

Jonathan Janson

*Radio Definido por Software: Estudo e
Realização de Teste com uma Plataforma Livre*

São José – SC
Fevereiro / 2012

Jonathan Janson

***Radio Definido por Software: Estudo e Realização
de Teste com uma Plataforma Livre***

Monografia apresentada à Coordenação do Curso Superior de Tecnologia em Sistemas de Telecomunicações do Centro Federal de Educação Tecnológica de Santa Catarina para a obtenção do diploma de Tecnólogo em Sistemas de Telecomunicações.

Orientador:

Cleiber Marques, Tecg.

Co-orientador:

Prof. Evandro Cantu, Dr. Eng.

CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE TELECOMUNICAÇÕES
CENTRO FEDERAL DE EDUCACAO TECNOLOGICA DE SANTA CATARINA

São José – SC

Fevereiro / 2012

Monografia sob o título “Radio Definido por Software: Estudo e Realização de Teste com uma Plataforma Livre”, defendida por Jonathan Janson e aprovada em 07 de março de 2012, em São José, Santa Catarina, pela banca examinadora assim constituída:

Cleiber Marques, Tecg.
Orientador

Prof. Evandro Cantu, Dr. Eng.
IFSC – São José

Prof. Elen Macedo Lobato Merlin, Dra. Eng.
IFSC – São José

*O insucesso é apenas uma oportunidade para recomeçar de novo
com mais inteligência.*

Henry Ford

Agradecimentos

Dedico meus sinceros agradecimentos a meus pais, Mateus e Valdete, que tiveram decisões difíceis na vida, mas sempre tendo a consciência que o estudo de seus filhos estivessem acima de qualquer outra prioridade.

A meu grande amigo Cleiber Marques, que me orientou neste trabalho, esteve sempre ao meu lado nos momentos de estudo durante o curso.

A minha família como um todo, por sempre acreditar sempre em meu potencial.

A minha irmã Julhiana Janson, mesmo longe incentivou e cobrou para que este trabalho estivesse concluído.

A minha namorada Mayara Pinheiro Martins, que teve total paciência nos momentos em que estava trabalhando neste trabalho, e ajudou para que estivesse concluído.

Ao professor Evandro Cantu, que providenciou todos os tramites para apresentação deste TCC, contribui muito com a escrita deste trabalho e foi um grande parceiro durante essa jornada acadêmica.

A todos os grandes amigos conquistados nesta instituição de ensino durante esta etapa acadêmica, os quais já passamos muitas histórias juntos.

Aos colegas de curso, que em algum momento contribuíram para um melhor entendimento dos conteúdos.

Por fim, gostaria de agradecer a todos os professores que tive contato, por passar todo seu conhecimento e em alguns momentos serem colegas de aprendizado.

Resumo

Os rádios definidos por *software* (RDS) são rádios onde o processamento do sinal é feito através de um *software* sendo executado em um processador, ou seja, sem a necessidade de *hardware* para que o sinal possa ser interpretado e seja entregue a sua aplicação. É uma tecnologia com um alto grau de flexibilidade, pois problemas que em rádios convencionais são tratados através de mudanças de *hardware*, passam a serem tratados em *software*.

RDS trata-se de uma tecnologia que tem um baixo custo de implementação, pois em uma estrutura ideal os únicos componentes de *hardware* encontrados são antenas e conversores A/D/A. Como uma topologia ideal está um pouco distante de ser realizada, o estudo do RDS deve ter um enfoque muito grande no que é conhecido de *Front End*, a interface que faz a ligação do mundo RF (Rádio Frequência) com o *Software* do rádio. Conhecer cada componente de um *Front End* faz com que o projeto do mesmo se torne viável para o projeto final de um RDS.

Existem muitas plataformas de implementação de rádios definidos por *software*, a GNU *Radio* é uma delas. É uma ferramenta *open source*, que pode ser implementada em linguagem de programação ou em interface gráfica, tornando-se assim uma plataforma muito amigável e de fácil utilização por qualquer pessoa que queira ingressar neste ramo de pesquisa. Para esta ferramenta também foi desenvolvido um *Front End* universal, a USRP, facilitando ainda mais as pesquisas para RDS.

Este trabalho visa apresentar de uma forma explicativa a tecnologia e o que compõe um RDS, mostrando cada elemento. Demonstra também um exemplo de aplicação combinando o GNU *Radio* e a plataforma USRP.

Sumário

Agradecimentos.....	4
Resumo.....	5
Sumário	6
Lista de figuras	8
1 Introdução	10
1.1 Motivação e Objetivos.....	11
1.2 Organização do Texto.....	11
2 Radio Definido por Software (RDS).....	12
2.1 Breve História do Rádio	12
2.2 Conceito de RDS	14
2.3 O RDS Ideal	15
3 Arquitetura Real de um RDS e seus Modulos	17
3.1 Arquitetura de um RDS	17
3.2 Front End RF	19
3.2.1 Frequência Intermediária (FI).....	20
3.2.2 Conversor Super Heteródino	20
3.2.3 Arquitetura de Conversão Direta	21
3.2.4 Arquitetura de Conversão Múltipla	23
3.3 Módulo de Processamento Digital.....	24
3.3.1 Conversores A/D e D/A (DAC E ADC).....	25
3.3.1.1 Teorema de Nyquist.....	26
3.3.1.2 Super Amostragem do Sinal	27
3.3.1.3 Subamostragem do Sinal	28
3.3.1.4 Arquitetura dos Conversores	29
3.3.2 DDC – <i>Digital Downconverter</i>	32
3.3.3 DUC – <i>Digital Up-Converter</i>	33
3.3.4 Processadores de Sinais	34
4 Uma Plataforma Livre para Desenvolvimento de RDS	35

4.1 <i>Universal Software Radio Peripheral Board (USRP)</i>	36
4.1.1 Conversor Analógico-digital	38
4.1.2 Conversor Digital-analógico.....	39
4.1.3 A FPGA	39
4.1.4 Placas Filhas	40
4.2 <i>GNU Radio</i>	40
4.2.1 GRC - <i>GNU Radio Companion</i>	42
4.3 Exemplo Prático de Aplicação	42
5 Conclusão.....	51
Referências.....	53
Apêndice A.....	56
Apêndice B.....	58

Lista de Figuras

Figura 2.1: Radio Definido por Software Ideal.....	15
Figura 3.1: Estrutura Básica de um RDS	18
Figura 3.2: RDS real com módulo de <i>front end</i> digital.....	19
Figura 3.3: Conversor Super Heteródino de um Estágio	21
Figura 3.4: Receptor com conversão direta em quadratura.....	22
Figura 3.5: Transmissor com conversão direta	22
Figura 3.6: Receptor de arquitetura de conversão múltipla	23
Figura 3.7: Transmissor de arquitetura de conversão múltipla.....	24
Figura 3.8: Módulo de processamento digital.....	25
Figura 3.9: Teorema de Nysquist.....	26
Figura 3.10: Filtro anti-aliasing atendendo o critério de Nyquist.....	27
Figura 3.11: Filtro anti-aliasing usando superamostragem.....	28
Figura 3.12: Conversor AD Paralelo.....	30
Figura 3.13: Conversor AD Sigma-delta	31
Figura 3.14: Conversor AD de Multiplo-estágio	32
Figura 3.15: Arquitetura de um DDC	32
Figura 3.16: Arquitetura de um DUC	34
Figura 4.1: USPR versão 1.....	37
Figura 4.2: diagrama de blocos da USPR	38
Figura 4.3: Arquitetura de uma DDC padrão na USPR.....	39
Figura 4.4: Arquitetura de uma DUC padrão na USPR.....	40
Figura 4.5: Transmissor AM-SSB LSB implementado em GRC	43
Figura 4.6: Espectro AM sem sinal de voz	45
Figura 4.7: Espectro AM com sinal de voz.....	45
Figura 4.8: Espectro AM-SSB LSB sem sinal de voz	46
Figura 4.9: Espectro AM-SSB LSB com sinal de voz.....	47
Figura 4.10: Transmissor AM-SSB LSB implementado em GRC	48
Figura 4.11: Espectro do sinal recebido sem sinal de voz	50

Figura 4.12: Espectro do sinal recebido com sinal de voz.....50

1 INTRODUÇÃO

Os campos de atuação das telecomunicações, envolvidas com a transmissão ou recepção de rádio frequência, sempre buscaram o avanço tecnológico a fim de customizar os equipamentos de rádio, tornando-os cada vez menores, mais rápidos, com menor custo e melhor qualidade.

O aprimoramento em outras áreas tecnológicas, como a área de *softwares*, circuitos integrados programáveis, processadores de alto desempenho, entre outras áreas, deram base para o início em pesquisas em Rádios Definidos por Software. Essa área de pesquisa visa o que muitos dos estudiosos de rádio frequência sempre almejavam, um menor custo com uma diminuição do *hardware* do equipamento.

A filosofia de um RDS é implementar o *hardware* de um rádio através de *software*, ou seja, processar o sinal transmitido ou capturado em um programa sendo executado dentro de um processador ou até mesmo um computador, tendo como único *hardware* no sistema a antena de transmissão ou recepção.

Os primeiros estudos nessa área foram de cunho militar, mas hoje em dia está bem difundida com vários ramos de pesquisas utilizando suas facilidades. Como toda tecnologia ela traz algumas limitações, que serão bem detalhadas no decorrer do texto, com todas maneiras de contornar essas limitações.

Muitos grupos começaram estudos de plataformas para o desenvolvimento dos RDS. A GNU¹ é uma delas, que dispõe a qualquer pessoa uma plataforma *open source* para implementação de um RDS. Isso ajuda muito nas pesquisas, pois dá acesso a instituições de ensino e acadêmicos a contribuir cada vez mais com o avanço dessa tecnologia.

¹ O projeto GNU teve objetivo inicial criar e distribuir um sistema operacional livre, onde qualquer pessoa possa usar, estudar, modificar ou redistribuir o programa ou seu código fonte, gratuitamente. Atualmente, o projeto expande esse conceito para várias aplicações.

1.1 Motivação e Objetivos

A motivação para esta monografia está em ampliar os campos de pesquisa na área de telecomunicações, no Campus São José do IFSC, incluindo tecnologias emergentes na área de radio transmissão.

Este documento tem como objetivo a apresentação da tecnologia de Rádios Definidos por *Software* (RDS), mostrando suas facilidades, suas limitações, como é seu funcionamento e arquitetura de cada etapa de sua implementação. Também objetiva a demonstração de uma plataforma de implementação RDS baseada em *Open Source*.

1.2 Organização do texto

O texto está organizado da seguinte forma: No Capítulo 2 é apresentado um breve histórico do rádio no Mundo e no Brasil, o conceito de RDS e sua arquitetura ideal. No Capítulo 3 é descrito a arquitetura real de um RDS, descrevendo cada módulo e algumas técnicas de como cada módulo pode ser implementado. No Capítulo 4 é apresentado a plataforma *Open Source* e um exemplo de implementação através dela. No capítulo 5 são apresentadas conclusões desta monografia.

2 RADIO DEFINIDO POR SOFTWARE (RDS)

Este capítulo é destinado ao estudo da tecnologia de rádios definidos por *software*, abordando seu conceito e arquitetura. Traz também um histórico do rádio em geral no Mundo e no Brasil.

2.1 Breve História do Rádio e Contextualização

Em 1887, surgiu o princípio da propagação radiofônica, através do professor alemão Heinrich Hertz que, baseado nos estudos do físico escocês James Clerk Maxwell, conforme escreve Paulo Ricardo Guedes Pinheiro [13]:

desenvolveu experiências em seu laboratório descobrindo que, ao fazer saltar faíscas entre duas esferas de cobre de seu aparelho oscilador (transmissor), também saltavam faíscas entre as pontas de um arco de metal colocado a uma certa distância, denominado ressonador (receptor). A experiência de *Hertz* comprovou a existência das ondas eletromagnéticas, previstas matematicamente por *Maxwell* anos antes.

Desta maneira Hertz havia descoberto as ondas de rádio e que, ainda que as ondas eletromagnéticas não pudessem ser visualizadas, estas eram detectadas eletricamente e podiam ser transmitidas e refletidas.

Em 1891 Nikola Tesla iniciou estudos e publicou um artigo intitulado “*The True Wireless*”. Posteriormente, em 1893 o pesquisador fez uma demonstração pública da comunicação sem o uso de fio, por meio do rádio [12]. Em pouco tempo as experiências de Tesla passaram a interessar diversos cientistas que se aprofundaram em seus estudos.

No ano de 1898, o físico Oliver Lodge conseguiu patentear uma invenção chamada de “*Electric Telegraphy*”, a qual produzia sinais de rádio através de bobinas de Tesla para o transmissor e um detector do tipo *coherer*.

Em dezembro de 1901, Guglielmo Marconi anunciou ter realizado uma transmissão de mensagem, a letra s pelo Código Morse, através do Oceano Atlântico, da Inglaterra ao Canadá. Em 1912, a patente do *Electric Telegraphy* de Olivr Lodge foi vendida para Guglielmo Marconi [12].

No Brasil, o padre e inventor Landell de Moura contribuiu muito para o desenvolvimento do rádio. Relatos demonstram que em 1894 Landell transmitiu a voz humana por 8 quilômetros em linha reta da avenida Paulista até o Alto de Santana, experiência por ele repetida no ano de 1900 na presença de jornalistas e de um representante do governo britânico[4].

Para transmitir a voz humana Landell usava dois veículos: um transmissor que utilizava um microfone eletromecânico que capturava as ondas sonoras por meio de uma câmara de ressonância, onde um diafragma metálico abria e fechava o circuito do primário de uma bobina de Ruhmkorff, e induzia no secundário dessa bobina uma tensão que era irradiada através de uma antena ou de duas esferas centelhadoras; e por meio de um aparelho de telefone sem fio, que usava a luz como uma onda portadora da informação de áudio [4].

Nos anos seguintes, algumas invenções e inovações ocorreram, entre as quais o desenvolvimento por Lee De Forest da válvula tríodo e o emprego do tríodo para amplificar e produzir ondas eletromagnéticas de forma contínua por Von Lieben e Armstrong [11].

O inventor Lee De Forest patenteou em 1907 a válvula tríodo e desenvolveu um detector eletrolítico para ondas de rádio. Através de sua criação foi possível amplificar um sinal de radiofrequência capturado pela antena antes do detector presente no receptor, o que permitiu a recepção de sinais mais fracos e ao final resultou na invenção de modulação em amplitude (AM). O inventor ainda foi o primeiro a utilizar a expressão “rádio”.

Por sua vez, o inventor Edwin Howard Armstrong patenteou algumas das invenções essenciais que tornaram possível o modelo de rádio existente hoje em dia: o circuito regenerativo, o circuito super heteródino e a modulação em frequência de banda larga, chamada de FM. Os circuitos regenerativos permitiram que os sinais pudessem ser ouvidos sem fones de ouvido por terem aumentado sua amplitude; os circuitos super heteródinos tornaram mais simples os receptores de rádio, eis que acabaram com a obrigação de controle de ajuste, o que tornou o rádio mais sensível e seletivo, e a modulação em frequência FM trouxe uma melhor qualidade de som quando comparada à modulação AM.

Muitas tecnologias atuais utilizam as comunicações em rádio frequência, entre elas a telefonia móvel, redes wireless, a televisão, e os avanços em pesquisa no campo da radiofrequência só aumentam por este motivo. Estes tipos de equipamentos são todos

baseados em hardware, ou seja, tem seu módulo de rádio transmissão todo projetado através de componentes eletrônicos, o que faz com que estes equipamentos tenham certo volume e certa dificuldade em ser projetados[32].

Uma filosofia emergente de implementação de equipamentos está em estudos nos dias atuais. Com origens em sistemas do departamento de defesa americano os rádios definidos por software vêm ganhando espaço nos estudos da área de radiofrequência. Para resolver um problema de interligação entre várias interfaces aéreas, uma solução que os americanos encontraram foi tentar desenvolver um equipamento de rádio transmissão que se adaptasse a várias faixa de frequência existentes sem a necessidade de mudança no hardware, implementando assim a filosofia de um rádio definido por software entre os anos de 1970 e 1980 [32].

2.2 Conceito de RDS

O conceito de radio definido por software (RDS) não deve ser confundido com rádios baseados ou controlados por software, pois hoje, praticamente todos os rádios se utilizam de software em sua concepção. Estes rádios baseados ou controlados por software necessitam de ajustes no hardware para qualquer mudança em interfaces baseadas em software [32]. Ou seja, com o software podem ser controlados parâmetros do rádio como, frequência de operação, modo de operação (AM, FM, SSB), controle de ganho, etc. O *software* é só uma interface para ajustes no próprio hardware. Já um RDS ideal deve ter todo o processamento de sinal feito através de software, exceto a digitalização do sinal que deve ser feita logo após a captação feita pela antena.

