

# Radio Data System em Rádio Definido por Software

Jéssika C. da Silva, Millena M. de M. Campos, Vicente A. de Sousa Jr. .

**Resumo**— A adesão do *Radio Data System* (RDS) por parte das emissoras de rádio e dispositivos comerciais cresceu durante os últimos anos, em grande parte devido à possibilidade de novas funcionalidades do RDS para os usuários de FM. Neste trabalho, a especificação do RDS da ANATEL é apresentada e discutida, servindo como base para a criação de um sistema de transmissão/recepção FM com RDS implementado em rádio definido por software. Como metodologia de teste, a transmissão/recepção real de um sinal FM é implementado no software GNU Radio, com o uso do hardware USRP como interface aérea. Pode-se comprovar que mesmo o RDS oferecendo mais vantagens para os usuários, não requer grande complexidade do sistema.

**Palavras-Chave**— RDS; GNU Radio; USRP.

**Abstract**— The *Radio Data System* (RDS) adherence by radio and commercial devices has grown over the last few years, largely because of the possibility of new RDS features for FM users. In this work, the ANATEL RDS specification is presented and discussed, serving as the basis for the creation of an FM radio transmission / reception system implemented in software-defined radio. The methodology used for the tests include the actual transmission / reception of an FM signal implemented in the GNU Radio software, using the USRP hardware as the air interface. It can be proven that even though RDS offers more advantages for the users, it does not require great system complexity

## I. INTRODUÇÃO

O RDS (*Radio Data System*) é um sistema de transmissão digital, pelo qual mensagens curtas de texto são transmitidas em paralelo à informação de áudio. O RDS foi desenvolvido pela EBU (*European Broadcasting Union*) com o intuito de trazer mais funcionalidades ao serviço de rádio FM, como por exemplo, a visualização do nome da emissora e da música na tela do receptor. São 20 funções padronizadas pela EBU, que além de trazer mais conforto para o ouvinte, podem tornar o receptor mais eficiente, por exemplo através do uso da opção de sintonização dinâmica de emissoras. No Brasil, o sistema é normatizado pela ANATEL [1] e está presente a mais de 20 anos, com número crescente de adeptos.

Implementar o RDS com a tecnologia de *Software-defined radio* (SDR) torna o sistema de rádio FM ainda mais flexível. O *Wireless Innovation Forum* [2], define SDR como: “O rádio no qual algumas ou todas as funções da camada física são definidas por software”. Isto significa tornar os dispositivos de comunicação mais versáteis e aumentar a interoperabilidade entre diferentes sistemas. Qualquer melhoria implementada não está atrelada a troca do dispositivo, mas apenas uma atualização do software, um dos motivos pelos quais a maioria dos sistemas de comunicação tendem a se tornar definidos por software, incluindo o RDS. Implementar o RDS em SDR abre espaço para um conjunto de possibilidades, como a sintonia

automática no tipo de programação desejada pelo ouvinte ou no sinal com melhor potência, sem a necessidade de novos circuitos analógicos.

O hardware de um dispositivo SDR consiste basicamente em uma plataforma genérica para processamento, como FPGA ou DSP, conectada a um *front-end* de rádio frequência, através de conversores ADC/DAC. Uma plataforma SDR bastante utilizada em pesquisas acadêmicas é a USRP (*Universal Software Radio Peripheral*) da Ettus Research, pois permite transmissão e recepção, e tem flexibilidade de operar em uma larga faixa do espectro. Além dos componentes básicos do SDR, a USRP possui interface de comunicação com um computador, que é encarregado, além de configurar o SDR, de realizar tarefas de processamento de sinais em software. Uma alternativa de menor custo é o *dongle* receptor de TV digital USB RTL-SDR. Ele opera entre 25 e 1,75 GHz, o que inclui aplicações como rádio FM, sinais ISM e sistemas celulares. Embora seja mais simples, é compacto e acessível, tornando-se viável principalmente para estudantes complementarem seu aprendizado fora de sala de aula.

