



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica

FELIPE BIZERRA FIDÉLES

Sistema de Identificação por Rádiofrequência sem Chip
em Rádio Definido por Software

Campina Grande, Paraíba

Outubro de 2013

FELIPE BIZERRA FIDÉLES

Sistema de Identificação por Rádiofrequência sem Chip em Rádio
Definido por Software

*Trabalho de Conclusão de Curso submetido ao
Departamento de Engenharia Elétrica da
Universidade Federal de Campina Grande para
obtenção do título do grau de Bacharel em
Ciências no domínio da Engenharia Elétrica*

Área de concentração: Sistemas de Comunicações

Orientador: Prof. Dr. Edmar Candeia Gurjão

Campina Grande, Paraíba

Outubro de 2013

FELIPE BIZERRA FIDÉLES

Sistema de Identificação por Rádiofrequência sem Chip em Rádio Definido por Software

*Trabalho de Conclusão de Curso submetido ao
Departamento de Engenharia Elétrica da
Universidade Federal de Campina Grande para
obtenção do título do grau de Bacharel em
Ciências no domínio da Engenharia Elétrica*

Aprovado em ___/___/___

Prof. Dr. Edmar Candeia Gurjão

Universidade Federal de Campina Grande
Professor Orientador

Professor Convidado

Universidade Federal de Campina Grande

Campina Grande, Paraíba

Outubro de 2013

AGRADECIMENTOS

Em primeiro lugar, agradeço à minha família por todo apoio e compreensão ao longo de todos esse anos.

Deixo aqui meu respeito e admiração aos bons professores que me acompanharam nesse processo de aprendizado. Destaco o Professor Edmar Gurjão pela orientação, apoio e amizade na graduação.

Aos meus amigos, sem eles nada disso teria dado certo. Muito obrigado pelas amizades e bons momentos compartilhados. Vocês estarão sempre comigo.

Ao pessoal da San Diego State University, pela boa recepção e companheirismo.

Aos funcionários do Departamento de Engenharia Elétrica, em especial a Adail, pela ajuda em momentos decisivos.

Tive contato com ótimas pessoas durante toda a graduação, pessoas essas que quero ver e rever em muitas ocasiões que estão por vir.

*“Devemos aprender durante toda a vida, sem
imaginar que a sabedoria vem com a velhice.”*

Platão

RESUMO

Este relatório é o resultado de um Trabalho de Conclusão de Curso realizado no Departamento de Engenharia Elétrica da Universidade Federal de Campina Grande. Teve como objetivo a implementação de um leitor para um sistema de Identificação por Rádiofrequência (RFID) que realiza a detecção de etiquetas passivas por meio de uma técnica de anticolisão, a ser desenvolvido em conjunto com outros dois trabalhos. Foi realizada geração de sinais do tipo *chirp* utilizando uma fonte de sinais e um VCO. Transmissões e recepções foram realizadas na frequência de 1 GHz através de um Rádio Definido por Software (RDS) implementado via uma plataforma de software GNU Radio e do hardware USRP (*Universal Software Radio Peripheral*). Verificou-se a eficácia das transmissões e limitações por parte do software utilizado.

Palavras-chave: RFID, *chirp*, sistema anticolisão, etiquetas passivas, Rádio Definido por Software.

ABSTRACT

This report is the result of a Senior Design Project developed at Electrical Engineering Department of Federal University of Campina Grande (UFCG). It was developed in order to implement a reader to a Radio Frequency Identification System that detects passive tags through an anti-collision technique, implemented together with other two works. Generation of chirp signals was performed using a signal generator and a Voltage Controller Oscillator (VCO). Transmission and reception of these signals were performed at 1 GHz through a Software Defined Radio implemented via open source software GNU Radio and USRP (Universal Software Radio Peripheral). Although the communication between receptor and transmission was done successfully, limitations in the software platform used were detected.

Keywords: RFID, chirp signals, anti-collision system, passive tags, software defined radio.

LISTA DE FIGURAS

Figura 1 – Interação entre componentes de um sistema RFID.....	4
Figura 2 – Vista superior de uma etiqueta ativa. É notável a presença de uma bateria em sua constituição.....	4
Figura 3 – Vista superior de uma etiqueta Passiva, que não possui nenhuma fonte de energia em seu projeto.....	5
Figura 4 – Faixa de frequência utilizada em sistemas RFID (100 kHz – 10 GHz) e frequências usuais de operação (125-134 kHz, 13.56 MHz, 915 MHz, 2.45 GHz).	6
Figura 5 - Antena <i>patch</i> que utiliza linhas de transmissão para atrasar o sinal recebido do leitor.	8
Figura 6 – Etiqueta (parte superior) e sua assinatura spectral (parte inferior).	8
Figura 7 - Uso da modulação da fase por retroespalhamento.	9
Figura 8 - (a) Janelamento do sinal (b) Variação da frequência instantânea em relação ao tempo (c) Sinal Chirp.....	11
Figura 9 - Sistema RFID com múltiplas etiquetas passivas.	12
Figura 10 - Rádio Definido por Software.	14
Figura 11 - Linhas de código em Python.....	15
Figura 12 – Diagrama de blocos do GNU Radio Companion.....	16
Figura 13 – Vista superior da USRP N200.....	18
Figura 14 - Sistema montado: USRPs, antenas e computadores.	20
Figura 15 - Implementação de um sinal chirp utilizando o GRC por meio de um VCO, tendo como entrada um sinal dente-de-serra.....	21
Figura 16 - Sinal <i>chirp</i> resultante da implementação no GRC.	22
Figure 17 - Sinal chirp no domínio da frequência para sinal de entrada dente-de-serra com frequência de 1 Hz em um VCO cuja sensibilidade é 6.28 M rad/seg/volt.	23
Figura 18 - Sinal chirp no domínio da frequência para sinal de entrada dente-de-serra com frequência de 10 kHz em um VCO cuja sensibilidade é 6.28 M rad/seg/volt.	24
Figura 19 - Sinal chirp no domínio da frequência para sinal de entrada dente-de-serra com frequência de 100 kHz em um VCO cuja sensibilidade é 6.28 M rad/seg/volt.	24
Figura 20 - Programa no GRC que realiza a transmissão de um sinal <i>chirp</i>	25
Figura 21 - Visualização em um analisador de espectro: sinal com 10 MHz de banda. 25	
Figura 22 – Diagrama de blocos para recepção do sinal <i>chirp</i> – GRC.....	26
Figure 23 - Recepção do sinal <i>chirp</i>	26
Figura 24 – Diagrama de blocos que mostra a adição de um filtro rejeita-faixas.	27
Figura 25 - Resposta em frequência após adição do filtro rejeita-faixas.....	27

Figura 26 - Armazenamento em arquivo	28
Figura 27 - Adição do bloco <i>Pulse Wave</i>	29
Figura 28 - FFT da multiplicação do sinal gerado pelo <i>Pulse Wave</i> e o sinal dente-de-serra.	29
Figura 29 - Falta de sincronização entre as fontes.	30
Figura 30 - Resultado da multiplicação de um sinal dente-de-serra e um pulso retangular dessincronizados.	30
Figura 31 - Erro exibido na saída do programa devido à alta taxa de amostragem.	31

SUMÁRIO

1. INTRODUÇÃO.....	1
2. REVISÃO BIBLIOGRÁFICA	3
2.1. Identificação por Rádiofrequência (RFID)	3
2.2. Etiquetas Passivas.....	7
2.3. Sinais <i>Chirp</i>	10
2.4. Sistema anti-colisão RFID.....	12
2.5. Rádio Definido por Software (RDS)	13
2.6. GNU Radio.....	14
2.7. USRP	17
3. IMPLEMENTAÇÃO.....	19
3.1. Sistema	19
3.2. Resultados	20
3.3. Limitações.....	30
4. CONSIDERAÇÕES FINAIS.....	32
5. REFERÊNCIAS	33
6. ANEXO – Instalação da USRP N200	36

1. INTRODUÇÃO

O sistema de Identificação por Rádiofrequência (RFID, do inglês, *Radio Frequency Identification*, RFID) é uma tecnologia que utiliza um leitor e etiquetas para identificação de itens e rastreamento [1]. Pesquisas estão sendo realizadas na linha de reconhecimento de diversas etiquetas passivas e redução de custos de tal forma que sejam utilizadas em sistemas complexos de alta escala [2].

O desafio de um sistema com múltiplas etiquetas passivas é realizar o reconhecimento de cada uma, pois o sinal recebido de pelo leitor será o resultado da colisão de vários sinais refletidos das etiquetas, sendo necessário um sistema anti-colisão, que separe a informação proveniente dessas etiquetas.

Para implementar esse sistema anti-colisão, é sugerida em [3] a transmissão de pulsos *chirp*, um tipo de sinal modulado linearmente em frequência que é utilizada em diversas aplicações de sistemas de radares e sonares.

Uma nova tendência de implementação de sistemas RFID é através do Rádio Definido por Software (RDS) [4], uma tecnologia que dispensa o aparato analógico necessário em rádios tradicionais e trata os sinais recebidos e transmitidos digitalmente. É utilizado devido à sua vasta linha de aplicações que envolvem a transmissão e recepção de sinais digitais, bem como o processamento dos mesmos, sem a necessidade de um grande aparato analógico.

Uma plataforma de software para Rádios por Software chamada GNU Radio foi utilizada, bem como o hardware específico para esse tipo de aplicação: USRP (*Universal Software Radio Peripheral*). Da escrita deste relatório, são apresentados os resultados, bem como as limitações encontradas durante a execução do projeto.

Na seção 2, será dada uma abordagem teórica aos tópicos relevantes do trabalho, para uma melhor compreensão das atividades desenvolvidas. Na seção

3, a implementação e os resultados serão detalhados. Por fim, as considerações finais apresentadas na seção 4 apresentam uma visão geral do trabalho, abordando propostas para estudos futuros.

2. REVISÃO BIBLIOGRÁFICA

2.1. Identificação por Rádiofrequência (RFID)

A Identificação por Rádiofrequência (RFID) é uma tecnologia que foi mencionada pela primeira vez em 1948, por Harry Stockman [5]. Entretanto, o seu sucesso teve que esperar o desenvolvimento de transistores, circuitos integrados, microprocessadores, comunicação por redes e as mudanças no mundo dos negócios [6], tão importantes para a sua popularização, que ocorreu a partir dos anos 1990.

A RFID é um sistema de comunicação sem fio que identifica objetos ou pessoas por meio de ondas eletromagnéticas. É constituído basicamente de três componentes [2]:

- Leitor (ou interrogador): composto por uma antena, módulos de rádiofrequência (RF) e de controle eletrônico. Tem a função de excitar a etiqueta, ler conteúdos, escrever dados e retransmití-los para a base de controle;

- Etiqueta: composta por uma antena e um chip, podendo ter uma bateria ou não. A sua função é retransmitir uma informação ao leitor;

- Central de controle: consiste em um computador que contém um software de controle e bancos de dados. É a central de processamento de dados do sistema.