Joseph Miota [25] define que “um rádio definido por software é um rádio cuja modulação de formas de onda do canal é definida em software. Isto é, as formas de ondas são geradas como sinais digitais amostrados, convertidas de digitais para analógicas através de conversor digital analógico (DAC) de banda larga, que captura todos os canais do nó do RDS. O receptor, por sua vez, captura o sinal, faz um abaixamento de frequência e demodula a forma de onda do canal por meio de um software que roda em um processador de uso geral”.

Em uma maneira mais simplificada, um rádio definido por software é um transceptor de rádio que tem seu princípio de funcionamento sendo executado através de um software, podendo ter seu funcionamento alterado com uma simples atualização deste mesmo software, sem a necessidade de nenhuma alteração de hardware.

O principal objetivo desta tecnologia é diminuir o processamento em hardware, digitalizando o sinal o mais próximo possível da antena. Idealmente, um RDS deve ter somente os conversores digital-analógico e analógico-digital entre o processador, onde é executado o software, e a antena, a qual deve ser uma antena inteligente² para acompanhar a reconfigurabilidade do software.

Desta forma, transformar problemas relacionados ao hardware em problemas relacionados ao software é um outro atrativo desta tecnologia [32].

2.3 O RDS Ideal

A concepção ideal de um rádio definido por software se dá através de três blocos, o bloco 1 composto pelas antenas, o bloco 2 composto pelos DAC (conversor digital-analógico) e ADC (conversor analógico-digital), e o bloco 3 responsável pelo processamento, como está ilustrado na figura 2.1.

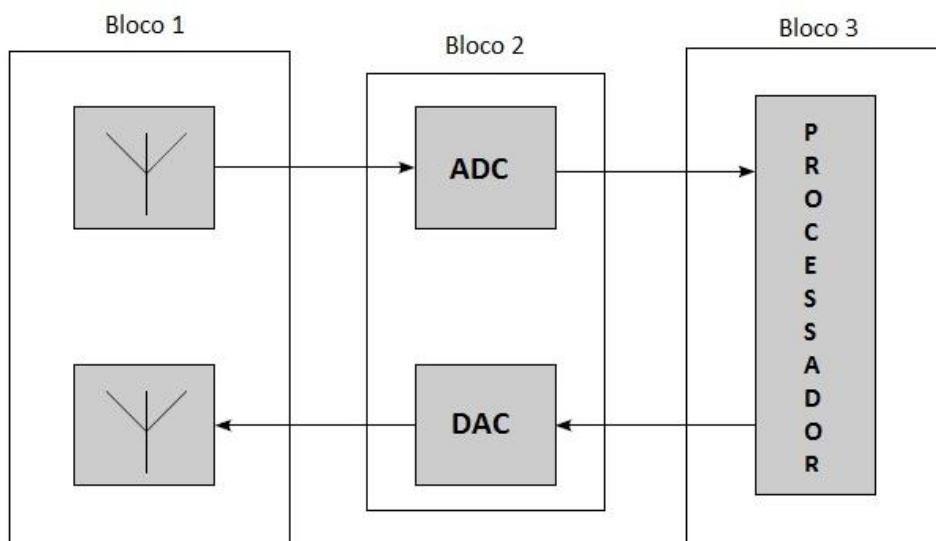


Figura 2.1: Radio Definido por Software Ideal

Segundo Letícia G. de Barros [2], um RDS busca três características principais:

² Antenas capazes de se ajustar dinamicamente ao meio de acordo com os requisitos do sistema são denominadas antenas inteligentes [32]

1. Reconfigurabilidade: Capacidade de alterar a funcionamento do rádio. Esta é uma característica promissora, pois a ideia de reconfigurar um rádio à distância, sem intervenção física é justamente a promessa do conceito de RDS.
2. Flexibilidade: Aceitar, sem mudanças na arquitetura do rádio, toda a reconfigurabilidade aplicada.
3. Modularidade: As partes que definem o sistema são executadas em módulos distintos. Essa característica implica na alta flexibilidade do sistema, já que se por ventura alguma modificação for feita em algum dos módulos, não afetará o sistema do RDS.

Algumas limitações impedem que o modelo ideal de RDS seja implementado, os requisitos necessários de ADC's e DAC's ficam muito além dos limites práticos existentes no que se refere a taxa de amostragem, largura de banda e faixa dinâmica³. Outra limitação para este modelo é o alto desempenho de processamento para executar todo esse software. Para solução desses problemas são feitas algumas alterações do sinal via hardware, enquanto diferentes processadores, tais como FPGAs, DSPs (processadores digitais de sinal) e processadores comuns ficam encarregados do processamento do software, para implementação de um modelo real de RDS [5].

³ Faixa dinâmica é a relação entre a tensão máxima do sinal de entrada e a amplitude mais elevada de erro que esse sinal tenha.

3 ARQUITETURA REAL DE UM RDS E SEUS MODULOS

Este capítulo traz a descrição da arquitetura real de uma RDS, mostrando suas limitações, sendo descrito cada módulo de seu funcionamento, não abordando o módulo de antenas. Será descrito o módulo de Front End RF e suas dificuldades de implementação, o módulo de processamento digital (DCA, ADC, DDC DUC), e os processadores de software de um RDS.

3.1 Arquitetura de um RDS

A arquitetura de um rádio convencional ou de um RDS é facilmente identificada e dividida em 2 componentes majoritários:

- *Front End* de rádio: Responsável por cuidar do recebimento e transmissão das frequências de rádio.
- *Back End* de rádio: Responsável pelo processamento do sinal de rádio.

Rádios não definidos em software dão a impressão de que o hardware é o elemento dominante tanto na subdivisão denominada *front end* quanto na subdivisão denominada *back end*. Tratando-se de rádios definidos por software, as atribuições que cabem a hardware e software mudam, com o software sendo o elemento dominante e o hardware tendo um papel de apoio [32]. Desta forma, na tecnologia RDS o hardware compõe o *front end* e o software domina o processamento no *back end*.

Em um RDS ideal, como está descrito no capítulo 2, o *front end* seria composto somente por antenas de conversores A/D e D/A, observando a figura 2.1 seria composto pelo bloco 1 e 2. Como já comentado devido a limitações da tecnologia desses conversores, o modelo ideal não é possível de implementação nos dias atuais. Algumas adaptações no sinal devem ser feitas, separando assim o denominado *front end*, que passa a ter a incumbência de tratar esse sinal entre a antena e os DAC e ADC. Então, a arquitetura básica de um RDS real está descrita na figura 3.1.

Nesta concepção de RDS real, o *front end* se torna um novo módulo, onde prepara o sinal para conversão AD (analógico/digital) para o processamento do sinal e/ou prepara o sinal para transmissão após a conversão DA (digital/analógico), tendo em vista que o RDS pode somente transmitir sinal, somente captar sinal, ou fazê-los ao mesmo tempo. Esta

preparação é feita através de uma amplificação do sinal, um controle de ganho, deslocamento para uma frequência intermediária (no caso de recepção) ou deslocamento para frequência original do sinal (no caso de transmissão), uma filtragem *anti-aliasing*⁴. Além de resolver o problema da tecnologia empregada nos conversores A/D e D/A, o *front end* pode ser projetado para otimizar o custo do projeto de um RDS, sendo ajustado para o uso de DAC's e ADC's de baixo custo.

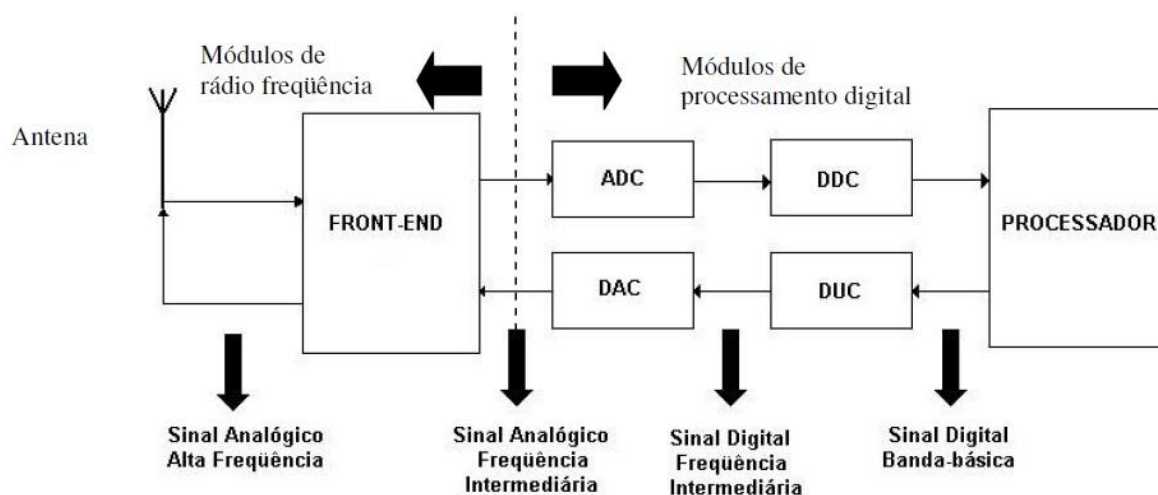


Figura 3.1: Estrutura Básica de um RDS

Na figura 3.1, surgem dois novos componentes, o *Digital Down Converter* (DDC) e o *Digital Up Converter* (DUC). Para que o processamento não seja efetuado em velocidades muito altas, após sua digitalização o sinal digital de frequência intermediária deve ser convertido para um sinal digital de banda básica, respeitando o Teorema de amostragem de Nyquist⁵ [2]. Então, a função do DDC é fazer a re-amostragem do sinal digital de frequência intermediária para um sinal digital de banda base (onde a menor frequência do sinal é igual a zero e a maior é igual à largura de banda do sinal). A função do DUC é efetuar o inverso do DDC, deixando o sinal novamente na frequência digital intermediária, pronto para o conversor DA.

⁴ Um filtro *anti-aliasing* consiste em um filtro passa baixa com frequência de corte igual ou menor que a metade da frequência de amostragem, com o intuito de retirar componentes de frequências indesejadas sobreponham o espectro do sinal amostrado.

⁵ Este teorema é detalhado na seção 3.2.1.1

Pode-se também fazer uma abordagem onde o *front end* denomina-se *front end* RF e, DAC, ADC, DDC e DUC passam a ser programados em FPGA's, denominando-se assim o *front end* digital, como demonstra a figura 3.2.

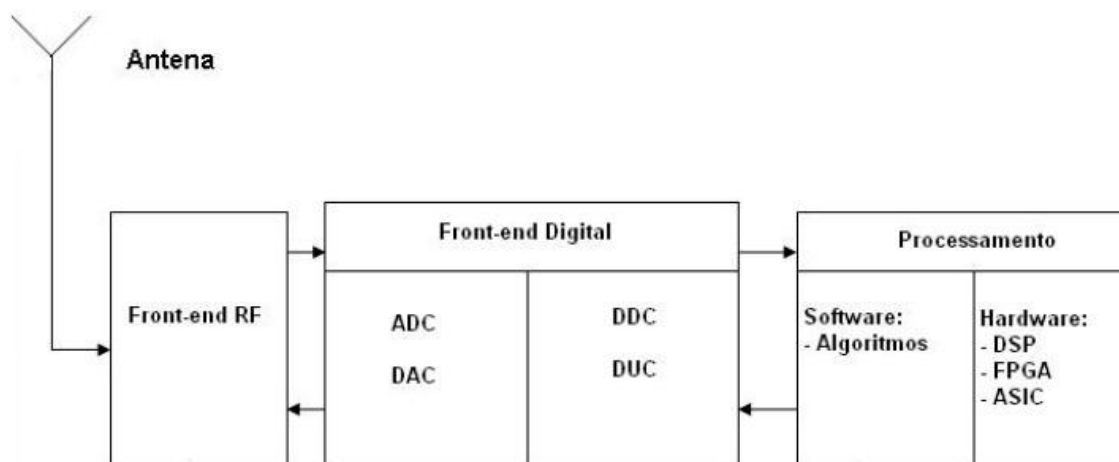


Figura 3.2: RDS real com módulo de *front end* digital

3.2 Front End RF

Como já abordado neste trabalho, um RDS ideal deve processar e transmitir qualquer sinal, independente da modulação, faixa de frequência, nível de potência ou largura de banda. Como este modelo está um pouco distante de ser implementado, devido as limitações tecnológicas da atualidade, o módulo de *front end* de um RDS deve ser estudado com muito cuidado e atenção [2].

Nota-se também que a principal limitação de um RDS está nos conversores A/D e D/A, fazendo com que o tratamento do sinal entre a antena e os DAC e ADC seja muito bem projetado, tornando este módulo um fator muito importante.

O *front end* RF possui duas tarefas. A primeira tarefa é de preparar o sinal para que o conversor A/D possa digitalizar o sinal recebido pela antena, e a segunda voltar o sinal para frequência original de transmissão antes de ser transmitido pela antena [32]. Lembrando que o *front end* RF tem 4 funções, amplificar o sinal, transladar a frequência do sinal, aplicar um controle de ganho e filtrar o sinal contra ruídos.

Para produzir um bom projeto de um *front end* RF, é necessário o estudo e entendimento de alguns conceitos, os quais serão detalhados nas próximas subseções.

3.2.1 Frequência Intermediária (FI)

A frequência intermediária (FI) do sinal, após o *front end* RF, não deve ser uma frequência alta, atendendo a frequência de operação dos DAC e ADC, que por limitações de tecnologia e custo são de baixa frequência em projetos de RDS. Esta frequência é conhecida como frequência intermediária, pois logo após a digitalização do sinal sofre mais um abaixamento de frequência tornando sinal digitalizado em banda base, ou seja, é uma frequência intermediária em relação ao sinal de entrada ou saída e o sinal processador pelo software.

Essa frequência deve variar entre 0Hz e a metade da frequência de amostragem dos conversores, seguindo assim o Teorema de Nyquist. Desta forma, os sinais de FI são processados em uma banda de frequência 0 e $x/2$ Hz, observando-se que a taxa de amostragem dos conversores é x Hz.

A frequência intermediária é obtida através de conversores de frequência. Um bom exemplo é o conversor super heteródino, que será detalhado na próxima seção. Este tipo de conversor é utilizado para que seja garantido o cumprimento do teorema de Nyquist, pois possui filtro antes do misturador do conversor, eliminando assim componentes indesejáveis do sinal e evitando a sobreposição dos espectros de frequência [19].

3.2.2 Conversor Super Heteródino

Everson [23] escreve que:

Nesta topologia, a informação contida no sinal de RF é transladada para um sinal de frequência fixa mais baixa na qual todos os processos de amplificação e filtragem necessários são realizados. Como a frequência do sinal é bem menor e fixa, o projeto dos circuitos torna-se mais simples (menor influência de elementos parasitas, maior estabilidade, etc.) e podem-se obter desempenhos iguais ou superiores (seletividade do filtro, ganho, etc.) a custos mais baixos que outras topologias.

Basicamente a estrutura desse conversor está descrita na figura 3.3.

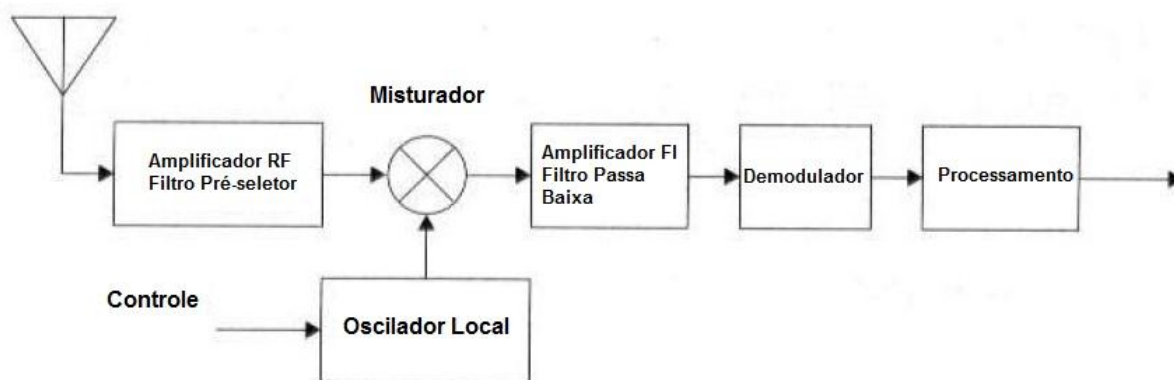


Figura 3.3: Conversor Super Heteródino de um Estágio

A estrutura funciona basicamente da seguinte forma:

1. O sinal de entrada é filtrado através de um filtro passa baixa;
2. O sinal já filtrado é amplificado;
3. No misturador o sinal é multiplicado pela frequência contida no oscilador (FI), transladando a informação do sinal de entrada para frequência intermediária;
4. O resultado dessa multiplicação é amplificado e filtrado por um filtro passa baixa;
5. Finalmente o sinal é demodulado e pode ser processado na frequência FI.

