and a SDR peripheral able to transmit and receive radio: HackRF One. For the transmission of the FM signal, we use the different blocks provided by GNU Radio, but for send a message we need to create our block. The FM radio, use a communications protocol standard for embedding small amounts of digital information, Radio Data System (RDS). RDS standardizes several types of information transmitted, as radio name and radio text.*

Un grand nombre de possibilités se sont ouverts à nous concernant l’utilisation de la radio logicielle nous nous sommes portés sur le développement d’une radio pirate, une attaque de type Replay et enfin l’écoute du traffic GSM. Pour cela nous avons utilisés le logiciel de traitement de signaux GNU Radio et une carte de contrôle capable d’emettre et de transmettre des ondes radios : La HackRF One.

# Abstract

**Every day, we are surrounded by radio frequencies. Through the FM radio, the weather service, cell phones and more recently in the connected objects, many of this objects use the radio frequencies to exchange data. Software Defined Radio (SDR) is a way to handle radio frequencies with software. In the context of ISIMA's open day, to represent our options, F5 Networks and IT Security, we need to demonstrate usage and threat about radio frquencies. This project aims to highlight the F5 options to ISIMA's open day.**

*Given the infinite number of applications that can be achieve with the software radio we decided to focus on the development of a pirate radio. With this, we can broadcast the music and messages through ISIMA. We used a free software development kit that provides signal processing tools to implement SDR and signal processing systems: GNU Radio, and a SDR peripheral able to transmit and receive radio : HackRF One. For the transmission of the FM signal, we use the different blocks provided by GNU Radio, but for send a message we need to create our block. The FM radio, use a communications protocol standard for embedding small amounts of digital information, Radio Data System (RDS). RDS standardizes several types of information transmitted, as radio name and radio text.*

Because of the complexity of creating a block in GNU Radio and the time to understand the documentation of the RDS standard, only the broadcast of music was performed.

The project allowed us to discover the SDR world and what is possible to do. When we experimented the different tools, we also discovered security vulnerabilities in some applications around us in our daily lives. For example, It is possible to hack the neighbour's garage door and open it at will. Keywords: Radio Data System, Radio, GNU Radio, Software Defined Radio, FM, HackRF One.

(310 words)

Table des matières

[Remerciements i](#_Toc444681749)

[Résumé iii](#_Toc444681750)

[Abstract iv](#_Toc444681751)

[Glossaire vi](#_Toc444681752)

[Introduction 1](#_Toc444681753)

[I - Qu'es ce que la radio logicielle ? 2](#_Toc444681754)

[A. Applications de réception 3](#_Toc444681755)

[B. Applications d’émission 3](#_Toc444681756)

[II – Les outils 5](#_Toc444681757)

[A. Gnu radio : 5](#_Toc444681758)

[B. HackRF One : 6](#_Toc444681759)

[III - Radio "pirate" 7](#_Toc444681760)

[A. Principe 7](#_Toc444681761)

[B. Protocole RDS 8](#_Toc444681762)

[1. Trames RDS 8](#_Toc444681763)

[2. Codage des informations 9](#_Toc444681764)

[3. Réalisation de la solution 11](#_Toc444681765)

[Replay Attack 12](#_Toc444681766)

[GNU Radio 13](#_Toc444681767)

[HackRF\_Transfer 15](#_Toc444681768)

[Inspectrum 16](#_Toc444681769)

[Sécurité 16](#_Toc444681770)

[L’écoute de communication GSM 18](#_Toc444681771)

[Gr-gsm 18](#_Toc444681772)

[Wireshark 18](#_Toc444681773)

[Sécurité 18](#_Toc444681774)

# Glossaire

# Introduction

Dans notre vie quotidienne, nous sommes entourés d’ondes. En passant par la radio FM, au service de météo, à la téléphonie et plus récemment aux objets connectés, de nombreux objets utilisent les ondes pour échanger des informations.

Dans le cadre des portes ouvertes de l’Isima, afin de représenter la filière F5, Réseaux et sécurité, nous devions présenter une application de radio logicielle.

Ce projet a pour but de mettre en avant la filière F5 aux portes ouvertes de l’Isima.

Etant donné le nombre infini d’applications, qu’il est possible de réaliser, nous nous sommes concentré sur la réalisation d’une radio « pirate ».

Dans ce rapport nous allons tout d’abord nous intéresser à la définition de la radio logicielle et à certaines applications existantes. Ensuite nous présenterons les outils que nous avons utilisé dans le cadre ce projet ainsi que l’application que nous avons réalisé.

# I - Qu'es ce que la radio logicielle ?

La radio classique est un assemblage complexe de composant électronique dont le but est de traiter un signal, aussi bien à la réception qu'à l'émission, afin de transmettre et de recevoir des informations. Le système radio est spécifique à un signal.

