

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد - تلمسان

Université Aboubakr Belkaïd - Tlemcen -
Faculté de TECHNOLOGIE



MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Télécommunications

Spécialité : Réseaux et Télécommunications

Par :

OUNADJELA Fatima Zohra

BOUHAFNA Nafissa

Sujet

Développement d'un récepteur large bande basé sur le dongle RTL-SDR

Soutenu le 27 juin 2018, devant le jury composé de :

Mme.HAFFAF
Hadjer

Université de
Tlemcen

Présidente

Mr.BOUABDELLAH
Reda

Université de
Tlemcen

Examinateur

Mr .BENADDA
Belkacem

Université de
Tlemcen

Encadrant

Année universitaire : 2017-2018



Remerciement

Nous remercions avant tout Allah tout puissant qui nous a comblé de ses bienfaits et nous a donné assez de force pour achever ce travail et de venir au bout de cette formation

Nous tenons tout d'abord à remercier la personne sans qui ce travail ne serai pas ; notre encadreur monsieur BENADDA BELKACEM, qui a accepté de diriger ce travail et nous a guidé tout au long,

Nous le remercies pour sa disponibilité, ses compétences qu'il a mises à notre service, et de son extrême gentillesse.

Nous remercions aussi nos familles, en particulier à nos parents pour leur tendresse et leur amour qui nous guidons aux quotidiens.

Enfin, nous tenons à exprimer notre reconnaissance à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail.

Résumé

Les systèmes SDR (Software Defined Radio) se présentent comme un nouveau paradigme pour les systèmes de télécommunications sans fils. En plus de tel systèmes entièrement reconfigurables permettent d'offrir des services variés sur la même plateforme matérielle. Dans ces systèmes soit un réseau ou un nœud sans fil, sont doté de logiciels pour modifier les paramètres de transmission ou de réception pour pouvoir communiquer efficacement en évitant les interférences avec les utilisateurs autorisés ou non autorisés. Dans ce travail, nous avons utilisé le dongle USB RTL-SDR sur une station de travail comme une plateforme de réception multi standard (3MHz-2GHz environ). La finalité consiste à concevoir des récepteurs dédiés aux télécommunications sans fils qui peuvent soutenir différents services. Une application spécifique à un monitoring des activités fréquentielles avoisinante est réalisée.

Mots clés : Radio définie par logiciel, SDR, RTL-SDR, Le chipset RTL2832U

Abstract

Software Defined Radio (SDR) systems are presented as a new paradigm for wireless telecommunications systems. In addition to such reconfigurable abilities, those systems are able to offer various telecommunications services on the same hardware platform. In this work, we will use the USB RTL-SDR dongle on PC workstation. The goal is to design wireless telecommunications receivers that can support different services. A specific application for nearby frequencies activities is realized.

Keywords: Software Defined Radio, **SDR, RTL-SDR ,the RTL2832U** Chipset.

ملخص

تعد الأنظمة اللاسلكية المبرمجة (SDR) كنموذج جديد لأنظمة الاتصالات اللاسلكية. بالفعل هذه الأنظمة قابلة لإعادة البرمجة الشيء الذي يسمح بتقديم خدمات متنوعة على نفس الجهاز اللاسلكي. في هذا المشروع، سنقوم باستخدام USB RTL-SDR على محطة عمل عادية لاستقبال والتحكم في الاشارات ذات التردد بين حوالي 3MHz-2GHz. الهدف هو تصميم مستقبلات الاتصالات اللاسلكية التي يمكنها دعم خدمات مختلفة. سيتم تنفيذ تطبيق لتشخيص نشاط الترددات المجاورة.

الكلمات المفتاحية: برنامج راديو محدد، SDR ، RTL-SDR ، مجموعة شرائح

RTL2832U

Table des matières

Introduction générale.....	1
Chapitre I : principe de la radio logicielle	3
I.1 Introduction.....	4
I.2 Historique.....	4
I.3 Présentation.....	5
I.3.1 Définition	5
I.3.2 Classification du radio logicielle selon leur niveau de configuration	6
I.3.2.1 La radio matérielle :	6
I.4 Objectifs de réalisation d'une radio logicielle.....	7
I.5 L'impact de la radio logicielle sur les équipements de télécommunication actuelle	7
I.6 Architecture du radio logiciel.....	7
I.6.1 Un équipement radio	7
I.6.2 La radio logicielle (RL).....	9
I.6.3 La radio logicielle restreinte (RLR) ou(SDR).....	10
I.7 Avantages.....	11
I.8 Inconvénients de SDR.....	12
I.9 Domaine d'application.....	12
I.10 Conclusion	12
Chapitre II : Architecture et fonctionnement d'un récepteur RTL-SDR.....	14
II.1. Introduction.....	15
II.2 Présentation.....	15
II.2.1 La radio logicielle et ses différents usages	15
II.2.1.1 Les radios logicielles tactiques	15
II.2.1.2 La radio intelligente (cognitive radio)	16
II.2.1.3 Radio amateurisme	17

Table des matières

II.2.1.4 RTL-SDR.....	18
II.3.1 Diagramme bloc d'un RTL-SDR.....	18
II.3.2 Le chipset RTL2832U	19
II.3.2.1 Définition.....	19
II.3.2.2 Le principe du chipset RTL2832U.....	19
II.3.2.3 L'architecture du chipset RTL2832U	20
II.3.2.4 Caractéristique du chipset RTL2832U.....	21
II.4 Les Caractéristiques du technique RTL-SDR.....	21
II.5 Les interfaces les plus connu pour le dongle RTL-SDR :	22
II.6 Domaine d'application.....	22
II.7 Conclusion	23
Chapitre III : Installation et configuration du logicielle SDR Sharp pour le dongle RTL-SDR.....	24
III.1 Introduction	25
III.2Materiel utilisé.....	25
III.3 Installation et mise en Service du dongle RTL-SDR	25
III.3.1 Configuration.....	26
III.3.2 Test du fonctionnement :	28
III.4 conclusion.....	30
Chapitre IV : Développement d'un récepteur de détection de l'activité spectrale	31
IV.1-Introduction.....	32
IV.2-Installation du package pour la radio RTL SDR.....	32
IV.3- Premier récepteur RTL-SDR sous Simulink.....	34
IV.3.1 La solution proposée	35
IV.3.1.1Les rôles des blocs du schéma réalisé	36
IV.3.2 Le masque utilisé dans le schéma bloc.....	37
IV.4 Programmation du Récepteur.....	42
IV.4.1Définition de la densité spectrale de puissance DSP.....	42
IV.4-2- initialisation des paramètres :	43

Table des matières

IV.5 Conclusion.....	48
Conclusion générale	49
Bibliographies	50

Liste des figures

Figure I.1 : Répartition matériel/logiciel dans une radio logicielle.....	18
Figure I.2 : classification du radio logicielle.....	18
Figure I.3 : principales architectures de récepteurs radio.....	20
Figure I.4 : Architecture classique superhétérodyne d'un récepteur par ondes radios.....	20
Figure I.5 : Architecture fonctionnelle idéale d'un système radio logicielle (RL)	21
Figure I.6 : Architecture fonctionnelle typique d'un système de radio logicielle restreinte (RLR)	22
Figure I.7 : Architecture superhétérodyne et RLR d'un système de radiocommunications multistandards	23
Figure II.1 la relation entre la radio cognitive et radio logicielle restreinte.....	29
Figure II.2 : diagramme bloc du récepteur RTL-SDR	30
Figure II.3 : schéma fonctionnel montrant l'architecture interne du R820T/R2832U RTL-SDR	32
Figure III.1 : Dongle RTL-SDR branché sur PC	37
Figure III.2 : Les paramètres da Zadig	39
Figure III. 3 : La mise a jour du driver	39
Figure III. 4 : installation du l'outil zading	40
Figure III.5 : Exemple d'exécution de l'application SDRSHARP.....	40
Figure III.6 : Les sources de logiciel SDR Sharp.....	41
Figure III.7 : RTL-SDR Controller	41
Figure III.8 : Exemple des chaines.....	42
Figure IV.1 : Emplacement pour l'installation du package RTL-SDR.....	44
Figure IV.2 installation du package	45
Figure IV.3 : le dongle USB RTL-SDR branche sur le PC.....	45
Figure IV.4 vérification d'installations du package RTL SDR.....	46
Figure IV.5 : la bibliothèque du MATLAB & Simulink.....	46
Figure IV. 6 le block système sous Matlab & simulink	47
Figure IV.7 : le paramètre de bloc sub-system.....	47
Figure IV.8 les paramètres du récepteur RTL SDR.....	48
Figure IV.9 :l'information sur le dongle RTL-SDR.	49

Liste des figures

Figure IV.10 : éditeur de masque	50
Figure IV.11: Paramètres et onglet de la boîte de dialogue	51
FigureIV.12 : Onglet Initialisation	52
Figure IV.13 : Onglet Documentation.....	53
Figure IV.14 les résultats obtenus à partir de programme essai.slx	54
Figure IV.15 : les paramètres initialisés dans le programme	57
Figure IV.16 :organigramme corespondant au script	59
Figure IV.17:le scanne d'analyseur de spectre en FFT et waterfall.....	60

Liste des abréviations

Liste des abréviations

ADC	: Analogue to Digital Converter
AFRL	: Air Force Rome Labs
AGC	: Automatic Gain Control (contrôle automatique de gain)
ASIC	: Application-Specific Integrated Circuit,
BTS	: Base Transceiver Station (Station de base)
CAN	: Convertisseurs Analogique/Numérique
COFDM	: Coded Orthogonal Frequency Division Multiplexing
DAB	: Digital Audio Broadcast
DAB+.	: Elle n'est rien d'autre qu'une version améliorée de DAB et elle utilise le tout dernier encodage audio.
DARPA	: Defense Advanced Research Projects Agency
dB	: Décibel. Le dB correspond au dixième de bel, unité de mesure Logarithmique du rapport entre deux puissances.
DMR	: du Digital Modular Radio
DoD	: Department of Defense, Department of Defense
DSP	: Digital Signal Processing
DVB-T	: Digital Video Broadcasting-Terrestrial
EDGE	: Enhanced Data rates for GSM Evolution
FFT	: Fast Fourier Transform
GNURadio	: GNU Radio est un logiciel gratuit et open source qui fournit des blocs de traitement du signal pour mettre en œuvre de la radio logicielle
GPS	: Global Positioning System
GSM	: Global System for Mobile Communications (système global de communication mobile)
HDSDR	: un logiciel freeware Radio (SDR)
HF	: Haute fréquence. La HF désigne les ondes radio dont la fréquence est comprise entre 3MHz et 30MHz
I2C	: Inter Integrated Circuit. I2C est un bus série et synchrone
ICNIA	: Integrated Communications, Navigation, and Identification Avionics,
IF	: Intermediate Frequency
IQ	: In Phase! Quadrature complex signal

Liste des abréviations

ISS	: International Space Station
JTRS	: Joint Tactical Radio System
MPEG2-TS	: Moving Picture Experts Group Transmission Stream
Msp/s	: Megasamples par second
NCOs	: Numerically Controlled Oscillator
NCW	: Network Centric Warfare
NOAA	: National Oceanic and Atmospheric Administration
RDS	: Le Radio Data System
RF	: Radio Frequency
RTL	: Realtek lab (laboratoire Realtek).
RLR	: radio logicielle restreinte
SDR	: Software-Defined Radio
TNS	: Traitement Numérique du Signal
TOP	: point de transfert programmable
UHF	: Ultra haute fréquence. L'UHF désigne les ondes radio dont la fréquence est comprise entre 300MHz-3GHz,
Viterbi	: Algorithme permettant de corriger, dans une certaine mesure, les erreurs survenues lors d'une transmission à travers un canal bruité. Il s'appuie sur la distance de Hamming afin de faire ressortir la plus faible métrique entre les différentes valeurs probables d'une séquence d'états
VCO	: Voltage-Controlled Oscillator (oscillateur contrôlé en tension)
VHF	: Very High Frequency, ou Très haute fréquence. La VHF désigne les ondes radio dont la fréquence est comprise entre 30 MHz et 300 MHz.
WNW	: Wideband Networking Waveform

Introduction Générale

On assiste actuellement à un développement rapide et un déploiement mondial d'infrastructures de communication sans fil large bande avec des modulations numériques. La diversité qui est introduites dans les bandes fréquentielles utilisées, le type de la modulation, ainsi que le mode de transmission, invite les utilisateurs à disposer de plusieurs catégories d'émetteur récepteur, qui est le premier désavantage et l'utilisation de la technologie de traitement numérique dans l'unité frontale et radiofréquence. Il n'est pas difficile de voir que la technologie du traitement numérique appliquée pour les émetteurs et les récepteurs de communication sans fil couvre un large éventail d'applications. Des sujets qui traitent la conversion analogique numérique, le filtrage numérique, ou éventuellement la densité spectrale de puissance, des technologies de traitement numérique qui offrent un certain nombre d'avantages en matière d'efficacité, de réduction des coûts, de délai d'exécution et de flexibilité. Il est évident qu'un traitement numérique ne peut être effectué que par le biais d'un logiciel pour la radio définie par logiciel (SDR). Le premier apport de ce paradigme et la facilité de prendre en charge de multiples normes et applications des télécommunications sans fils. Contrairement au traitement en bande de base, pour les systèmes SDR les fronts de réception radiofréquence sont étroitement connectés aux systèmes de traitement numériques, ce qui impose de grandes limitations et difficultés sur la vitesse de traitement, la mémoire, la capacité de calcul, la puissance, la taille, les interfaces et les largeurs de bande.

Le dongle RTL-SDR se présente comme une plateforme de la radio logicielle SDR bon marché, ce dongle coûte environ 200DZD. Les RTL-SDR sont initialement utilisés comme des tuners TV DVB-T avec des puces RTL2832. Cependant le RTL-SDR peut également servir de scanner radio large bande. Les récepteurs RTLSDR utilisent un convertisseur analogique numérique en quadrature Realtek RTL2832 en combinaison avec un oscillateur programmable (le plus souvent un Elonics E4000, Fitipower FC0013, FC0012, FC2580 et R820T). Les expériences indiquent que ces périphériques fonctionnent assez bien sans aucune modification matérielle, il suffit de développer les bons logiciels et pilotes, par exemple : GNU Radio ou HDSDR.

Introduction Générale

Dans ce travail le matériel RTL-SDR se compose d'une clé USB et une antenne omnidirectionnelle. Le dongle USB utilisé est basé sur la puce Realtek RTL2832U.

Notre mémoire est organisé en quatre chapitres, un premier chapitre aborde les systèmes de la radio logicielle, nous avons abordés cette étude sur la radio logicielle pour expliquer le contexte sur lequel évolue notre travail. Le chapitre 2 détail le fonctionnement du front end RTL-SDR un outil que nous allons utiliser pour développer notre solution. Destiné en premier à être utilisé comme récepteur TV DVB-T, convertir le RTL-SDR en une plateforme SDR qui nécessite une configuration et mise en service particulière qui fait l'objet du chapitre 3. Le chapitre 4 est consacré au développement de notre application récepteur qui se limite à détecter les activités spectrales avoisinantes.

Chapitre I : principe de la radio logicielle

I.1 Introduction

Aujourd'hui, la radio logicielle est la convergence entre la radiocommunication et l'informatique. La radio logicielle exploite la puissance de traitement des technologies informatiques modernes pour simuler le comportement d'un circuit radio. En effet, une radiologicielle est un système de radiocommunication configurable utilisant des techniques de traitement logiciel sur des signaux radio. Elle se présente comme un système qui permet à un terminal de pouvoir interagir avec son environnement, et doit pouvoir aussi traiter plusieurs normes de communications radio ayant des spécifications différentes, à savoir des canaux de communications à bande étroite ou large. Cette dernière est capable de percevoir son environnement, et de s'y adapter. Il pourra donc détecter les fréquences et les utiliser. Dans ce chapitre, nous allons étudier l'architecture, le fonctionnement d'une antenne RTL-SDR et leur différent domaine d'utilisations.

I.2 Historique

La radio logicielle n'est pas vraiment une nouvelle technologie, mais plutôt une évolution logique et une convergence des radios numériques et des technologies logicielles. Le concept de la radio logicielle est né de la recherche militaire américaine. L'origine de la conception radio logicielle est apparu aux années 1987, à partir de la "Air Force Rome Labs" (AFRL) qui a financé le développement de la première radio programmable comme une étape d'évolution après de l'architecture intégrée de communications, de navigation, et d'identification (ICNIA) [1].