Leticia [2] afirma que, “normalmente a conversão de sinais é feita em dois estágios, pois a arquitetura precisa de menor qualidade nos filtros e não necessita de isolamento entre os misturadores da entrada e os osciladores locais. Porém, o acréscimo de estágios pode levar a um consumo de potência”.

3.2.3 Arquitetura de Conversão Direta

A técnica de conversão direta é assim denominada, quando existe somente um estágio de conversão, ou seja, quando o oscilador ligado ao misturador tem a própria frequência intermediária no caso de um *front end* RF de um RDS, tanto na recepção como na transmissão [22].

Em um RDS o mais comum a se usar são conversores diretos observados nas figuras 3.4 e 3.5, onde após o translado do sinal tem-se a obtenção de dois sinais em quadratura na frequência FI [17].

A figura 3.4 ilustra um receptor com conversão direta de frequência, onde o sinal é filtrado e amplificado antes de passar pelo conversor em quadratura, e depois é novamente

filtrado antes de serem digitalizado pelos conversores AD. Este comportamento caracteriza uma conversão direta, pois apesar de o sinal ser multiplicado duas vezes, é multiplicado pela mesma frequência defasada em 90° .

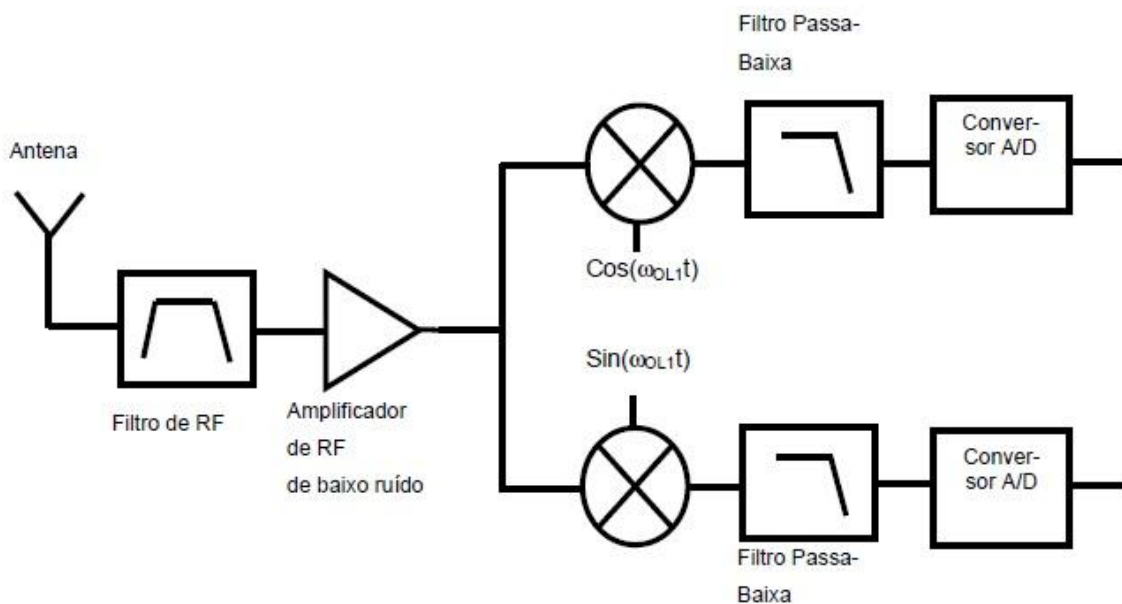


Figura 3.4: Receptor com conversão direta em quadratura

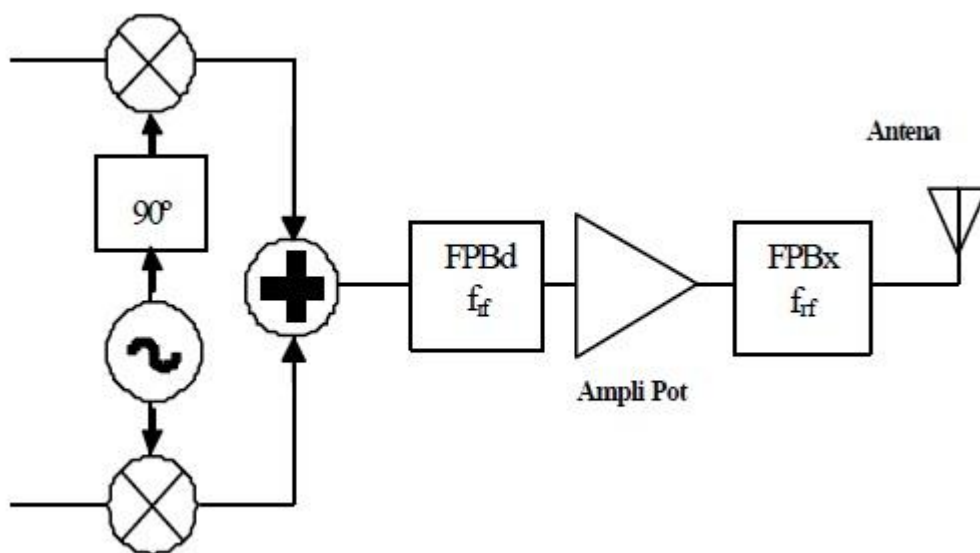


Figura 3.5: Transmissor com conversão direta

A figura 3.5 segue a mesma lógica inversa da figura 3.4, onde dois sinais em quadratura sofrem a conversão para frequência desejada e depois são somados, amplificados e filtrados para transmissão. A característica da conversão direta está demonstrada quando os sinais em quadratura são convertidos em uma mesma frequência defasada em 90° .

Essa arquitetura tem certas vantagens e desvantagens. Como escreve [22], “como a translação é direta para banda base, toda a circuitaria de filtragem e amplificação pode ser mais facilmente integrada, eliminando a necessidade de elementos externos”. Já ao que se refere à desvantagens, Letícia [2] afirma:

Quanto á desvantagem está a necessidade de um oscilador local de frequência variável que gere duas senóides em quadratura e balanceadas em amplitude, considerando um grande range de faixa de frequências. Além disso, tem-se problema no espalhamento espectral dos osciladores e no retorno das frequências indesejadas à antena, essas frequências serão irradiadas ou retransmitidas ao receptor, causando um nível DC invariante no tempo no sinal.

3.2.4 Arquitetura de Conversão Múltipla

Como o próprio nome já diz, um sistema com conversão múltipla tem o conceito de transladação do sinal em vários estágios, ou seja, quando o sinal é misturado a frequências variáveis em cada estágio até chegar a frequência desejada [32].

Um receptor e um transmissor utilizando esta arquitetura estão ilustrados nas figuras 3.6 e 3.7 [32] respectivamente:

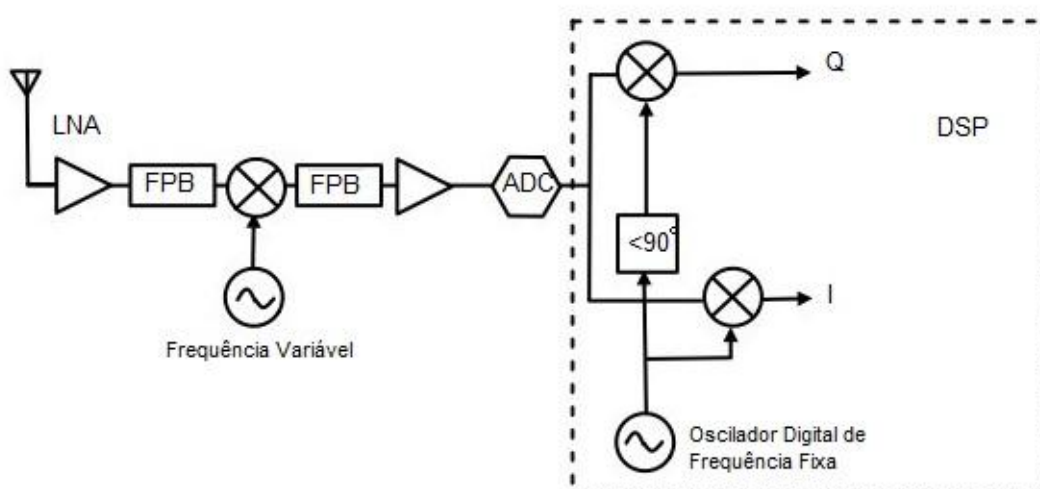


Figura 3.6: Receptor de arquitetura de conversão múltipla

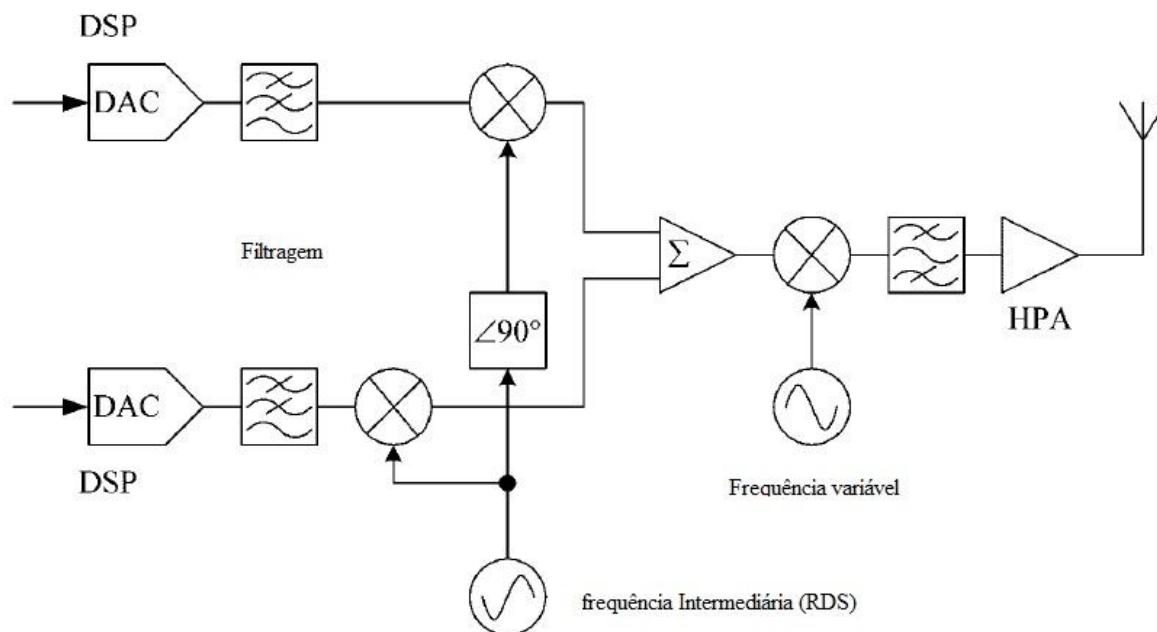


Figura 3.7: Transmissor de arquitetura de conversão múltipla

Nas duas figuras fica exposto o princípio da conversão em múltiplo estágio, pois tanto na transmissão quanto na recepção, o sinal é convertido em um estágio com frequências fixas em quadratura. E no outro estágio é convertido em uma única frequência variável.

A maior vantagem dessa arquitetura está em fazer o segundo estágio digitalmente em um DSP, pela facilidade de a conversão dos sinais de quadratura estar sendo feito em uma frequência fixa. Como desvantagens desse sistema está o grau de complexidade, a utilização de vários osciladores locais, e como a implementação de vários filtros é necessária a, implementação em um único circuito integrado se torna inviável [32].

3.3 Módulo de Processamento Digital

Esta seção irá detalhar as partes componentes de um módulo de processamento digital, mostrado na figura 3.1. Como visto anteriormente, este é o módulo onde o sinal captado pela antena e transladado para frequência intermediária é digitalizado e levado a um sinal digital de banda base, estando assim pronto para o processamento no caso de sinal recebido. Na transmissão de sinal, este módulo é responsável por converter o sinal digital em banda base, que acabou de ser processado, na frequência intermediária digital e logo após transformá-lo novamente para um sinal analógico.

Serão apresentados e detalhados cada elemento deste módulo, que é composto por conversores A/D e D/A, DDC e o DUC, como ilustrado na figura 3.8.

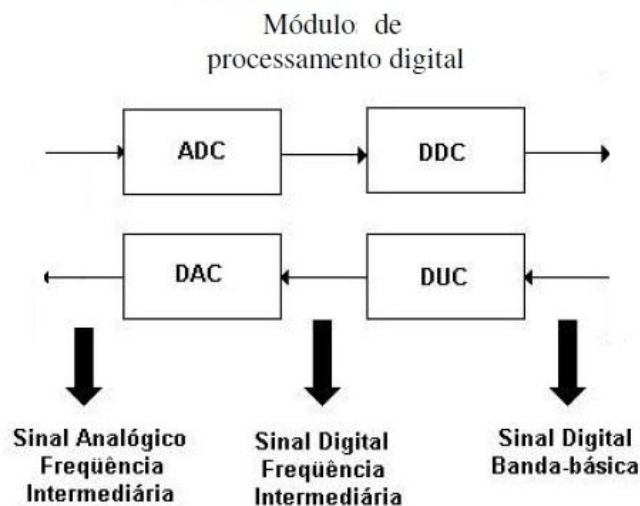


Figura 3.8: Módulo de processamento digital

3.3.1 Conversores A/D e D/A (DAC E ADC)

Como já visto neste trabalho os conversores analógico-digital e digital-analógico são componentes determinantes em um RDS e estão entre os mais importantes no projeto. Em muitos dos casos, eles definem a largura de banda, a faixa de frequência e o consumo de energia do rádio. Sabe-se que a largura de banda e a faixa dinâmica de frequência do sinal analógico tem que ser compatível com os conversores, no caso de um RDS, que na maioria trabalha na casa de GHz, os DAC e ADC se tornam um limitador. Por exemplo, um sinal de 1 GHz de frequência necessita de uma faixa dinâmica de 20 bits para ser amostrado, além de outros fatores limitadores dos conversores, o que na tecnologia atual não é muito comum de encontrar, pois infelizmente os avanços nos estudos de conversores está muito a quem das outras tecnologias relacionadas com RDS [17].

3.3.1.1 Teorema de Nyquist

O teorema de Nyquist é um princípio para o estudo da conversão analógico-digital. Segundo afirma o Dr. André Gustavo Monteiro Lima [21]:

O conceito básico para o entendimento das técnicas de conversão analógico-digital é o critério de Nyquist, o qual estatui que: “um sinal com largura de banda f_a deve ser amostrado a uma taxa $f_s > 2.f_a$ ou as informações do sinal serão perdidas”. Este teorema baseia-se no fato de que quando um sinal analógico contínuo no tempo é amostrado uniformemente, o espectro do sinal é repetido nos múltiplos inteiros da frequência de amostragem. Cada repetição do espectro define uma zona de Nyquist.

A figura 3.9 demonstra que se $f_s = 2.f_a$ os espectros do sinal se repetem um do lado do outro, sem espaçamento nem sobreposição entre eles. Observando isso, nota-se que quanto mais f_s for maior que f_a , mais distantes as repetições de espectro se tornam, ou seja, só haverá sobreposição de espectro de f_s for menor de $2.f_a$.

Após ser observado isso, podemos definir a largura de banda de Nysquist como sendo o espectro da primeira zona de Nyquist, localizada entre o nível DC e $f_s / 2$. Um fator importante a se comentar é a hipótese de sinais indesejados acima da frequência do sinal alvo, este sinal aparecerá como imagem na primeira zona de Nysquist, caracterizado como ruído *aliasing*.

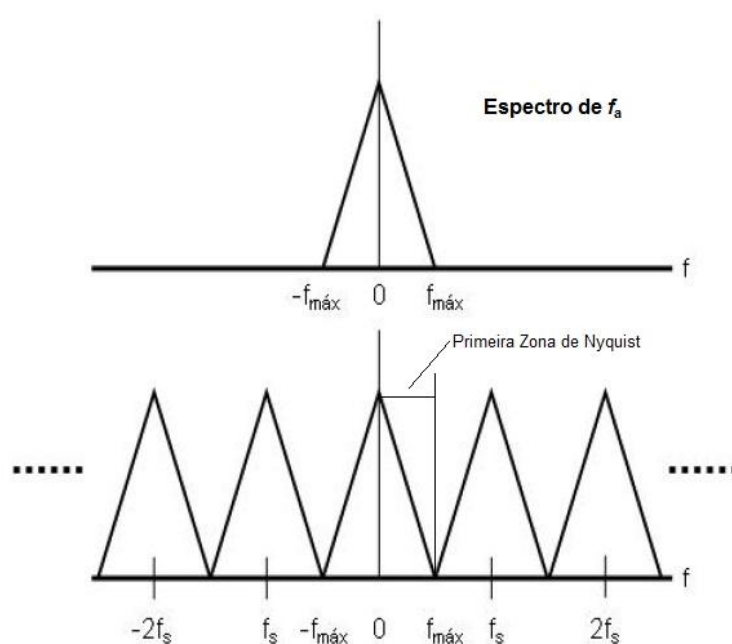


Figura 3.9: Teorema de Nysquist

A filtragem deste sinal, com um filtro denominado *anti-aliasing*, diminui muito a probabilidade de sinais imagem aparecerem indesejavelmente. A especificação e projeto desse filtro devem ser muito bem estudados em relação ao seu custo x benefício [21].