O sistema RFID funciona da seguinte forma: quando um objeto que contém uma etiqueta entra na zona de operação do leitor, este a excita através da transmissão de um sinal RF, que é retransmitido contendo uma informação, seja ela um número serial ou uma instrução [7]. Quando o leitor recebe esses dados, eles são repassados à central de controle para serem inseridos em um banco de dados, por exemplo [7]. A Figura 1 ilustra esse processo.

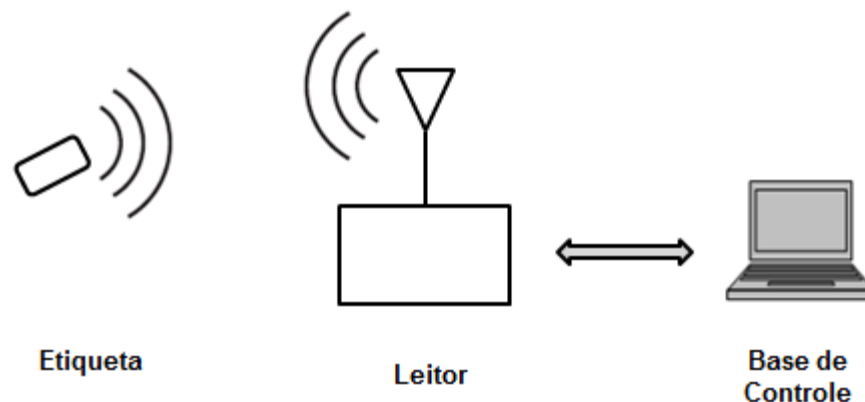


Figura 1 – Interação entre componentes de um sistema RFID.

Uma base de controle pode estar ligada a vários leitores, bem como um leitor pode excitar várias etiquetas.

As etiquetas podem ser ativas ou passivas [2]. As ativas possuem uma fonte de energia, transmitindo assim um sinal com maior potência e, conseqüentemente, aumentando o raio de abrangência para a leitura de dados. Possuem dimensões maiores e são mais caras, sendo melhor aplicadas em objetos grandes para serem rastreados a longa distância [8]. Têm alta capacidade de memória, em torno de quilobytes e suas baterias podem durar de dois a sete anos [7]. A Figura 2 ilustra uma etiqueta ativa.



Figura 2 – Vista superior de uma etiqueta ativa. É notável a presença de uma bateria em sua constituição.

Fonte: <http://newtech010.blogspot.com.br/2009/06/rfid-technology-new-revolution.html>

No caso das etiquetas passivas, que servem de base para este trabalho, não possuem uma fonte de energia e por isso se tornam mais baratas e menores, podendo ser integradas facilmente a diversos materiais e produtos. Entretanto, a

sua distância de rastreamento não supera 3 metros [2]. Atualmente, essas etiquetas podem conter em torno de 2 quilobits de memória [7] e o seu tamanho impossibilita um maior armazenamento de memória. A Figura 3 ilustra uma etiqueta passiva.

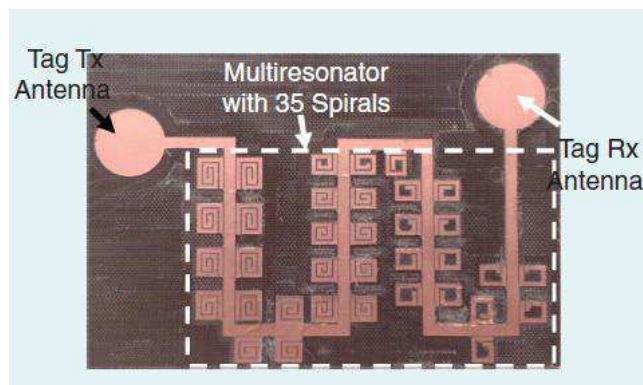


Figura 3 – Vista superior de uma etiqueta Passiva, que não possui nenhuma fonte de energia em seu projeto.

Fonte: [2]

Padrões do RFID regularizam, por exemplo, os protocolos de dados e sistema, as regras das operações RFID (segurança e teste) e identificação. O EPC (Código Eletrônico do Produto, do inglês, *Electronic Product Code*) é um deles e padroniza a comunicação entre o leitor e a etiqueta, bem como o formato dos dados de identificação e informações de armazenamento para banco de dados [7].

Vale observar que a frequência de operação pode variar de 100 kHz a 10 GHz, como mostrado na Figura 4. Este parâmetro afeta a taxa de transmissão de dados, o tipo e tamanho da antena e o nível de interferência resultante de outros sistemas de rádio [7].

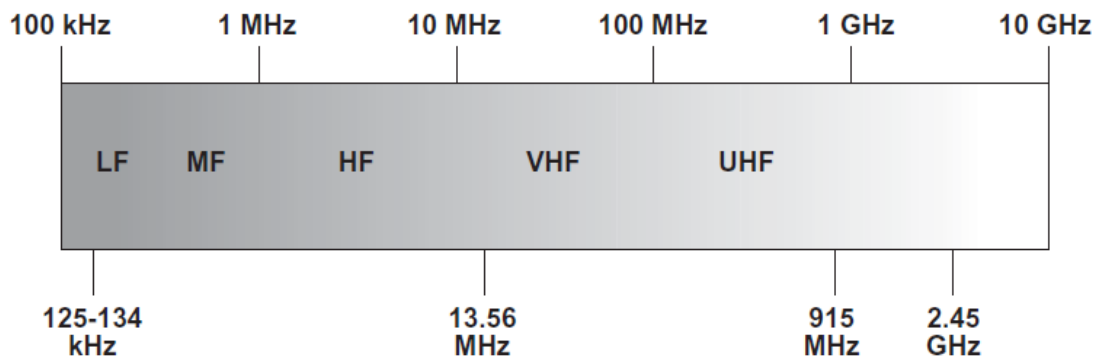


Figura 4 – Faixa de frequência utilizada em sistemas RFID (100 kHz – 10 GHz) e frequências usuais de operação (125-134 kHz, 13.56 MHz, 915 MHz, 2.45 GHz).

Fonte: Texas Instruments

Os sistemas RFID possuem inúmeras aplicações, dentre elas [9]:

- Logística: etiquetas são colocadas em itens que serão rastreados por leitores desde a fábrica até pontos de vendas. Empresas de correio como DHL e Fedex utilizam essa tecnologia nos seus serviços de entrega. Além disso, por meio do RFID é possível que uma loja tenha informações à respeito de quantos e quais produtos foram os mais vendidos;

- Sistemas de pedágio: permitem que carros passem por pontos de coleta de pedágio sem que seja necessário paradas;

- Ingressos: a exemplo da Olimpíada de Beijing, todos os ingressos continham etiquetas contendo informações pessoais de cada comprador, como fotografia, número de passaporte, endereço, e-mail e número de telefone;

- Saúde: uso de etiquetas para coleta de informações da saúde dos pacientes, bem como ajudar na distribuição e manuseio de medicamentos;

- Segurança: identificação pessoal. Documentos como passaporte e identidade podem conter elementos RFID para armazenamento de informações dos portadores.

2.2. Etiquetas Passivas

Como já mencionado anteriormente, as etiquetas passivas não apresentam uma fonte de alimentação no seu circuito, o que as torna mais baratas, sendo chamadas de “códigos de barras RF” [10].

É possível que o RFID substitua os códigos de barras ópticos. No caso das etiquetas ativas, isso ainda não ocorreu porque custam o dobro de preço dos códigos de barras [11] devido ao seu caro processo de fabricação, que inclui etapas de projeto de circuitos integrados específicos e testes.

No projeto de etiquetas passivas há o desafio de como codificar dados sem a presença de um chip. De acordo com [8], é possível dividi-las em três categorias: baseadas na reflectometria no domínio do tempo, na assinatura espectral e na modulação de retroespalhamento de amplitude e fase.

Para as etiquetas passivas baseadas na reflectometria no domínio do tempo, o leitor envia um sinal na forma de um pulso e recebe o sinal refletido pela etiqueta, que retorna na forma de um trem de pulsos, utilizado para codificar dados. Embora o custo seja reduzido e essa seja uma boa alternativa para melhorar o reconhecimento da posição por parte do leitor, o número de bits que podem ser codificados é limitado.

Um exemplo é mostrado na Figura 5, uma antena *patch* que possui linhas de transmissão que atrasam o pulso do sinal de entrada. De acordo com o tempo de atraso, pode ser realizada uma codificação para diferentes etiquetas.

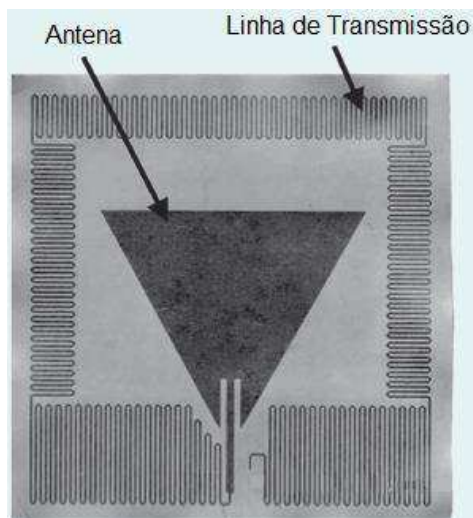


Figura 5 - Antena *patch* que utiliza linhas de transmissão para atrasar o sinal recebido do leitor.

Fonte: [12] - Editada

No caso da assinatura espectral, as etiquetas codificam dados utilizando estruturas ressonantes [8]. Cada bit é detectado de acordo com a presença ou ausência de um pico ressonante em uma certa faixa de frequência. Essa técnica garante uma maior capacidade de armazenamento de dados, mas requer uma faixa de espectro muito alta para isso. A Figura 6 ilustra um exemplo, que cria cinco picos na seção transversal de radar da antena.

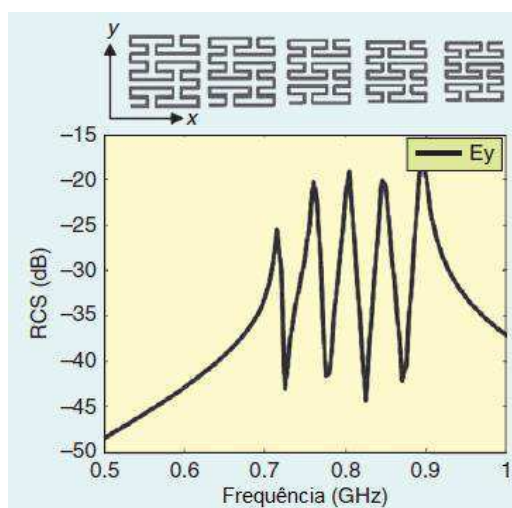


Figura 6 – Etiqueta (parte superior) e sua assinatura espectral (parte inferior).