Le programme ICNIA initié en 1970 dans le but d'introduire les premières radios VHF multi-modes dans l'armée américaine.[1]. Au début des années 1990, l'AFRL et la "Défense Advanced Research Projects Agency" (DARPA) financent le programme SPEAKEasy afin de pallier aux problèmes d'incompatibilités entre les moyens de communication des différentes branches de l'armée américaine [2].

Les technologies Speakeasy ont ensuite évoluées en 1998 vers le programme "Joint Tactical Radio System" (JTRS) financé par le Joint Program Office (JPO) de l'US Département of Défense (DoD) [1]. En 1998 l'US Navy est doté du Digital Modular Radio (DMR) devenu une radio logicielle restreinte multi modes à quatre canaux RF full-duplex dans la bande 2 MHz-2 GHz [1]. Aujourd'hui connu sous le nom de Software Defined Radio (SDR) forum

Le terme « radio logicielle » a été inventé officiellement en 1991 par Dr Joseph Mitola, qui travaillé en tant que « program manager » pour la défense américaine, il publié le premier article sur le sujet en 1992[2].

I.3 Présentation

I.3.1 Définition

Une radio logicielle, en anglais *Software Radio*, est un système de radiocommunication configurable utilisant des techniques de traitement logiciel au lieu de dépendre du matériels sur des signaux radiofréquences ; Utilisant des circuits numériques programmables pour réaliser le traitement du signal. Sa flexibilité lui permet de s'adapter à un large spectre de réseaux, protocoles et techniques de radiocommunication.

Une radio logicielle met en œuvre deux notions fondamentales : la plate-forme et la forme d'onde.

I.3.1.1 Une plate-forme : est l'ensemble du matériels radio et des systèmes de traitement qui hébergent une ou plusieurs formes d'onde. Dans une radio logicielle, une forme d'onde est l'ensemble des transformations algorithmiques appliquées à l'information pour la convertir en signal radio [3].

I.3.1.2 Une forme d'onde : est entièrement logicielle et définit les caractéristiques du signal RF, dont la fréquence, la modulation et le format.

Puisque les formes d'onde sont entièrement logicielles, la radio logicielle permet l'implémentation sur une plate-forme existante grâce à une mise à jour, mais à condition qu'elles soient portables sur la plate-forme considérée, d'où l'importance des recherches actuelles dans le développement de standards ouverts [3].

En plus, les radios logicielles modernes mettent également en œuvre des fonctions cryptographiques, de correction d'erreurs et de codage de source de la voix, vidéo ou des données.

Une radio logicielle est composée de deux sous-systèmes : un sous-système matériel et un sous-système logiciel. La répartition des opérations réalisées dans les sous-systèmes matériels et logiciels est illustrée dans le schéma suivant.

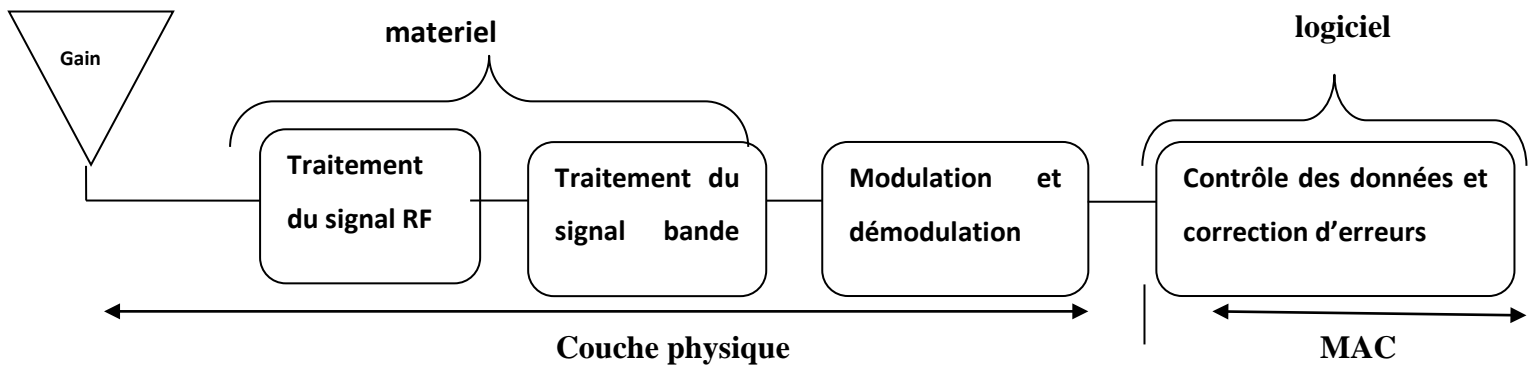


Figure I.1 : Répartition matériel/logiciel dans une radio logicielle

La partie logicielle nous offre la propriété de la fréquence porteuse, démodulation, la bande passante du signal et d'accès réseau.

I.3.2 Classification du radio logicielle selon leur niveau de configuration

On peut classier la radio logicielle sur cinq catégories différentes [4] :

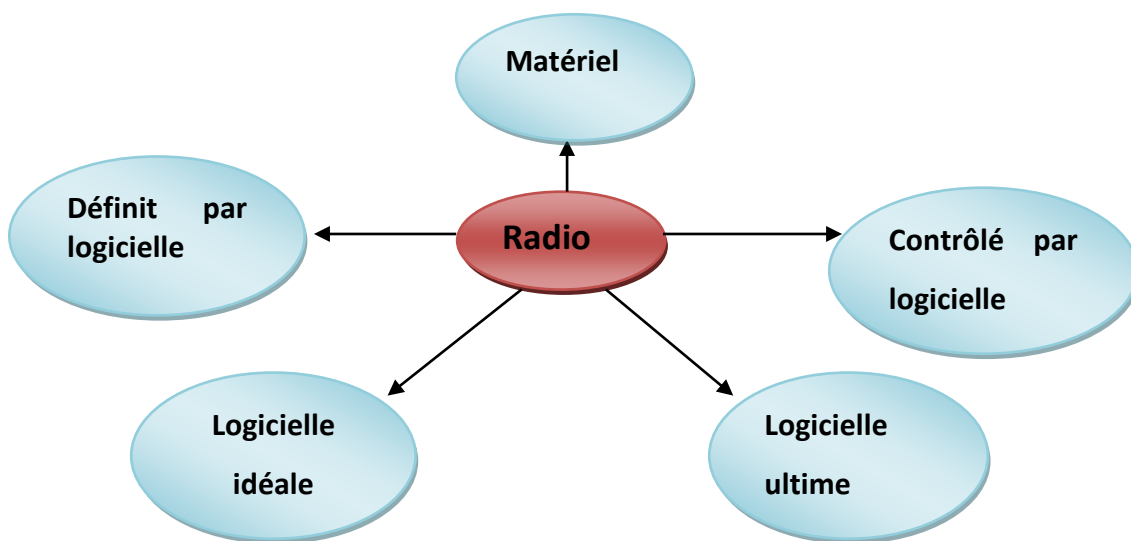


Figure I.2 : classification du radio logicielle

On peut faire la catégorisation du type de radio logicielle selon le degré de configuration comme suit :

I.3.2.1 La radio matérielle : entièrement matérielle c'est-à-dire on ne peut pas être modifiée par logiciel donc la reconfiguration se faire par l'échange de composants.

I.3.2.2 La radio contrôlé par logiciel : reconfiguration logicielle limitée

I.3.2.3 La radio définie par logicielle : reconfiguration des formes d'ondes, fréquence, bande passante, contrôlé logicielle

I.3.2.4 La radio logicielle idéale : conversion analogique au niveau d'antenne, du haut-parleur et du microphone ; tout le reste est logicielle

I.3.2.5 La radio logicielle ultime : comprend tout type de trafic et d'information de contrôle, et supporte la plupart des applications et circuits d'antennes.

I.4 Objectifs de réalisation d'une radio logicielle

- remplacer des contraintes matérielles en contraintes logicielles, plus faciles à résoudre et moins onéreuses.
- La conversion analogique/numérique est ainsi positionnée au plus proche de l'antenne, Tant à l'émission qu'à la réception, pour effectuer un maximum d'opérations dans le domaine numérique [3].

I.5 L'impact de la radio logicielle sur les équipements de télécommunication actuelle

La radio logicielle permet de :

- réduire de manière importante le coût et l'investissement en matériel de radiocommunication.
- bénéficier de la puissance d'un PC pour traiter (presque) n'importe quel signal radio...
- mettre à jour par logicielles BTS pour supporter de nouvelles modulations (exemple : migration de la norme GSM à la norme EDGE)
- d'émettre (et de recevoir) des signaux simultanément dans plusieurs canaux, à l'aide d'un unique émetteur/récepteur, etc. [2]

I.6 Architecture du radio logiciel

I.6.1 Un équipement radio

Un récepteur radio se compose de plusieurs parties : une antenne qui reçoit le signal radiofréquence, un sous-système radiofréquence/fréquence intermédiaire (RF/FI) qui convertit et filtre le signal dans la bande spectrale désirée, et un démodulateur/décodeur qui convertit ce signal sous un format exploitable. Le résultat obtenu est transmis à une application et enfin à l'utilisateur [3].

Le terme radio logicielle se décline en plusieurs expressions, suivant l'architecture du transmetteur.

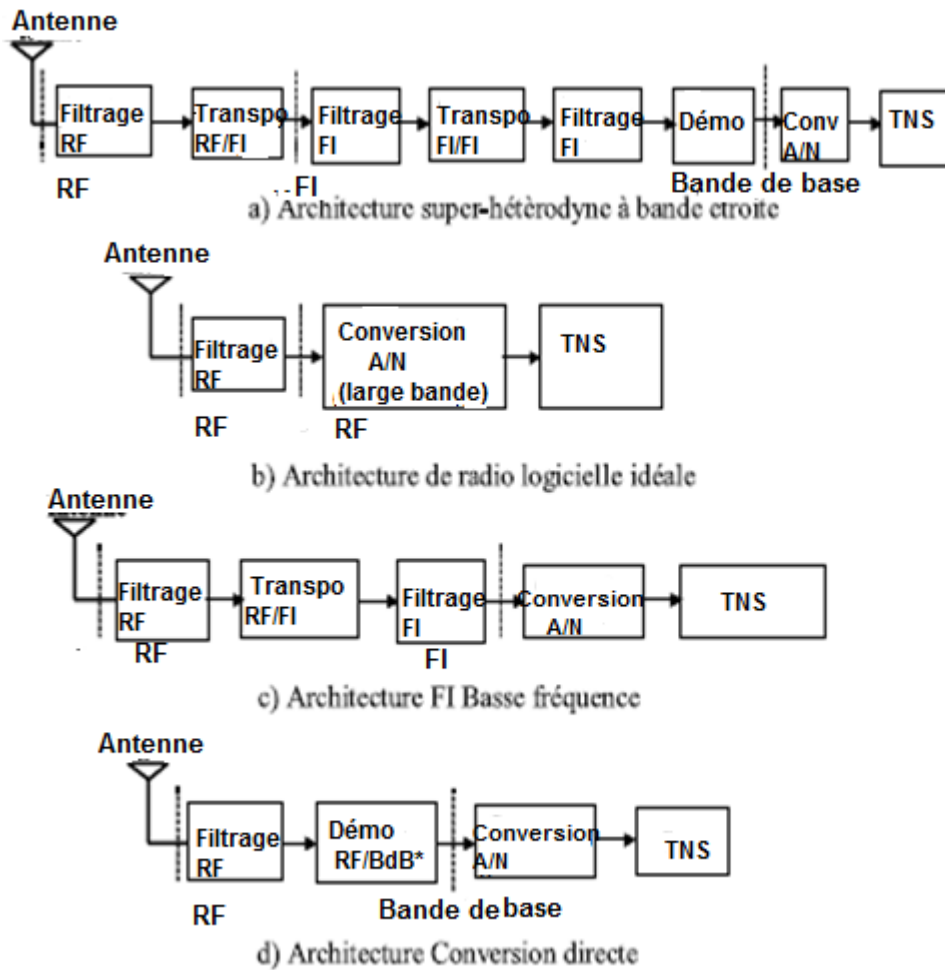


Figure I.3 : principales architectures de récepteurs radio [5]

La Figure I.4 montre une architecture classique dite superhétérodyne d'un récepteur de radiocommunication comprenant un traitement RF, un traitement en FI et un traitement en bande de base. L'architecture de l'émetteur est symétrique [6].

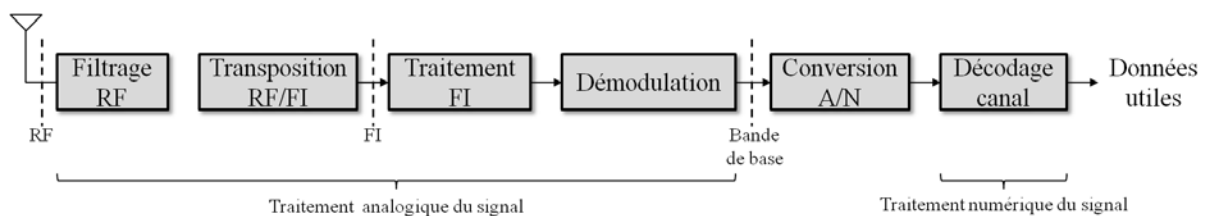


Figure I.4 : Architecture classique superhétérodyne d'un récepteur par ondes radios

Pour supporter une telle architecture fonctionnelle la plate-forme matérielle est le plus souvent hétérogène et peu flexible.

Aujourd'hui, de nombreux systèmes de radiocommunications supportent plusieurs standards de communications. Ce qui, le plus souvent implique une multiplication de la chaîne de traitement du signal et donc une multiplication des composants.

I.6.2 La radio logicielle (RL)

La RL idéale, principalement définie pour un récepteur, consiste à effectuer la numérisation du signal dès la sortie de l'antenne de réception. Ainsi, tous les traitements de filtrages, les traitements de transpositions fréquentielles et les traitements en bande débrase sont effectués par traitements logiciels et non plus par des composants dédiés.

La Figure I.5 met en évidence l'architecture fonctionnelle RL idéale d'un système de radiocommunication. Contrairement à une architecture superhétérodyne (Figure.4) l'ensemble des traitements appliqués sur le signal réceptionné est réalisé dans le monde numérique. Cela permet de bénéficier de la flexibilité du traitement numérique totalement mise en œuvre en logiciel, supporté par une plate-forme matérielle composée de processeurs [4].

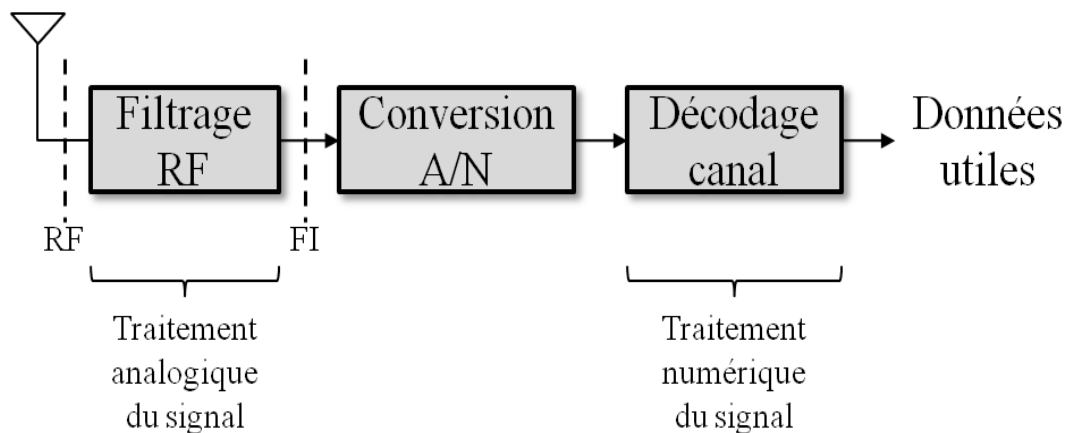


Figure I.5 : Architecture fonctionnelle idéale d'un système radio logicielle (RL)

Il existe encore de nombreux challenges à relever (notamment au niveau du traitement RF), avant de pouvoir prétendre à proposer des systèmes RL idéaux. Afin

de réduire la marche entre un système de radiocommunication superhétérodyne et un système RL, une étape intermédiaire est apparue : la RLR (*Radio Logicielle Restreinte* ou encore plus couramment nommé SDR – *Software Defined Radio*).