O projeto de um filtro *anti-aliasing* deve atender critério de Nyquist. Para que isso ocorra o filtro tem que apresentar uma transição de atenuação máxima no espaço compreendido entre X e $(f_s - X)$, onde X é a frequência de corte do filtro, veja também que a atenuação máxima do sinal deve ser considerada no início do intervalo ou faixa dinâmica do conversor AD [33], como é descrito na figura 3.10.

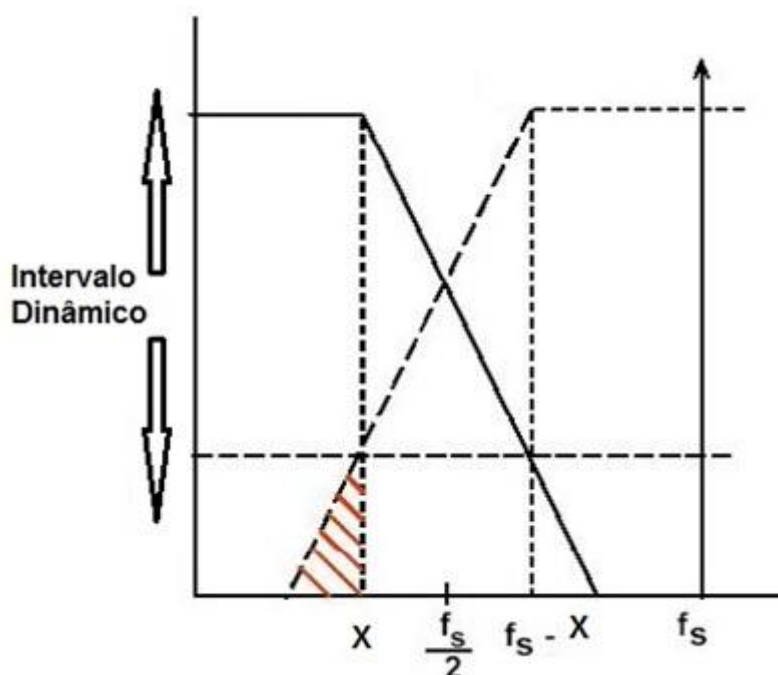


Figura 3.10: Filtro *anti-aliasing* atendendo o critério de Nyquist

3.3.1.2 Super Amostragem do Sinal

Um facilitador para o projeto de um filtro *anti-aliasing* é a técnica de super amostragem de um sinal. Esta técnica está baseada em usar uma frequência de amostragem maior, desta maneira a distância entre duas zonas de Nyquist consecutivas será maior enquanto o intervalo dinâmico não é modificado. Essa afirmação fica clara na figura 3.11, onde temos um filtro com a caída de corte menos brusco.

A desvantagem desse método é o uso de conversores que tenham uma taxa de amostragem maior, ou seja, uma maior capacidade de processamento.

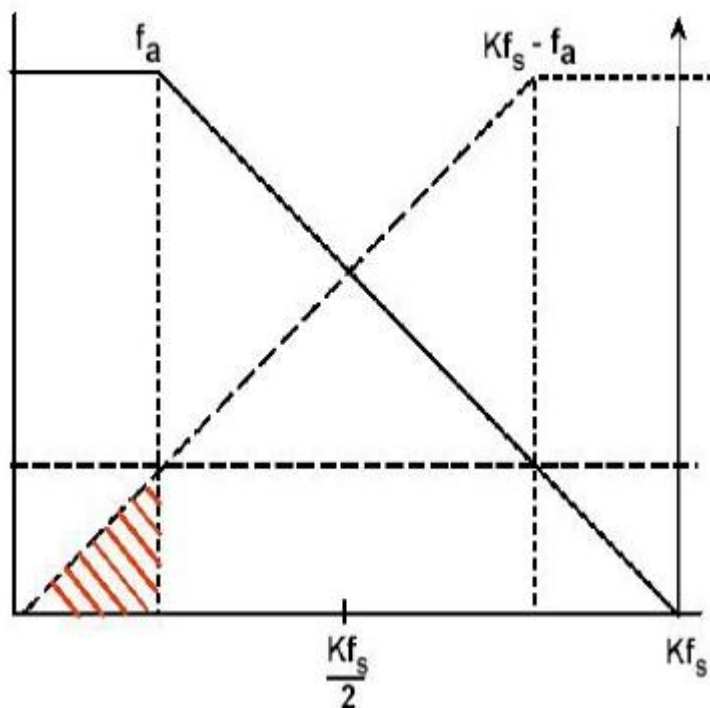


Figura 3.11: Filtro *anti-aliasing* usando Superamostragem

3.3.1.3 Subamostragem do Sinal

Com a condição de a taxa de amostragem do sinal ser no mínimo duas vezes a largura de banda do sinal amostrado, o método de subamostragem do sinal pode ser utilizado, este método também é conhecido como amostragem passa-banda ou amostragem harmônica. Seu princípio é o submeter o sinal à um filtro passa-banda e o replicá-lo no domínio da frequência em cada zona de Nyquist [32].

As frequências mínima e máxima do sinal são determinantes para que não haja interferência de sinais indesejados, e as equações 3.1 a 3.6 devem ser respeitadas [32]:

$$f_s > 2 \cdot (f_{max} - f_{min}) \quad (3.1)$$

$$2 \cdot (f_{max} - f_{min}) \leq f_s \leq 2 \cdot f_{max} \quad (3.2)$$

$$\frac{2 \cdot f_{max}}{k} \leq f_s \leq \frac{2 \cdot f_{min}}{k-1} \quad (3.3)$$

$$k \in Z \quad (3.4)$$

$$2 \leq k \leq \frac{f_{max}}{f_{max} - f_{min}} \quad (3.5)$$

$$f_{max} - f_{min} \leq f_{min} \quad (3.6)$$

3.3.1.4 Arquitetura dos Conversores

A busca por melhoras nos conversores fez com que fossem criados vários tipos de arquiteturas diferentes. Nos estudos de RDS é importante conhecer como funcionam os conversores A/D e seus tipos de arquitetura, algumas delas serão abordadas nesta parte do trabalho.

A arquitetura de conversão mais veloz do mercado são os conversores paralelos, também conhecidos como arquitetura *flash* [21]. Utilizam de lógica combinacional para efetuar a conversão e tem tensões de referência para comparar as tensões de entrada, descrevendo a saída de forma binária. Por se tratar de uma arquitetura paralela, essa arquitetura necessita de vários comparadores para aumentar a precisão de saída.

A figura 3.11 mostra a arquitetura de um conversor em dois estágios. O primeiro ele compara a entrada analógica com as tensões de referência e depois ele aplica uma lógica decodificadora transformando em um código de n bits.

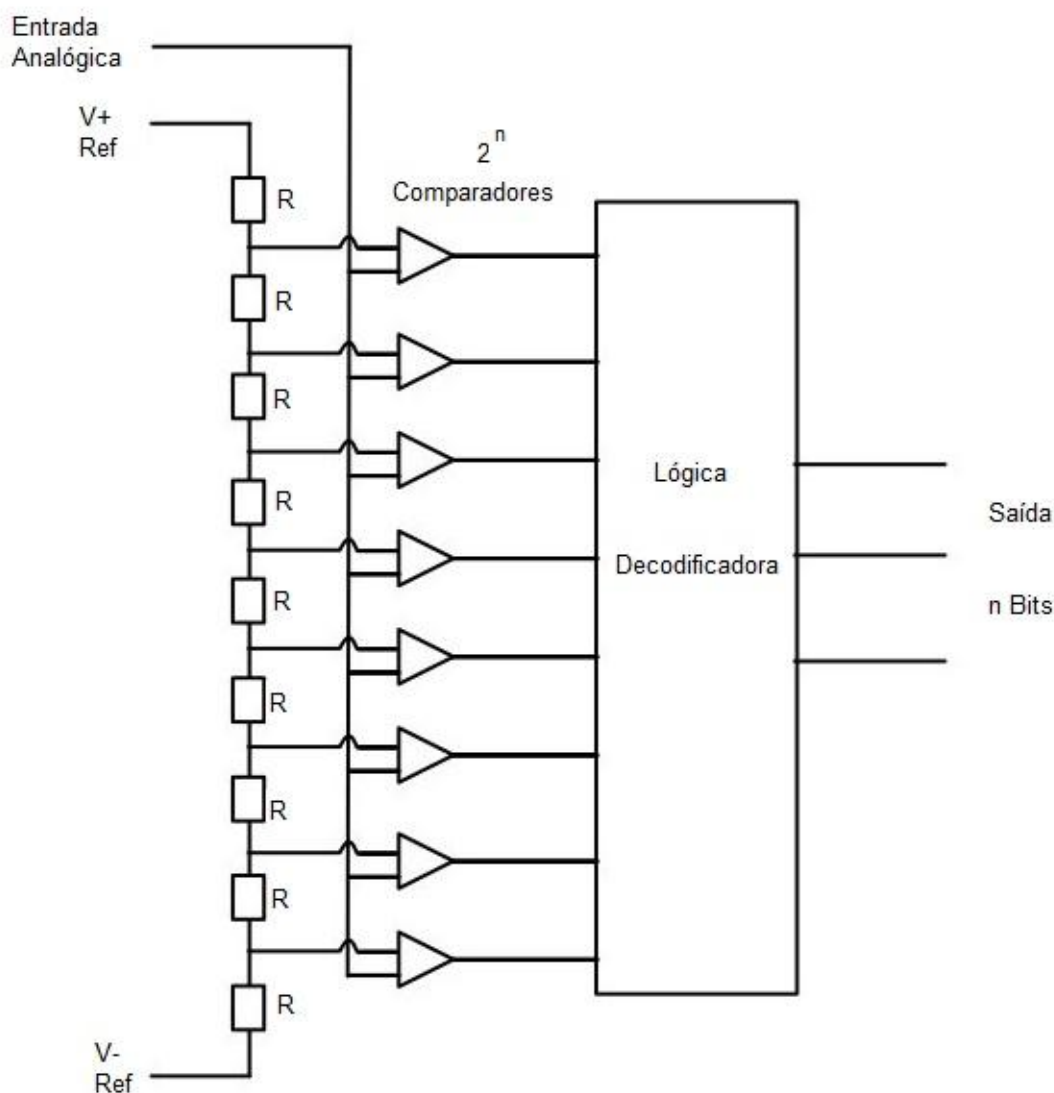


Figura 3.12: Conversor AD Paralelo

Conversores muito utilizados em sistemas de comunicação hoje em dia são os conversores sigma-delta, com arquitetura simples eles utilizam um comparador para verificar se o sinal de entrada é maior que o de saída, se a entrada for menor que a saída, esta é decrementada. Por trabalharem com uma frequência de amostragem superior a taxa de amostragem de Nyquist fazendo com que o ruído de quantização seja próximo a zero [32].

A figura 3.13 mostra a arquitetura de um conversor sigma-delta, onde pode ser observado que sua principal vantagem é a simplicidade do circuito, acarretando em um baixo custo, o que justifica o porquê de serem os conversores mais desenvolvidos tecnologicamente.

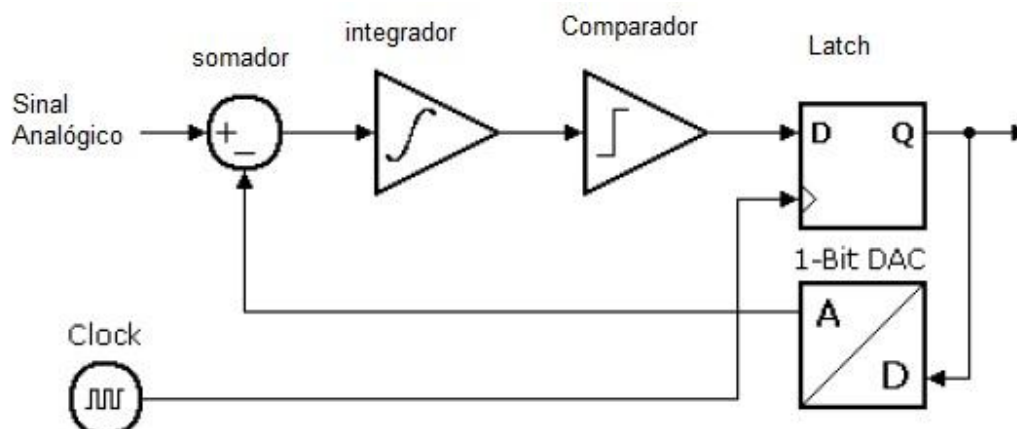


Figura 3.13: Conversor AD Sigma-delta

O sinal entra no integrador, onde cria uma forma de onda triangular proporcional à tensão do sinal analógico. Esta forma de onda triangular encontrada na saída do integrador é comparada com o zero volt pelo comparador, ele pode ser considerado um conversor A/D de 1 bit, já que sua saída terá dois estados, alto ou baixo. A saída do comparador é armazenada em um flip-flop do tipo D. A saída do flip-flop é usada para realimentar o circuito através de um conversor D/A de um bit. Este conversor D/A converterá basicamente o “0” ou o “1” armazenado no flip-flop em uma tensão de referência positiva ou negativa para ser somado na saída do somador. Portanto, o somador efetuará a soma da próxima amostra com o resultado da amostra anterior (uma tensão positiva ou negativa), com o objetivo de manter zero na saída do integrador. O resultado é que na saída do flip-flop teremos uma série de zeros e uns que correspondem ao dado amostrado: o nível médio da cadeia de bits representa a tensão média do sinal de entrada analógico. Como o clock usado no flip-flop é muito alto, o dado é amostrado várias vezes, quanto maior o clock, maior a precisão de um conversor A/D sigma-delta.

Outra arquitetura bem popular é o conversor de múltiplo-estágio, que tem como principal ideia a escalabilidade do sistema. Seu funcionamento básico consiste em digitalizar o sinal, transformá-lo novamente em analógico e fazer uma subtração desse sinal adquirido do sinal de entrada, esse resultado passará para o próximo estágio que segue essa mesma filosofia, como mostrado na figura 3.14.

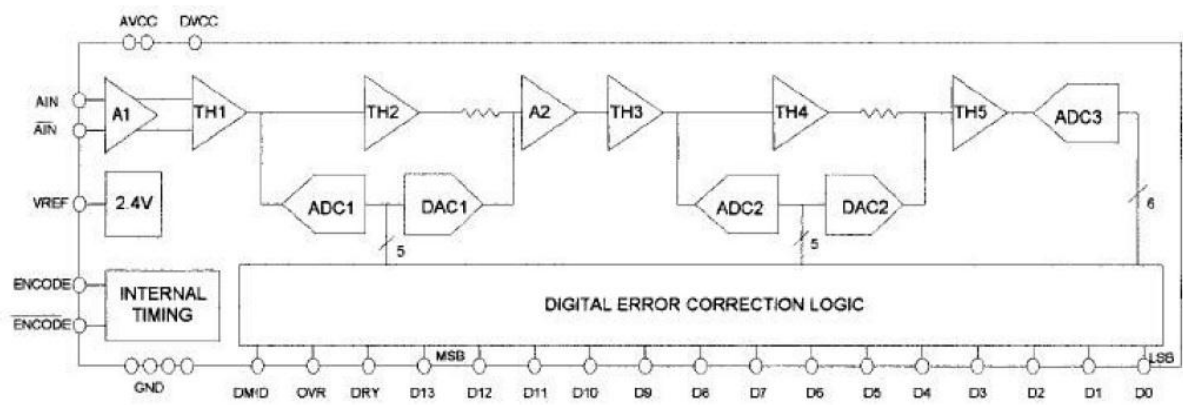


Figura 3.14: Conversor AD de Múltiplo-estágio

3.3.2 DDC – *Digital Downconverter*

Na sequência de estudo do módulo de processamento digital, está o *digital downconverter* (DDC), responsável por mais um tratamento no sinal antes de ser processado pelo processador do RDS. Este tratamento consiste em trazer o sinal, que já está digitalizado em uma frequência intermediária, para uma frequência em banda-base.

Seu funcionamento consiste em mixar o sinal, logo após ser digitalizado no ADC, em duas senóides geradas por um oscilador local, com frequência intermediária e em quadratura, decompondo assim o sinal em componentes complexas. Para seleccionar a faixa de frequência de interesse, o sinal passa por um filtro decimador, reduzindo assim a taxa de dados enviados ao processador [30]. A arquitetura de um DDC está demonstrada na figura 3.15.