Fonte: [13] - editada

Na técnica de modulação em amplitude e fase por retroespalhamento, a codificação dos dados se dá pela variação da amplitude ou fase do sinal refletido pela etiqueta por meio da carga reativa conectada a esta. No caso da Figura 7, o sinal recebido é refletido a cada discontinuidade, de tal forma que a informação é codificada de acordo com a fase do sinal refletido em relação a uma fase referência. Nesse caso, a variação de amplitude é minimizada.

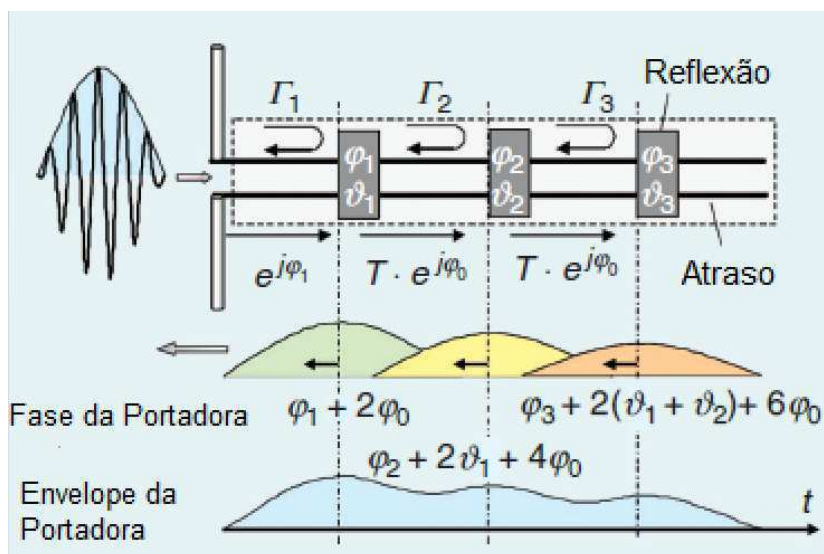


Figura 7 - Uso da modulação da fase por retroespalhamento.

Fonte: [14] – editada

Na literatura, é notável o uso de pulsos de curta duração como sinal enviado pelo leitor. Eles são usados pois seu largo espectro facilita a aplicação das técnicas descritas, além de possuir baixa magnitude, não interferindo em outros sistemas que estejam em operação. Pulsos linearmente modulados em frequência, chamados de sinais *chirp*, podem ser usados nesse tipo de transmissão [3].

2.3. Sinais *Chirp*

Radares e sonares necessitam detectar objetos muito pequenos a distâncias muito grandes com uma boa resolução angular [15]. Como o tipo de sinal utilizado nas transmissões é um pulso, há limitações nesse quesito com relação à potência transmitida, pois a duração do impulso é limitada. Além disso, a resolução do radar em relação ao número de alvos detectáveis a uma distância pequena entre si também é limitada.

É necessário um sinal que combine um alto valor de amplitude e que garanta uma boa resolução para o radar em pulsos de curta duração. Sinais *chirp* são ideais nesse caso.

O princípio básico para sinais desse tipo é varrer linearmente uma largura de banda B em um período de T segundos (Figura 8). Dessa forma, a frequência instantânea é modelada linearmente numa banda B , mas centrada na frequência f_0 .

Para demonstrar matematicamente esse sinal [16], considere uma senóide dada por:

$$x(t) = A \cos(\theta(t)) \quad (1)$$

De tal forma que a frequência instantânea seja dada por:

$$\omega_i(t) = \frac{d}{dt} \theta(t) \quad (2)$$

Seja a fase quadrática [16]:

$$\theta(t) = 2\pi\mu t^2 + 2\pi f_0 t + \varphi \quad (3)$$

Logo,

$$\omega_i(t) = \frac{d}{dt} \theta(t) = 4\pi\mu t + 2\pi f_0 \quad (4)$$

$$f_i = 2\mu t + f_0 \quad (5)$$

É possível notar que a frequência muda linearmente com o tempo. Analisando a Figura 8(b), a equação da reta relativa à frequência instantânea pode ser escrita como:

$$f_i = \frac{f_1 - f_2}{T} t + f_0 \quad (6)$$

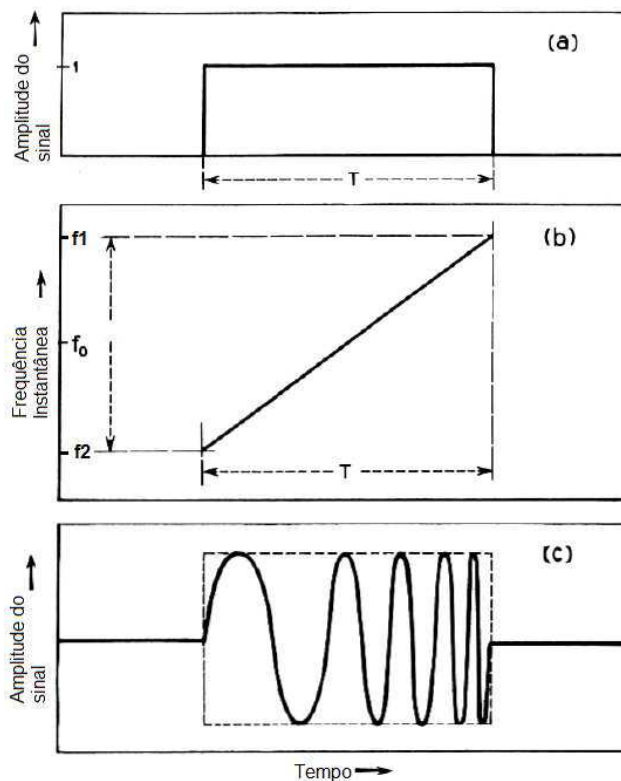


Figura 8 - (a) Janelamento do sinal (b) Variação da frequência instantânea em relação ao tempo (c) Sinal Chirp.

Fonte: [15] - editada

O sinal *chirp* pode ser gerado por meio de um VCO (Oscilador Controlador de Tensão, do inglês, *Voltage Controller Oscillator*), que será abordado nas seções seguintes.

2.4. Sistema anti-colisão RFID

Um sistema RFID que opere com diferentes etiquetas passivas (Figura 9) terá como resposta do sinal enviado pelo leitor a soma da dos sinais refletidos de várias etiquetas, sendo necessário um método que separe os sinais de identificação das mesmas, chamado de anti-colisão.

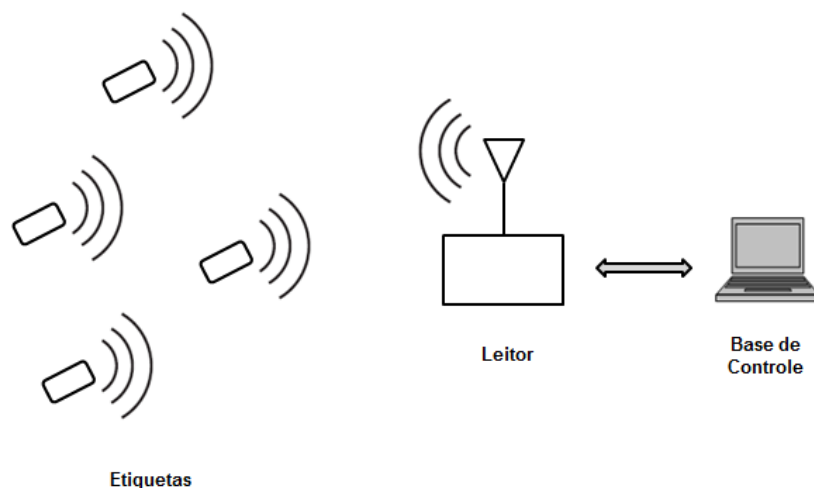


Figura 9 - Sistema RFID com múltiplas etiquetas passivas.

A ideia é que o leitor utilize sinais *chirp*, como descrito em [3] e a Transformada Fracional de Fourier [3] [17] para separação de sinais desse tipo. Quando o sinal modulado é refletido das etiquetas, este é transformado para um domínio de ordem fracional α , tal que

$$\alpha = \tan^{-1}\left(\frac{f_s/N}{2r}\right)$$

Em que f_s é a frequência de amostragem, N o número de pontos utilizados e r a taxa de mudança de frequência do sinal *chirp*.

A transformada comprime o sinal no domínio fracional citado e mostra picos em seu gráfico. O número de picos corresponde ao número de sinais *chirp* presentes, ou seja, ao número de etiquetas retransmitindo o sinal enviado pelo leitor [3].

Após detectar os picos nos gráficos, o próximo passo é estimar o número de etiquetas presentes por meio do janelamento e decodificá-las transformando os sinais resultante para o domínio do tempo e novamente para o domínio da frequência por meio da FFT [3]. Desse modo, é possível detectar a etiqueta específica de cada retransmissão.

Por fugir do escopo desse trabalho, sugere-se, caso necessário, que o leitor leia os trabalhos [3] e [17] para uma melhor abordagem.

2.5. Rádio Definido por Software (RDS)

O RDS consiste em uma tecnologia que tem recebido grande reconhecimento na indústria de Telecomunicações devido à sua flexibilidade. Ele facilita a implementação de módulos funcionais em sistemas de rádio como modulação e desmodulação, geração de sinais e codificação [18], desenvolvidos em software, dispensando todo um aparato analógico que realize essas funções. Além disso, por meio do processamento no domínio digital, esquia-se da variação de temperatura e envelhecimento dos componentes, presente em rádios tradicionais.

Proposto por Joseph Mitola III [19], o RDS consiste basicamente em receber um sinal, modificá-lo por meio de de ADCs (Conversores Digital-Analógico, do inglês, *Analog to Digital Converters*) e em seguida para a banda básica via DDCs (*Digital Down-Converters*) para o processamento via software. O processo inverso é realizado na transmissão de sinais via DUCs (*Digital Up-Converters*) e DACs (Conversores Analógico-Digital, do inglês, *Digital to Analog Converters*).

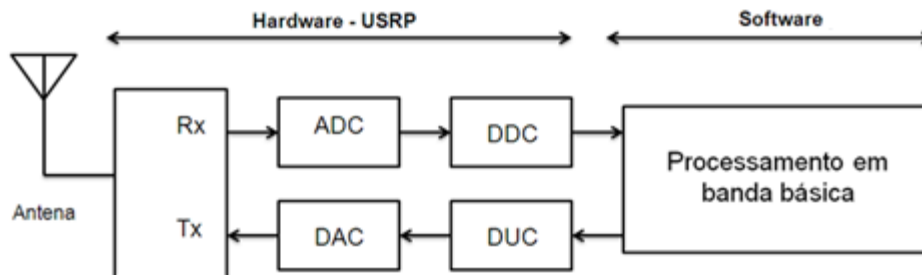


Figura 10 - Rádio Definido por Software.