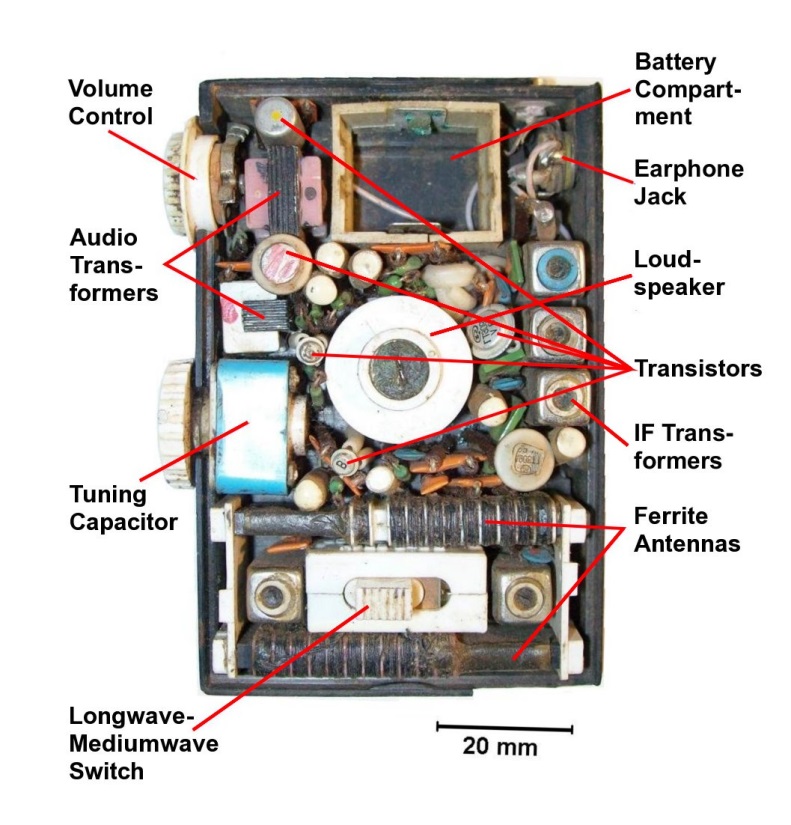


Figure . Radio analogique

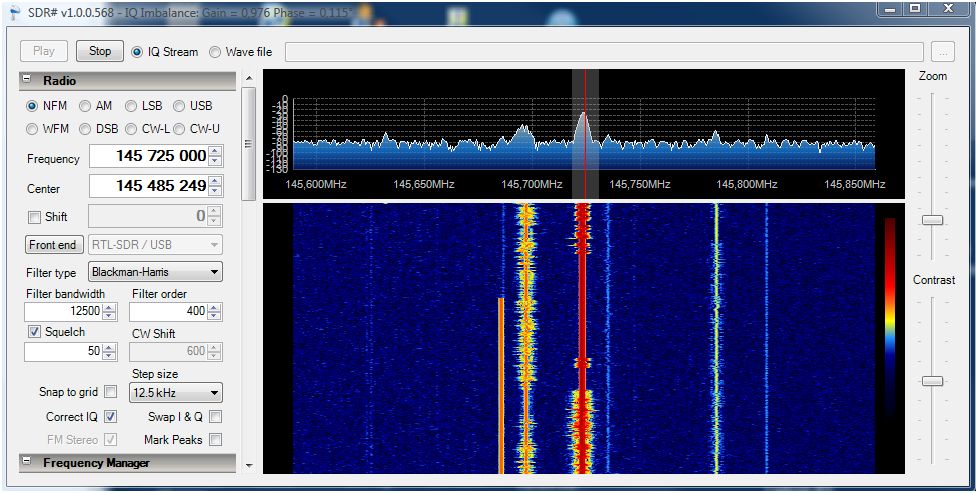
Une radio logicielle, au contraire, fonctionne de façon numérique à l'aide d'un processeur classique d'ordinateur. Un convertisseur analogique-numérique, lié à une antenne, permet de recevoir ou d'émettre sur une large bande de fréquences. Les signaux reçus seront ensuite traiter selon le type d'application. Il est donc possible d'utiliser le même équipement pour traiter une multitude de signaux, sous différentes formes et fréquences, en ne changeant que la partie logicielle.



La radio logicielle est beaucoup plus adaptable que la radio classique.

## Applications de réception

Les applications sont nombreuses et variées. Il est possible à l'aide d'une antenne adaptée de récupérer des informations de satellites, d'avions, de bateaux, … Il est possible de récupérer les signaux de n'importe quelle application qui émet des ondes. Par exemple, il est possible de récupérer les messages envoyés par les smartphones, messages envoyés par les services publiques comme la police, les pompiers, informations des avions, ...