I.6.3 La radio logicielle restreinte (RLR) ou(SDR)

Contrairement à la RL, dans un système RLR la numérisation du signal n'est pas effectuée directement en sortie de l'antenne mais une fois le signal transposé en FI, voire en bande de base, après filtrage analogique RF [6].

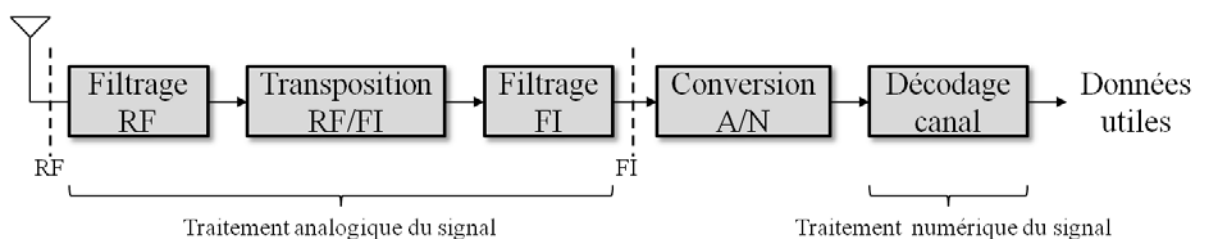


Figure I.6 : Architecture fonctionnelle typique d'un système de radio logicielle restreinte (RLR)

La Figure I.6, illustre l'architecture d'un système RLR (SDR). Contrairement à une architecture RL, le traitement analogique du signal réceptionné n'est pas réduit à son plus simple élément. Une plate-forme matérielle d'un système RLR est donc nécessairement

Constituée d'un composant dédié ASIC (Application-Specific -Integrated Circuit) au traitement analogique. De plus contrairement à la RL idéal, le traitement numérique du signal n'est pas totalement effectué par traitement logiciel mais conjointement par traitement logiciel et matériel.

La Figure I.7 met en évidence l'évolution entre une architecture classique superhétérodyne et une architecture RLR d'un système de radiocommunication multistandards [6].

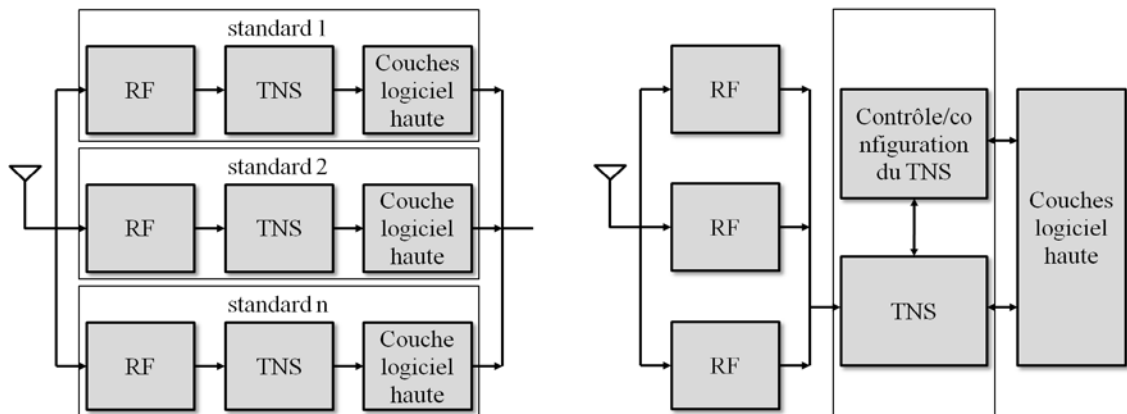


Figure I.7 : Architecture superhétérodyne et RLR d'un système de radiocommunications multistandards

La principale évolution de la RLR, par rapport à un système radiocommunication classique, vient de la partie Traitement Numérique du Signal (TNS) qui est reconfiguré selon le standard utilisé à un instant donné, et en un lieu donné. Pour contrôler et configurer la chaîne de TNS, un module de gestion commun aux différents standards est nécessaire.

I.7 Avantages

La radio logicielle est une évolution logique des systèmes de radiocommunication, elle apporte de nombreux avantages pour l'ensemble des acteurs du domaine [4].

La radio logicielle constitue tout d'abord un avantage pour les fabricants et équipementiers, grâce à la disparition de certains composants analogiques coûteux et encombrants (filtres, oscillateurs, etc.), ainsi que la possibilité d'implémenter une famille de produits " radio " sur une unique plateforme matérielle et d'utiliser un même code logiciel dans différents équipements [4].

- La radio logicielle offre au plus grand nombre la capacité (légitime ou non) d'étudier divers protocoles et communications sans fil à l'aide d'équipements abordables techniquement et financièrement [4].
- performances accrues en réception par rapport à des technologies radio traditionnelles possibilité de mettre à jour logiciellement les BTS pour supporter de nouvelles modulations (exemple : migration de la norme GSM à la norme EDGE) [2].
- Adaptation à son environnement en radio intelligente.

- Une partie matérielle réduite au strict minimum : TRX très petits (hors ordinateur)
- Un filtrage numérique puissant, souple, évolutif.
- L'implémentation par software de différents types de modulations et standards avec un seul équipement hardware.
- La possibilité d'enregistrer toute une bande et rejouer tout le trafic qui s'y est écoulé et pas uniquement une fréquence, comme classiquement.
- Une linéarité importante de la chaîne de réception (Une fois calibré avec un générateur, on dispose d'un S-mètre pour une fois précis au dBm près (jusqu'à saturation de la carte son, il y a bien évidemment des limites à tout !))
- La possibilité et le plaisir de réaliser soi-même un récepteur qui égale voire surpasse les équipements commerciaux.

I.8 Inconvénients de SDR

- Latence (Particulièrement gênante en conversion indirecte)
- Fréquence image (Avec la conversion indirecte du signal RF vers la bande de base).
- Coût supplémentaire par le besoin d'un PC (A ajouter au coût total du TRX).
- Limitations dues à certains composants pas encore disponibles à un coût raisonnable [2].

I.9 Domaine d'application

La radio logicielle a une importance dans une plusieurs domaines, parmi ses domaines d'applications :

I.9.1 Réseaux militaires : avec la radio logiciel les paramètres de la communication sans fil peuvent être adaptés de manière dynamique en fonction de temps.

I.9.2 Réseaux d'urgence : les réseaux de sécurité publique et d'urgence peuvent profiter des concepts de la radio pour fournir la fiabilité et la flexibilité de communication sans fil.

I.10 Conclusion

L'utilisation de radio logicielle dans les systèmes de radiocommunication est en plein essor. Cette technologie possède des avantages économiques importants : elle permet de réaliser rapidement et à moindre coût les développements nécessaires au support des nombreux protocoles existants ainsi que de ceux en cours de spécification.

L'amélioration des architectures matérielles dédiées aux applications de radio logicielle ainsi que la mise en place de standards internationaux promet à cette

technologie un vaste champ d'utilisation (outils de métrologie, équipement de télécommunications, systèmes tactiques . . .).

Les évolutions à venir de la radio logicielle prenant en compte les limitations des plates-formes actuelles en feront certainement un outil de plus en plus incontournable dans l'analyse des réseaux sans-fil.

Dans le prochain chapitre nous allons présenter un dispositif dédié initialement à la DVB pour la radio et la télévision, qu'il est possible de reconfigurer à l'image des systèmes de la radio logicielle.

Chapitre II : Architecture et fonctionnement d'un récepteur

RTL-SDR

II.1. Introduction

La technologie radio logicielle a connu des améliorations importantes ces dernières années. Il est possible maintenant de réaliser des plates-formes à faible coût tel que les réalisations de SDR radioamateur. Des plateformes SDR sont également vendues par différentes entreprises spécialisées dans la conception d'équipements d'instrumentation, de test ou autre.

Dans le cadre de projets de recherche, plusieurs plateformes radios logicielles expérimentales ont pu voir le jour. On distingue deux catégories de dispositifs :

- Dispositifs spécifiques : conçus et développés pour répondre à un besoin particulier, relativement peu flexibles et souvent coûteux. On retrouve notamment dans cette catégorie les équipements de test et de métrologie (5G, SDR d'industriels...) et les plateformes de recherche à gros budget, mais également certaines fabrications artisanales visant une norme de radiocommunication ou un protocole particulier.
- Dispositifs flexibles, généralement plus accessibles financièrement, mais avec une précision moyenne et des performances contenues. On retrouve notamment dans cette catégorie des montages « libres », et des plateformes de recherche.

Dans ce chapitre nous allons aborder une étude sur les différentes plateformes SDR. Une attention particulière sera accordée à la plateforme RTL-SDR en présentant son architecture ainsi que son mode de fonctionnement.

II.2 Présentation

II.2.1 La radio logicielle et ses différents usages

II.2.1.1 Les radios logicielles tactiques

Les militaires ont des besoins évidents de radios logicielles. Une radio logicielle peut changer son codage, sa modulation et/ou son chiffrement en fonction du besoin, ce qui lui donne un avantage important sur le champ de bataille. Les radios logicielles sont considérées comme un élément clé de la communication radio dans le domaine tactique de la « guerre réseau centrée » (NCW - *Network Centric Warfare*).

Le concept de radio logicielle offre une capacité lors d'opérations interarmées et interalliées où les unités combattantes peuvent utiliser différents types de systèmes de communication sans fil implémentant ou non de nouvelles technologies de transmission dans les bandes HF à UHF. Par ailleurs, les radios logicielles permettent des transmissions haut débit à l'aide de formes d'onde spécifiques (WNW – *Wideband Networking Waveform*) [3].

II.1.1.2 La radio intelligente (cognitive radio)

La radio intelligente s'appuie sur le fondement de la radio logicielle, à savoir la représentation algorithmique et le traitement des signaux radio, pour obtenir des informations sur l'environnement proche afin d'améliorer ses performances radio. La radio intelligente apporte de nouvelles capacités aux radios logicielles tel que la gestion fine des ressources, des protocoles réseau, prestations de services et certification des téléchargements. Pour être intelligente, une radio doit d'abord être étudiée son état est définie à l'aide d'informations qui lui sont ajustées par des capteurs.

La possibilité de reconfigurer un réseau de communication en fonction de son environnement est la clé de voûte de la radio intelligente. Les radios logicielles apportent de la flexibilité au niveau des blocs de communication utilisés alors que les radios intelligentes évaluent l'environnement d'exploitation afin de maximiser leurs performances par rapport à un (ou plusieurs) objectif(s) de communication [7].

❖ La relation entre la radio cognitive et la radio logicielle

Bien que de nombreux modèles différents soient possibles, l'un des plus simples modèles conceptuels qui présente la relation entre la radio cognitive et la radio logicielle restreinte est illustré dans la figure II.1. Dans ce modèle, les éléments de la radio cognitive entourent le support radio logicielle restreinte.

la partie chargée de l'optimisation ou du contrôle du module radio logicielle restreinte représentée par Le “**moteur cognitive**” en se basant sur quelques paramètres d'entrée tels que les informations issues de la perception sensorielle ou de l'apprentissage de l'environnement radio, du contexte utilisateur, et de l'état du réseau. La figure suivante résume la relation entre la radio intelligente et la radio logicielle [7].

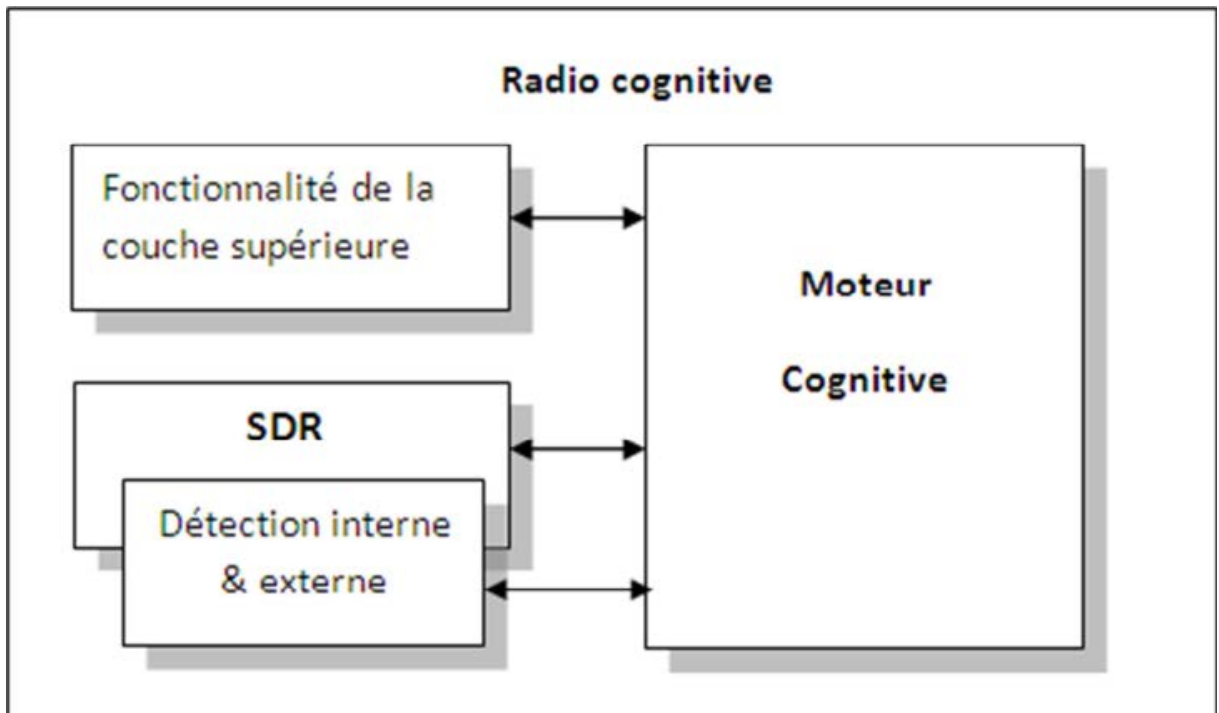


Figure II.1 la relation entre la radio cognitive et radio logicielle restreinte

II.2.1.3 Radio amateurisme

Des radioamateurs ont mis au point leur propre technologie radio logicielle au cours de ces dernières années. Une configuration typique utilise un récepteur à conversion directe basé sur la détection et l'échantillonnage de signaux en quadrature. Les performances du récepteur sont directement liées à la dynamique des convertisseurs analogique/numérique (CAN) utilisés. Les signaux radiofréquence sont transposés dans la bande de fréquences audio, puis échantillonnés par des CAN à hautes performances. Une première génération d'équipements utilisait une carte son de PC pour fournir des fonctionnalités de CAN. La dernière génération utilise des CAN embarqués plus performants, offrant une meilleure dynamique d'entrée et une plus grande résistance au bruit et aux interférences RF.

Un PC performant effectue ensuite les opérations de traitement numérique du signal à l'aide d'un logiciel spécifique capable d'effectuer tout type de modulation/démodulation, filtrage, amélioration du signal, etc., ouvrant la voie à de multiples expérimentations[3].

II.2.1.4 RTL-SDR

Le RTL-SDR est une plateforme radio logicielle extrêmement bon marché (environ 200DZD), basée sur des d'ongles de récepteur USB DVB-T TV (Digital HDTV) avec la puce RTL2832U. Il a été découvert par le pirate informatique Eric Fry, le développeur de pilotes Linux Antti Palosaari et l'équipe d'Osmocom qui développaient leur propre SDR que la puce RTL2832U avait un mode qui permettait le fonctionnement comme plateforme SDR [8].

II.3 Architecture du dongle RTL-SDR

II.3.1 Diagramme bloc d'un RTL-SDR

La figure II.2 [8] représente un schéma fonctionnel d'un dongle RTL-SDR qui peut être connecté par le port USB à un PC.