As operações de um RDS podem ser modificadas em tempo real de execução, de acordo com as necessidades do sistema [20], além de que podem ser utilizados junto com diferentes sistemas de rádio-frequência como GPS, WiMAX, Bluetooth, entre outros. Para sua implementação, são necessários uma plataforma de software, um hardware e uma antena.

2.6. GNU Radio

O GNU Radio [21] consiste em uma plataforma de software livre para Rádios Definidos por Software que pode ser instalada em um computador de uso geral.

Junto com um hardware específico, ele pode ser utilizado largamente em ambientes acadêmicos e comerciais na implementação prática de rádios e em pesquisas de sistemas de comunicação sem fio.

Essa plataforma realiza todo o processamento de sinais por meio de blocos de processamento, que podem possuir entradas ou saídas de dados em tempo real, que são recebidos ou transmitidos por um hardware. Existem blocos prontos e qualquer desenvolvedor pode projetar filtros, moduladores, equalizadores, entre outras aplicações que estejam faltando em um projeto.

Como cada bloco realiza uma operação diferente, eles são conectados realizando um fluxo contínuo de dados, o que facilita na correção de erros e reutilização do código. Tais dados podem ser de diversos tipos, inclusive de bits e bytes e o desafio é entender como esses dados passam do bloco inicial, que pode

ser uma fonte de sinais ou um arquivo, até o ponto de transmissão, por exemplo, quando um sinal analógico é utilizado para transferir o sinal projetado para uma frequência específica.

A arquitetura do GNU Radio é dividida entre as linguagens C++ e Python. Os blocos de nível baixo são escritos em C++, pois essa linguagem garante maior velocidade de processamento. O Python é utilizado para instanciar e conectar os blocos, facilitando a implementação dos projetos. A Figura 11 ilustra um exemplo do GNU Radio.

```

Roteiro.txt ✕ scope_signal.py ✕
#!/usr/bin/env python

from gnuradio import gr
from gnuradio.wxgui import stdgui2,fftsink2,scopesink2
import wx

class gnuradioGUI(stdgui2.std_top_block):
    def __init__(self,frame,panel,vbox,argv):
        stdgui2.std_top_block.__init__(self,frame,panel,vbox,argv)

        #fft = fftsink2.fftsink_c(panel, title="FFT display", fft_size=512, sample_rate=100000)
        #vbox.Add(fft.win,4,wx.EXPAND)

        scope = scopesink2.scope_sink_c(panel, title="Oscilloscope", sample_rate=100000)
        vbox.Add(scope.win,4,wx.EXPAND)

        signal = gr.sig_source_f(100000,gr.GR_SAW_WAVE,500,10,0)
        freq_mod = gr.frequency_modulator_fc(0.35)
        #signal = gr.pwr_squelch_ff(0, 0.2, 1000, True)
        throttle = gr.throttle(gr.sizeof_float,100000)
        self.connect(signal,throttle)
        #self.connect(throttle,fft)
        self.connect(throttle, freq_mod)
        self.connect(freq_mod,scope)
        #self.connect(freq_mod,fft)

if __name__ == '__main__':
    app = stdgui2.stdapp(gnuradioGUI,"A simple GNU Radio GUI")
    app.MainLoop()

```

Figura 11 - Linhas de código em Python.

Vários projetos foram executados utilizando essa plataforma, como um completo sistema RFID [4] e um de detecção do espectro [22], ambos do Departamento de Engenharia Elétrica da Universidade Federal de Campina Grande.

Os blocos também podem ser conectados por meio de uma interface gráfica desenvolvida chamada de GRC (GNU Radio Companion), similar ao Simulink, do programa MATLAB®. Linhas de código em Python são geradas automaticamente, de forma implícita, sendo necessários apenas cliques do usuário.

A Figura 12 mostra um exemplo em que um gerador de sinais é somado ao ruído e visualizado num osciloscópio.

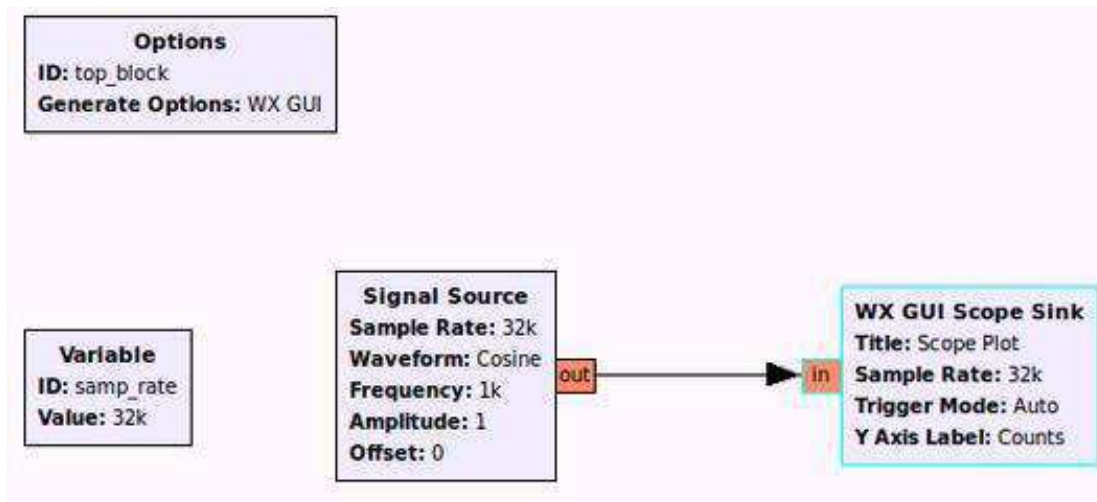


Figura 12 – Diagrama de blocos do GNU Radio Companion.

O GRC facilita a implementação para iniciantes, já que não exige o uso do Python de forma explícita, pois o fluxo de dados se torna mais claro de forma gráfica, além de economizar tempo. Os blocos desenvolvidos em C++ podem ser facilmente instalados e desinstalados nessa interface.

Um ponto fraco de sempre utilizá-lo é que há pouca versatilidade nas suas implementações, dificultando o desenvolvimento de sistemas mais complexos [23].

2.7. USRP

A USRP (*Universal Software Radio Peripheral*), desenvolvida pela Ettus Research [24], é um hardware de baixo custo utilizado em RDS por centros de pesquisas, empresas e universidades [21]. Esse dispositivo se conecta a um computador de uso geral por meio de um cabo USB 2.0 ou Gigabit Ethernet e é utilizado para transmitir e receber sinais. Para conectá-lo à plataforma GNU Radio, é necessária a instalação de um driver chamado UHD (*USRP Hardware Driver*).

Há diversos tipos de USRP, que possuem arquitetura parecida: FPGA, ADCs, DACs, geração de *clock* e sincronização e regulação de potência, que são componentes que realizam o processamento em banda básica. Há necessidade de *front-ends*, chamados de placas-filhas, utilizadas para operações analógicas como filtragem e conversão de sinais. Cada placa-filha opera em uma faixa específica de frequências, o que permite que a USRP opere com sinais de até 6 GHz.

Apesar da alta flexibilidade em relação às faixas de frequências de operação, possuem largura de banda de transmissão limitada até 100 MHz dependendo do tipo de dispositivo utilizado, que pode ser reduzida caso o computador utilizado não tenha um bom processador.

Além do GNU Radio, a USRP pode ser utilizada em conjunto com o Simulink e LabVIEW.

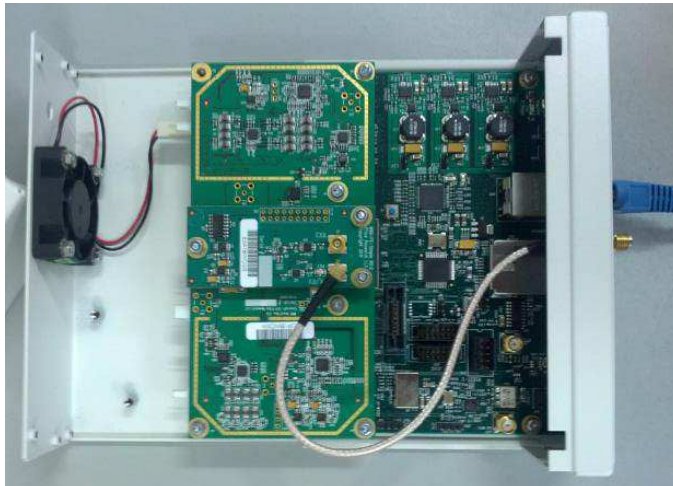


Figura 13 – Vista superior da USRP N200.

3. IMPLEMENTAÇÃO

3.1. Sistema

Este trabalho em conjunto com outros dois [25] [26] formam um sistema de identificação RFID com etiquetas passivas. Neste projeto foi desenvolvido o leitor para excitar a etiqueta por meio de pulsos *chirp*, o [25] fabricaria as etiquetas e o [26] ficaria responsável pelo sistema anti-colisão.

No sistema implementado, o GRC gera o sinal *chirp* em banda básica, uma USRP o transmite na frequência de 1 GHz e outra realiza a recepção, possibilitando a visualização do sinal no domínio da frequência.

Foram utilizadas duas USRPs N200, duas antenas log-periódicas que operam entre 800 MHz e 2,6 GHz e dois computadores Datasonic TAURUS I Q6.4 D5 com Processador Intel Core i5 3470 3.2 GHz, placa mãe ASUS P8H61-M-Lx e memória DDRIII 4 GB, além de possuir placa de rede Ethernet Gigabit, compatível com o hardware do RDS.

A placa-filha utilizada foi a WBX FE, que pode ser utilizada tanto para transmissões e recepções na faixa de 50 MHz a 2,2 GHz.