## Applications d’émission

Pour émettre des ondes, cela est plus compliquées car les ondes sont protégées et réglementées. Il n'est pas possible de faire tout ce qu'on veut sous peine de sanction. Les fréquences radioélectriques appartiennent au domaine public de l'Etat. L'ANFR, l'Agence Nationale des Fréquences, gère et contrôle l'utilisation de ces fréquences. Cependant, en passant le coté légal des applications, il est possible de contrôler un drone avec un PC via une interface radio, ouvrir fermer une porte de garage, …



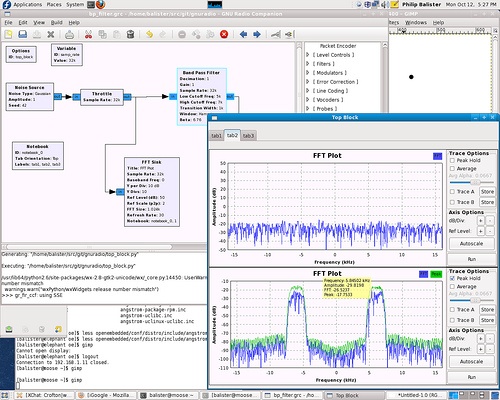
# II – Les outils

## 

## Gnu radio :

Afin de réaliser des applications de radio logicielle, nous avons utilisé une suite logicielle dédiée à leurs implémentations, Gnu Radio. L'utilisation du logiciel s'articule autour de blocs qu'il suffit d'interconnecter les uns aux autres. Les blocs comprennent des sources (récepteur radio, fichier, …), des blocs de traitements de signal, de l'affichage graphique, ainsi que des blocs de sorties (audio, graphique, …).

Pour réaliser une chaîne de traitement du signal, il est possible de le réaliser en le codant dans un des nombreux langages pris en charge par le logiciel, c++, python, … ou en utilisant une interface graphique, Gnu Radio Companion, qui permet d'assembler les modules graphiquement.



La sonde spatiale internationale Cometary Explorer, qui a été lancé en 1978 par la NASA pour surveiller l’activité du soleil, est actuellement sous le contrôle de GNU RADIO. En 1999, la NASA abandonna le satellite sur l’orbite du soleil. En 2008, le satellite était toujours opérationnel, un groupe de scientifique et de programmeur réussirent à reprendre le contact du satellite, en 2014, en adaptant GNU RADIO au protocole utilisé dans les années 1970.

## HackRF One :

Le HackRF One est un émetteur-récepteur pour la radio logicielle. Il couvre avec les antennes adaptées les fréquences de 1MHz à 6GHz.

HackRF One preliminary
photo

Développé par Micheal Ossman le but de cette carte a été de recréer des outils d’espionnage utilisé par la NSA, grâce aux fuites d’informations majeures apportées par Edward Snowden. Le projet a été ensuite « kickstarté » pour permettre de fournir un outil puissant à des coûts plus faibles que la concurrence. Une carte de type USRP est de l’ordre de plusieurs milliers de dollars tandis que la HackRF One est disponible pour mois de 400$.

# III - Radio "pirate"

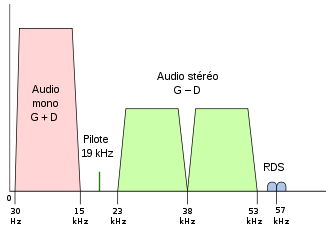
Dans le cadre de ce projet, nous avons décidé de développer une radio "pirate". L'application permet de diffuser de la musique et de diffuser des messages. Nous nous sommes fixés comme objectif de diffuser de la musique et d’afficher le nom de l’école comme nom de radio et un petit message lors des portes ouvertes.

## Principe

Au départ, la radiodiffusion consistait à moduler en fréquence la porteuse par un signal audio monophonique, d’une largeur de 15kHz. Avec l’arrivé de la stéréophonie et d’autres services, tels que le RDS (Radio Data System), le signal est devenu un signal multiplexe.