Um software muito utilizado para desenvolvimento em SDR é o GNU Radio. Ele tem código aberto e oferece um conjunto de blocos funcionais (moduladores, codificadores, entre outros), existindo a possibilidade de desenvolver novos blocos ou alterar os existentes de acordo com a necessidade do usuário.

Apesar de ser uma tendência nas rádios FM, o termo RDS não é muito popular, nem sua tecnologia é difundida. Outros trabalhos, com o mesmo objetivo de mostrar o funcionamento do RDS, serviram de base para a implementação e os testes realizados neste artigo. Os autores de [3] apresentam as especificações e o funcionamento do RDS para seu país, e disponibilizam uma apresentação matemática sobre o processo de transmissão/recepção. Eles utilizam o Matlab® como ferramenta de prototipagem, fazendo a adição de ruído branco e análise da relação sinal-ruído.

Outro estudo mais próximo ao apresentado neste artigo é o [4], pois utiliza equipamentos para a transmissão FM com RDS e a recepção é finalizada por interface de computador. Os autores não apresentam a modelagem do sistema RDS, mas fazem a distinção do sistema RDS (europeu) e RBDS (americano). Contudo, o protótipo construído não é tão viável para pesquisa, pois os equipamentos não são flexíveis a mudanças.

Este artigo utiliza uma plataforma de SDR juntamente com o GNU Radio, e cria uma cadeia de transmissão e recepção do sinal FM com RDS, além de detalhar receptor e transmissor feitos no software, relacionando os procedimentos do RDS com os blocos utilizados. Uma atenção especial é dedicada a descrição do processo de codificação RDS, algo não encontrado de maneira acessível na literatura.

Os autores são da Universidade Federal do Rio Grande do Norte, Natal, RN. Emails: {jessika,millena,vicente}.gppcom@gmail.com

## II. Radio Data System (RDS)

Para elaboração das normas definidas na Resolução nº 349/2003 [1], a ANATEL utilizou como base a especificação do RDS para radiodifusão sonora em VHF/FM, na faixa de frequências de 87,5-108 MHz, do comitê europeu de normalização eletrotécnica (CENELEC) [5]. Esta seção descreve alguns aspectos da especificação do RDS no Brasil, incluindo os tipos de informações transmitidas, sua disposição no espectro de rádio e como se dá a formação dos grupos de transmissão que carregam as informações de texto.

### A. Modulação e Disposição do Sinal no Espectro

Na transmissão FM estéreo comercial, um sinal de áudio com faixa de 30 Hz à 15 kHz é a entrada de um sistema de multiplexação, no qual a soma dos canais direito (R) e esquerdo (L) é transmitida sem alterações. Já a subtração dos canais (L-R) é modulada usando DSB-SC (*Double Side Band-Supressed Carrier*) em uma portadora de 38 kHz (segunda harmônica de 19 kHz). Um sinal piloto com frequência de 19 kHz é transmitido para indicar a presença de transmissão estereofônica e que no receptor seja feita a demodulação coerente. Para que não haja interferência do sinal de áudio com o RDS, este é também modulado usando DSB-SC em uma portadora de 57 kHz (terceira harmônica), conforme ilustrado na Fig. 1.

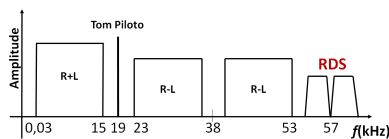


Fig. 1. Espectro do sinal FM estéreo comercial com RDS.

Antes da modulação em banda-passante, a mensagem RDS em banda-base é formatada a uma taxa de 1187,5 bps, codificada diferencialmente e em seguida é feita a codificação Manchester. Com a codificação diferencial elimina-se a necessidade de detecção coerente no receptor, já que a informação fica contida na mudança de bits. Já a codificação Manchester garante auto-sincronização receptor/transmissor.

### B. Codificação de Faixa-Base

Como definido em [1], uma codificação de faixa-base é realizada pelo *RDS encoder*, especificando que as mensagens do RDS são organizadas em grupos, como mostrado na Fig. 2.