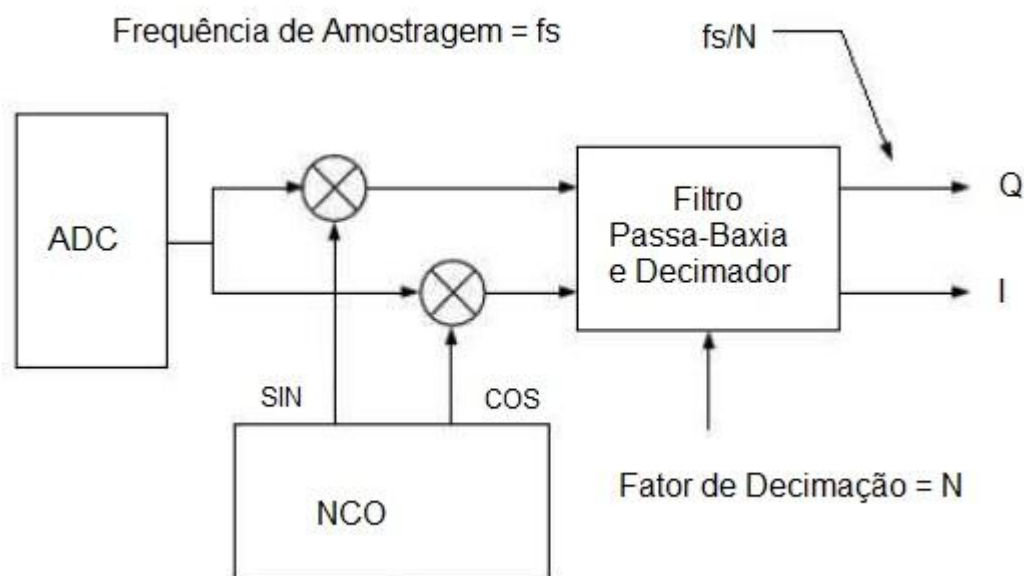


Figura 3.15: Arquitetura de um DDC

As duas senóides, mencionadas no parágrafo anterior, são geradas através de um componente comandado pelo próprio processador do RDS, o NCO (*Numerically Controlled oscillator*). Segundo Rafael Schena [28],

Um NCO é um bloco que, basicamente, tem como entrada um *clock* e um incremento de fase, a cada ciclo do *clock* ele incrementa o seu acumulador de fase e

apresenta na saída o seno e o cosseno da fase armazenada no acumulador. Há também a possibilidade de se ter entradas para sinais digitais para modulações em frequência ou fase das portadoras digitais geradas pelo NCO, abrindo a possibilidade de usar o NCO como um modulador digital de frequência ou de fase.

O NCO pode ser implementado em várias arquiteturas, como uma tabela de *look-up* em uma memória ROM ou utilizando um algoritmo CORDIC.

As arquiteturas de tabela de *look-up*, podem se basear em grandes ou pequenas ROM's. A de grande ROM armazenam valores de senos e cossenos para todos os 360 graus do ciclo trigonométrico, já a de pequena ROM armazena valores de seno e cosseno para 45 graus do ciclo trigonométrico, calculando os outros valores através de relações trigonométricas. O CORDIC⁶ é um algoritmo numérico que calcula funções trigonométricas através de rotações fasoriais iterativas, calculando coordenadas cartesianas de um vetor que roda sobre um ângulo arbitrado, e é de excelente desempenho e uma saída para sistemas que não possuem memória interna [28].

3.3.3 DUC – Digital Up-Converter

O *Digital Up-converter* (DUC) tem a função contrária do DDC, ele transporta o sinal processado pelo RDS, que está em banda-base, para frequência intermediária novamente, mas ainda como um sinal digital. A figura 3.16 está ilustrando a arquitetura de um DUC.

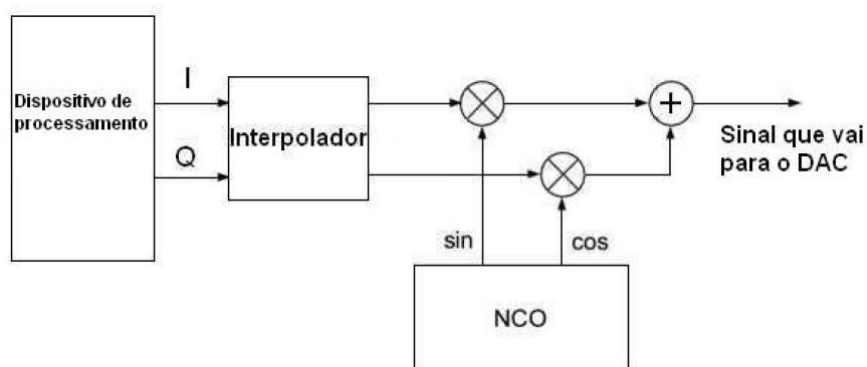


Figura 3.16: Arquitetura de um DUC

⁶ Maiores explicações sobre o algoritmo CORDIC podem se acessadas em <http://www.andraka.com/cordic.htm>.

A unidade de processamento do RDS envia dois sinais complexos ao DUC, que por sua vez interpola os dois sinais para obter um maior número de amostras, depois mixa os sinais interpolados com suas senóides em quadratura e subtrai os dois sinais para que o sinal se torne real e não mais complexo [32].

3.3.4 Processadores de Sinais

A exemplo dos conversores A/D e D/A, os processadores de sinais são outra parte limitadora de um RDS. Sua capacidade de processamento que determina o uso do DDC e DUC, mas ao contrário dos conversores são bem desenvolvidos e seguem uma evolução muito rápida.

Um RDS dispõe de vários tipos de processadores para execução de seu software, como os DSPs (processadores digitais de sinais), os FPGAs (conjunto de portas lógicas programáveis) e os ASICs (circuitos integrados de aplicações específicas) [6].

Entretanto, um rádio definido por software pode ser executado em um microcomputador qualquer, através de plataformas livres⁷ e *front-ends* específicos desenvolvidos para pesquisa e desenvolvimento nesta área. A GNU *Radio* e a USRP são um exemplo de plataformas de pesquisa muito difundidos. Trabalhando em conjunto, a GNU *Radio* para o desenvolvimento do software do rádio, que será executado no próprio computador, e a USRP serve como interface com mundo RF, ou seja, é uma interface de *front-end* universal. Estes dois componentes são o objetivo de estudo do próximo capítulo deste trabalho [6].

⁷ Segue o mesmo conceito de software livre. O GNU Radio é um software livre, mas usamos o conceito de plataforma livre por trabalhar em conjunto com uma placa universal desenvolvida para este *software*. O conceito de software livre se refere a um *software* do “código aberto”, onde qualquer pessoa pode adaptá-lo as suas necessidades e expor as mudanças feitas no código, contribuindo para o desenvolvimento do *software*.

4 UMA PLATAFORMA LIVRE PARA DESENVOLVIMENTO DE RDS

No campo de pesquisas em RDS muitas plataforma podem ser utilizadas. Este trabalho demonstrará a GNU *Radio* [10], uma plataforma livre, flexível e de fácil implementação, que a torna uma ótima ferramenta para desenvolvimento em pesquisas com rádio definido por software. A plataforma GNU *Radio*, trabalha em conjunto com a plataforma USRP, um componente de *hardware* que tornou-se indispensável na implementação de um RDS com a plataforma GNU *Radio*.

4.1 *Universal Software Radio Peripheral Board (USRP)*

Desenvolvida por Matt Ettus [6] para atender pesquisadores de RDS que utilizam GNU *Radio*, a *Universal Software Radio Peripheral board* conhecida como USRP, é uma interface de *hardware* que está entre a antena e o computador onde está sendo executado o RDS. É uma placa que incorpora o ADC e o DAC, uma FPGA para processamentos do sinal antes de o sinal ir para o GNU *Radio*, placas denominadas filhas (interfaces onde o sinal é captado e transmitido através de antenas) e uma interface de comunicação com o computador onde é executado o GNU *Radio*.

A USRP é existente em duas versões a USRP e a USRP2, onde as melhorias da segunda versão em relação a primeira são melhorias nos conversores A/D e D/A, melhorias na FPGA e a mudança de interface de comunicação com o computador que passou de USB 2.0 para interface Ethernet, dando uma maior velocidade ao RDS a ser implementado. Neste trabalho será abordada a primeira versão da USRP, a qual está ilustrada na figura 4.1 [30].

O *hardware* da USRP é composto por uma placa mãe que pode comportar 4 placas filhas. A placa mãe tem em sua composição quatro conversores AD de 12 bits e 64 Msamples/s e quatro conversores DA de 14 bits e 128 Msamples/s que são conectados com os canais de entrada e saída (placas filhas) e controlados por uma FPGA Cyclone EP1C12 da fabricante Altera [30].

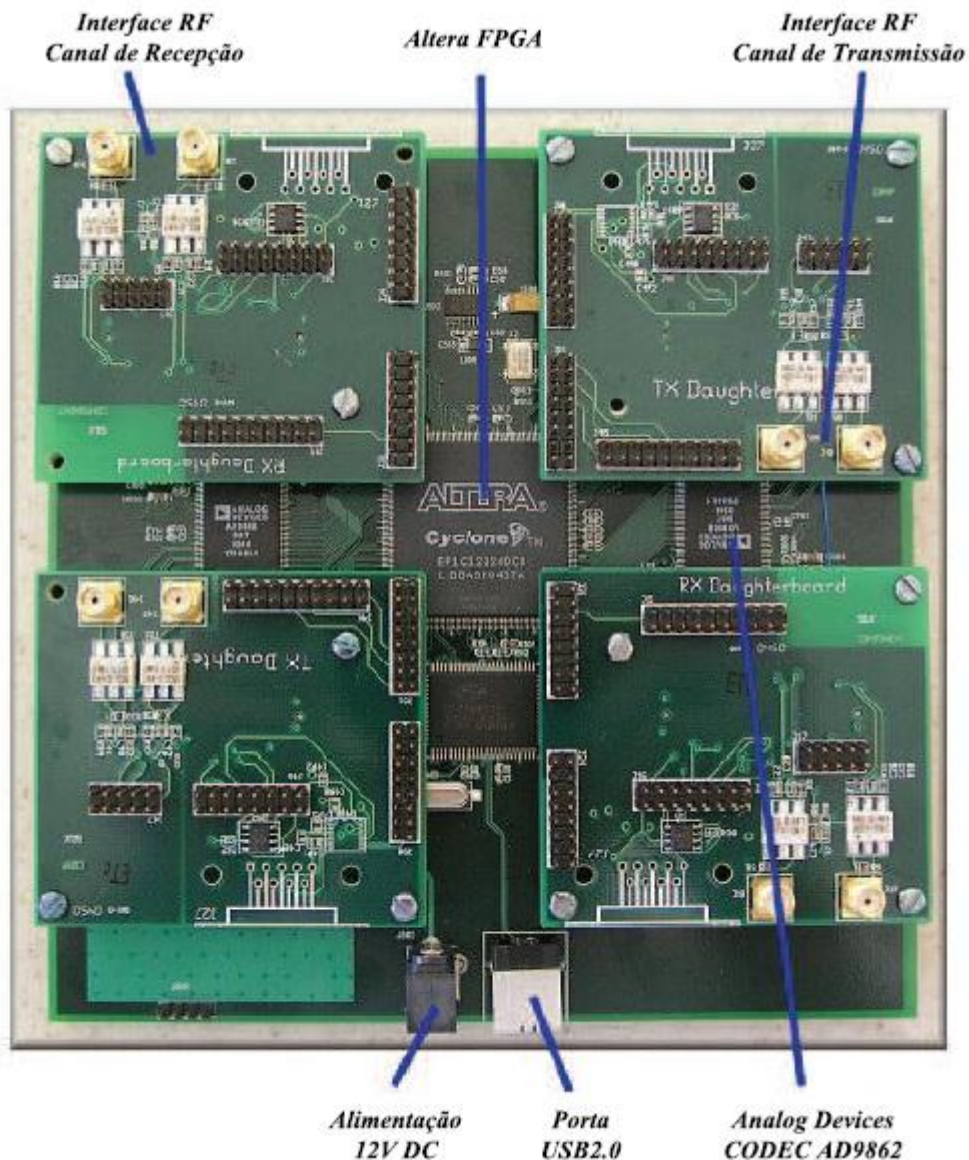


Figura 4.1: Placa USPR versão 1 [6]

Uma particularidade muito importante sobre esta versão de USPR é que sua comunicação com o computador se dá somente através de uma interface USB 2.0, que tem velocidade de 32 MB/s na prática deixando a largura de banda em uma faixa de 8MHz com amostras complexas de 4 bytes [6]. A figura 4.2 mostra o diagrama de blocos da USPR. A USPR pode trabalhar de forma *full duplex*, deixando transmissão e recepção independentes, com a única observação de que a soma das taxas de Tx e Rx não pode ultrapassar os 32 MB/s, pela limitação da comunicação USB 2.0 [30].

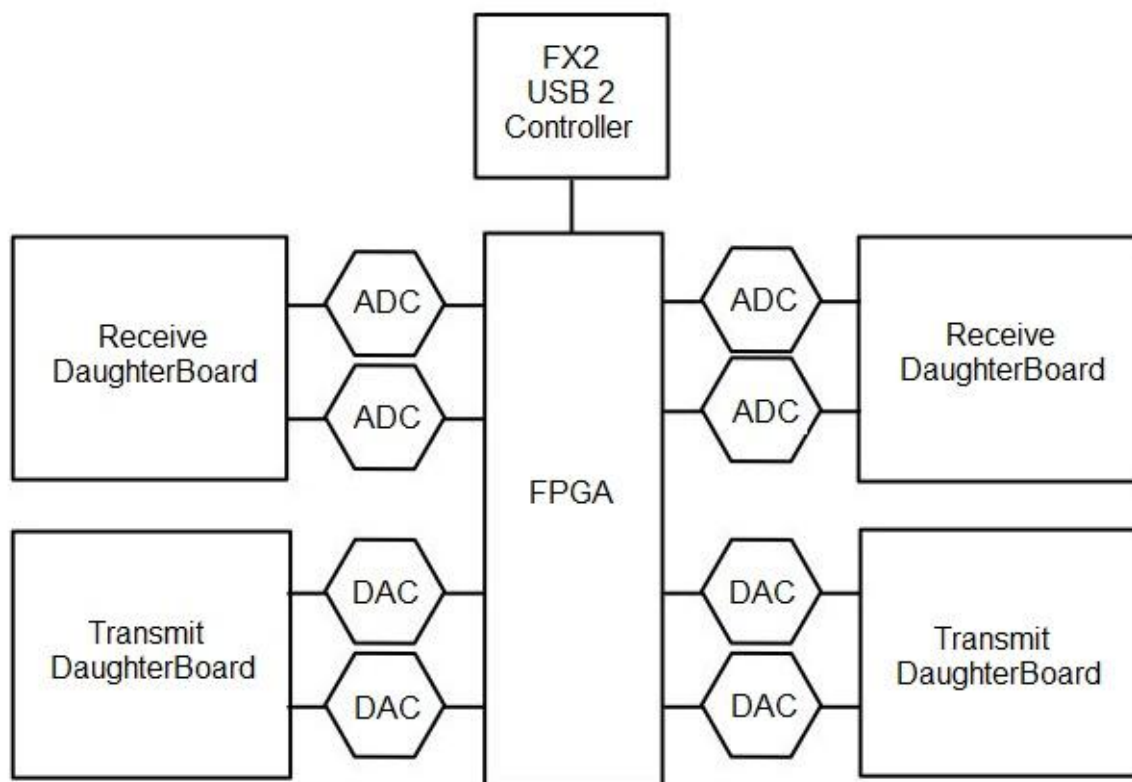


Figura 4.2: diagrama de blocos da USRP

4.1.2 Conversor Analógico-digital

A USRP contém 4 ADC's de 12 bits e taxa de amostragem de 64M amostras por segundo. Esses conversores devem converter no máximo um sinal de 32 MHz para atender o teorema de Nyquist, mas se o sinal for amostrado com uma frequência intermediária superior a esse valor máximo, ocorrerá o fenômeno de *aliasing*, assim mapeando uma banda entre -32 MHz e 32 MHz, o que pode ser útil, por exemplo, em uma aplicação para ouvir frequências de rádio FM sem a utilização de um *front end* RF. Os conversores têm uma faixa dinâmica de 2V pico a pico e impedância de entrada de 50 ohms. Se o sinal de entrada for fraco pode ser amplificado por um amplificador de ganho programável conhecido como PGA, que é ajustado por software e pode amplificar em até 20 dB o sinal de entrada [30].

4.1.3 Conversor Digital-analógico

Quatro DAC's de 14 bits compõe também a placa mãe da USPR para transmissão de sinais, podem converter 128M amostras por segundo. Tem os mesmos 50 ohms de impedância dos ADC's e uma tensão de saída de 1V de pico e corrente variável entre zero a 20 mA. Também dispõe de um PGA de 20 dB [30].

4.1.4 A FPGA

A FPGA contida na placa mãe da USPR tem a função básica de realizar operações matemática em alta velocidade de processamento, e o entendimento dela é uma parte fundamental para os usuários da GNU *Radio*. Como mostrado na figura 4.2, a FPGA é ligada diretamente aos conversores A/D e D/A e tem o papel de reduzir a taxa de dados que vai ao computador através da interface USB 2.0.