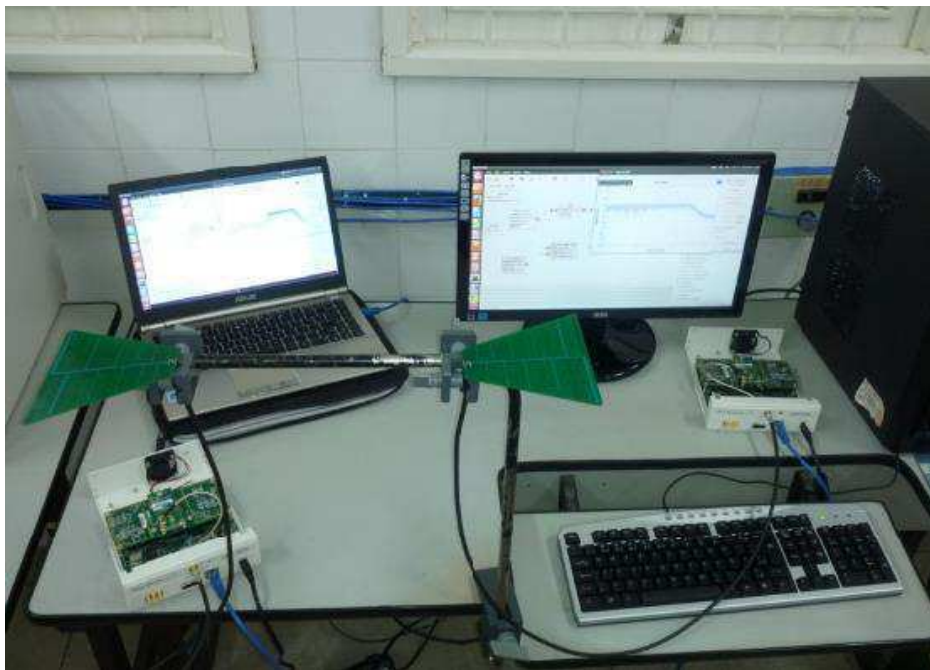


Figura 14 - Sistema montado: USRPs, antenas e computadores.

3.2. Resultados

O primeiro passo consiste na implementação de um sinal *chirp* por meio do GRC (Figura 15). É utilizada uma taxa de amostragem de 25 MS/s (executável pelo hardware). Como mencionado, um sinal *chirp* pode ser simulado por meio de um VCO, que está implementado no GNU Radio na forma de um bloco de processamento. Há dois outros blocos: o *Throttle*, que evita que a CPU do computador use sua velocidade máxima de operação; e o *WX GUI Scope Sink*, que consiste em um simples visualizador de sinais.

O VCO possui três parâmetros: taxa de amostragem, sensibilidade e amplitude. A sensibilidade refere-se à variação em frequência: como esse parâmetro é medido em rad/seg/volt, para cada volt de entrada, a frequência irá variar de 1 rad/seg. Como a sensibilidade do exemplo a seguir é de 6,28M rad/seg/volt e a forma de onda de entrada é uma dente-de-serra com frequência 1 Hz, para efeitos

de visualização. A frequência do sinal de saída irá variar de DC a 1 MHz em 1 segundo (período do sinal de entrada).

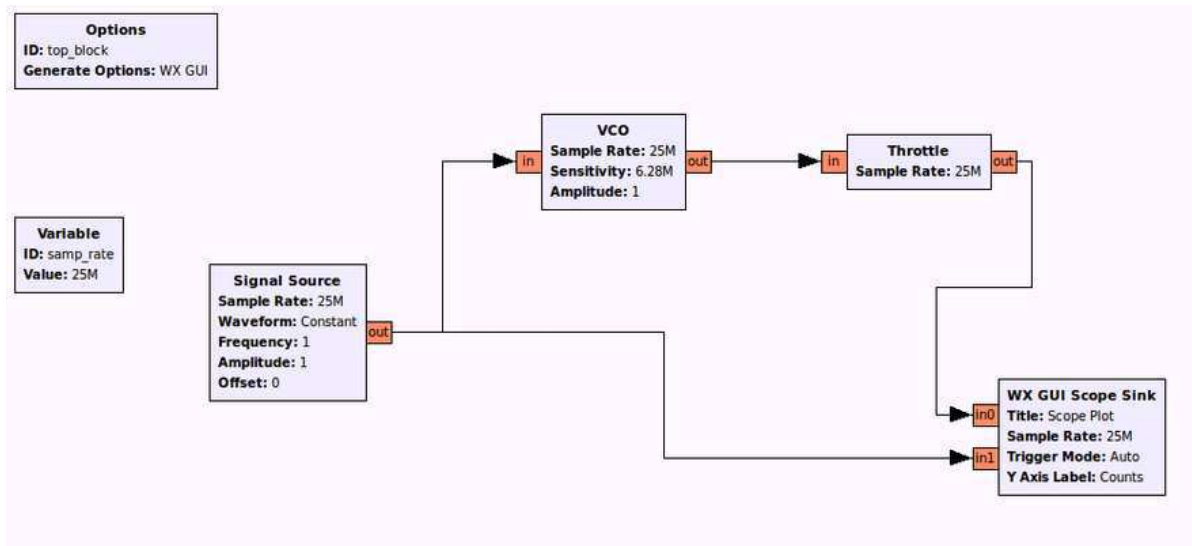


Figura 15 - Implementação de um sinal chirp utilizando o GRC por meio de um VCO, tendo como entrada um sinal dente-de-serra.

A saída no visualizador de sinais é mostrada na Figura 16. Vale observar o quanto a frequência cresce em relação à variação da onda dente-de-serra.

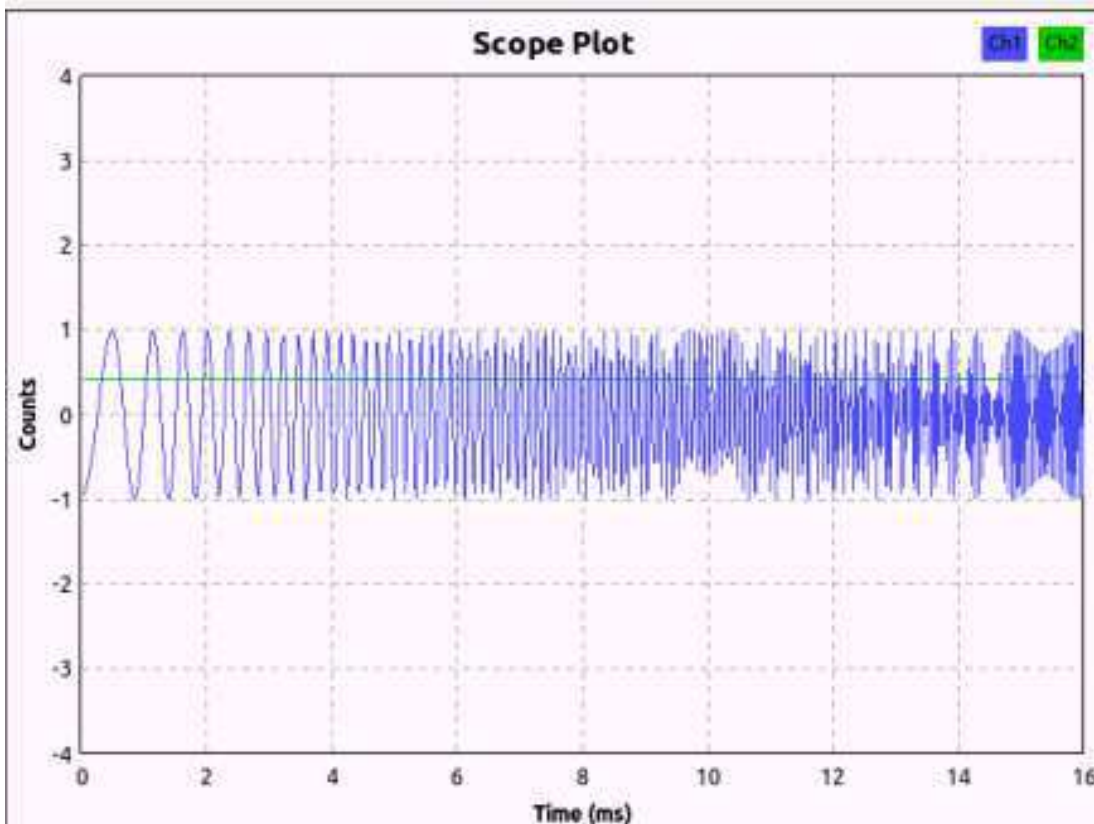
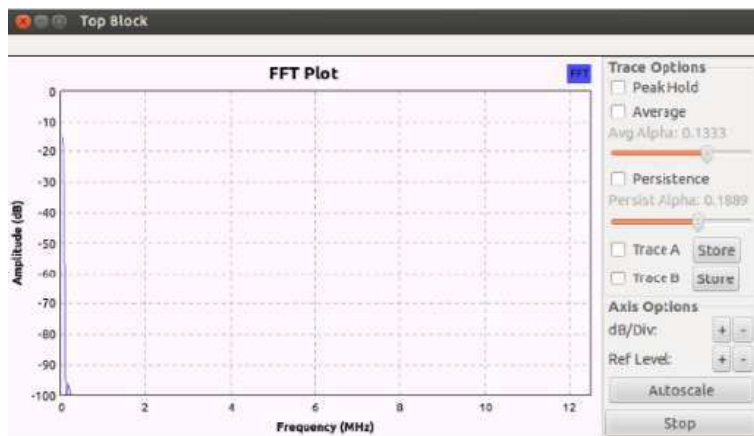


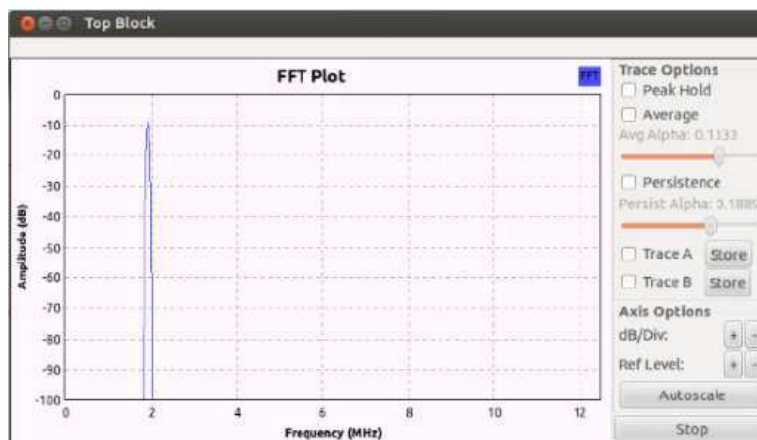
Figura 16 - Sinal *chirp* resultante da implementação no GRC.

Outra forma interessante de visualizar o sinal *chirp* gerado é no domínio da frequência (Figura 17). Como a frequência do sinal de entrada é 1 Hz, é possível ver a variação da frequência instantânea, em forma de um pulso, que varia do DC a 10 MHz, como já comentado.