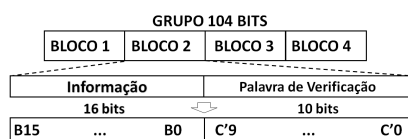


Fig. 2. Codificação de Faixa-Base do RDS para cada grupo (adaptado de [1]).

Existem 16 grupos classificados em tipos A e B. Cada tipo de grupo é responsável por uma aplicação específica e carrega mensagens necessárias à sincronização: PI, TP, PTY e

identificadores de grupo. Como mensagens de sincronização, elas estão presentes em todos os grupos e ocupam posições fixas no primeiro bloco, de forma que o receptor não precisa de informações adicionais para decodificá-las.

Cada grupo tem 4 blocos de 26 bits (104 bits no total), dos quais o bit mais significativo de cada bloco é transmitido primeiro. Apenas 16 bits são de informação, os restantes constituem a palavra de verificação e a palavra agregada, responsáveis pelo sincronismo de blocos e correção de erros.

### C. Tipos de Informação

A primeira função a ser transmitida é o *programme identification* (PI). Trata-se de um código que serve para identificar um conjunto de emissoras inter-relacionadas numa determinada área geográfica. Ele não é usado para exposição ao ouvinte, mas para leitura no receptor. Os primeiros 16 bits (16 MSBs) de todos os grupos indicam 4 caracteres hexadecimais que constituem o código PI. O primeiro caractere identifica o país (B, C, D e 3 indicam o Brasil), segundo a ANATEL [1]. O segundo caractere indica a área coberta pelo sinal das emissoras inter-relacionadas (local, regional, nacional, internacional, supra regional). Os outros dois caracteres constituem um código específico atribuído pela ANATEL, destinado para cada conjunto de emissoras. Duas emissoras diferentes só terão o mesmo código PI se transmitirem programas idênticos e estiverem na mesma área geográfica.

Um uso interessante para o PI é que caso exista uma má recepção do sinal. O receptor pode procurar outras frequências transmitindo o mesmo código PI e fazer a comutação de acordo com a potência do sinal recebido em cada uma. Este é um exemplo de aplicação mais facilmente implementada em SDR, melhorando a recepção sem a necessidade de adicionar componentes analógicos.

O *PTY (Program Type)* identifica o tipo de programação que está sendo transmitida por meio de 5 bits que ocupam da 7ª até a 11ª posição do segundo bloco. Sua relação com a programação está especificada na Tabela A3 da resolução da ANATEL [1]. O *PTY* pode ser alterado de acordo com o programa transmitido e também pode ser utilizado para sintonia automática no receptor. Pode ser facilmente implementado em GNU Radio um bloco em que o ouvinte seleciona o tipo de programa que deseja escutar, e o receptor automaticamente varre o espectro FM a procura de uma frequência que esteja transmitindo o correspondente *PTY*.

O *TP (Traffic Announcement)* é o indicador de informações de trânsito. Ele ocupa o 6º bit do bloco 2 e serve para indicar se a emissora transporta informações de trânsito. Uma função diretamente relacionada é a *TA (Traffic Program Indication)* - ocupa a 12ª posição no grupo 0), que indica se a emissora está transportando informações de trânsito no exato momento. Se *TP* for 1 e *TA* for 0, é porque a emissora não está transmitindo informação de trânsito, mas eventualmente poderá transmitir. Se *TP* for 0 e *TA* for 1, a emissora não transmite informações de trânsito, mas outra emissora inter-relacionada está transmitindo. Nesse modo, o receptor pode sintonizar na outra emissora por meio de uma lista de frequências alternativas.

Os 5 MSBs do segundo bloco se destinam a especificação de qual grupo está sendo enviado e servirá para que o decodificador possa identificar o grupo e ler seus dados corretamente. Os 4 MSBs indicam o tipo, conforme tabela presente na página 10 de [1]. O quinto bit é 0, se o grupo for do tipo A, ou 1, se for do tipo B.