Por padrão essa FPGA vem configurada com alguns componentes. Dentre eles estão os DDC's que são implementados com 4 Filtros CIC (*Casaded Integrator-Comb*), filtros de alto desempenho implementados através de somas e atrasos. Para rejeição de sinais fora da banda desejada, está implementado um filtro de 31 coeficientes em cascata com os filtros CIC. Por padrão 2 DDC's, mas há a possibilidade de implementar até 4 DDC's [30].

A arquitetura do DDC implementado no padrão da FPGA pode ser visualizada na figura 4.3, onde cada DDC em uma entrada I e uma entrada Q que está em quadratura. Qualquer um dos 4 ADC's pode ser enviado para entrada dos DDC's, que pode receber sinal de mais de um ADC, intercalando a saída em sequência [30].

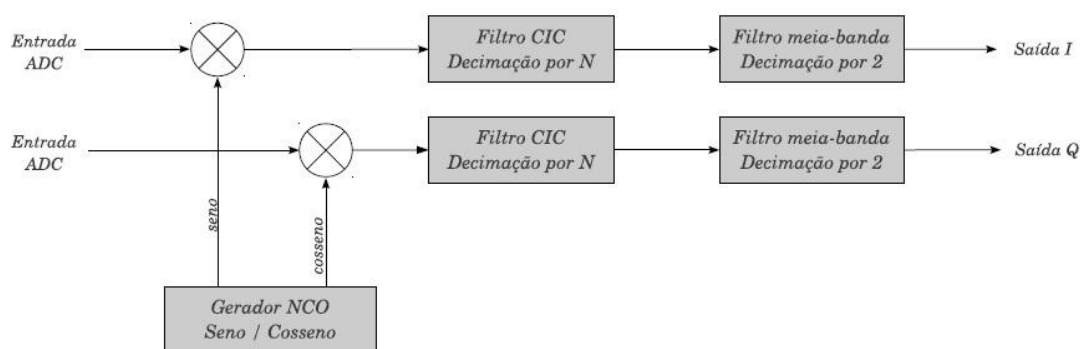


Figura 4.3: Arquitetura de uma DDC padrão na USPR

No caminho inverso, o caminho de transmissão, a configuração padrão ocorre da mesma maneira. O DUC interpola o sinal que veio através da interface USB 2.0 e o converte para frequência intermediária. Como apresentado na figura 4.4.

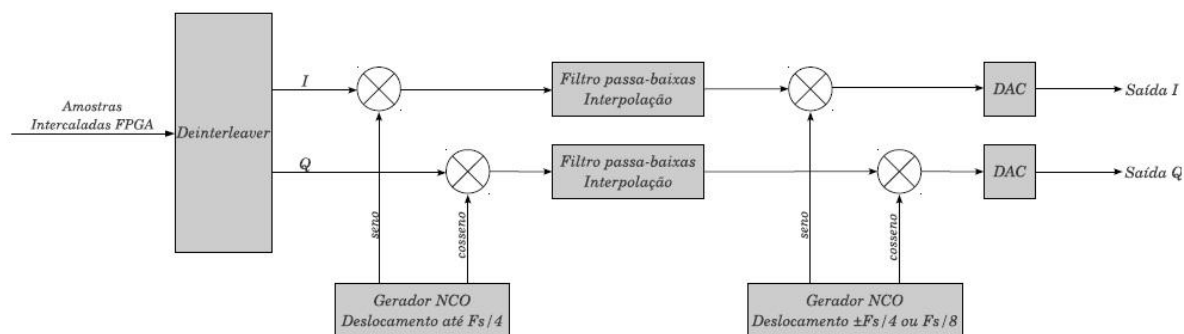


Figura 4.4: Arquitetura de uma DUC padrão na USRP

4.1.4 Placas Filhas

A placa mãe de um USRP pode comportar quatro placas filhas, duas placas de transmissão e duas placas de recepção, elas são utilizadas para ligar o mundo RF à USRP, e podem ser encontradas em várias frequências de operação diferentes, podendo trabalhar em frequências de até 5,9 GHz [6].

4.2 GNU Radio

A fim de demonstrar o funcionamento de um RDS através de uma plataforma de software livre, com flexibilidade e facilidade de programação, este trabalho apresentará a GNU Radio [9], que é muito utilizado no campo de pesquisa dessa tecnologia. Licenciada pela GPL (*General Public License*), qualquer pessoa pode implementar um RDS usando os blocos de processamento de sinal contidos no GNU Radio.

Segundo Eric Blosson [3], um dos fundadores do projeto, a GNU Radio tem como objetivo “trazer o código mais próximo possível da antena, transformando assim problemas de *hardware* em problemas de *software*”.

Este *software* foi projetado para processar sinais digitais complexos em banda base, que como já vimos é todo preparado pela USRP. Seu funcionamento consiste em ligar

blocos de elementos, encontrados em sistemas de rádios transformados em *software* através de operações matemáticas, já pré-implementados no próprio GNU *Radio*, mas se o usuário precisar de um bloco específico pode ele mesmo implementá-lo [27]. Dentre os blocos que compõe está plataforma de *software* livre, podemos mencionar:

- Operações matemáticas, como soma, multiplicação, subtração, logaritmo, entre outras;
- Portas lógicas;
- Moduladores, como OFDM, QAM, DPSK, entre outros;
- Filtros FIR, IIR, passa-banda, passa-baixa, etc;
- Interpolação e decimação;
- Blocos de ligação com a USPR e USPR2;
- Controles de ganho;
- *Scramblers* (embaralhadores de sinal);
- Corretores de erros como alguns tipos de treliças;
- A FFT (Transformada Rápida de Fourier);

Ou seja, são muitos blocos de elementos já implementados, o que torna as pesquisas e projetos em RDS muito mais rápidos de ser implementados.

Esses blocos são denominados de:

- *Flow Graphs* ou grafos, são blocos que tem como função o processamento de sinais, são interligados entre si formando assim um fluxo para o sinal.
- *Sources* chamados de fontes, como o próprio nome já diz são fontes de sinal.
- *Sinks* chamados sorvedouros, são blocos sem portas de saída, somente recebem sinal.

A implementação de um RDS em GNU *Radio* é feita através de duas linguagens de programação. A linguagem C++ é destinada para o processamento dos sinais, por ser uma linguagem de baixo nível e um alto desempenho de processamento. Já a interligação dos blocos é feita através da linguagem Python, uma linguagem de programação em alto nível que dá uma maior facilidade e agilidade para fazer as ligações. A utilização das duas linguagens de programação é possível através do uso da SWIG (*Simplified Wrapper and Interface Generator*), que cria uma interface entre as duas linguagens [10].

Existem quatro tipos de dados para representação das amostras que estão nos blocos do GNU *Radio*, que ainda podem ser representados através de vetores [27]. Os quatro tipos são:

- *Complex*: dados de 8 bytes representados pela letra c.
- *Float*: dados de 4 bytes representados pela letra f.

- *Short int*: dados de 2 bytes representados pela letra s.
- *Char*: dados de 1 byte representados pela letra b.

4.2.1 GRC - GNU *Radio Companion*

A implementação de um RDS pode se tornar ainda mais rápida. A GRC (GNU *Radio Companion*) nada mais é do que uma interface gráfica criada para o GNU *Radio*. Esta ferramenta trouxe uma maior facilidade nas implementações de RDS, por ser mais intuitiva do que a programação em GNU *Radio*, principalmente para pesquisas acadêmicas onde há necessidade de uma maior velocidade na criação de protótipos.

4.3 Exemplos Práticos de Aplicação

Esta sessão apresentará um exemplo simples e funcional de rádios definidos em *software*, implementados por Alexandru Csete e dispostos em seu site [14]. Esse dentre outros exemplos estão dispostos com os códigos em GNU *Radio* e sua forma gráfica através do GNU *Radio Companion*, além de ser demonstrado seu funcionamento através de vídeos comprovando que são implementações funcionais. Para o funcionamento deste exemplo, foram feitas algumas modificações no software implementado, pois as placas filhas utilizadas não eram as mesmas que Alexandru Csete usou em seus exemplos.

Para execução deste exemplo foram utilizados dois códigos disponibilizados por Csete em seu site, um receptor de rádio AM e um transmissor AM-SSB LSB⁸. As figuras 4.5 e 4.10 mostram os blocos implementados já com as modificações necessárias.

A transmissão do sinal se deu através de duas placas USRP com placas filhas WBX. Essas placas trabalham com frequências entre 50 MHz e 2.2 GHz, e o exemplo foi implementado para transmissão e recepção com frequência de 432,15 MHz [6]. O apêndice B trará a especificação de algumas placas sugeridas e das duas versões de USRP, informando como adquiri-las.

⁸ AM-SSB significa você transmitir o sinal AM com uma só banda, no caso do LSB somente a banda inferior do sinal.

Os códigos foram executados no GNU *Radio Companion* (GRC) versão 3.2.2 na distribuição Ubuntu 11.04. As instruções de instalação e execução do GRC estão descritas no apêndice A.

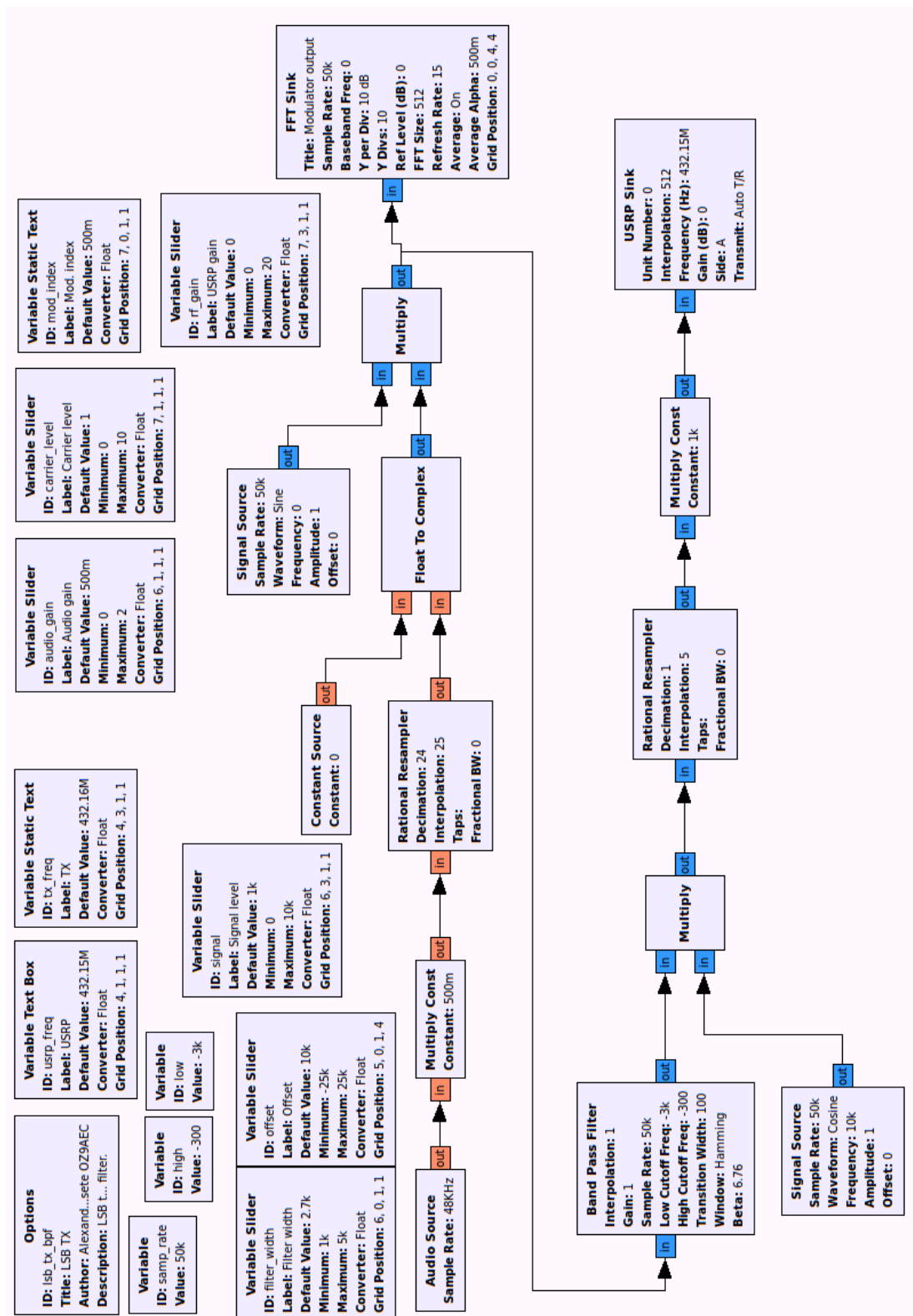


Figura 4.5: Transmissor AM-SSB LSB implementado em GRC.

O código ilustrado na Figura 4.5 mostra um transmissor AM-SSB LSB. O bloco “*Options*” é o bloco principal do Rádio, onde constam as configurações da janela do rádio. As variáveis dão parâmetros a configuração e ajuste do rádio.

O sinal foi capturado pela placa de som do computador através do bloco “*Audio Source*”. Após isso, o sinal é processado e modulado em AM com uma portadora de 50 KHz no primeiro bloco “*Multiply*”, sendo ilustrado com e sem a presença de voz nas figuras 4.6 e 4.7, respectivamente. Essa ilustração é feita através de um *plot* do espectro do sinal modulado no bloco “*FFT Sink*”, onde está implementado a Transformada Rápida de Fourier. Os principais blocos do transmissor ilustrado na figura 4.5 são:

- O bloco *Audio Source*: responsável por capturar o sinal de voz na placa de som;
- Os dois blocos *Signal Source*: responsáveis por criar as portadoras do modulador;
- Os dois blocos *Multiply*: responsáveis por multiplicar o sinal pelas portadoras;
- O bloco *FFT Sink*: que aparece uma vez só na figura, mas foi utilizado duas vezes para capturar o espectro do sinal em pontos diferentes;
- O bloco *Pass Band Filter*: implementa um filtro passa faixa, utilizado para eliminar a portadora e a banda superior do sinal AM;
- O bloco *USRP Sink*: que trata o sinal e sincroniza o programa com a placa USRP, desta forma o sinal pode ser transmitido ao meio físico;
- Os blocos de variáveis e variáveis *slider*, são parâmetros do rádio. As variáveis são fixas durante a execução do programa, enquanto as variáveis *slider* tem parâmetros de início e fim que podem ser modificados durante a execução do programa;
- Os demais blocos servem para tratar o sinal, para que possam ser processados pelos blocos *Flow Graphs*.

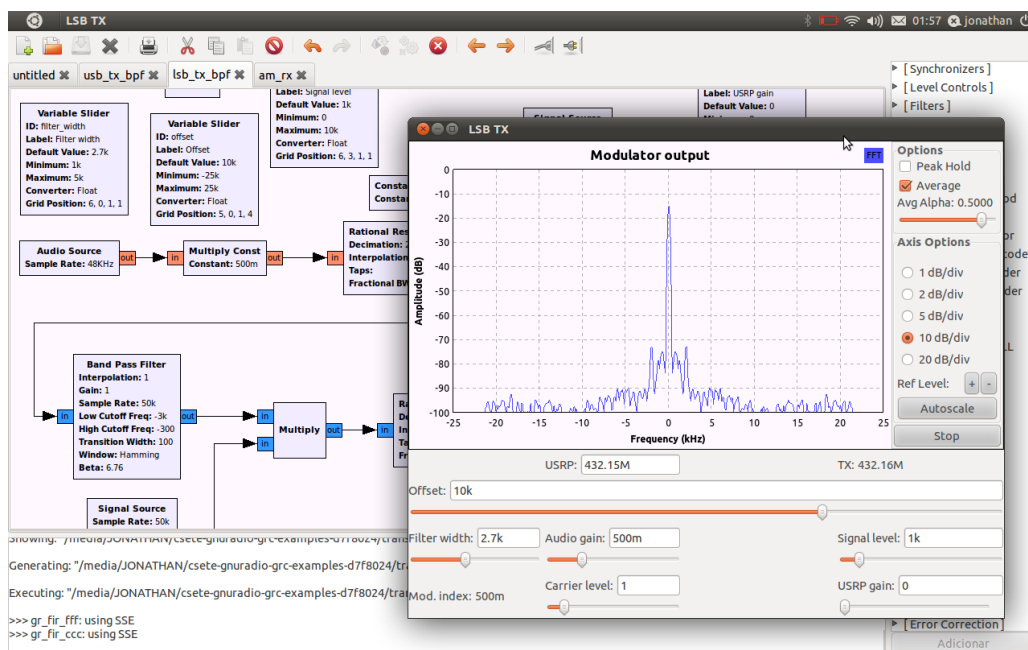


Figura 4.6: Espectro AM sem sinal de voz

Na figura 4.6 o espectro mostra a portadora e as bandas laterais de um sinal AM, onde pode-se notar uma pequena amplitude nas bandas laterais caracterizadas por um pequeno ruído de fundo causado pelo ambiente e pela má qualidade do microfone.