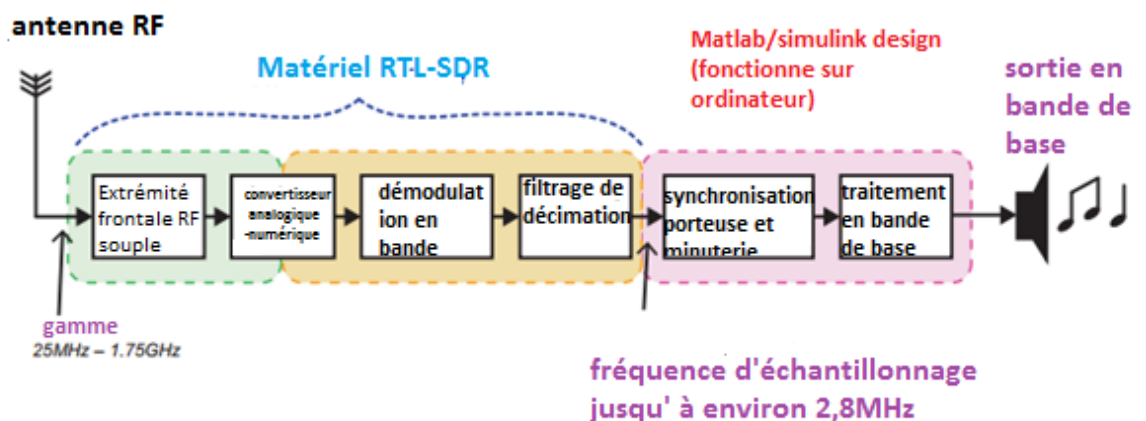


Figure II.2 : diagramme bloc du récepteur RTL-SDR

Les signaux RF sont reçus à l'antenne, les signaux sont convertis par le dongle RTL-SDR, et des échantillons In Phase/Quadrature Phase (IQ) sont transmis à l'ordinateur connecté, exécutant un logiciel de traitement à l'image de MATLAB. La conception du récepteur est mise en œuvre à l'aide des algorithmes DSP appropriés pour démoduler le signal en bande de base et extraire le signal d'information. Il peut s'agir de signaux audio, vidéo, images ou de données quelconques.

II.3.2 Le chipset RTL2832U

II.3.2.1 Définition

Le RTL2832U est un démodulateur DVB-T COFDM haute performance qui charge le mode 2K ou 8K avec une bande passante de 6, 7 et 8 MHz [9]. Les modulation, par exemple le débit de code et l'intervalle de garde, sont automatiquement détectés. Ce chipset prend en charge les tuners à fréquence intermédiaires IF : 36,125 MHz, 4,57 MHz ou Zéro-IF à l'aide d'un cristal 28,8 MHz. Ceci lui permet de rendre en charge la radio FM les transmissions DAB et DAB+. Intégré avec un ADC avancé (convertisseur analogique-numérique), le RTL2832U offre une grande stabilité en réception.

Cette puce peut être convertie pour réaliser de la radio logicielle restreinte (RLR) à faible couts puisqu'il permet de transmettre les échantillons bruts I (In-phase) et Q (Quadrature-phase) à la station de travail connectée. Les échantillons I/Q en sortie du dongle RTL-SDR sont codés sur 8bits avec une fréquence d'échantillonnage théorique de 3.2MHz. Néanmoins, la meilleure fréquence d'échantillonnage limitant fortement les pertes est de 2.56MHz. Généralement, on utilisera une fréquence d'échantillonnage de 2.048MHz [10].

II.3.2.2 Le principe du chipset RTL2832U

Le RTL2832U est un démodulateur COFDM, qui fait des traitements :

- synchronisation de symboles.
- Réglage fin de fréquence.
- Rotation de phase.
- Estimation et correction du canal.
- Désentrelocement interne et externe.
- Décodage de **Viterbi**.
- Décodage RS et correction d'erreur vers l'avant.
- Rejet d'interférence de canal adjacent et Co-canal.
- Annulation de bruit d'impulsion.
- Correction de la fréquence d'échantillonnage.

II.3.2.3 L'architecture du chipset RTL2832U

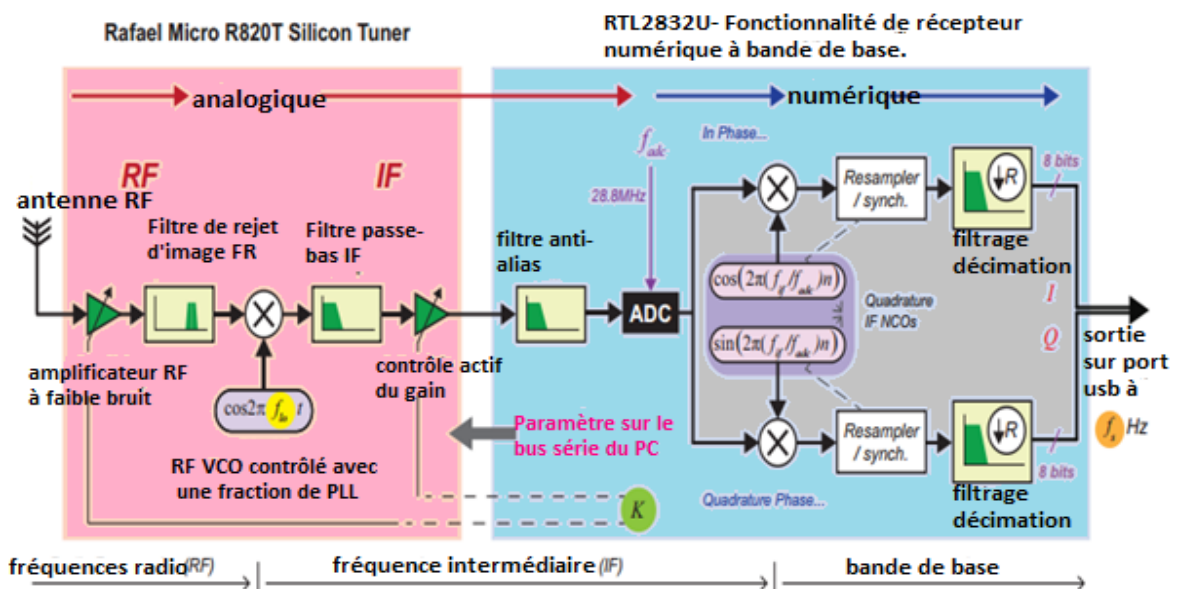


Figure II.3 : schéma fonctionnel montrant l'architecture interne du R820 T/R2832U RTL-SDR [8]

La figure II.3 montre un diagramme de flux de traitement du signal des principales étapes effectuées sur le RTL SDR. Les signaux RF entrant dans le tuner sont convertis vers le bas en une faible fréquence intermédiaire FI à l'aide d'un oscillateur commandé en tension (VCO).

Le VCO est programmable et est contrôlé par le RTL2832U via une interface de circuit intégré (I2C). Après une phase de contrôle actif du gain (AGC), qui ajuste dynamiquement l'amplitude du signal d'entrée en fonction de la plage de fonctionnement de l'appareil, le signal FI doit ensuite être ramené à la bande de base [11].

La méthode classique consiste à faire passer le signal FI à travers un filtre anti-alias, échantillonner la sortie avec un ADC, puis le convertir en bande de base à l'aide d'oscillateurs à commande numérique en quadrature (NCOs) (c.-à-d. un sinus et un cosinus oscillant à la fréquence FI).

Lorsque le RTL 2832U fonctionne normalement les données en bande de base IQ seraient démodulées DVB-T et un flux vidéo MPEG2-TS (Moving Picture Experts Group Transmission Stream) est émis sur l'interface USB. Toutefois, lorsqu'il est en "mode test", l'étape finale de démodulation est ignorée les données en bande de base sur 8 bits sont émises sur l'interface USB.

II.3.2.4 Caractéristique du chipset RTL2832U

- COFDM conforme à NorigUnified 1.0.3, D-book 5.0 et ETSI 300-744
- Mode de transmission automatique et détection d'intervalle de garde
- Circuits d'annulation de bruit impulsif
- Récupération automatique de porteuse sur un large écart ($\pm 800\text{KHz}$)
- Performances supérieures avec profils d'écho pré / post / long
- Circuit de rejet d'interférence adjacent et Co-canal intégré
- AGC retardé avec point de transfert programmable (TOP)
- ADC 8 bits pour la mesure de niveau des signaux RF
- Filtres PID MPEG-2 matériels
- Huit ports d'E / S à usage général
- Interface USB 2.0

II.4 Les Caractéristiques du technique RTL-SDR

II.4.1 Le convertisseur analogique numérique RTL-SDR (ADC)

ADC est un acronyme pour "Analog to Digital Converter". C'est une micro puce qui numérise le signal analogique en entrée. Plus un ADC possède de bits, plus la numérisation n'est précise. Par exemple, un ADC 8 bits peut mettre à l'échelle la tension d'entrée analogique en valeurs comprises entre -127 et +127, tandis qu'un ADC 12 bits peut mettre à l'échelle la tension d'entrée de -2047 à +2047[12].

Avec un faible nombre de bits, certains petits détails de l'entrée analogique, tels que des signaux faibles, peuvent être perdus lors de la numérisation. Ceci est particulièrement vrai s'il y a des signaux forts et faibles à proximité. Le RTL-SDR dispose d'un ADC 8 bits, qui est assez faible, mais juste assez grand pour donner des performances décentes.

La plage dynamique est la plage entre les valeurs les plus grandes et les plus petites possibles. La plage dynamique d'un ADC peut être calculée approximativement avec : nombre du bits * 6 dB. Cela donne au RTL-SDR une plage dynamique d'environ 50 dB. Cependant, la plage dynamique peut être légèrement améliorée grâce à l'astuce de sur échantillonnage qui peut être effectuée par logiciel.

II.4.2 Bande passante RTL-SDR

La bande passante maximale du RTL-SDR est de 3,2 MHz, mais la plus grande bande passante stable est de 2,4 MHz ou 2,8 MHz. Le réglage d'une bande passante trop large peut entraîner la perte d'échantillons.

La plupart des logiciels compatibles RTL-SDR nous permettent de choisir notre bande passante qui est parfois appelée aussi fréquence d'échantillonnage. Bien que la fréquence d'échantillonnage et la bande passante ne soient pas les mêmes, dans le RTL-SDR régler la fréquence d'échantillonnage à 2 Msps (Megasamples per second) nous donne 2 MHz de bande passante.

En le réglant à 2,8 Msps, nous obtenons une bande passante de 2,8 MHz. c'est parce que le RTL-SDR utilise l'échantillonnage I/Q avec deux ADC) [12].

II.4.3 Impédance d'entrée

Comme les dongles RTL2832U sont conçus pour être utilisés avec du matériel TV, ils ont tous une impédance d'entrée de 75 Ohms. La plupart des équipements radio amateurs et professionnels fonctionnent avec un câblage de 50 Ohm, des connecteurs et des adaptateurs [12].

II.5 Les interfaces les plus connus pour le dongle RTL-SDR :

- SDR#
- HDSDR
- GNU Radio
- SDR-console (de SDR-Radio. com.)
- SDRUNO

De nombreux autres programmes, y compris des programmes spécialisés

Certains s'appuient sur l'installation de SDR#, d'autres sur une installation spécifique.

II.6 Domaine d'application

Un SDR à large bande ouvre de nombreux projets intéressants. Certaines applications du RTL-SDR incluent les applications suivantes, dont certaines seront discutées plus en détail dans les chapitres suivants

- écouter les conversations hertziennes non cryptées.

Chapitre II : Architecture et fonctionnement d'un récepteur RTL-SDR

- Regarder la télévision analogique.
- renifler les signaux GSM.
- l'utilisation de RTL SDR sur votre appareil Android comme scanner radio portable.
- Réception et décodage des signaux GPS.
- Utiliser le dongle RTL SDR comme analyseur de spectre.
- Réception d'images satellites météorologiques de la NOAA.
- L'écoute des satellites et de l'ISS.
- Radioastronomie.
- Surveillance de la diffusion des météorites.
- Écoute de la radio FM et décodage des informations RDS

II.7 Conclusion

Nous avons présenté dans ce chapitre les différents concepts liés à des notions importantes concernant la radio logicielle qui est un système capable d'analyser le comportement spectral de son environnement dans le but de déterminer les caractéristiques de celui-ci, de détecter les ressources fréquentielles disponibles. Celui-ci doit alors adapter ses caractéristiques et son comportement en fonction de l'ensemble des paramètres externes qu'il a détectés et les besoins de son utilisateur. La radio logicielle doit alors réagir d'une façon autonome et instantanée.

Comme nous l'avons présenté le dongle basé sur une puce RTL2832U permet de capter les signaux hertziens à sa portée. Dans le chapitre suivant nous allons expliquer comment nous avons mis en service le dongle RTL-SDR.

**Chapitre III : Installation et configuration du logicielle SDR
Sharp pour le dongle RTL-SDR**

Chapitre III : installation et configuration du logicielle SDRSharp pour le dongle RTL-SDR

III.1 Introduction

SDR est un récepteur radio qui converti ce qu'il reçoit en un signal numérique qui peut être traité de plusieurs façon au niveau du dispositif logiciel. Dans le cas du dongle RTL-SDR les données sont reçues via un simple bus USB. Cependant il est bien que le fonctionnement parait simple il faut configurer correctement le dongle RTL-SDR afin de pouvoir lire la bande désirée. Ce chapitre aborde la mise en service que nous avons effectué pour faire fonctionner le dongle RTL-SDR et la manière de l'interfacer avec le logiciel de traitement.

III.2 Matériel utilisé

Dans notre travail nous avons utilisé une version pas chère du dongle RTL-SDR connecté à une antenne omnidirectionnelle. Il est important de connecter avec le dongle RTL-SDR une antenne adéquate, accorder sur les bandes de fréquences à écouter. Une station de travail assez rapide ; processeur i3 3GHz, RAM 4 Go, ports USB 2.0 avec WINDOWS7 est utilisée pour l'acquisition des échantillons et le traitement.



Figure III. 1 : Dongle RTL-SDR branché sur PC

III.3 Installation et mise en Service du dongle RTL-SDR

Sous Windows 7 et 10 il est important d'installer les drivers et les logiciels SDR en mode administrateur.

Chapitre III : installation et configuration du logiciel SDRSharp pour le dongle RTL-SDR

Notre première mise en service fait intervenir les deux logiciels :

- Le logiciel [Zadig](#) pour Windows.
- Le logiciel [SDR Sharp](#) pour Windows 64 bits (X86)

En premier il est important d'installer Microsoft .NET redistribuable pour configurer le système à base de la RTL-SDR. Si dans les PC Windows moderne le .NET est installé par défaut, certain PC peuvent avoir besoin d'une version récente du .NET qui peut être téléchargée à partir du lien :

« <http://www.microsoft.com/engb/download/details.aspx?id=21> »

Les drivers sont fournis par le package [sdr_install.Zip](#) Téléchargeable à partir du <http://zadig.akeo.ie>

III.3.1 Configuration

Une fois le dongle RTL-SDR branché, tel que montré sur la figure III-1, il est important de le configurer en installant les drivers adéquats. Le driver utilisé sous Windows est installé à partir de l'outil Zadig.

Etape 1 :

L'outil Zadig est utilisé pour chaque port USB connecté avec un dongle RTL-SDR, en cas de changement de port USB, il est important de réexécuter Zadig pour faire la mise à jour du driver à nouveau.