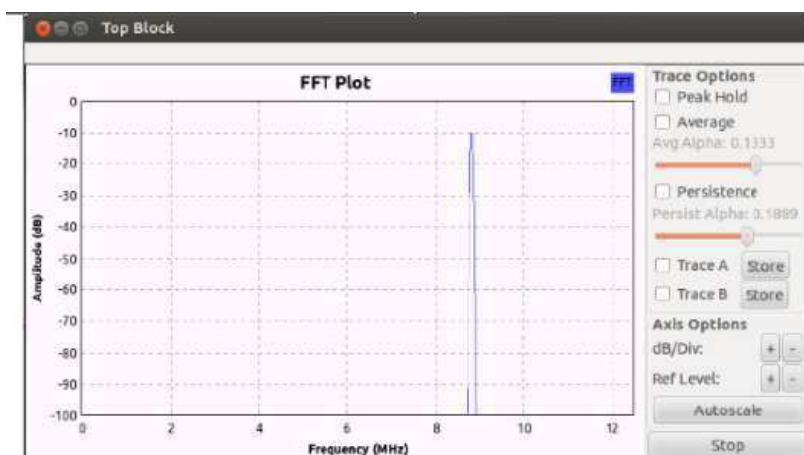
Se a frequência da dente-de-serra for aumentada para valores de 10 kHz (Figura 18) e 100 kHz (Figura 19) a variação da frequência instantânea será tão rápida que, no domínio da frequência, o pulso não estará mais visível, sendo possível visualizar um sinal que tem componentes até 10 MHz, correspondendo à resposta esperada de um sinal *chirp*



(a)



(b)



(c)

Figure 17 - Sinal chirp no domínio da frequência para sinal de entrada dente-de-serra com frequência de 1 Hz em um VCO cuja sensibilidade é 6.28 M rad/seg/volt.

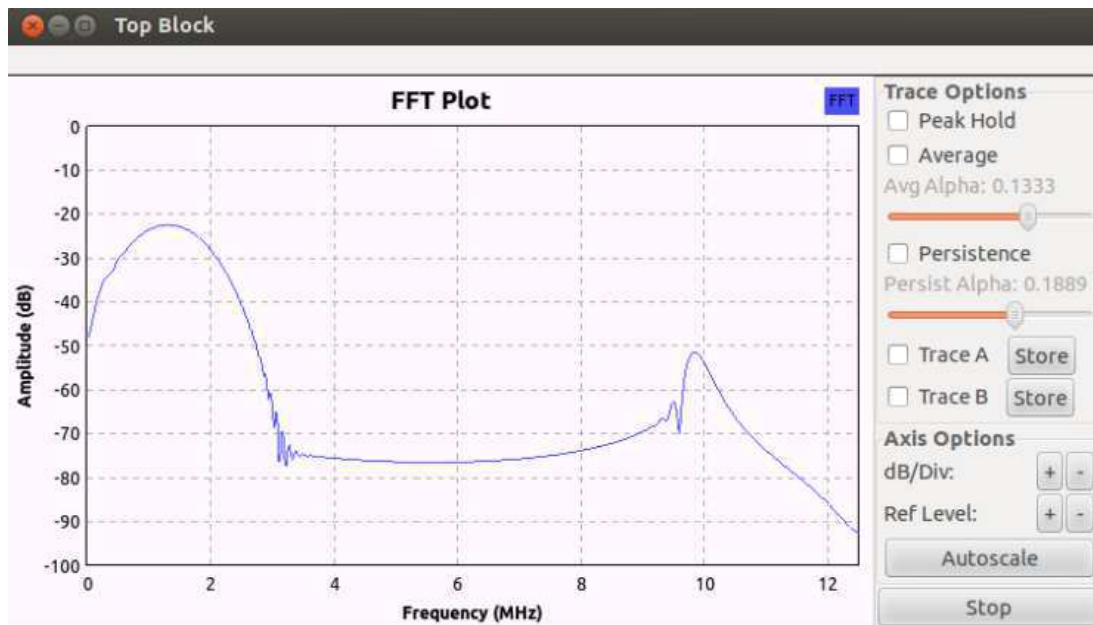


Figura 18 - Sinal chirp no domínio da frequência para sinal de entrada dente-de-serra com frequência de 10 kHz em um VCO cuja sensibilidade é 6.28 M rad/seg/volt.

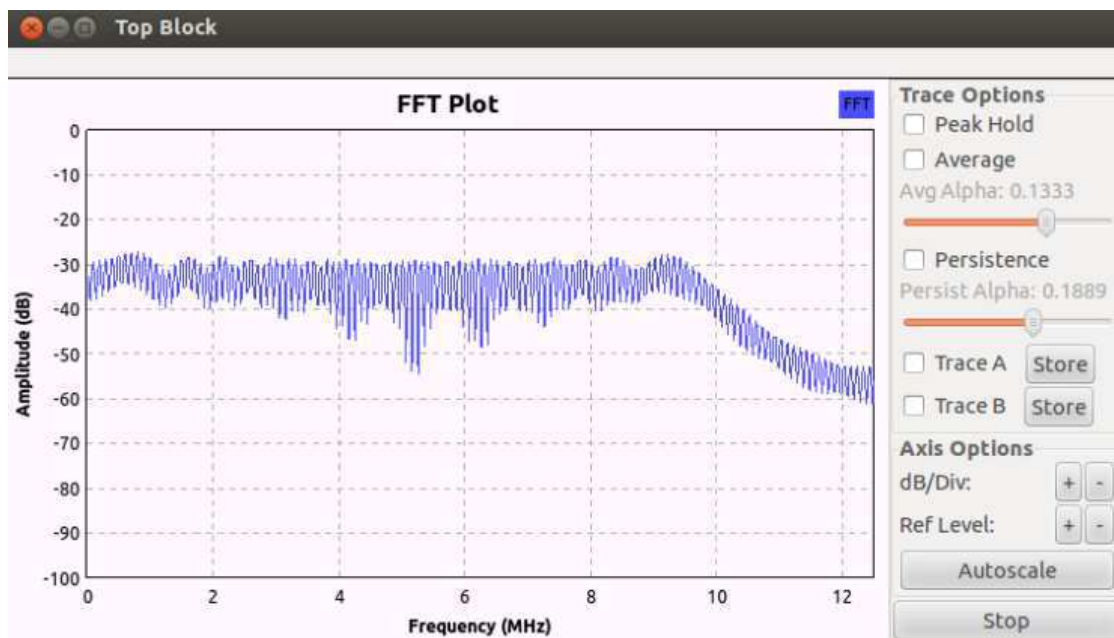


Figura 19 - Sinal chirp no domínio da frequência para sinal de entrada dente-de-serra com frequência de 100 kHz em um VCO cuja sensibilidade é 6.28 M rad/seg/volt.

Esse sinal foi transferido para uma frequência mais alta, no caso, 1 GHz (Figura 20). Isso se faz adicionando o bloco *UHD: USRP Sink*, em que podem ser ajustados parâmetros como ganho e o tipo da antena utilizada. O bloco *Hilbert* é

usualmente utilizado em aplicações do GNU Radio para transformar dados do tipo *float* para complexo.

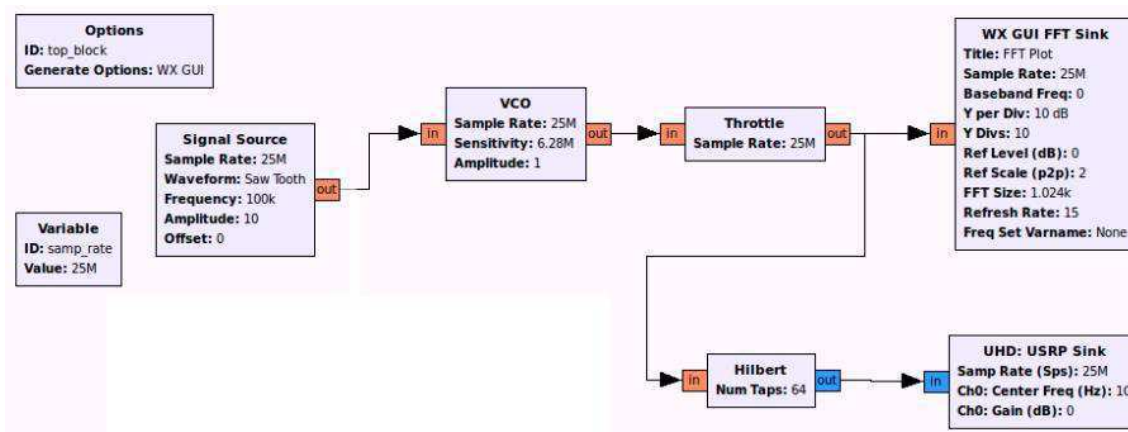


Figura 20 - Programa no GRC que realiza a transmissão de um sinal *chirp*

A Figura 21 mostra esse sinal sendo visualizado num analisador de espectro.

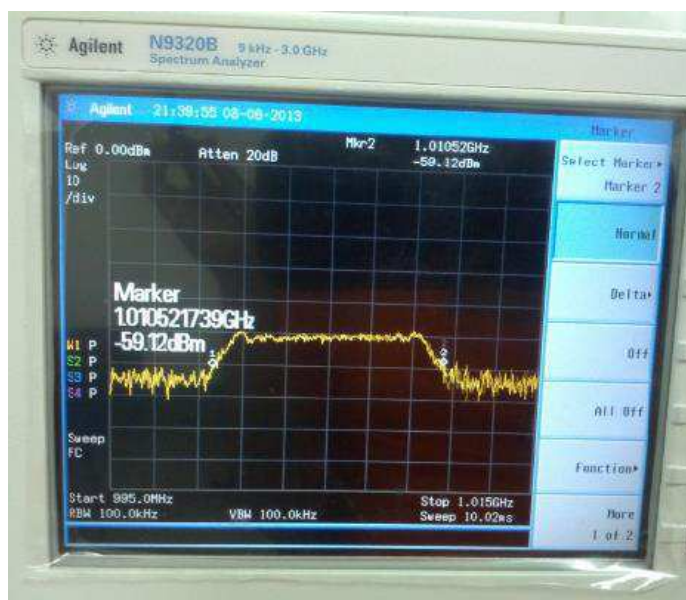


Figura 21 - Visualização em um analisador de espectro: sinal com 10 MHz de banda.

Utilizando outra USRP para a recepção, observa-se o programa na Figura 22 e a resposta em frequência do sinal recebido em banda básica na Figura 23.

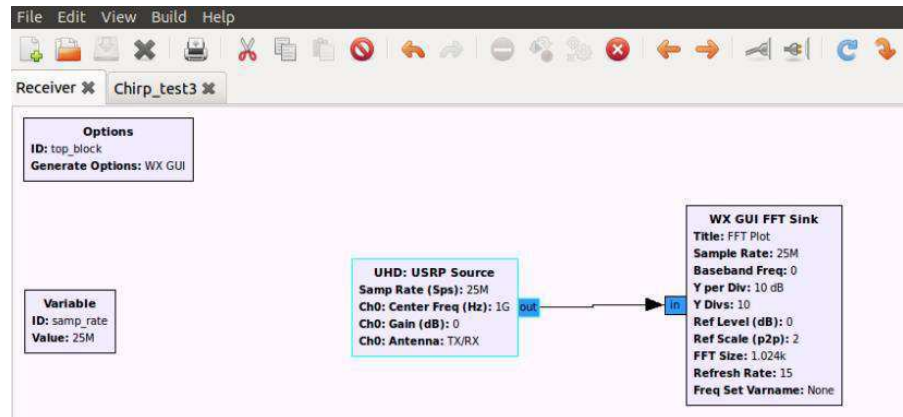


Figura 22 – Diagrama de blocos para recepção do sinal *chirp* – GRC.