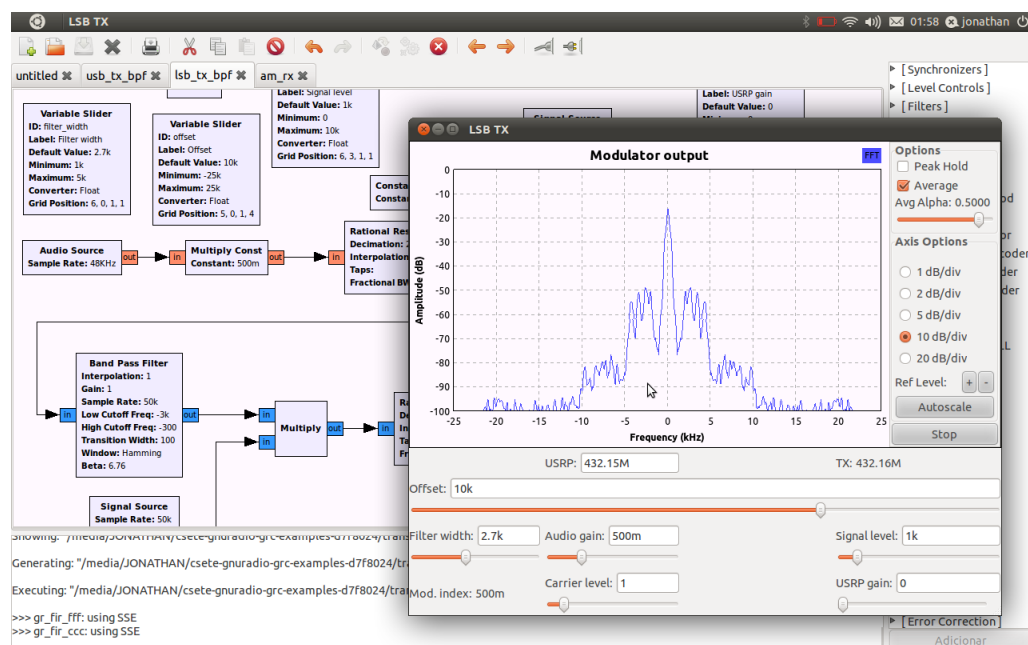


Figura 4.7: Espectro AM com sinal de voz

Na figura 4.7 o espectro mostra a portadora e as bandas laterais de um sinal AM, onde as bandas laterais mostram o sinal de voz sendo modulado.

Para ser modulado em AM-SSB LSB o sinal passa por um filtro passa-faixa e é mais uma vez multiplicado por uma portadora de 50 KHz, e pode ser visualizado com e sem a presença de voz na figuras 4.8 e 4.9, respectivamente. Os gráficos foram obtidos transferindo a entrada do bloco “FFT Sink”, da saída do primeiro para o segundo bloco “Multiply”. Após isso, o sinal é transmitido para a placa USRP via USB, para que isso seja possível foi configurado o bloco “USRP Sink”, onde consta uma frequência de 432.15 MHz a qual o sinal é transmitido no meio.

Na figura 4.8 é observada uma pequena amplitude na banda inferior caracterizada pelo ruído de fundo do ambiente e má qualidade do microfone. Já na figura 4.9, observa-se o sinal de voz modulado na banda inferior do sinal AM.

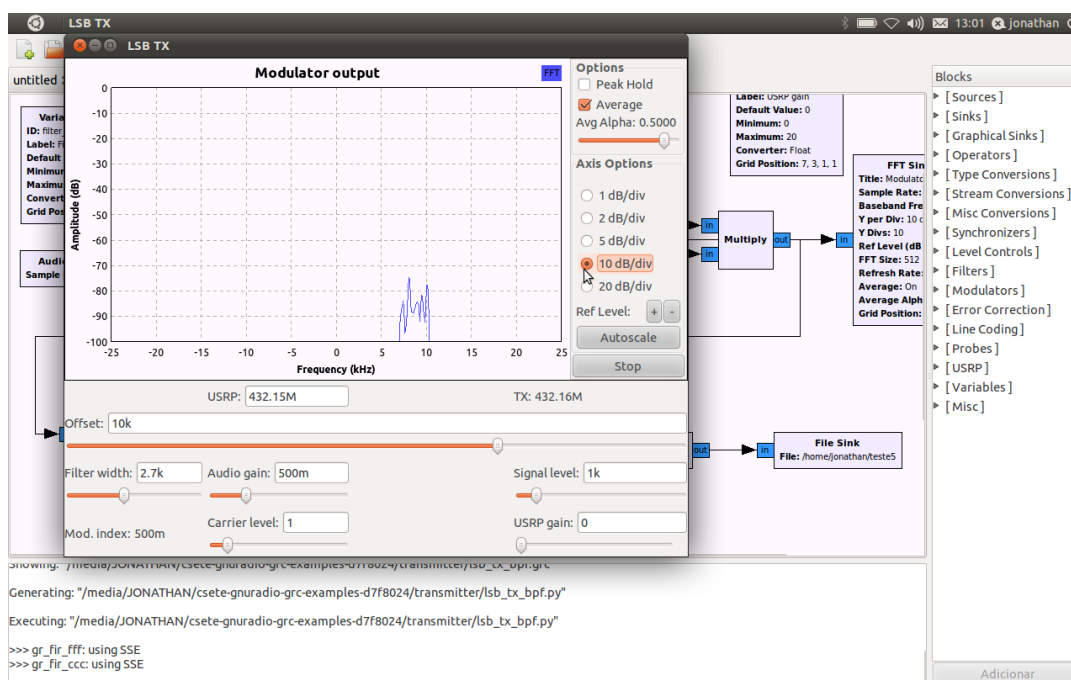


Figura 4.8: Espectro AM-SSB LSB sem sinal de voz.

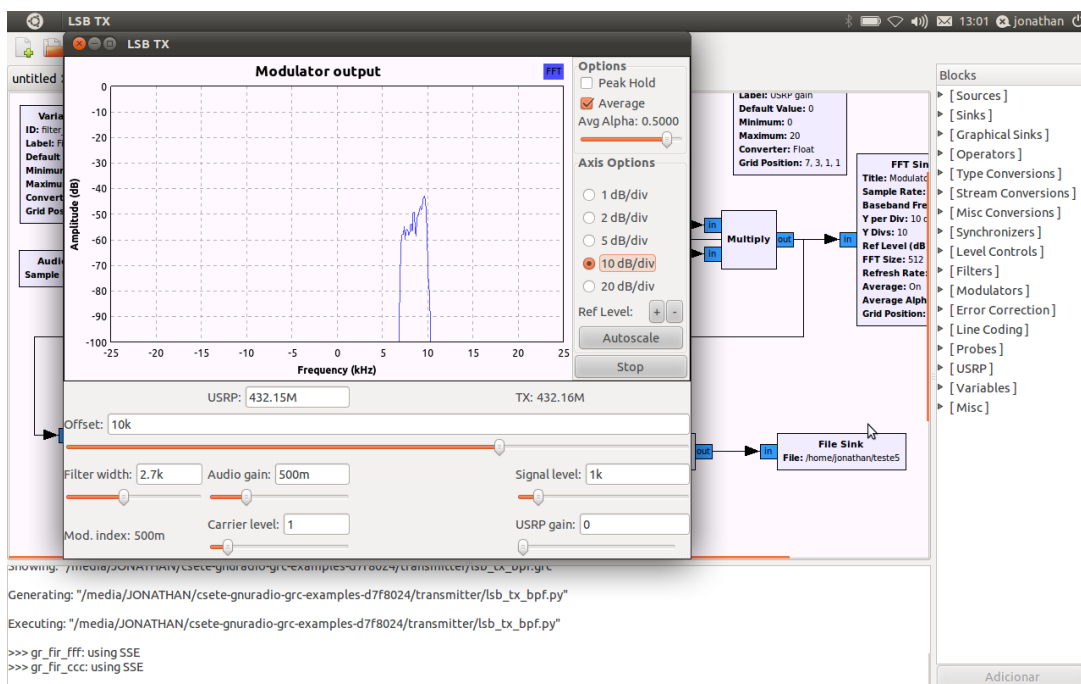


Figura 4.9: Espectro AM-SSB LSB com sinal de voz.

Seguindo para segunda parte do exemplo, a recepção de um receptor com demodulador AM, a figura 4.10 mostra os blocos do rádio. Este exemplo foi modificado do exemplo original desenvolvido por Csete, onde a USRP recebe o sinal na frequência de 7.2 MHz, como o sinal foi transmitido em 432,15 MHz foi modificado a variável de configuração do bloco “USRP Source” que está localizada no bloco “Variable Static Text”.

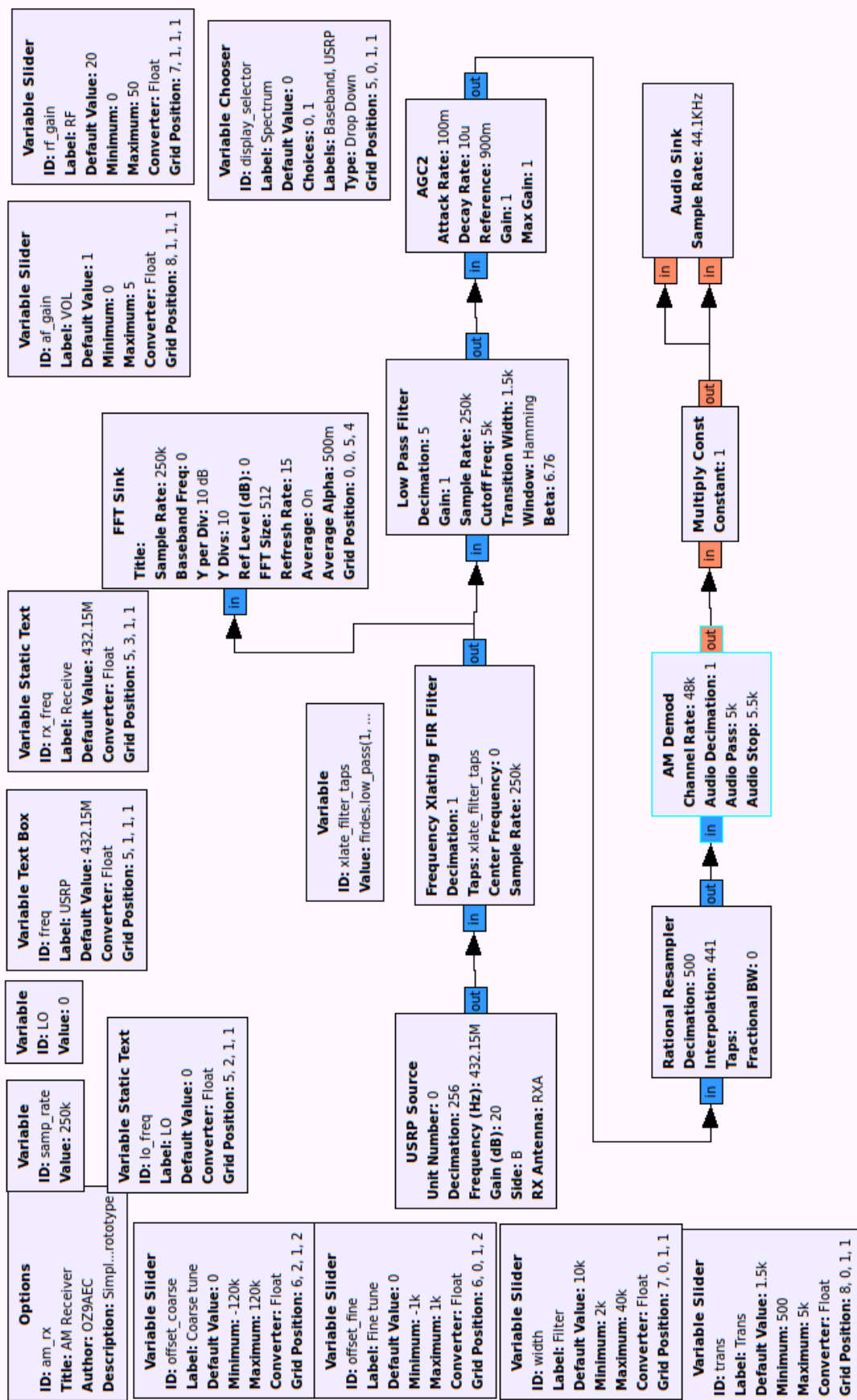


Figura 4.10: Transmissor AM-SSB LSB implementado em GRC.

O sinal obtido na placa USRP é sincronizado com o GNU *Radio* através do bloco “USRP *Source*”. O espectro do sinal recebido pode ser visualizado através da implementação do bloco “FFT *sink*” e está ilustrado nas figuras, 4.11 sem sinal de voz e 4.12 com sinal de voz. Após isso, o sinal é filtrado no bloco “*Low Pass Filter*” e passa por um ajuste de ganho no bloco “AGC2” antes de ser demodulado no bloco “AM Demod”. O bloco “AM Demod” é um demodulador AM já implementado no GRC. O áudio do sinal recebido é escutado através da própria placa de som do computador através do bloco “*Audio Sink*”. Os principais blocos da figura 4.10 são:

- O bloco USRP *Source*: sincroniza o programa com a placa USRP e recebe o sinal, desta forma o sinal pode ser processado pelos blocos do programa;
- O bloco FFT *Sink*: utilizado para capturar o espectro do sinal recebido pela USRP;
- O bloco *Low Pass Filter*: implementa um filtro passa baixa, utilizado para que seja capturado somente as frequências de interesse;
- O bloco AGC2: é um controle de ganho no sinal;
- O bloco AM Demod: é um demodulador AM já implementado pelo GRC;
- O bloco *Audio Sink*: responsável por enviar o sinal de voz para placa de som;
- Os blocos de variáveis e variáveis *slider*, são parâmetros do rádio. As variáveis são fixas durante a execução do programa, enquanto as variáveis *slider* tem parâmetros de início e fim que podem ser modificados durante a execução do programa;
- Os demais blocos servem para tratar o sinal, para que possam ser processados pelos blocos *Flow Graphs*.

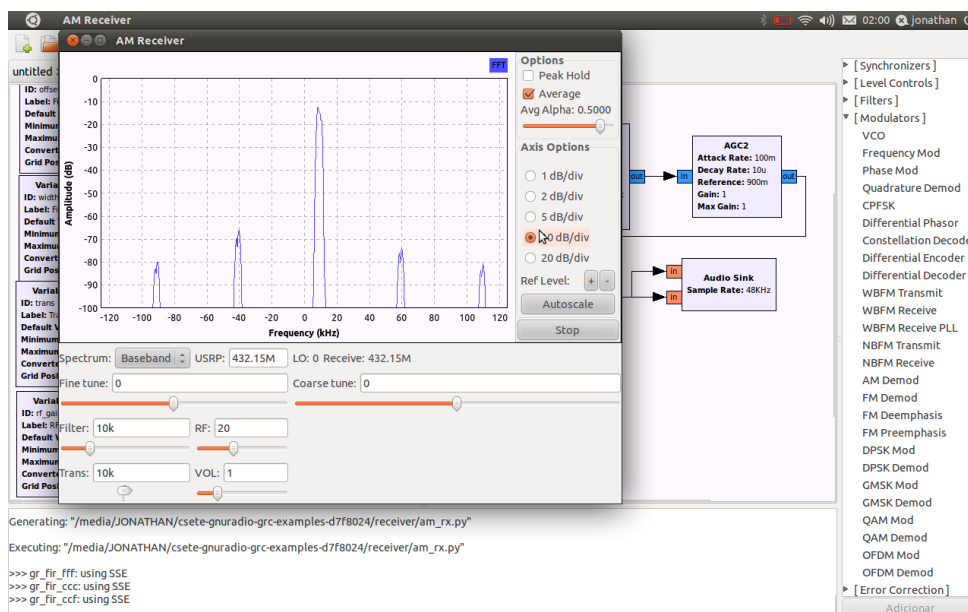


Figura 4.11: Espectro do sinal recebido sem sinal de voz.