Etape 2 :

Le menu sur “*Options*“, tel que montré par la figure-III .2, permet d'activer le type des devices, il est important de cocher “*List all devices*“

Chapitre III : installation et configuration du logiciel SDRSharp pour le dongle RTL-SDR

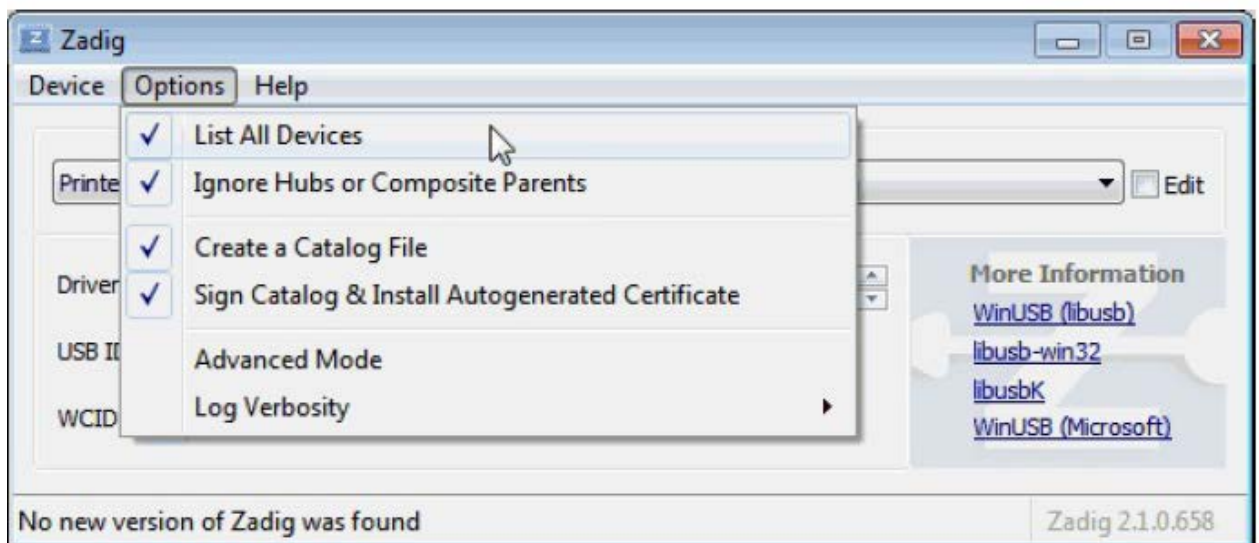


Figure III.2 : Les paramètres de Zadig

Etape 3 :

Tel que le montre la figure-III .3, nous Sélectionnons dans le menu déroulant “Bulk-In, Interface (Interface 0)” ou quelque chose comme “RTL2832U”. Par la suite nous avons Lancé la mise à jour du driver Win USB en cliquant sur “Replace drivers”

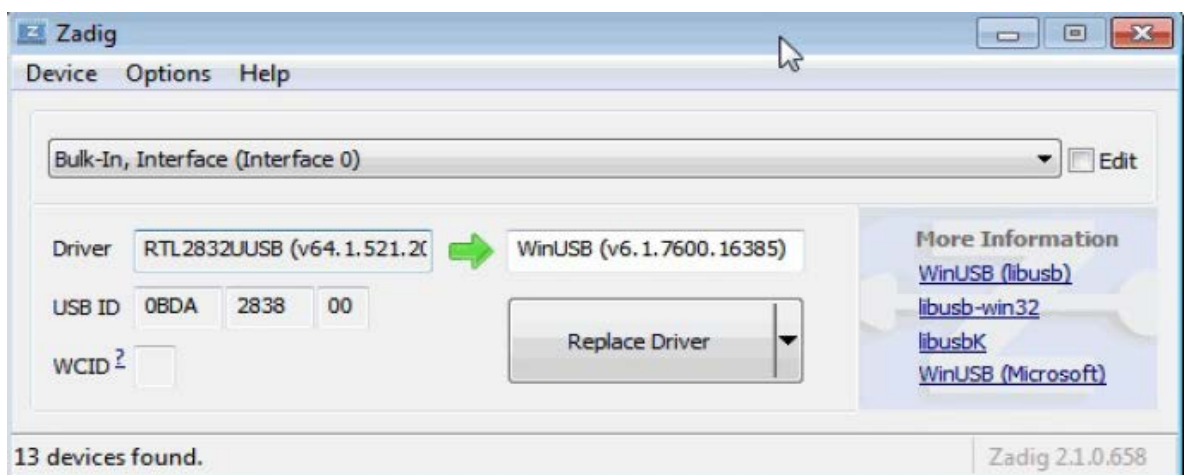


Figure III. 3 : La mise à jour du driver

Chapitre III : installation et configuration du logicielle SDRSharp pour le dongle RTL-SDR

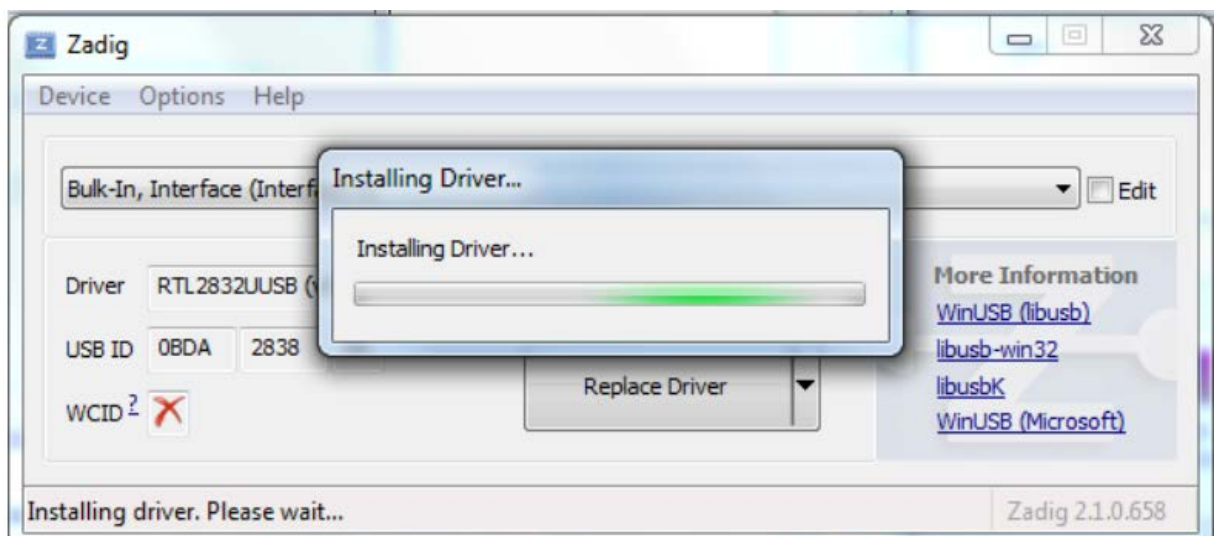


Figure III. 4 : installation du l'outil Zadig

III.3.2 Test du fonctionnement :

Le test du bon fonctionnement du dongle nous avons utilisé l'application SDRSharp.exe en mode administrateur.

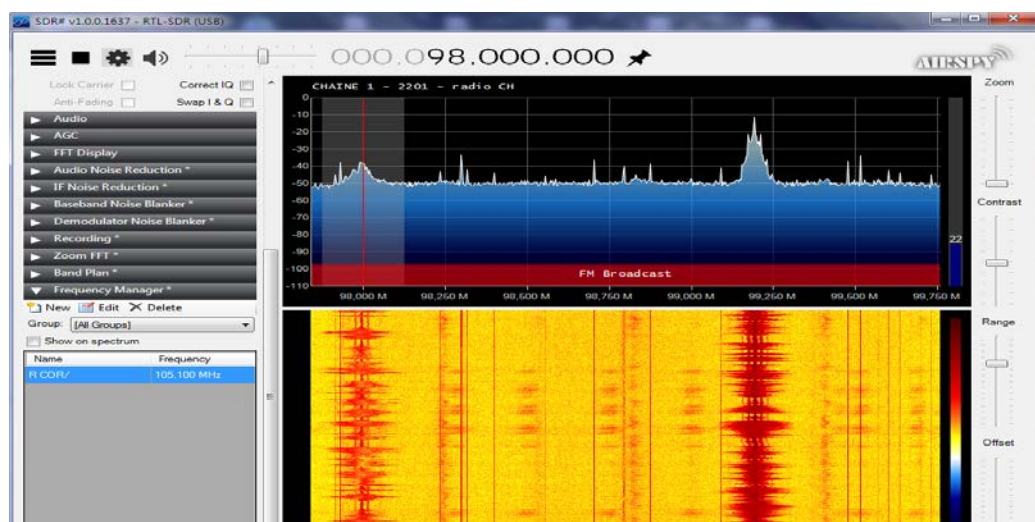


Figure III.5 : Exemple d'exécution de l'application SDRSHARP

La figure-III.5 montre un exemple d'exécution de l'application SDRSHARP :

Chapitre III : installation et configuration du logiciel SDRSharp pour le dongle RTL-SDR

- dans le menu déroulant la source il faut choisir “RTL-SDR USB“, tel que précisé par la figure-III.6.

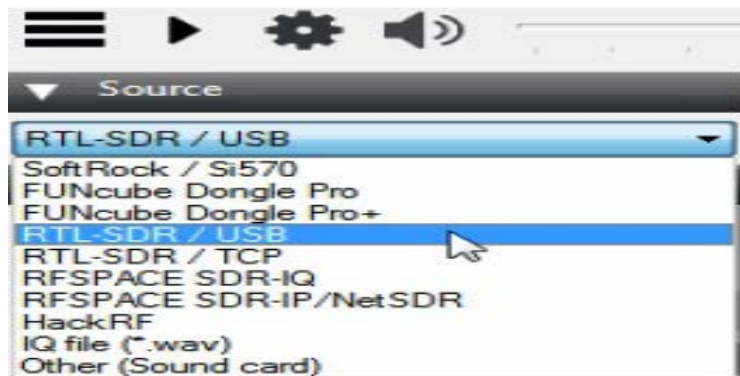


Figure III.6 : Les sources de logiciel SDR Sharp

- Définir le gain en réception (figure III.7).

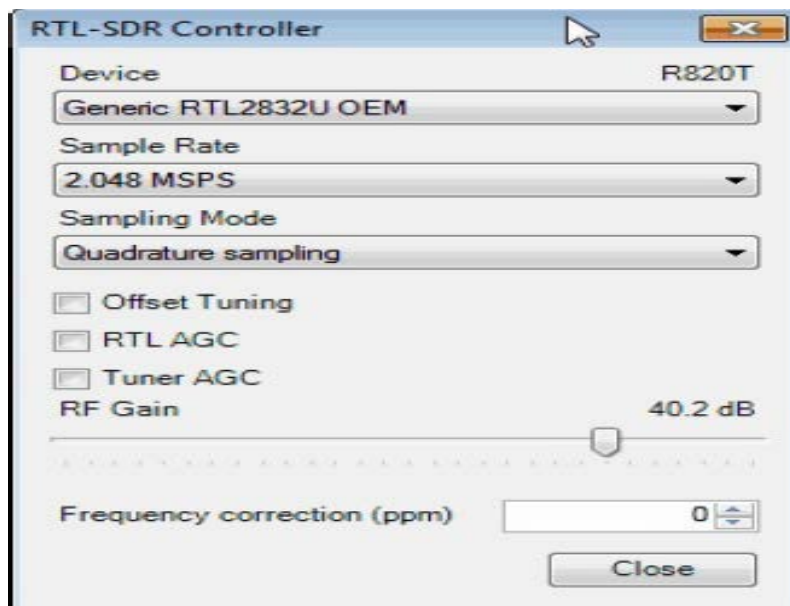


Figure III.7 : RTL-SDR Controller

- Définir la fréquence centrale à écouter, exemple : 88.4MHz et 108MHz tel que représenter par la figure III.8.

Chapitre III : installation et configuration du logicielle SDRSharp pour le dongle RTL-SDR

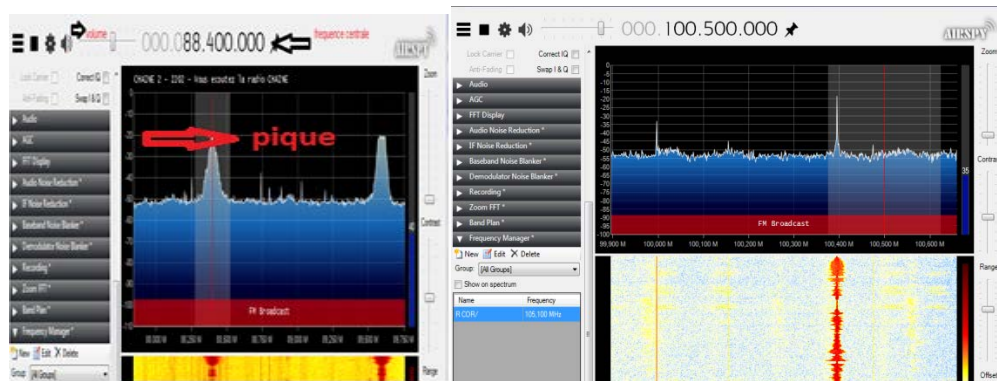


Figure III.8 : Exemple des chaines

- Cliquez sur “Play” pour démarrer l’acquisition

III.4 conclusion

Nous avons montré dans ce chapitre le mode de fonctionnement du dongle RTL-SDR, qui permet de visualiser l’activité spectrale allant de 30MHz à 1700MHz, sur une station de travail universelle (un simple PC portable). Pour cela un driver doit être installé afin de configurer correctement le dongle. Le logiciel SDRSharp nous permet d’effectués les écoutes des stations radios. Nous nous intéressant concevoir une solution à l’image de SDRSharp en exploitant directement les échantillons en quadrature fournis par le dongle.

**Chapitre IV : Développement d'un récepteur de détection de
l'activité spectrale**

Chapitre IV : Développement d'un récepteur de détection de l'activité spectrale

IV.1-Introduction

Après avoir présenté les outils associés avec RTL-SDR ainsi que le logiciel SDRSHARP et ses différents paramètres, nous sommes parvenus à l'importance de développer et proposer une solution SDR à base d'un autre environnement de développement tel que MATLAB & Simulink, un environnement qui nous permet d'implémenter les fonctions et algorithmes de traitement des signaux mathématiques et les systèmes sous forme de diagramme en blocs, et de simuler le fonctionnement de ces systèmes. Dans ce chapitre nous allons utiliser ce dernier environnement pour développer une solution dédiée au dongle RTL-SDR pour scanner les activités fréquentielle environnantes.

IV.2-Installation du package pour la radio RTL SDR

Avant de pouvoir travailler sur l'environnement MATLAB nous avons installé le package de support pour la radio RTL-SDR. Le support package permet d'utiliser la radio USB RTL-SDR en tant que périphérique autonome pour la réception de signaux RF en direct. Les étapes à suivre sont :

1. Dans l'onglet MATLAB → Accueil, dans la section Environnement, nous utilisons l'onglet Modules complémentaires « AddOns » → Obtenir des packages de support matériel «figure IV .1 ».

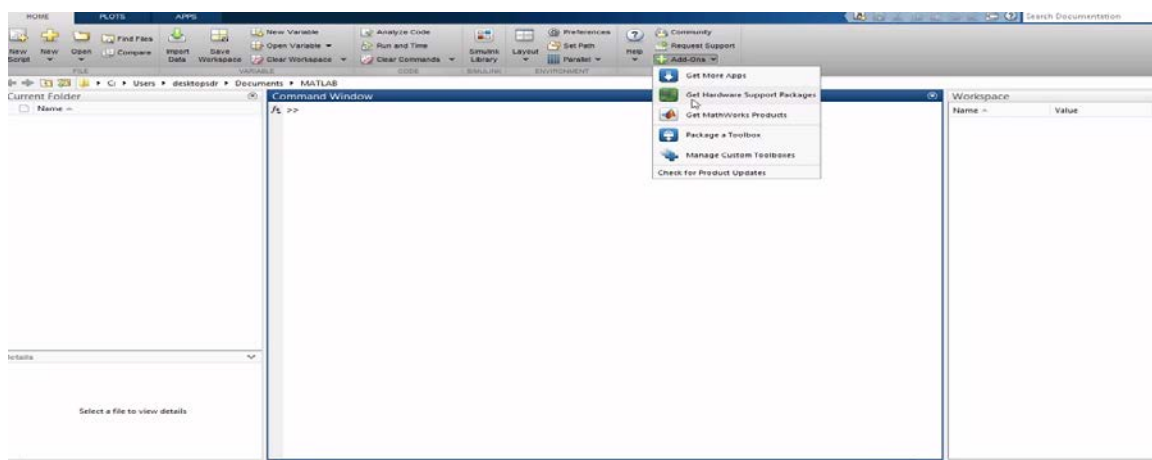


Figure IV.1 : Emplacement pour l'installation du package RTL-SDR

Chapitre I V : Développement d'un récepteur de détection de l'activité spectrale

Après avoir cliqué sur Install, nous avons choisi dans les supports le package RTL SDR radio comme il est illustré dans la figure IV-2 pour installer ce support.