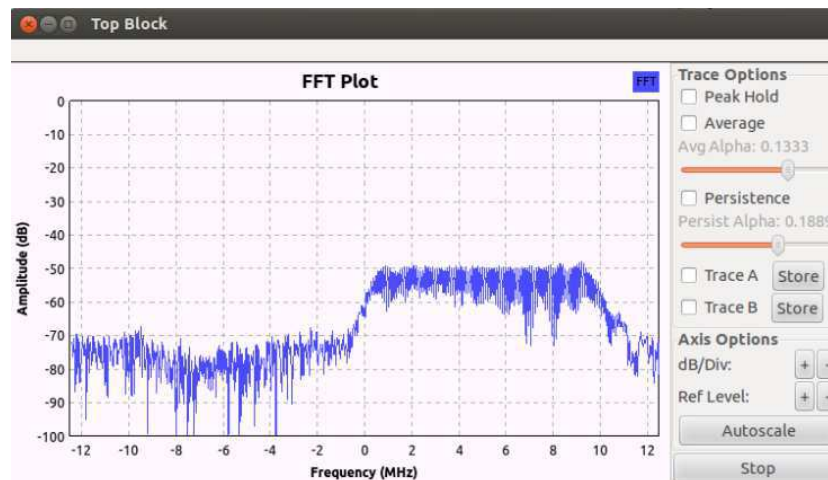


Figure 23 - Recepção do sinal *chirp*.

Uma forma de simular uma etiqueta passiva, é adicionar um filtro rejeita-faixas após a recepção, como exibido na Figura 24, de tal forma que seria possível codificar bits a partir da sua resposta em frequência, como mostrado na Figura 25, em que o nível alto é considerado como um bit 0 e o nível baixo como um bit 1. A diferença de níveis ocorre devido à adição de um filtro que rejeita a faixa de 3,5 MHz – 7,5 MHz.

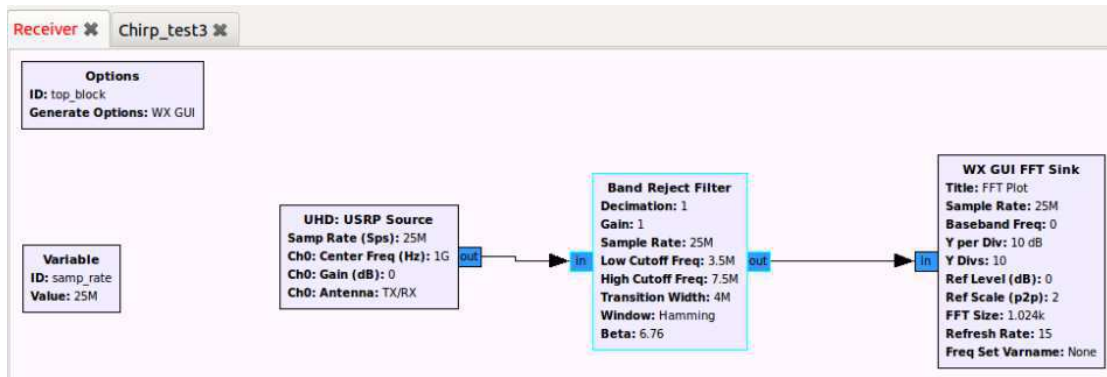


Figura 24 – Diagrama de blocos que mostra a adição de um filtro rejeita-faixas.

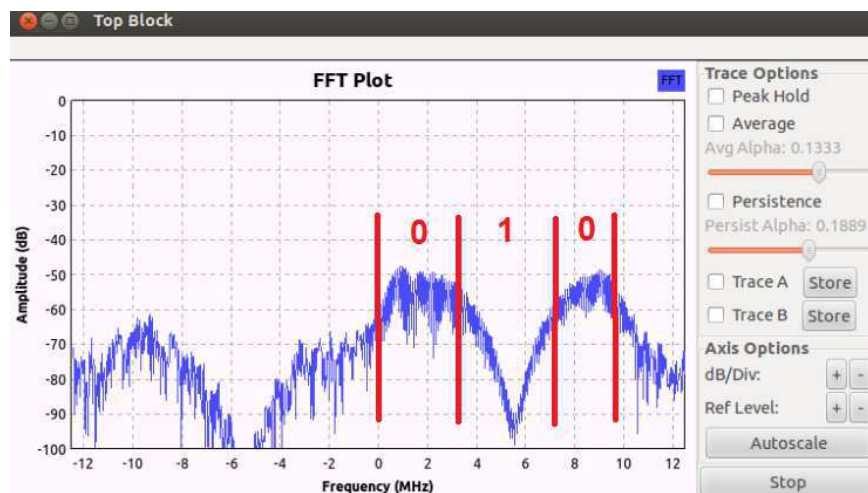


Figura 25 - Resposta em frequência após adição do filtro rejeita-faixas.

Para análises *offline*, sugere-se guardar a informação dos sinais recebidos em um arquivo do tipo *.dat* (Figura 26). É possível visualizar os dados via GRC ou MATLAB®.

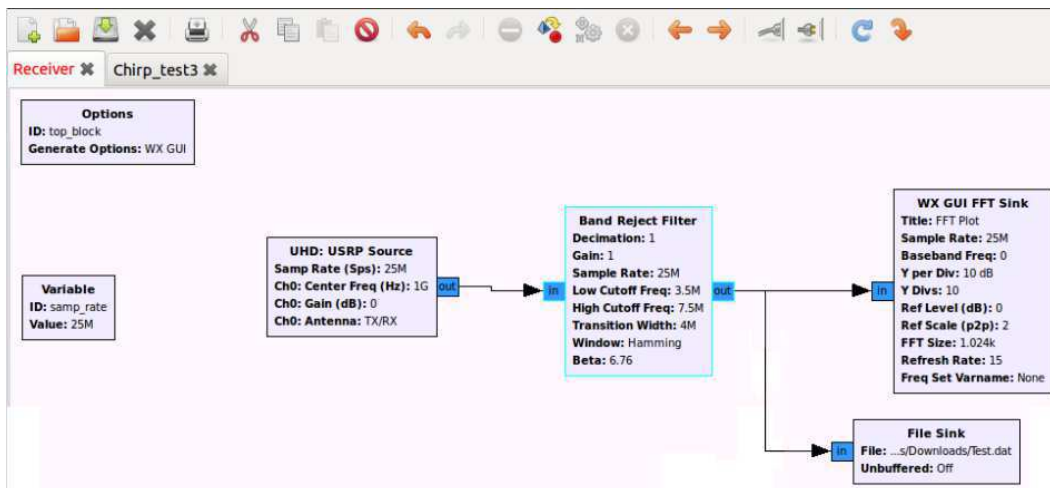


Figura 26 - Armazenamento em arquivo

Partindo da ideia de que para o sistema colisão seja necessária a utilização de pulsos de sinais *chirp*, nesse projeto foi implementado um bloco de processamento que ainda não fazia parte do escopo do GNU Radio. Ele foi nomeado de *Pulse Wave*, que simula uma fonte com ciclo de trabalho ajustável. Além desse parâmetro, o bloco recebe do usuário as informações de amplitude e frequência.

No esquema montado na Figura 27, a frequência do sinal gerado pelo bloco *Pulse Wave* é igual a um quarto da frequência do sinal dente-de-serra, ou seja, um período do *Pulse Wave* corresponde a quatro do dente-de-serra. Com ciclo de trabalho 0.25, multiplicando esses dois sinais resulta em um sinal em que um quarto do seu período corresponde a um período de dente-de-serra, enquanto outros três quartos permanecem nulos. Dessa forma, executando o programa, a resposta em frequência desse sinal varia de valores nulos à resposta mostrada na Figura 28.

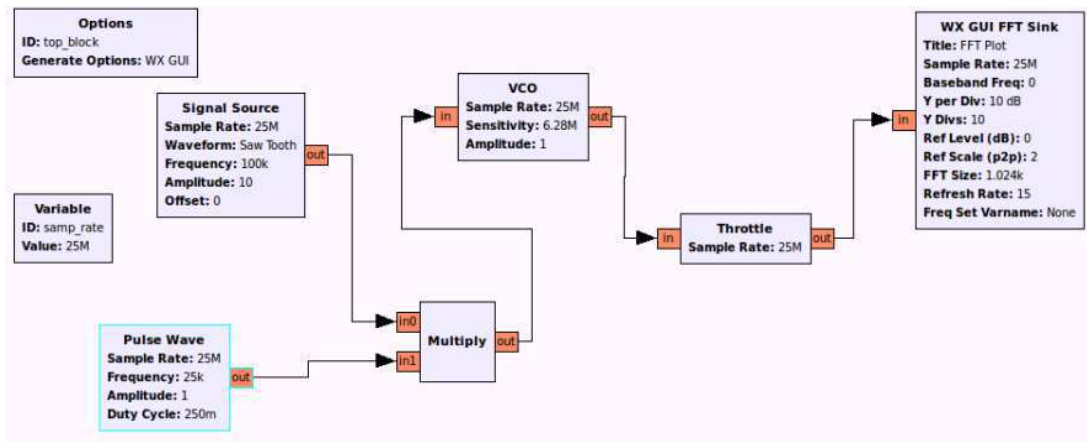


Figura 27 - Adição do bloco *Pulse Wave*.

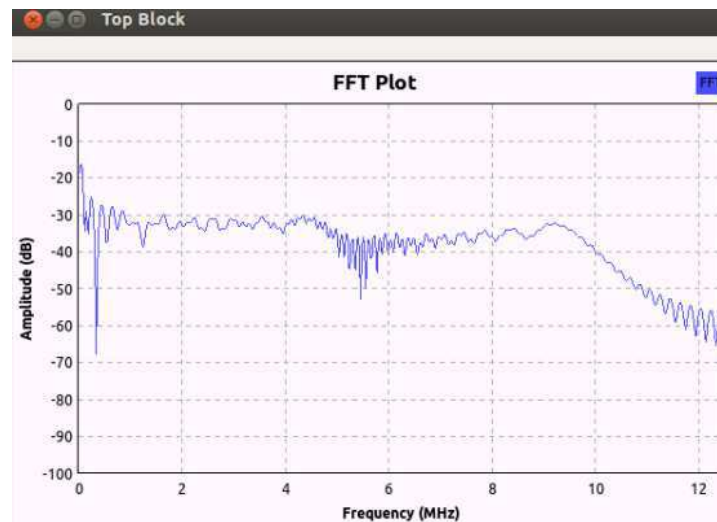


Figura 28 - FFT da multiplicação do sinal gerado pelo *Pulse Wave* e o sinal dente-de-serra.