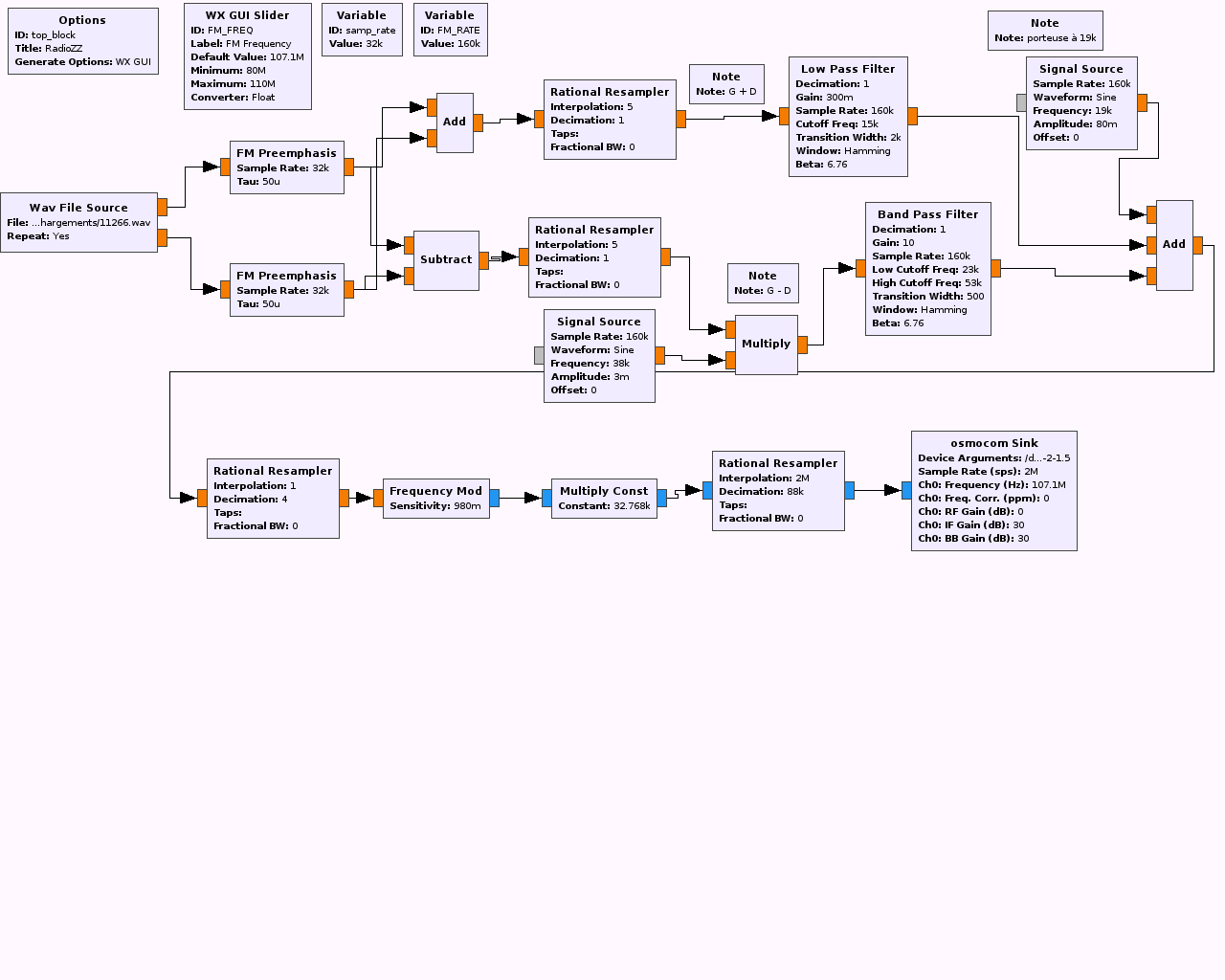
Pour garder la compatibilité avec les récepteurs monophoniques, un signal est émis sans modulation de 30Hz à 15kHz. Dans les émissions en stéréo, ce signal monophonique représente en réalité la somme des deux canaux stéréo gauche et droite (G+D). De cette façon un récepteur monophonique dispose bien de l’ensemble des sons émis. En stéréo, la différence des voies (G – D) est en plus transmise en modulant une sous-porteuse à 38kHz. Un signal pilote à 19kHz est ajouté au signal pour indiquer la présence de la stéréo et permettre le décodage de sous-porteuse sans risque de déphasage du son. Pour obtenir les voies gauche ou droite, le récepteur stéréophonique effectue respectivement la somme ou la différence des deux signaux G+D et G-D. La somme permet d’obtenir le résultat 2G et la différence 2D.

Les informations du RDS sont transmises sous forme numérique, via une sous porteuse à 57kHz.

[](https://commons.wikimedia.org/wiki/File:Modulation_radiodiffusion_FM.svg?uselang=fr)

## Transmission de la stéréo

Pour transmettre la stéréo, nous avons réalisé le montage suivant dans GnuRadio companion.



Nous sommes partis d’une source WAV, qui va lire un fichier audio et le décomposer en deux canaux, un pour le son gauche et un pour le son droit. Les blocs Pre-emphasis sont là pour améliorer la qualité du signal.

Pour transmettre la stéréo, il faut additionner le signal G+D, le signal G-D multiplier par une sous porteuse à 38 kHZ et une porteuse à 19 kHZ.

Pour le signal G+D (signal monophonique) et le G-D, il suffit, respectivement d’additionner et de soustraire les deux canaux entre eux. Le signal G+D est sans modulation, est doit être compris entre 30Hz et 15kHz. Dans notre cas, nous voulons envoyer un signal, la partie avant les 30Hz n'interfère pas avec les autres signaux, nous pouvons donc appliquer un filtre passe bas qui coupe à 15 kHZ.