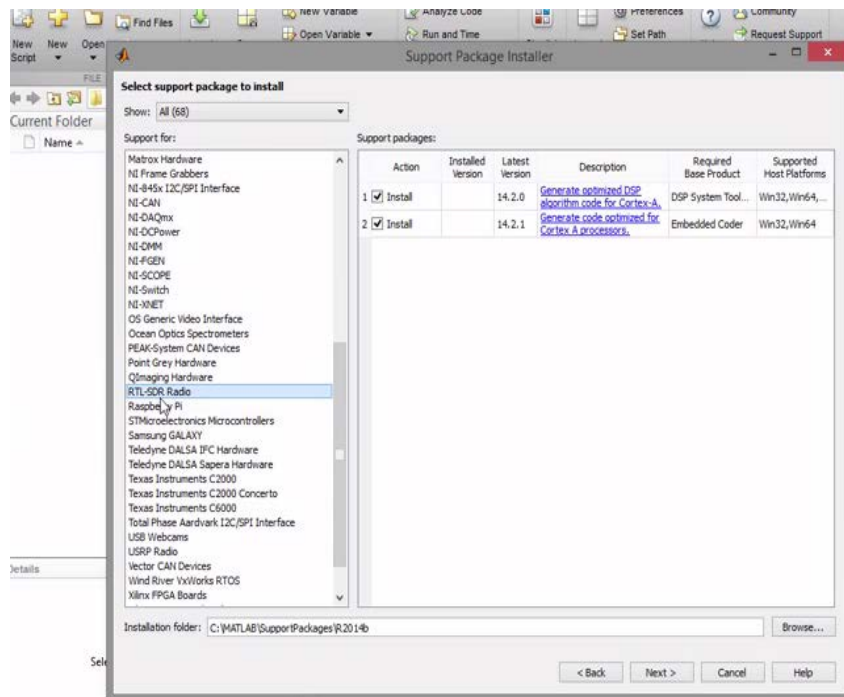


Figure IV-2 installation du package

Pour continuer l'installation il faut brancher le dongle comme il est illustré dans la figure IV.3 et cliquer suivant.

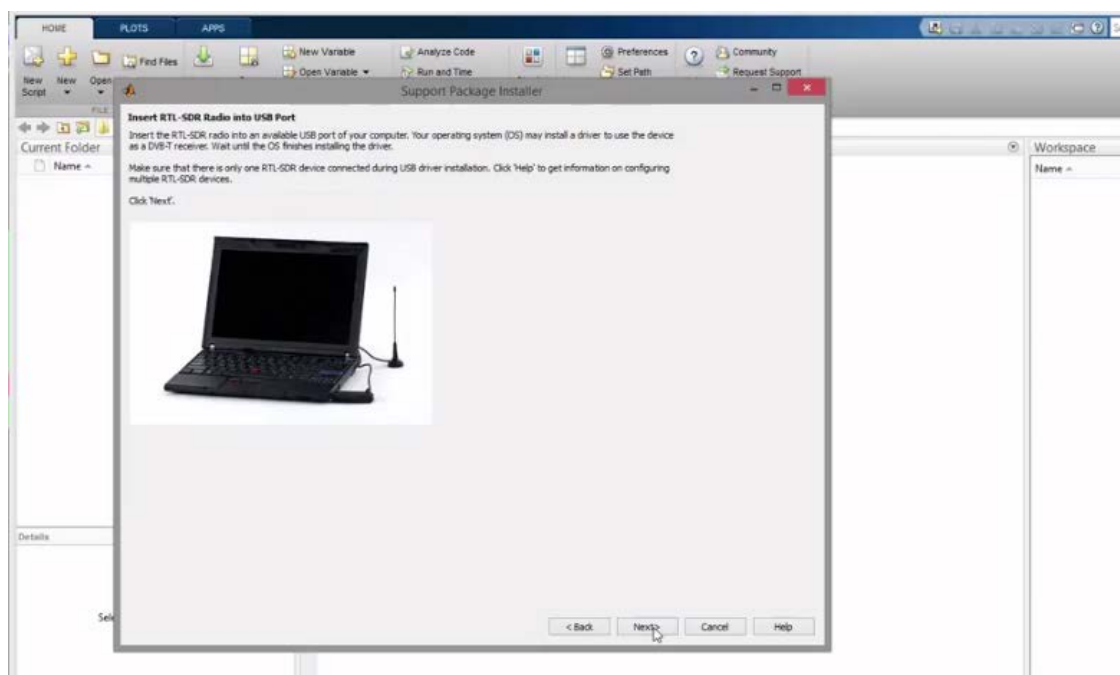


Figure IV.3 : le dongle USB RTL-SDR branche sur le PC

Chapitre I V : Développement d'un récepteur de détection de l'activité spectrale

Puis il faut compléter toutes les étapes pour terminer l'installation du package.

D'une autre part, nous avons besoin ouvrir simulink → library, après nous allons vérifier que le package est bien installer « figure IV. 4 ».

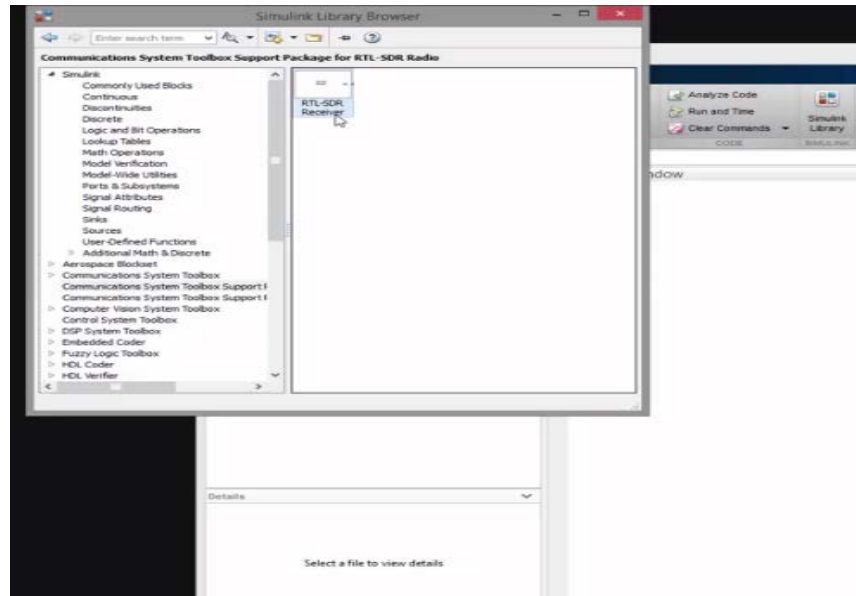


Figure IV.4 vérification d'installations du package RTL SDR

Si l'installation est réussite ; nous ajoutons dans la bibliothèque du Simulink notre « RTL SDR Library », comme il est illustre dans la figure IV.5

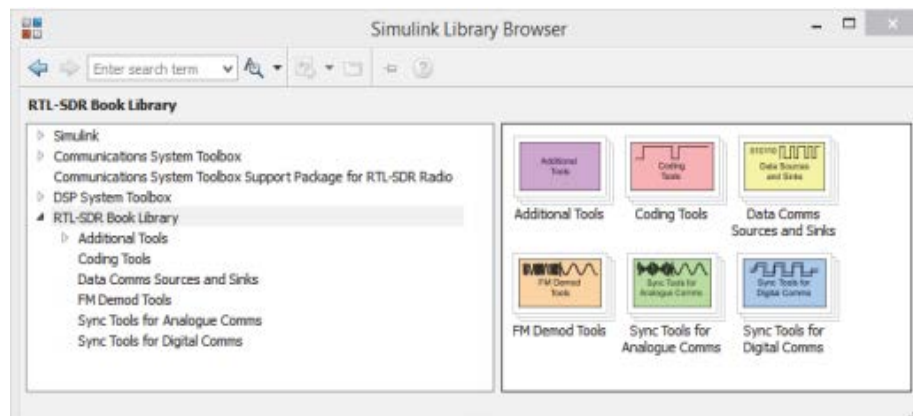


Figure IV.5 : la bibliothèque du MATLAB & Simulink

IV.3- Premier récepteur RTL-SDR sous Simulink

Après l'installation du package RTL SDR nous avons commencé de faire le premier pas dans Matlab & Simulink et dessiner le schéma bloc approprié pour un récepteur universel, comme il est illustre dans la figure IV. 6.

Chapitre IV : Développement d'un récepteur de détection de l'activité spectrale

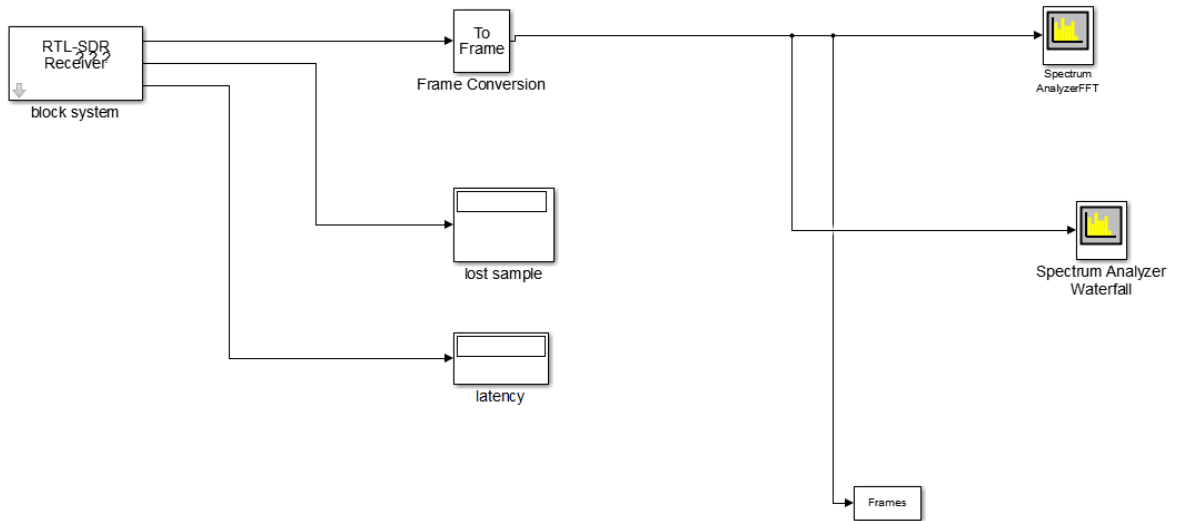


Figure IV. 6 le block système sous Matlab & Simulink

A l'intérieur du block système nous avons un autre schéma qui appartient au récepteur RTL-SDR et la fréquence centrale que nous avons spécifiée sa valeur par 98.0MHz figure IV.7.

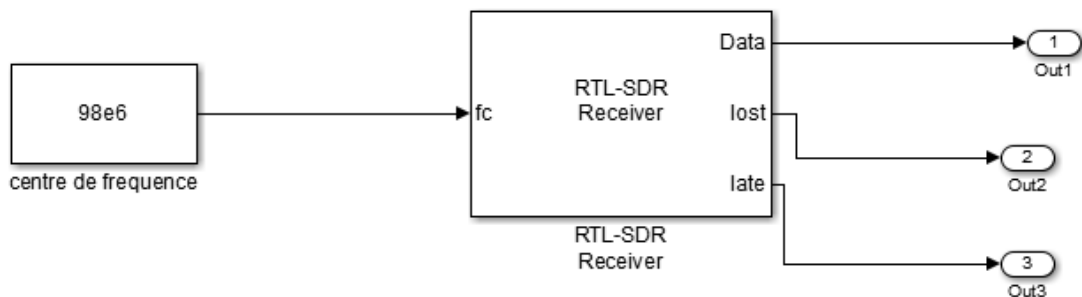


Figure IV.7 : les paramètres de bloc sub-system

IV.3.1 La solution proposée

Dans un premier temps, nous avons créé tous les blocs nécessaires qui sont proposés dans les figures précédentes. Le bloc récepteur RTL-SDR est important pour manipuler les échantillons. En effet, ce bloc matérialise le dongle connecté au pc, la sortie du bloc fournit directement les échantillons phase et en quadrature (I/Q) au format nombre complexe. Les échantillons sont regroupés en frame pour être présentés à l'analyseur de spectre.

IV.3.1.1 Les rôles des blocs du schéma réalisé

-Le bloc récepteur RTL-SDR: est considéré comme une source de signal qui reçoit les données directement à partir du dongle RTL-SDR et produit un signal de vecteur de colonne de longueur fixe indiqué par les échantillons par paramètre d'armature:

Nous avons ajouté tous les paramètres du notre récepteur RTL-SDR ; en choisissant un gain du 60 dB, une fréquence d'échantillonnage de 2.8MHz car elle est la valeur maximum qui peut capter le dongle RTL-SDR et une fréquence de correction égale à 0ppm (sans correction). Les frames sont élaborés à partir de 1024 échantillons.

1 fft=1024 échantillons

Dans la partie configuration de transfert de données nous avons coché les deux ports de sortie :

-échantillon perdu (lostsample) : demander au bloc de récepteur de RTL-SDR pour produire le nombre d'échantillons perdus pendant les transferts de données.

-latence (latency) : qui demande au bloc de récepteur de RTL-SDR d'indiquer la latence pendant les transferts de données en nombre de frames.

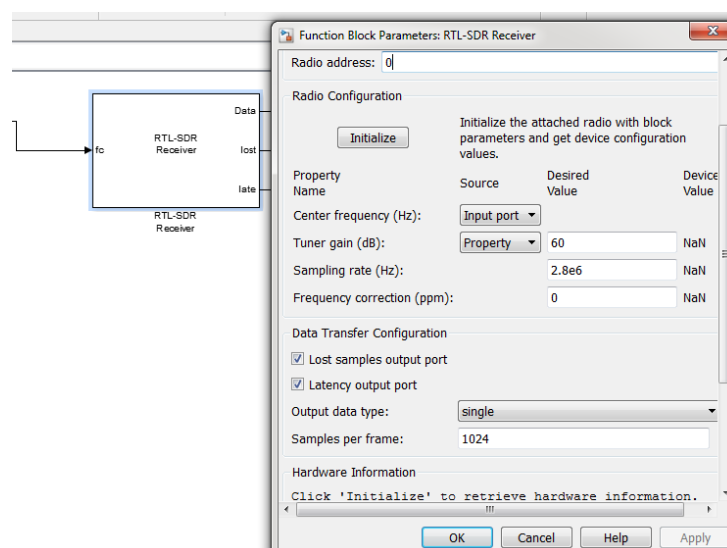


Figure IV.8 les paramètres du récepteur RTL SDR

Chapitre I V : Développement d'un récepteur de détection de l'activité spectrale

Une fois l'insertion des paramètres terminé il faut vérifier que le dongle est bien connecter avec le PC et que l'environnement Matlab communique avec ce dernier en tapant la commande `my_rtlsdr= sdrinfo`« Figure IV.9 ».

```
>> my_rtlsdr = sdrinfo
my_rtlsdr =
    RadioName: 'Generic RTL2832U OEM'
   RadioAddress: '0'
  RadioIsOpen: 0
    TunerName: 'R820T'
  Manufacturer: 'Realtek'
    Product: 'RTL2841UHIDIR'
   GainValues: [29x1 double]
RTLCrystalFrequency: 28800000
TunerCrystalFrequency: 28800000
  SamplingMode: 'Quadrature'
  OffsetTuning: 'Disabled'
```

Figure IV-9 :l'information sur le dongle RTL-SDR

Pour faciliter notre travail nous avons proposé une solution pratique qui consiste à développer un **masque** qui est une interface utilisateur faite sur mesure pour plusieurs blocs en le faisant apparaître comme bloc atomique avec sa propre zone de dialogue d'icône et de paramètres.