Devido à importância dos identificadores e dessas 3 funções, esses dados são transmitidos em todos os grupos. Existem outras funções que são transmitidas a uma taxa menor, mas cuja transmissão é obrigatória por serem necessárias à sintonização. O grupo 0, tipo A ou B, mostrados na Fig. 3, reúne todas essas informações, e por isso, sua transmissão também é obrigatória, enquanto a transmissão dos demais grupos é opcional.

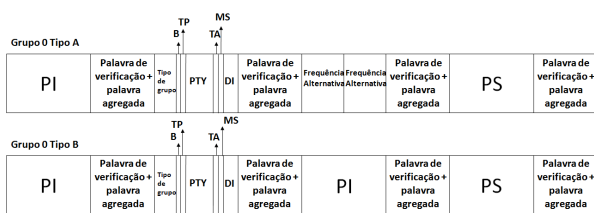


Fig. 3. Formato do Grupo 0 dos tipos A e B.

A 13ª posição do bloco 2 do grupo 0 (1 bit) indica se o áudio transmitido é de música ou locução. Se o áudio for de locução, não há necessidade de um filtro com as mesmas qualidades de um filtro utilizado para música. Em SDR os parâmetros do filtro podem ser mudados em função deste bit.

Por fim, os três últimos bits do bloco 2 no grupo 0 indicam o modo de operação de áudio da emissora. Os dois últimos bits (D1D0) indicam a opção de modo (mono ou estéreo, comprimido ou não comprimido, PTY dinâmico ou estático, e outra opção não usada no Brasil). Já o bit anterior (D2) é usado para indicar o estado de operação dessa opção de modo. Por exemplo, para D1D0 = 00, D2 = 1 se o áudio é estéreo, D2 = 0 se é mono. Todas as opções de modo devem ser enviadas, o que torna necessário a transmissão de pelo menos 4 grupos.

Com o desenvolvimento em SDR, o receptor pode ter a demodulação mono e a estéreo, sem a necessidade de dois circuitos analógicos específicos para cada uma.

No terceiro bloco são transmitidos dois bytes para indicar duas frequências alternativas. A primeira opção 00000000 não é usada, a segunda 00000001 significa 87,6 MHz, e para obter as seguintes soma-se 0,1 MHz. A frequência máxima é 107,9 MHz obtida para o byte 11001100. A posição 224 indica que não existe frequência alternativa. As outras opções não tem significado atribuído relativo à frequência alternativa. A lista de frequências alternativas é montada pela emissora, e consiste no conjunto de frequências, nas quais emissoras relacionadas estão transmitindo o mesmo programa.

No quarto bloco é transmitido o nome da emissora (PS) através de 16 bits que indicam 2 caracteres alfanuméricos. É recomendado que o nome da emissora seja formado por oito caracteres alfanuméricos, que é o número de caracteres geralmente disponíveis no visor do receptor, o que torna necessária a transmissão de pelo menos 4 grupos para formar

o nome da rádio. Para transmitir além do nome da emissora, pode-se utilizar a função de *radio text*, específica do grupo 2.

### III. RDS NO GNU RADIO

Este trabalho utilizou um repositório de código aberto [6] que disponibiliza os blocos funcionais de codificação e decodificação do RDS no GNU Radio. Foi utilizada a versão do GNU Radio 3.7 no Ubuntu 14.04. Com o uso dos blocos funcionais e alguns exemplos disponíveis no próprio repositório, foi possível criar os experimentos deste trabalho no GNU Radio Companion, os quais são descritos nesta seção.

#### A. O Transmissor RDS

O primeiro bloco do transmissor é o RDS *encoder*, responsável pela codificação de faixa-base. Sua configuração é similar ao mostrado na Fig. 4, sendo possível configurar os dados da emissora. A sua saída é a sequência de bits codificada em grupos de acordo com as especificações do RDS (taxa de amostragem de 1187,5 amostras/s).