Canal G

Canal D

+

G + D

Filtre passe bas à 15 kHz

Le signal G-D est multiplié par la sous porteuse à 38 kHZ (qui correspond à 2 fois la porteuse). Sur ce signal nous appliquons un filtre passe bande autour de 23 kHZ et 53 kHZ.

Canal G

Canal D

-

G - D

Filtre passe bande à 23 kHz à 53 kHz

Sous porteuse à 38 kHz

X

Pour avoir le signal stéréo il suffit d’additionner le signal G+D, la porteuse et le signal G-D.

G + D

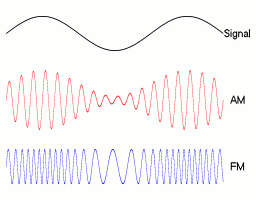
G - D

+

Porteuse à 19 kHz

Signal stéréo

Une fois le signal généré, il ne reste plus qu’à le moduler avant de le transmettre. En modulant en fréquence, l’information est portée par une variation de fréquence. Pour coder un « 1 » les fréquences sont plus rapprochées alors que pour coder un « 0 », les fréquences sont plus éloignées.



La modulation en fréquence est plus robuste que la modulation en amplitude pour transmettre un message dans de mauvaises conditions (bruits, atténuations, …).

Signal stéréo

Modulation en fréquence

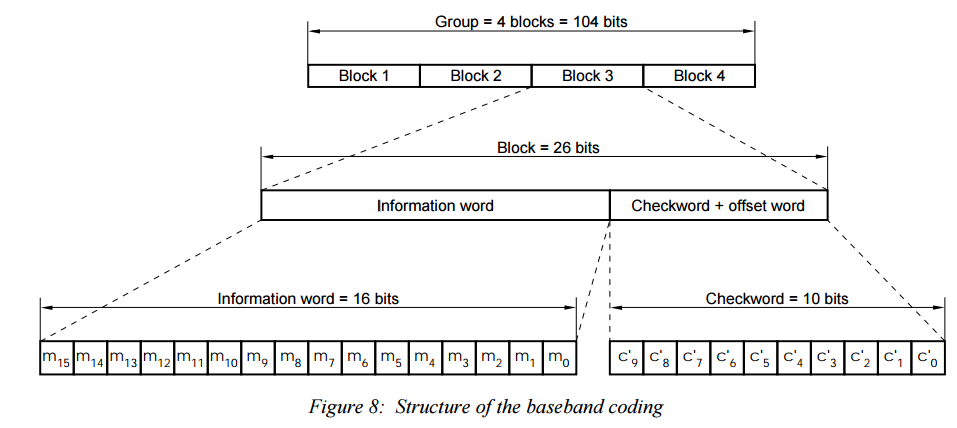
Hack RF One

## Protocole RDS

Le protocole RDS est un service de transmission de données, il permet l’écoute d’une station sans interruption lors d’un déplacement, en prenant en charge automatiquement le passage d’une fréquence à l’autre, il fournit une identification des stations par leur nom, des messages textuels, des flashes routier, etc.

### Trames RDS

La transmission d’une trame RDS est constituée de 4 blocs de données. Ils sont transmis en continu sans interruption. Un bloc est constitué d’une donnée sur 16 bits associée à un mot de contrôle de 10 bits. Les blocs sont transmis dans l’ordre A, B, C et D. Ces 4 blocs forment un groupe. Un groupe fait donc 104 bits (4\*26).



Le bloc A est utilisé pour transmettre le code d’identification de la radio, le bloc B permet d’identifier le type de données transmis et les blocs C et D contiennent les données.

Les fonctionnalités essentielles et les plus utilisées du RDS sont :

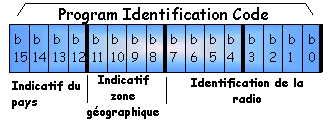
* La transmission du nom de la station sur 8 caractères (code PSN : Program Service Name)
* La transmission d’une liste de fréquences correspondant à des émetteurs voisins diffusant également la station en cours d’écoute (code AF – Alternative Frequency). Cette liste permet d’assurer la continuité de l’écoute d’une station même en mouvement.
* La transmission de messages pouvant atteindre 64 caractères (code RT – Radio Text). Elle permet, par exemple de donner des informations sur l’artiste et le nom du morceau qui joué.
* La transmission d’informations relative au trafic sur le réseau routier (code TMC – Traffic Message Channel). Ces informations permettent de prévenir, les automobilistes, au sujet de bouchons, de zones de travaux ou d’accidents.