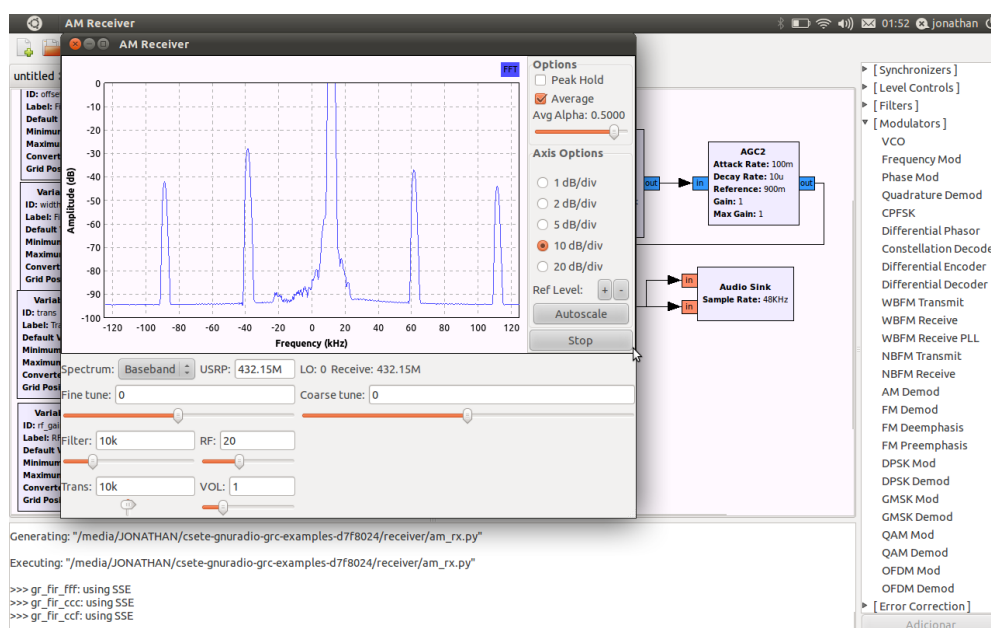


Figura 4.12: Espectro do sinal recebido com sinal de voz.

Na figura 4.11, observa-se o sinal AM captado sem o sinal de voz, onde estão presentes somente a portadora e suas sub-portadoras. Na figura 4.12, pode-se notar que o espectro recebido muda, onde aparece o sinal de voz modulado na faixa inferior do sinal AM.

5 CONCLUSÃO

A tecnologia de Rádios Definidos por *Software* é uma área nova e ampla de pesquisas. Neste trabalho foram abordados os principais conceitos sobre RDS, muitos dos quais estudados no Curso Superior em Sistemas de Telecomunicações oferecido pelo Instituto Federal de Santa Catarina (IFSC) campus São José, o que facilita e ajuda no entendimento da tecnologia.

Analisando o RDS, conclui-se que, apesar de algumas limitações, seus benefícios são explícitos para os desenvolvedores de sistemas, pois pode-se modificar toda estrutura de um equipamento de rádio frequência sem a necessidade de modificar o *hardware*. Ou seja, fazer essa modificação apenas trocando algumas linhas de programa ou até mesmo modificando algum parâmetro em uma interface gráfica. As preocupações e cuidados ao projetar um RDS são recompensados por se estar criando um receptor ou transmissor de rádio-frequência altamente flexível.

Dentre suas vantagens estão a flexibilidade do sistema, o baixo custo de implementação, o tamanho do *hardware* (que é altamente reduzido), a facilidade e velocidade de implementação de um projeto. As principais limitações de um RDS estão em dois componentes, os conversores A/D/A que ainda não tem sua tecnologia muito avançada, e os processadores que precisam ter alta performance de acordo com a frequência dos dados a serem processados. Por estes motivos, um RDS ideal não pode ser implementado nos dias de hoje.

Este trabalho também apresentou uma plataforma *open source* para implementação de um RDS, o GNU *Radio* que juntamente com um *hardware* periférico, utilizado para ligar o computador ao mundo RF, mostram uma maneira muito didática de estudar e projetar arquiteturas de equipamentos de rádio.

Além dos estudos teóricos e conceitos sobre RDS, este trabalho demonstrou um exemplo de uso de uma plataforma RDS. A plataforma testada é uma excelente opção para os laboratórios de IFSC, permitindo que novos trabalhos e pesquisas possam ser realizados.

Para realização de trabalhos futuros pode-se citar algumas sugestões, como por exemplo:

- O estudo de técnicas de rádio localização e implementação através de RDS;
- O estudo de rádios cognitivos, que utilizam RDS como base;
- A implementação de DDC's e DUC's em um FPGA.

REFERÊNCIAS

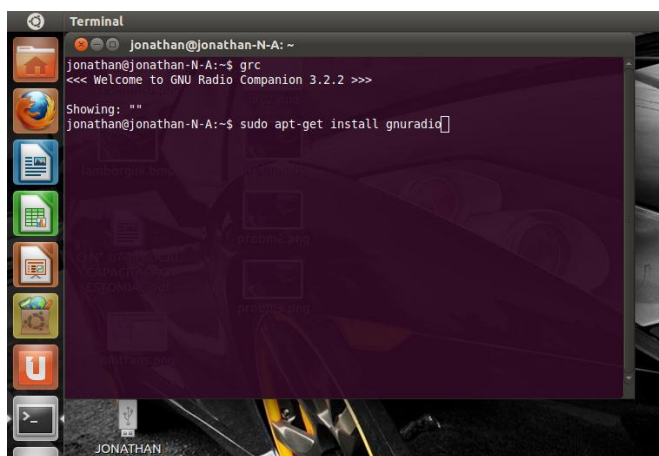
- [1] ALMEIDA, Erika P. L. de. Desenvolvimento de técnica de sensoriamento do espectro embasada em detecção de energia para aplicações em sistemas rádio-cognitivo. Brasília: ENE/UnB, 2010.
- [2] BARROS, Letícia Garcia de. O Rádio Definido por Software. Brasília: ENE/UnB, 2007.
- [3] BLOSSOM, E. (2009). Gnu radio. <http://www.gnu.org/software/gnuradio>.
- [4] CAMPOS, G. O outro inventor do rádio. *Revista Superinteressante*, Julho 2007.
- [5] DUTRA, Rafael Barros. Avaliação do desempenho da modulação OFDM em canais HF para rádios cognitivos. Rio de Janeiro: COPPE/UFRJ, 2010.
- [6] ETTUS, M. (2009). Universal software radio peripheral. <http://www.ettus.com/>.
- [7] FARIAS, Aldo; MELO, Carlos E. N.; DIAS, Ricardo V.. Uma abordagem a métodos de sensoriamento do espectro aplicados a rádio cognitivo. Brasília: ENE/UnB, 2010.
- [8] FÁVERAS, Daniel. Conversor analógico digital sigma-delta para um sistema em chip. Brasília. UnB, 2003.
- [9] GARRATT, G. R. M. *The Early History of Radio: from Faraday to Marconi*. London, UK: Institution of Electrical Engineers, in association with the Science Museum, 1994.
- [10] GNU RADIO. <http://www.gnuradio.org>. Acessado em 14/12/2011.
- [11] <http://www.microfone.jor.br/historia.htm>. Acessado em: 31/01/2012.
- [12] <http://www.andraka.com/cordic.htm>. Acessado em: 10/02/2012.
- [13] http://www.radioclubecampobelo.com.br/historia_do_radio.htm. Acessado em: 31/01/2012.

- [14] <http://www.oz9aec.net/index.php/gnu-radio/grc-examples>. Acessado em: 06/02/2012
- [15] <http://www.teleco.com.br/DVD/PDF/tutorialciclos.pdf>. Acessado em: 31/01/2012.
- [16] <http://www.historiadomundo.com.br/idade-contemporanea/landell-de-moura-o-inventor-do-radio.htm>. Acessado em: 31/01/2012.
- [17] ISOMÄKI, P.; AVESSATH, N. *na overview of software defined radio technologies*. [S.l.], 2004. Disponível online em www.tucs.fi/publications/attachment.php?fname=TR625.pdf.
- [18] KENINGTON, W. *RF and baseband techniques for software defined radio*. [S.l.]: Artech House, 2005.
- [19] KUHNEN, C. A. *Tópicos de Eletromagnetismo - Aspectos Históricos*. Florianópolis: UFSC, 2001.
- [20] LATHI, B. P. *Modern Digital and Analog Communication Systems*. Oxford University Press, 1998.
- [21] LIMA, André G. M. *Reconfigurabilidade em um Sistema de Comunicação Móvel Baseado na Arquitetura de um Rádio Definido por Software*. Brasília: UnB, 2006.
- [22] LUO, Fa-long. *Digital Frnt-End in Wireless Communications and Broadcasting*. [S.l.]: Cambridge, 2011.
- [23] MARTINS, Everson. *Projeto de Misturador com Topologia Célula de Gilbert Utilizando PHEMT*. São Paulo: UNICAMP, 2002.
- [24] MATOS, Roberto de; FRÖHLICH, Antônio Augusto; BECKER, Leandro Buss. *Otimizando o Desempenho de Rádios Definidos por Software Através do Deslocamento de Canais*. Florianópolis: UFSC.
- [25] MITOLA, J. *software radio architecture*. [S.l.]: Wiley, 2000.
- [26] MITOLA, J. *Cognitive Radio – An integrated agent architecture for software defined radio*. Suécia: Wiley, 2000.

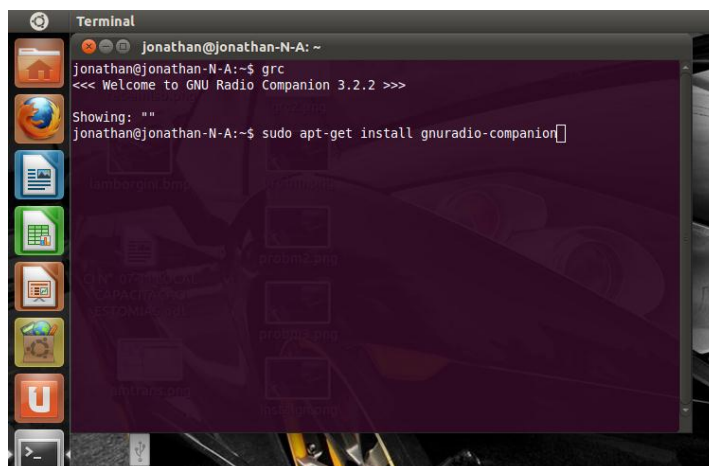
- [27] REINALDO, Pierry Moreno. Rádio localização utilizando Rádio Definido por Software. Florianópolis: UFSC, 2011.
- [28] SCHENA, Rafeal. Desenvolvimento de um Digital Down Converter (DDC) para Protótipo Embarcado de Rádio Definido por Software. Brasília: ENE/UnB, 2007.
- [29] SDR FORUM. <http://www.sdrforum.org>. Acessado em 31/11/2011.
- [30] SHEN, Dawei. Tutorial 4: The USRP Board. [S.l.: S.n.], 2005.
- [31] SJÖSTRÖM, U; CARLSSON, M; HÖRLIN, M. Design and implementation of a digital down converter chip. IN: The European Association for Signal and Image Processing. [S.l.: s.n.], 1996.
- [32] TUTTLEBEE, Walter. *Software Defined Radio*. [S.l.]: Wiley, 2002.
- [33] VASCONCELLOS, Rodrigo Durães de. Projeto de um conversor analógico/digital por aproximações sucessivas de 12 bits. Belo Horizonte: UFMG, 2011.

APÊNDICE A – Instalação GNU Radio e GRC

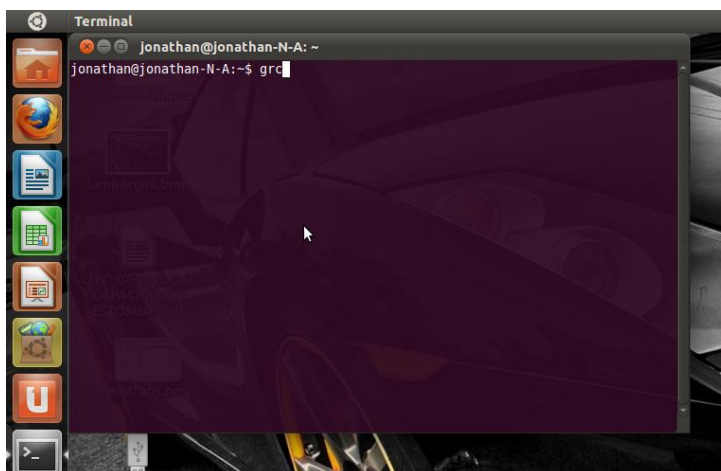
Para instalação do GNU Radio no Ubuntu 11.04 abra o um terminal e digite o comando “apt-get install gnuradio” se estiver como administrador, caso contrário digite o comando “sudo apt-get install gnuradio” (conforme figura abaixo) e digite a senha quando for pedido.



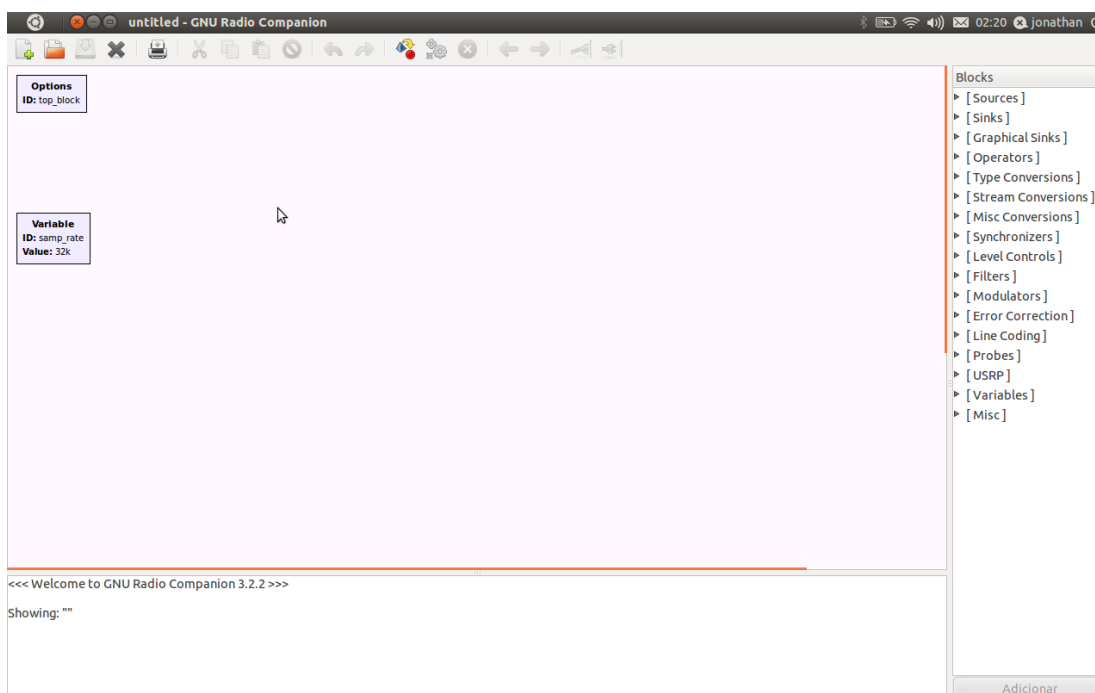
Para instalação do GNU Radio Companion no Ubuntu 11.04 abra o um terminal e digite o comando “apt-get install gnuradio-companion” se estiver como administrador, caso contrário digite o comando “sudo apt-get install gnuradio-companion” (conforme figura abaixo) e digite a senha quando for pedido. Não é preciso instalar o GNU Radio e depois instalar o GRC, pois a instalação do GRC já instala automaticamente o GNU Radio.



Para execução do GRC, abra o terminal e digite o comando “grc” (conforme figura abaixo).



O programa será executado através da janela abaixo.



APÊNCIDE B – Aquisição da USRP

Tanto a USRP quanto as placas filhas são adquiridas diretamente no site www.ettus.com.



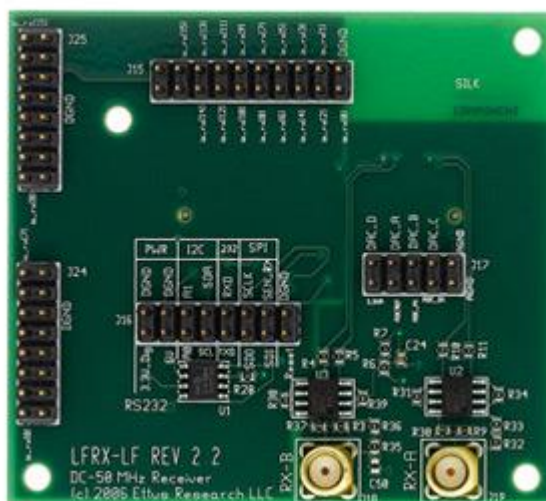
USRP

O *datasheet* da USRP está disponível no link: https://www.ettus.com/content/files/Ettus_USRP1_DS_FINAL_1.27.12.pdf.

Dentre as opções de placas filhas estão:

- LFRX Daughterboard 0-30 MHz Rx

Link: <https://www.ettus.com/product/details/LFRX>



- LFRX Daughterboard 0-30 MHz Tx

Link: <https://www.ettus.com/product/details/LFTX>



- WBX 50-2200 MHz Rx/Tx

Link: <https://www.ettus.com/product/details/WBX>



- RFX1800 1.5-2.1 GHz Rx/Tx

Link: <https://www.ettus.com/product/details/RFX1800>