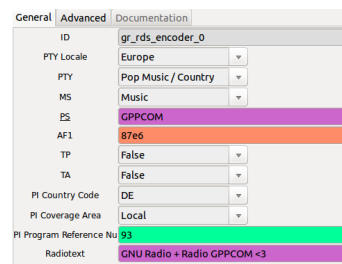


Fig. 4. RDS Encoder implementado no GNU Radio.

Outra parte importante do transmissor é mostrada na Fig. 5, com destaque para a codificação em banda-base composta pela codificação diferencial (*Differential encoder*) e pela codificação Manchester.

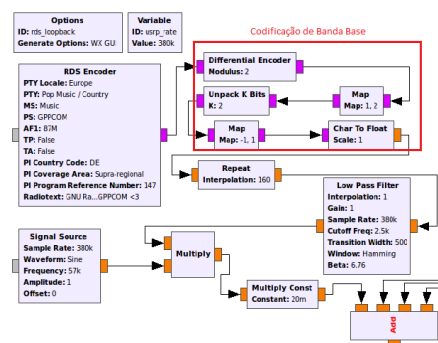


Fig. 5. Modulação DSB-SC, codificações diferencial e Manchester.

A codificação diferencial equivale a uma operação *xor* entre a entrada atual e a saída anterior, confinando a informação na mudança dos bits. Na codificação Manchester, cada bit é representado por uma mudança entre dois bits. Assim, se a informação de entrada for 0 com tempo de bit  $\tau$ , então a saída será representada por dois bits 10, com tempo de bit igual a  $\tau/2$ . Após a codificação Manchester, cada bit terá metade do tempo de bit original, com o dobro

da taxa de amostragem. Este aumento traz o benefício da auto-sincronização no receptor e da eliminação da componente DC.

Concluída a codificação, o sinal, com taxa de amostragem de 2375 amostras/s, é filtrado para limitar a banda do RDS em 5 kHz. Em seguida é modulado por uma portadora senoidal de 57 kHz (multiplexação com o sinal de áudio FM). É necessário casar a taxa da saída do codificador com a taxa da portadora, função realizada pelo bloco *Repeat*. Após a filtragem e a modulação, o sinal RDS é multiplexado com os outros sinais da transmissão FM: piloto, canal direito e esquerdo, como mostrado na Fig. 6.

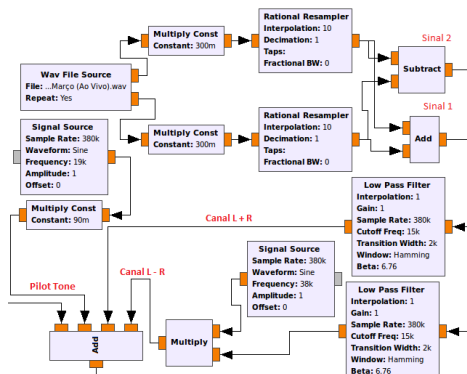


Fig. 6. Geração de áudio estéreo implementada no GNU Radio.

O bloco *Wav File Source*, responsável pela abertura de um arquivo de áudio, possui duas saídas (canais R e L), as quais são somadas para compor o sinal 1 centrado em 0 Hz, e subtraídas para compor o sinal 2 que será centrado em 38 kHz. O bloco *Rational Resampler* adéqua a taxa de amostragem do arquivo de áudio para 380 kHz (mesma taxa do restante do sistema). Posteriormente, a largura de banda do sinal é limitada pelo bloco *Low Pass Filter*, um filtro passa baixa, que com os parâmetros definidos, faz a filtragem com uma frequência de corte de 15 kHz e banda de transição igual a 2 kHz. Esta banda de transição é relativamente alta, mas valores pequenos resultam em maior processamento. O passo seguinte é o deslocamento do sinal (L-R) para 38 kHz para compor o sinal 2 e finalmente todos os sinais são somados, modulados na frequência e transmitidos via USRP por meio do bloco *UHD: USRP sink*, conforme Fig. 7.

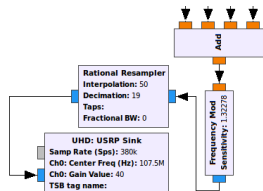


Fig. 7. Modulação FM e transmissão com USRP via GNU Radio.