Nous allons expliquer cette solution étape par étape :

IV.3.2 Le masque utilisé dans le schéma bloc

Étape 1 : Ouvrir l'éditeur de masque

1. nous avons ouvrir le modèle dans lequel nous souhaitons masquer un bloc.
2. nous et sélectionnons **Masque > Créer un masque** sous le bloc Sous-système

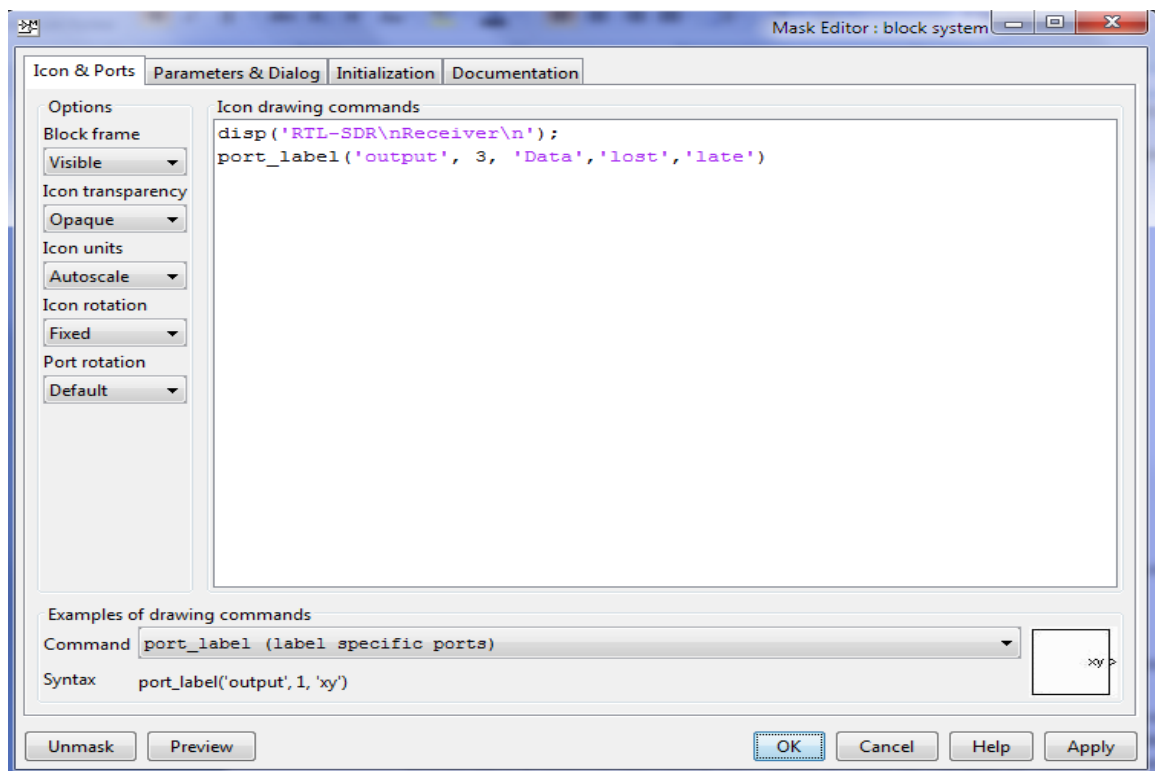


Figure IV-10 : éditeur de masque

Étape 2 : Définir le masque

L'éditeur de **masque** contient quatre onglets permettent de définir le masque du bloc et de personnaliser la boîte de dialogue du masque.

Nous allons renseigner tout d'abord les onglets « Icône et Documentation ». Nous verrons ensuite l'onglet « Initialisation » qui crée la boîte de dialogue et définit les variables de paramétrage du bloc masqué.

- **Icône et onglet Ports**

Nous avons utilisé cet onglet pour créer une icône pour le masque du bloc. Nous pouvons utiliser le volet **Options** à gauche pour spécifier les propriétés de l'icône et la visibilité de l'icône

- **Paramètres et onglet de la boîte de dialogue**

1. Nous avons tapé le nom de l'objet inséré et son titre dans les colonnes Prompt et Name, respectivement. La valeur de la colonne invite apparaît dans la boîte de dialogue du masque.

2. Dans le volet droit, sous éditeur des propriétés, nous fournissons les valeurs dans les sections Propriétés, boîte de dialogue et mise en page.

Chapitre IV : Développement d'un récepteur de détection de l'activité spectrale

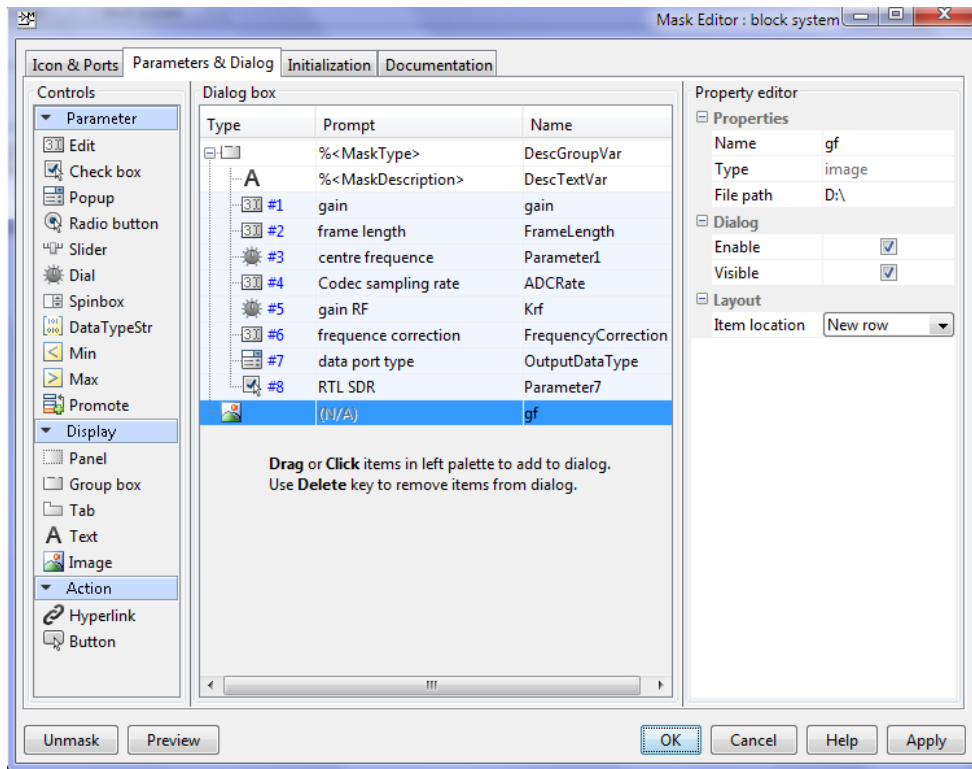


Figure IV-11 : Paramètres et onglet de la boîte de dialogue

- **Onglet Initialisation**

Nous avons utilisé ce volet pour spécifier le code MATLAB pour contrôler les paramètres du masque. Nous pouvons ajouter même des conditions pour les valeurs spécifiées par nous, particulièrement pour le gain.

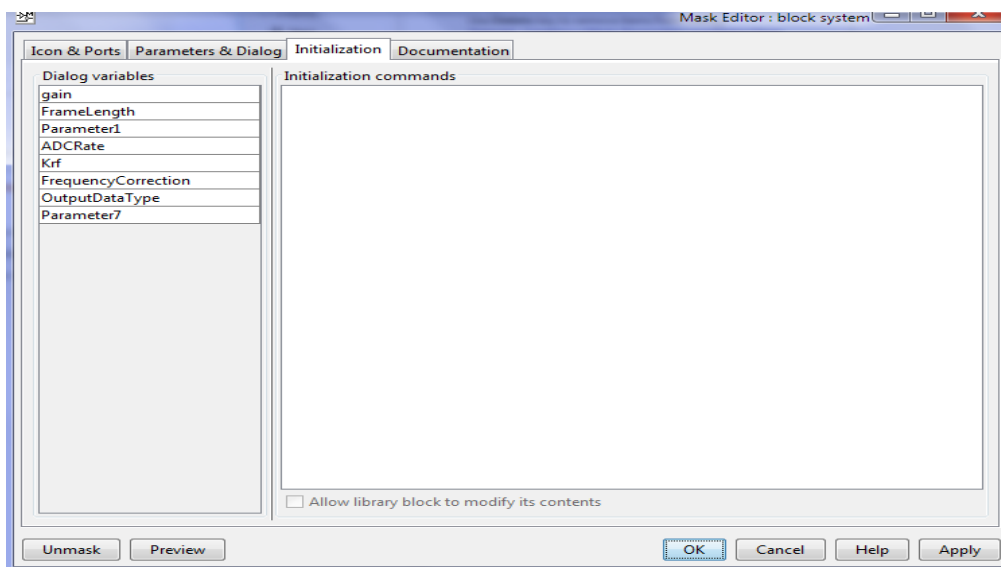


Figure IV.12 : Onglet Initialisation

Chapitre IV : Développement d'un récepteur de détection de l'activité spectrale

- **Onglet Documentation**

Nous avons utilisé cet onglet pour ajouter un nom, une description et des informations supplémentaires pour le masque.

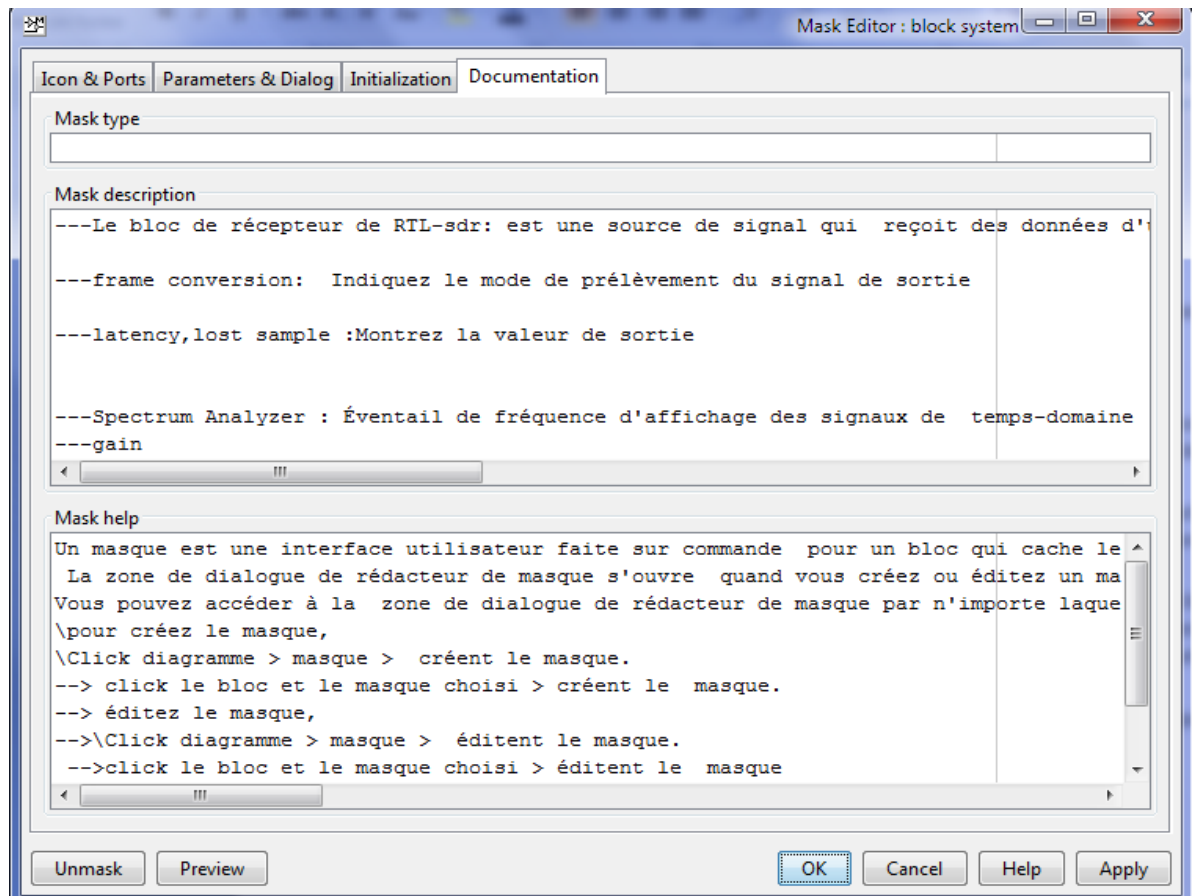


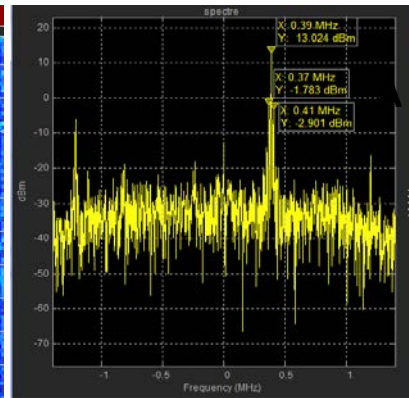
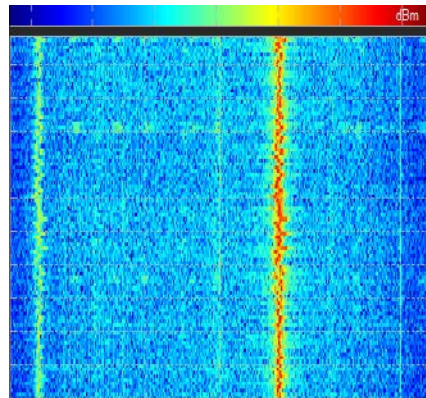
Figure IV-13 : Onglet Documentation

Une fois la simulation lancée, notre récepteur Simulink commence à visualiser les activités spectrales, La figure suivante exprime tous les résultats obtenus sur des bandes fréquentielles différentes en fixant la valeur de Gain a40dB :

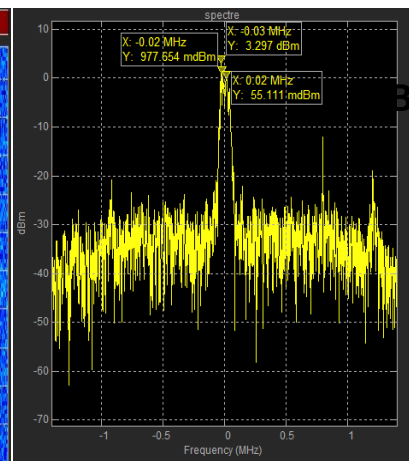
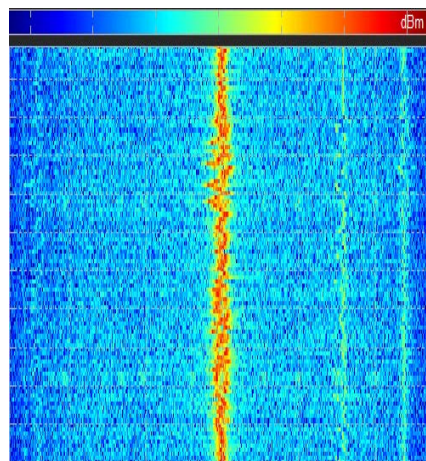
Chapitre I V : Développement d'un récepteur de détection de l'activité spectrale

analyseur de spectre Waterfall

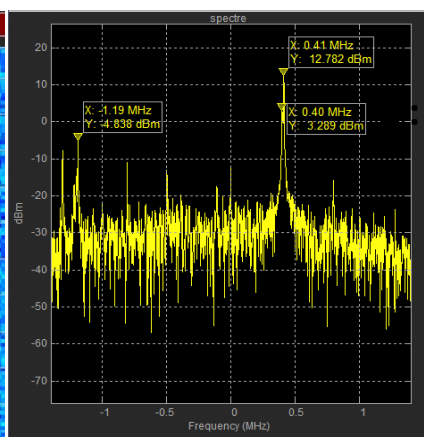
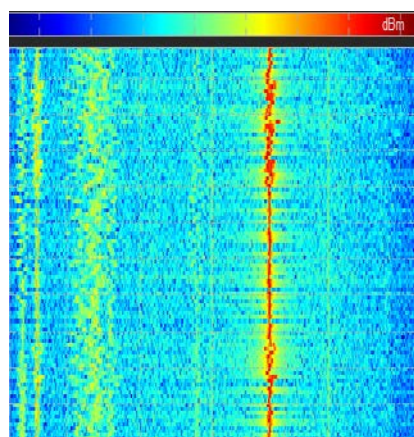
analyseur de spectre en FFT



fc=88MHz



fc= 98MHz



fc= 100MHz

Figure IV.14 les résultats obtenus sous simulink

IV.4 Programmation du Récepteur

Bien que Simulink soit simple d'utilisation il ne permet pas de traiter les concepts SDR au bas niveau. Nous avons développé un script qui permet de scanner l'activité spectrale en exploitant les théorèmes du traitement numérique du signal et le principe de la Densité Spectrale de Puissance DSP.