Apesar da FFT estar aparentemente correta, foi possível notar uma dessincronização entre as fontes de sinais, o que pode levar a um erro na geração de pulsos de sinal *chirp*. Para teste, foram analisadas as visualizações de um sinal dente-de-serra e o gerado pelo *Pulse Wave*. Nota-se que a cada vez que o programa é executado, a diferença de fase entre as ondas geradas muda, indicando falta de sincronização (Figura 29). A Figura 30 ilustra o resultado da multiplicação de fontes.

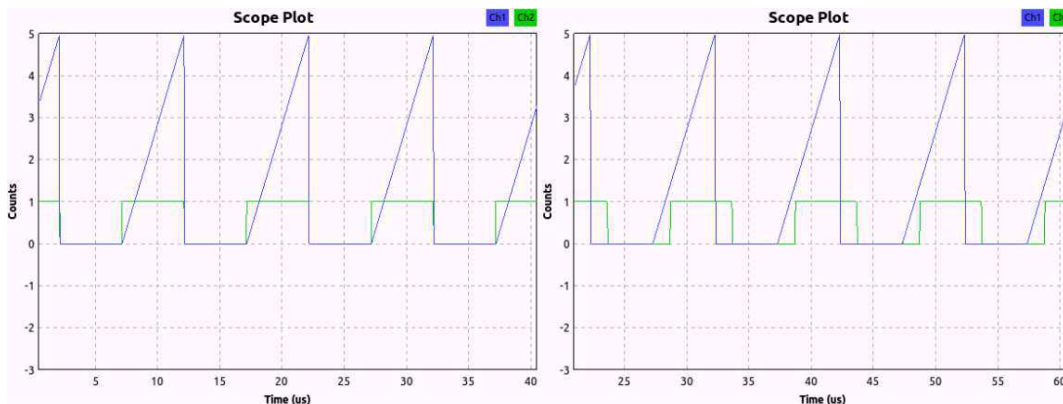


Figura 29 - Falta de sincronização entre as fontes.



Figura 30 - Resultado da multiplicação de um sinal dente-de-serra e um pulso retangular dessincronizados.

3.3. Limitações

Diante dos resultados apresentados, é possível listar algumas limitações que dificultem a interação do transmissor implementado com os outros trabalhos citados anteriormente:

- A falta de sincronização entre as fontes de sinais do GNU Radio impede que seja gerado um pulso *chirp* corretamente, sendo necessário um bloco que implemente um *clock* que sincronize tais fontes;
- O limite de largura de banda determinado pela USRP é de 100 MHz;

4. CONSIDERAÇÕES FINAIS

Como mencionado em [3], diante de um sistema RFID com muitas etiquetas, o sinal recebido pelo leitor resultará na colisão dos sinais de identificação de várias etiquetas, sendo necessário um sistema anti-colisão que separe a informação de cada etiqueta e as identifique.

Neste trabalho, foi implementado um transmissor de sinais *chirp* utilizando um RDS que pode ser utilizado posteriormente como leitor em um Sistema de Identificação RFID para excitar etiquetas passivas.

Limitações foram encontradas, como limite da largura de banda do hardware e processamento por parte dos computadores e falta de sincronização do sistema, o que impossibilitou a implementação de um sistema *online* de identificação.

Sugere-se um estudo sobre sincronização de fontes do GNU Radio, por meio do desenvolvimento de blocos que implementem um *clock* que regularize a fase de duas ou mais fontes utilizadas, bem como da sincronização entre USRPs, através do estudo e testes dos blocos já existentes para esse propósito, para melhoria da eficiência do transmissor.

Por fim, neste trabalho foram consolidados e aplicados conhecimentos obtidos ao longo do curso de graduação.

5. REFERÊNCIAS

- [1] SOLIC, P., RADIC, J., ROZIK, N., *Software Defined Radio Based Implementation of RFID Tag in Next Generation Mobiles*. IEEE Transactions on Consumer Electronics, 2012, p. 1051-1055.
- [2] PRERADOVIC, S., KARKAMAR, N., “*Chipless RFID Bar Code of The Future*”, *IEEE Microwave Magazine*, Dezembro de 2010, pp. 87-97.
- [3] AZIM, R., KARMAKAR, N., “*A Collision Avoidance Methodology for Chipless RFID Tags*”. Proceedings of the Asia-Pacific Microwave Conference 2011, p. 1514-1517.
- [4] BRIAND, A., ALBERT, B., GURJÃO, E. C., “*Complete Software Defined RFID System Using GNU Radio*”, *IEEE International Conference on RFID – Technologies and Applications*, 2012, pp. 287 – 291.
- [5] STOCKMAN, H., “*Communication by Means of Reflected Power*”, *Proceedings IRE*, Outubro de 1948, pp. 1196-1948.
- [6] LANDT, J., “*The History of RFID*”, *IEEE Potentials*, Outubro/Novembro de 2005, pp. 8-11.
- [7] HUNT, V., PUGLIA, A., PUGLIA, M., “*RFID A Guide to Radio Frequency Identification*”, Hoboken, Nova Jersey, EUA, 2007, Ed. Wiley.
- [8] WEINSTEIN, R., “*RFID: A Technical Overview and Its Application to Enterprise*”, *IT Pro*, Maio/Junho de 2005, pp. 27-33.
- [9] WU, D., NG, W., YEUNG, D., DING, H., “*A Brief Survey on Current RFID Applications*”, *Eighth International Conference on Machine Learning and Cybernetics*, Baoding, Julho de 2009, pp. 2330-2335.

- [10] PERRET, E., et. al., “*RF and THz Identification Using a New Generation of Chipless RFID Tags*”, *Radioengineering*, Vol. 20, No. 2, Junho de 2011, pp. 380-386.
- [11] BOEK, H., “*Some Hot North American RFID applications*”. Pode ser acessado em: < <http://www.rfidradio.com/?p=9> >. Último acesso em: 28 de Setembro de 2013.
- [12] SHRETHA, S., et. al., “*Transmission Line Reflection and Delay-based ID Generation scheme for RFID and other applications*”, *International Journal of Radio Frequency Identification Technology Applications*, Vol. 1, No. 4, 2007, pp. 401-416.
- [13] MCVAY, J., et. al., “*Space-filling curve RFID tags*”, *IEEE Radio and Wireless Symposium*, San Diego, CA, 2006, pp. 199-202.
- [14] SCHULER, et. al., “*Phase Modulation Scheme for Chipless RFID and Wireless Sensor Tags*”, *Asia Pacific Microwave Conference*, 2009, Singapura, 2009, pp. 229-232.
- [15] KLAUDER, J. R., et. al., “*The Theory and Design of Chirp Radars*”, *The Bell System Technical Journal*, Vol. 39, No. 4, 1960, pp. 745-808.
- [16] SMITH, T., Slides da Disciplina CMPT 468, *Simon Frase University*. Link: < http://mariobon.com/Glossario/___Teoria_dei_segnali/Chirp_1.pdf >. Último acesso: 30 de Setembro de 2013.
- [17] CANDAN, Ç., KUTAY, M., OZAKTAS, H., “*The Discrete Fractional Fourier Transform*”, *IEEE Transactions on Signal Processing*, Vol. 48, No. 5, Maio de 2000, p. 1329-1337.
- [18] *Software Defined Radio – White Paper*. Wipro Technologies. Disponível em: < http://www.ab4oj.com/dl/sdr_wipro.pdf >. Último acesso: 13 de Junho de 2013.
- [19] MITOLA III, J. *Software Radio Architecture, Object-Oriented Approaches to Wireless System Engineering*. 1a Edição, 2000. John Wiley & Sons.

[20] SELVA, A. F. B., et. al., *“Introduction to the Software-defined Approach”*, *IEEE Latin America Transactions*, Vol. 10, No. 1, Janeiro de 2012, pp. 1156-1161.

[21] Página oficial do GNU Radio: www.gnuradio.org. Último acesso: 03 de Outubro de 2013.

[22] FIDELES, F. B., GURJÃO, E. C., “Detecção da Ocupação do Espectro de Frequências usando Rádio Definido por Software”, VII Congresso de Iniciação Científica da UFCG, 2012.

[23] BRAGA, D. D., “Desenvolvimento de Software para Análise Gráfica em Tempo Real de Sistemas de Comunicação Digital Definidos por Software”, Trabalho de Conclusão de Curso do Departamento de Engenharia Elétrica da UFCG, Junho de 2011.

[24] Página da Ettus Research: www.ettus.com. Último acesso: 05 de Outubro de 2013.

[25] CARDOSO, M. H. B., “Projeto de Antenas e Setup para Medições de Tags RFID sem Chip”, Trabalho de Conclusão de Curso do Departamento de Engenharia Elétrica da UFCG, em desenvolvimento, 2013.

[26] LOPES, C., Transformada Fracionária de Fourier aplicada à Separação de TAGs em Sistemas RFID sem Chip. Dissertação de mestrado do Departamento de Engenharia Elétrica da UFCG, em desenvolvimento, 2013.

6. ANEXO – Instalação da USRP N200

É importante saber se seu computador possui placa de rede Ethernet Gigabit.

Para conectar a USRP N200 ao computador, primeiramente conceda à máquina um IP fixo, por meio do ícone conexões no canto direito da tela.

Clique "Editar Conexões" e em seguida "Editar" a Conexão Cabeada. Na aba "Configurações IPv4", edite o IP da máquina para 192.168.10.1 e máscara 255.255.255.0 de tal modo que o PC e a USRP estejam na mesma rede.

IP da USRP N200 = 192.168.10.2

Para testar a conexão, use os comandos

```
ping 192.168.10.2
```

```
uhd_find_devices
```

ou

```
uhd_usrp_probe
```

Neste último, caso seja a primeira vez que o dispositivo seja conectado ao PC, talvez seja necessário instalar o firmware e imagem do FPGA. Logo, localize tais arquivos. Neste exemplo, eles se encontram em

```
/usr/local/share/uhd/images
```

Usar *usrp_n200_fw.bin* e *usrp_n200_r4_fpga.bin*

Observar o número de revisão atrás do equipamento (Rev.: 4.0).

Pasta do programa *usrp_n2xx_net_burner.py*:

```
/usr/local/lib/uhd/utils
```

Uso:

```
./usrp_n2xx_net_burner.py --addr=192.168.10.2 --  
fw=/usr/local/share/uhd/images/usrp_n200_fw.bin --  
fpga=/usr/local/share/uhd/images/usrp_n200_r4_fpga.bin
```

Resete o dispositivo e teste-o novamente com o comando *uhd_usrp_probe*.