### B. O Receptor RDS

O primeiro passo da recepção é a coleta e tratamento do sinal FM, como mostrado na Fig. 8. Ele é recebido pela USRP (bloco *UHD: USRP Source*), e em seguida filtrado

para preservar somente o sinal da emissora desejada (bloco *Frequency Xlating FIR Filter*). Por fim, é feita a demodulação FM pelo bloco *WBFM Receive*, retornando o sinal em banda-base.

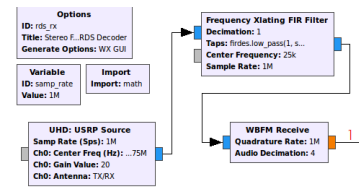


Fig. 8. Recepção do sinal FM com RDS implementado no GNU Radio.

A decodificação do sinal RDS é mostrada na Fig. 9. Inicialmente, um filtro (*Frequency Xlating FIR Filter*) extrai o sinal da terceira harmônica (57 kHz). Posteriormente, sinal é formatado por filtro cosseno elevado (*Root Raised Cosine Filter*) e a taxa de amostragem é ajustada para 2375 Hz. O bloco *MPSK Receiver* demodula o sinal, incluindo estratégias de sincronização fase, frequência e símbolo. Como parte final da demodulação, os blocos *Complex To Real*, *Binary Slicer*, *Keep 1 in N* e *Differential Decoder* fazem a decodificação Manchester e a decodificação diferencial. O bloco *RDS Decoder* realiza a decodificação RDS e os blocos restantes apresentam as informações de texto obtidas pelo receptor.

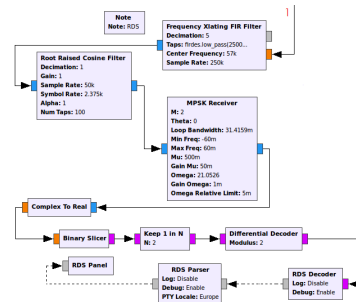


Fig. 9. Demodulação do sinal RDS implementado em GNU Radio.

O sinal piloto é extraído pelo *Decimating FIR Filter*, centrado em 19 kHz, e o conjunto de blocos mostrados na Fig. 10. Essa operação resulta em uma versão limpa do sinal piloto com o dobro da frequência devido ao bloco *Multiply*, resultando em uma portadora em 38 kHz.

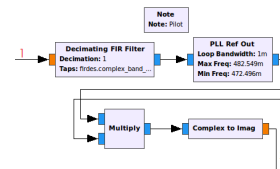


Fig. 10. Tratamento do sinal piloto no receptor.

A extração do sinal estéreo é mostrada na Fig. 11. Um filtro passa-baixa extrai o sinal (L+R) e um filtro passa-faixa de 23 kHz até 38 kHz recupera o sinal (L-R), o qual é demodulado com a portadora de 38 kHz resultante do sinal piloto e filtrado, para um melhor resultado na extração deste sinal.

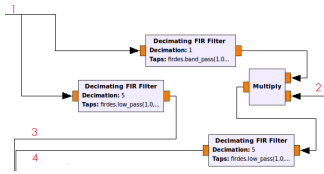


Fig. 11. Extração dos sinais (L+R) e (L-R) implementado no GNU Radio.

As componentes L e R são separadas e levadas ao bloco *Audio Sink* para reprodução audível, de acordo com a Fig. 12. Foram utilizados os blocos *FM Deemphasis* e *Multiply Constante* para melhorar a qualidade do sinal e por último o bloco *Rational Resampler* para adequar a taxa de amostragem para a placa de som do computador.

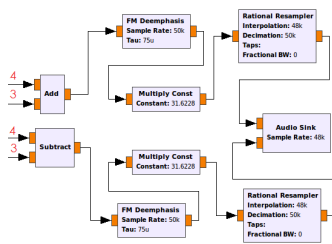


Fig. 12. Obtenção do áudio estéreo FM e reprodução no computador.