Dans le projet, nous nous sommes intéressés au PSN et au Radio Text. Nous voulions émettre, lors de la journée porte ouverte de l’Isima, le nom de l’école comme nom de radio et un petit message pour mettre en avant notre filière, la filière F5.

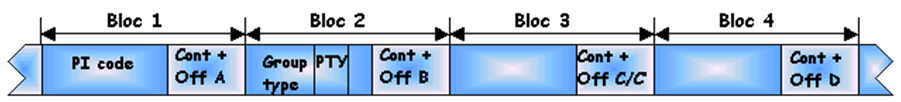
### Codage des informations

Les règles de codage de la norme RDS, sont définies dans la norme EN50067.

Le premier bloc de chaque groupe contient toujours la même information, le code identificateur de la radio, le PI code. Ce code est formé de 16 bits et est attribué par le CSA, le conseil Supérieur de l’audiovisuel. Les 4 premiers bits permettent d’identifier le pays émetteur, les 4 suivants à indiquer la couverture de l’émission et les 8 derniers à identifier la radio. Dans notre cas, le PI code n’a aucune importance, car celui-ci ne saura diffuser qu’au sein de l’Isima.

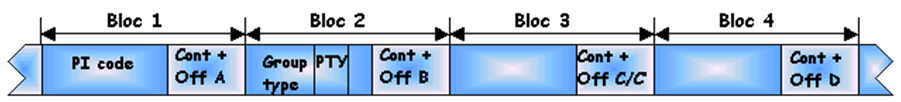


Le bloc suivant va identifier le type de données qu’on va envoyer. Dans notre cas, nous voulons afficher le nom de station, le PSN et une chaine d’information, Radio text. La norme indique que le groupe 0 permet de transmettre le PSN, pour le radio text le groupe est 2. Ensuite il faut choisir la version, soit A soit B. La version A permet de transmettre les fréquences alternatives. Ces fréquences sont envoyées dans le bloc C. La version B permet envoyer des données sans ces fréquences alternatives. Dans ce cas le bloc C sera égal au bloc A.



**AF**

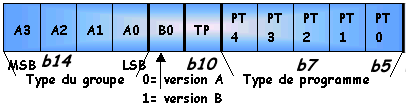
**Données**



**Données**

**Données**

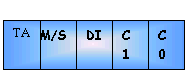
Les champs suivants sont pour indiquer si on envoie des informations sur le trafic routier et le type de programme diffusé par la radio, par exemple Sports, Information, Rock, …



Les autres bits du bloc correspondent aux informations de base (Basic tuning). Les informations de base dépendent du groupe est de la version choisit.

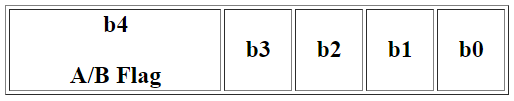
Pour le PSN, les informations de bases sont les suivantes :

* TA (Traffic Annoucement) : permet de savoir si des informations routières sont en cours de diffusion ou non.
* M/S (Music / speech) : Si une transmission de type discussion est en cours de transmission le flag a pour valeur 0, sinon c’est émission musicale et le flag est à 1.
* DI (Decoder Information) : Ce paramètre permet de donner des informations au récepteur sur le type de transmission utilisé, par exemple stéréo non compressé.
* Les 2 derniers bits permettent d’ordonner les groupes. Il est donc possible d’envoyé que 4 (2^2) groupes séquentiellement.



Pour le Radio Text, les informations de bases sont :

* Un flag qui permet d’effacer l’affichage en cours.
* Les 4 derniers bits permettent de déterminer l’adresse des caractères contenus dans le bloc D.



Pour le bloc C, comme on n’envoie pas de fréquence alternative il sera égale au bloc A, c'est-à-dire il contiendra l’identifiant de la radio.

Pour le bloc D, les données envoyées, le nom de la station ou le radio text, sont codé en ASCII. Comme un bloc de données comporte que 16 bits, il est possible d’envoyer que 2 caractères dans un groupe. Pour le PSN, le nom de la radio est limité à 8 caractères (4 adresses possibles \* 2 caractères) et le radio text à 32 caractères (2 ^4 = 16 adresses possibles \* 2 caractères).

### Réalisation de la solution

Gnu Radio ne permet pas de traiter le RDS de base. Nous avons recherché tout d’abord si des modules complémentaires à Gnu Radio existaient. Nous avons trouvé un module permettant de décoder et de coder du RDS. Ce module, gr-rds, réalisé par Bastian Bloessl, permet de décoder une trame rds et d’afficher les informations contenues dans le signal.

<Image>

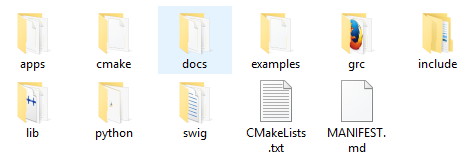
Cependant la partie encodage ne fonctionnait pas. Nous avons donc décidé de créer notre propre module pour GnuRadio companion.