IV.4.1 Définition de la densité spectrale de puissance DSP

La densité spectrale de puissance $\gamma_X(f)$ d'un processus aléatoire stationnaire $X(t)$ est la transformée de FOURIER de sa fonction d'autocorrélation calculée par [13] :

$$\gamma_X(f) = \int_{-\infty}^{+\infty} \Gamma_{XX}(\tau) e^{-2\pi j f \tau} d\tau \quad (1)$$

$$\Gamma_{XX}(\tau) = \int_{-\infty}^{+\infty} \gamma_X(f) e^{2\pi j f \tau} df \quad (2)$$

IV.4.1.1-Propriétés

-la valeur à l'origine de la densité spectrale de puissance d'un processus aléatoire stationnaire est égale à l'aire sous le graphe de la fonction d'autocorrélation:

$$\gamma_X(0) = \int_{-\infty}^{+\infty} \Gamma_{XX}(\tau) d\tau \quad (3)$$

-la valeur moyenne du carré d'un processus aléatoire stationnaire est égale à l'aire sous le graphe de la densité spectrale de puissance:

Chapitre I V : Développement d'un récepteur de détection de l'activité spectrale

$$E \{ X^2(t) \} = \Gamma_{XX} (0) = \int_{-\infty}^{+\infty} \gamma_X(f)df \quad (4)$$

Cette relation est importante car elle indique que la puissance d'un signal aléatoire, qui n'est autre que $E\{X^2(t)\}$, peut se calculer par l'intégrale de la densité spectrale de puissance. C'est bien entendu à cette propriété fondamentale que la fonction $\gamma_X(f)$ doit le nom de densité spectrale de puissance.

-la densité spectrale de puissance d'un processus aléatoire stationnaire à valeurs réelles est une fonction paire de la fréquence:

$$\gamma_X(-f) = \gamma_X(f) \quad (5)$$

Remarque :

La transformée de FOURIER d'un signal aléatoire est elle-même aléatoire, et de ce fait, on ne peut pas la calculer. On peut par contre déterminer sa densité spectrale de puissance car cette fonction est déterministe -on suppose bien sûr la stationnarité au sens large ainsi que l'ergodicité de $X(t)$.

IV.4-2- initialisation des paramètres :

- **La section des paramètres**

On Inspecte les paramètres. Un certain nombre de variables sont initialisées dans la section 'Paramètres', avec des valeurs correspondant à celles réglées dans le bloc Récepteur RTL-SDR.tel qu'illustré dans la figure suivant.

Chapitre IV : Développement d'un récepteur de détection de l'activité spectrale

```
function rtl_sdr_rx_startup_matlab
%% Paramètres
rtl_sdr_id      = '0';           % IDENTIFICATION DE RTL-SDR
rtl_sdr_tunerfreq = 25e6;        % de RTL-SDR de fréquence de tuner en hertz
rtl_sdr_gain     = 40;           % de RTL- SDR de gain de tuner dans dB
rtl_sdr_fs       = 2.4e6;        % de RTL- SDR de taux de prélèvement en hertz
rtl_sdr_frmrlen  = 4096;        % de RTL- SDR de données de rendement encadrent la taille
rtl_sdr_datatype = 'single';    % de RTL- SDR de rendement de type de données
rtl_sdr_ppm      = 0;           % de RTL- SDR de pièces de tuner par million de correction
sim_time        = 5;           % de temps de simulation en secondes
```

Figure IV.15 : les paramètres initialisés dans le programme

Les valeurs définies dans figure IV.13 incluent la longueur de la trame de données de sortie «**rtl_sdr_frmrlen**», le type «**rtl_sdr_datatype**» qui nomme IQ et la durée d'exécution de la simulation «**sim_time**».

Alors, premièrement nous avons Examine tous les objets de système. Comme il s'agit d'un modèle purement "de démarrage" et qu'il n'y a que très peu d'éléments de base. À ce récepteur codé, seuls trois objets système sont initialisés. Le premier est pour le RTL-SDR, il utilise simplement les valeurs créées plus tôt lors du réglage de chacun des paramètres de l'objet. Le second est pour un analyseur Spectrum, le même type de portée qui était utilisé dans le récepteur Simulink. Il est configuré pour agir en tant qu'analyseur de spectre FFT. Troisièmement, un autre analyseur de spectre est initialisé, également configuré pour agir en tant que spectre Analyseur FFT, mais dans le style cascade.

- **la section des calculs.**

La valeur de **rtl_sdr_frmtime** est trouvée en calculant le temps qu'il faut pour traiter une trame de données, et ce temps est utilisé plus tard pour contrôler l'image et la durée de la simulation.

- **la section sur la simulation.**

La dernière section est l'endroit où la "simulation" est effectuée et où le récepteur est implémenté. Après avoir vérifié deux fois que MATLAB peut communiquer avec votre RTL-SDR, une boucle de temps est instanciée qui tourne en boucle pendant le nombre de secondes spécifié par l'attribut **sim_time**.

A l'aide de la fonction **step**, une trame de données est extraite du RTL-SDR.

Ensuite, elle est stockée dans la variable **rtl_sdr_data** qui est la quadrature/phase **IQ**. Le cadre est une matrice de longueur **rtl_sdr_frmrlen**, il contient 4096 échantillons par défaut.

Chapitre I V : Développement d'un récepteur de détection de l'activité spectrale

Utilisation de la fonction `step` encore une fois, permet transmise cette trame de données aux analyseurs de spectre, ce qui les incite à effectuer les opérations suivantes FFTs sur le nouveau lot de données et de mettre à jour leurs écrans.

Le compteur de temps d'exécution qu'il a boucle est mise à jour pour incorporer le temps de traitement associé à la démodulation une autre image du RTL-SDR.

Enfin, il faut vérifier que notre dongle RTL SDR est communiqué avec MATLAB par la commande `sdrinfo` comme il est expliqué précédemment, après nous exécute le script.

D'abord l'exécution du programme, Nous allons présenter un organigramme résumant les étapes expliquées précédemment :

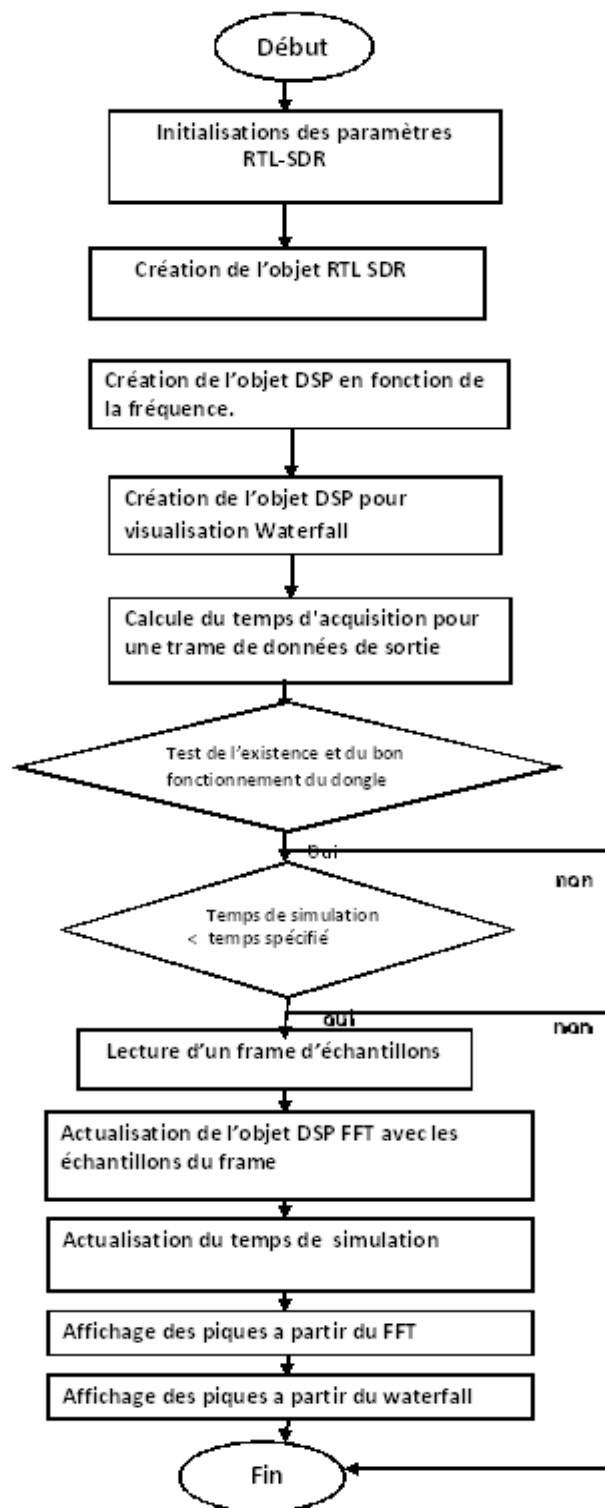


Figure IV. 16 :organigramme correspondant au script

Les résultats obtenus sont presque identiques aux résultats précédents, la figure IV.17 illustre ses résultats en obtenant différentes fréquences telles que la fréquence de la bande GSM [900-1800] :

Chapitre I V : Développement d'un récepteur de détection de l'activité spectrale

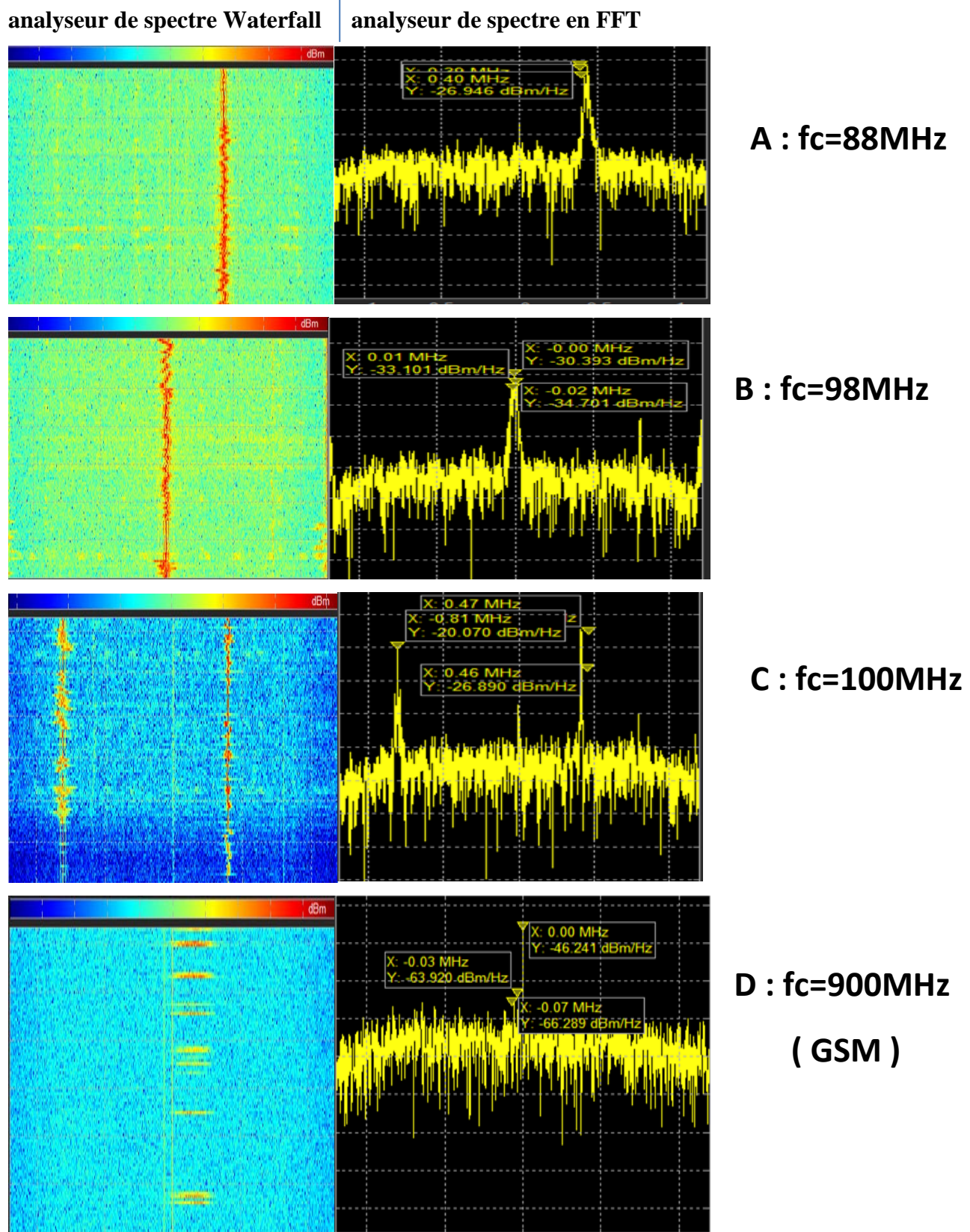


Figure IV.17 : le scanne d'analyseur de spectre en FFT sous Matlab

IV.5 Conclusion

Dans ce chapitre nous avons présenté la partie essentielle de notre projet fin d'étude, l'idée de notre travail consiste à donner une solutions SDR qui scanne les signaux environnants avec précision avec l'environnement de développement MATLAB.

D'une part, Sous simulink nous avons appris la méthode qui simplifie notre travail (masque) et se limiter les paramètres les plus importants du scan du schéma block sous Simulink.

D'une autre part, la programmation sous MATLAB nous a permis de développer les techniques de traitement numériques à l'image de la densité spectrale de puissance.

Conclusion générale

Le travail présenté dans ce projet a pour objectif le développement d'un récepteur universel à faible coût pour des systèmes de types radio logicielle. Le développement de notre système vise essentiellement à étudier la reconfigurabilité dans les terminaux mobiles et les stations de base pour répondre à une diversité de standards et de fonctionnalités.

Dans un premier temps, nous avons brièvement introduit les concepts de la radio logicielle. Nous avons présenté l'architecture utilisée avec ce concept ainsi que ses avantages, objectifs dans le domaine de radiocommunication. Un aperçu général sur les différentes plateformes SDR ainsi que la relation entre la radio logicielle et la radio cognitive est également présentée.

Ensuite, pour compléter les études théoriques nous avons présenté le récepteur RTL-SDR avec ses caractéristiques pour mesurer les signaux RF.

L'implémentation commence avec la manière de l'interfacer le dongle RTL-SDR avec notre station de travail du PC i3 et le logiciel de traitement SDRSHARP qui nous permet de capter des signaux réels par exemple de la radio FM.

Nous nous sommes intéressés dans le chapitre 4 de proposer une solution SDR à base d'un autre environnement de développement tel que MATLAB & Simulink, une fois en construisant un schéma bloc du récepteur RTL-SDR qui nous permet d'identifier les fréquences utilisées. Comme nous avons développé sous la forme d'un programme un script simplifié qui effectue le scanne des d'activités spectrales.

Les perspectives de notre travail consistent à détecter automatiquement en plus de la fréquence le type de la modulation utilisé afin de pouvoir interpréter correctement l'information transmise.

En plus, cette Expérience est un bon complément de notre formation de base, elle nous a apporté dans un premier temps des nouvelles connaissances dans le domaine de la radio logicielle et les méthodes de négociation pour obtenir le spectre. Ainsi de voir en pratique des concepts plus avancés tel que l'application SDRSHARP.

Bibliographies

- [1] B. Fette "Cognitive radio technology", Academic Press, Elsevier, Second Edition, 2009
- [2] Christian BARTHOD "Présentation et mise oeuvre du SDR", juin 2014.
- [3] Michaël Nicolas. Radio logicielle : analyse d'architectures matérielles et outils informatiques. Electronique. 2011.
- [4] C Kasmi, A Ebalard et Pierre-Michel Ricordel "De la radio matérielle à la radio logicielle" 2011.
- [5] Jean-Philippe Delahaye "Plate-forme hétérogène reconfigurable : application à la radio logicielle", 2007.
- [6] Stéphane LECOMTE "Méthodologie de conception basée sur des modèles de haut niveau pour les systèmes de radio logicielle", 2011.
- [7] H. Arslan "Cognitive radio, Software defined radio, and adaptive wireless systems", Springer, 2007.
- [8] Robert W. Stewart, Kenneth W. Barlee, Dale S. W. Atkinson, Louise H. Crockett "Software Defined Radios using MATLAB® & Simulink® and the RTL-SDR", September 2015 by Strathclyde Academic Media Version 1.15.0922.
- [9] <http://www.realtek.com.tw/products/productsView.aspx?Langid=1&PFid=35&Level=4&Conn=3&ProdID=257>
- [10] <http://osmocom.org/projects/sdr/wiki/rtl-sdr>
- [11] T. J. Roupheal, RF and Digital Signal Processing for Software. Defined Radio: A Multi-Standard Multi-Mode Approach, Newnes, 2009.
- [12] CARL LAUFER "The Hobbyist's Guide to the RTL-SDR: Really Cheap Software Defined Radio".
- [13] Christian JUTTEN "Théorie du signal", novembre 2009