### C. Loopback

O *Loopback* consiste na montagem do receptor e do transmissor RDS em apenas um arquivo, com objetivo de analisar o funcionamento da implementação em GNU Radio sem o uso da USRP. Existe apenas a conexão direta entre o blocos do receptor e transmissor descritos anteriormente, sem o uso dos blocos *UHD*.

Ao rodar o código são mostrados o gráfico do espectro RDS e as informações do RDS (Fig. 13). É possível verificar a decodificação das informações configuradas na Fig. 4.

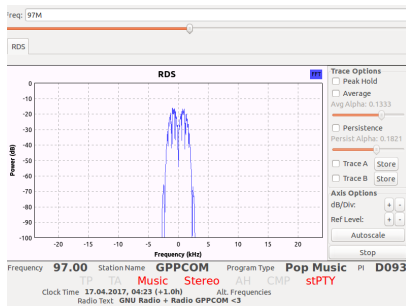


Fig. 13. Saída do *loopback* RDS implementado em GNU Radio.

### D. RDS em plataforma de SDR

O experimento foi feito utilizando o receptor de FM com RDS no GNU Radio e a USRP, sintonizando em uma emissora que possui RDS, a 96.7FM, em Natal, RN. A conexão da USRP com o computador é feita via ethernet, como mostrado na Fig. 14. Após a execução do código, obtém-se a tela da

Fig. 15. Além de poder selecionar a frequência de recepção sem precisar alterar o código, esta interface mostra o espectro FM e como alternativa o espectro RDS, mostra as informações de frequência, nome da emissora, *radio text*, PI, PTY, tipo de programação, etc.



Fig. 14. Setup do Receptor FM-RDS com a USRP.

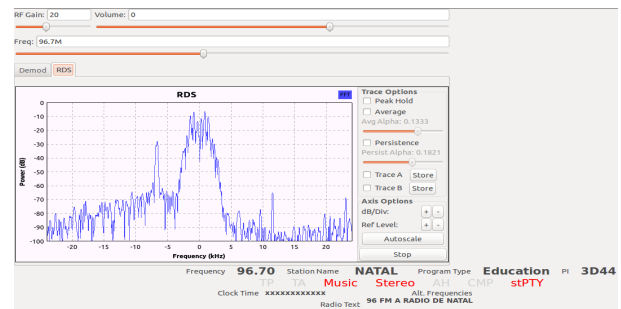


Fig. 15. Interface do receptor FM-RDS na USRP sintonizado em 96.7 MHz.

## IV. CONCLUSÕES E COMENTÁRIOS FINAIS

O sistema RDS é uma realidade nas rádios FM brasileiras e em alguns equipamentos de rádio, este artigo evidencia que é possível a implementação de sistemas comerciais em SDR utilizando plataformas de baixo custo com um arcabouço de programação de alto nível como o GNU Radio. Como trabalhos futuros, espera-se utilizar o sistema RDS implementado em GNU Radio para explorar outras aplicações, tais como: sistema com sintonia automática no tipo de programação desejada, sistema de anúncio de climáticos e de emergência via RDS, entre outros.

## REFERÊNCIAS

- [1] ANATEL, "Resolução nº 349, de 25 de setembro de 2003," 2003.
- [2] WirelessInnovationForum. (2017) What is software defined radio? [Online]. Available: <http://www.wirelessinnovation.org/assets/documents/SoftwareDefinedRadio.pdf>
- [3] C. Roppel and A. O. Perez, "An efficient implementation of an fm/rds software radio," *4th European DSP in Education and Research Conference*, 2010.
- [4] C. Barca, C. Neamtu, H. Popescu, S. Dumitrescu, and A.-S. Sandu, "Implementation of RDS platform solutions for an emergency system," *IEEE*, 2013.
- [5] CENELEC, "Specification of the radio data system (RDS) for VHF/FM sound broadcasting in the frequency range from 87,5 to 108 Mhz," 1998.
- [6] B. Bloessl. (2016) FM RDS/TMC transceiver. [Online]. Available: <https://github.com/bastibl/gr-rds>