Dans GnuRadio tout est considéré comme un bloc. Un bloc est une unité qui permet de réaliser une tâche. Par exemple, la lecture dans un fichier se fait grâce à un bloc, l’affichage de données dans un autre. En fonction de sa tâche, il existe trois types de blocs :

* Les blocs de type source. Ce type de bloc produit des données
* Les blocs de type sink. Ce type de bloc consomme des données.
* Les blocs de traitements. Ce bloc permet de transformer les données. Il possède une ou plusieurs entrées et sorties.

Pour la réalisation de notre bloc, nous avons eu le choix de le réaliser soit en Python soit en C++. Nous avons choisi la première solution car il est plus rapide de développer le bloc et de le tester avec ce langage. En effet dans un langage de script, il n’est pas nécessaire de compiler le programme à chaque modification.

Pour réaliser un bloc et de l’intégrer à companion, GnuRadio fournit un outil, gr\_modtool, qui permet de générer un squelette de module.



Le dossier include et lib vont contenir les fichiers C++, respectivement les headers et les fichiers sources. Le dossier python va contenir les scripts python. Le dossier grc contient la description du bloc au format xml pour pouvoir être intégrer à companion. Le fichier xml contiendra la description des paramètres du bloc ainsi que les entrées et les sorties. Le dossier apps contient les applications réalisé dans GnuRadio.

L'outil gr\_modtool permet également de générer automatiquement le squelette du code qui sera utilisé pour le bloc.



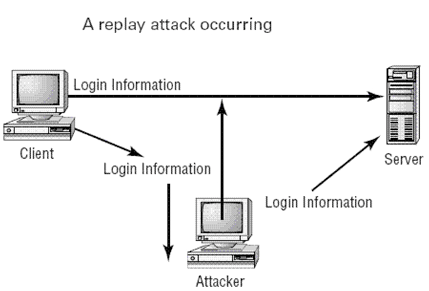
Le module comporte deux méthodes, une méthode init et une méthode work. La méthode init permet d'initialiser les entrées et les sorties du bloc. C'est dans la méthode work que le traitement sera effectué.

Dans notre bloc, nous nous sommes concentrés sur la transmission du nom de la station, le PSN et l'envoie d'un message text, le radio text. Malgré nos efforts nous ne sommes pas parvenues à le faire fonctionner. Au moment de l'exécution du bloc, nous avons une erreur au niveau de la capacité de stockage des données.

# Replay Attack

Après avoir effectué des expérimentations avec la radio nous nous sommes tournés vers des exploitations de failles de sécurité.

Notre première décision a été de nous tourner vers des choses simples comme intercepter un signal et ensuite le retransmettre de façon identique. Ce genre de technique concerne tous les équipements communiquants par Radio Fréquence comme les portes de garages, la centralisation des voitures ou encore certaines sonnette de maison.

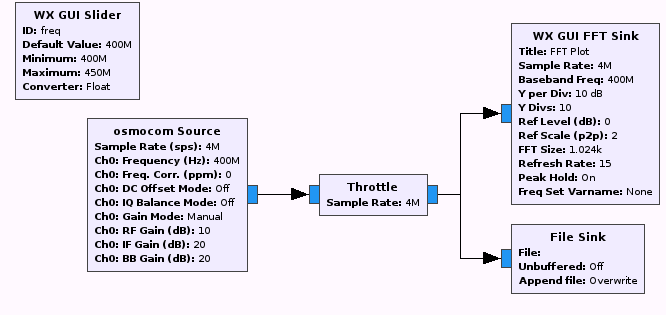


La première chose à faire a été de faire le choix vers quel équipements se tourner lors d’une réunion de projet nous nous sommes aperçu que nous disposions d’une clef de voiture qui émet aux alentours des 433 MHz, fréquence situé dans notre gamme de possibilités.

Donc notre objectif est de pouvoir ouvrir la centralisation d’une voiture uniquement par l'intermédiaire de la HackRF One.

## GNU Radio

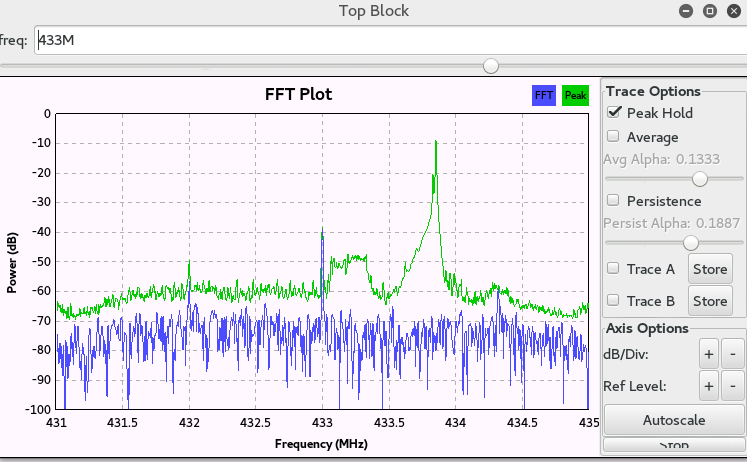
La première étape est donc d’intercepter le signal à reproduire pour cela, nous avons mis en place ce schéma simple sous GnuRadio.



Notre schéma est composé de plusieurs éléments:

* Le WX GUI Slider correspond à une variable permettant de balayer les fréquences pour ainsi se positionner sur les fréquence à observer on l’a ici positionné pour observer les fréquences entre 400 à 450 MHz.
* L’Osmocom Source correspond à la carte HackRF One qui observera les fréquences balayés sur une fenêtre de 4 MHz (taux d'échantillonnage). Les différents gains et paramètres ont été laissés par défaut pour le moment.
* Le bloc throttle permet lui une stabilité de l’application. En effet le traitement et l’affichage des données en temps réel peut être une lourde charge pour un processeur d’ordinateur. C’est pour cela que ce bloc limite les transferts à la vitesse du taux d'échantillonnage.
* Le WX GUI FFT est un bloc d’observation permettant d’observer le signal par une Transformée de Fourier rapide (Fast Fourier Transform). Il permet ainsi d’analyser les signaux dans le domaine fréquentiel pour observer les évènements concernant les signaux.
* Enfin le file sink permet de stocker les signaux capturés dans un fichier pour pouvoir être réutilisé ultérieurement. Nous le sauvegarderons sous le nom /root/sdr/clefFord.

Une fois le script généré et exécuté voici ce que nous obtenons



L’option “peak hold” permet de sauvegarder les valeurs maximales du signal et donc sauvegarder visuellement les évènement. Nous pouvons observer un pic non négligeable lorsqu’on appui sur la clef de voiture.

Le signal ayant été capturé nous allons donc le retransmettre en état pour voir si on l’on peut observer l’ouverture de la voiture pour cela nous allons utiliser ce schéma GnuRadio.

# 

Voici donc le schéma de transmission, du signal capturé. Il est composé de trois élément :

* Le file source est le signal sauvegardé à l’étape précédente
* L’Osmocom Sink est le matériel de diffusion (Carte HackRF One) ce bloc est configuré pour restituer le signal à l’identique donc il est centré à 433MHz et dispose du même taux d’échantillonnage.
* Enfin on observera le signal émis pour vérifier qu’il est identique à celui capturé.

Le signal est effectivement identique mais malheureusement aucun évènement ne s’est produit. Pour analyser l’échec de la tentative nous allons observer les données transmises par la clef.

## HackRF\_Transfer

Plutôt que d’utiliser Gnuradio il est possible d’utiliser une commande inclus dans la librairie hackrf il s’agit de la commande hackrf\_transfer qui permet en effet de capturer ou de diffuser des signaux. Pour capturer un signal voici la commande à taper :

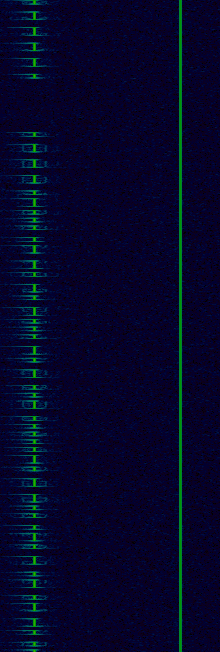
hackrf\_transfer -r clefFord.iq -f 435000000 -s 4000000

* l’option “r” correspond au fichier de sortie
* l’option “f” la fréquence sur laquelle se centrer
* l’option “s” le taux d’échantillonnage (fenêtre d’observation sur le plan fréquentiel)

## Inspectrum

Il est ainsi possible de voir les données envoyées par la clef de voiture en utilisant un outil appellé inspectrum (utilisé par MIcheal Ossman concepteur de la HackRF One). Ce logiciel permet donc de faire une analyse temps-fréquence.

Voici donc ce que l’on obtient avec inspectrum à gauche on peut reconnaître ce qui pourrait correspondre à une entête et enfin les données avec une alternance de signaux court et de signaux plus long correspondant à du binaire.



## Sécurité

Les clefs de centralisation de voiture utilise le système de code tournant(Rolling Code) pour pouvoir se prémunir de la l’attaque par rejeu. Il s’agit d’un système de sécurité

# L’écoute de communication GSM

Le Global System for Mobile Communication (GSM) est une norme de communication pour la téléphonie mobile.

## Gr-gsm

## Wireshark

## Sécurité

# Conclusion