

UNIVERSIDADE ESTADUAL DO MARANHÃO
CENTRO DE CIÊNCIAS TECNOLÓGICAS
CURSO DE ENGENHARIA DE COMPUTAÇÃO

JOÃO RENATO FREITAS MENDES

**PROPOSTA DE PROTÓTIPO DE RÁDIO DE COMUNICAÇÃO EM 250 MHZ COM
TECNOLOGIA DE RÁDIO DEFINIDO POR SOFTWARE PARA INTERNET DAS
COISAS**

SÃO LUÍS – MA

2019

JOÃO RENATO FREITAS MENDES

**PROPOSTA DE PROTÓTIPO DE RÁDIO DE COMUNICAÇÃO EM 250 MHZ COM
TECNOLOGIA DE RÁDIO DEFINIDO POR SOFTWARE PARA INTERNET DAS
COISAS**

Trabalho apresentado ao curso de Graduação em Engenharia de Computação na Universidade Estadual do Maranhão como pré-requisito para obtenção do título de Bacharel sob orientação do Prof. Dr. Carlos Henrique Rodrigues de Oliveira.

SÃO LUÍS - MA

2019

Mendes, João Renato Freitas.

Proposta de protótipo de rádio de comunicação em 250 MHz com tecnologia de rádio definida por software para internet das coisas / João Renato Freitas Mendes. – São Luís, 2019.

68f.

Monografia (Graduação) – Curso de Engenharia de Computação, Universidade Estadual do Maranhão, 2019.

Orientador: Prof. Dr. Carlos Henrique Rodrigues de Oliveira.

1.Rádio. 2.Software. 3.Indústria 4.0. 4.Internet das coisas. 5.Interoperabilidade.
I.Título

CDU: 654.16:004.4

JOÃO RENATO FREITAS MENDES

**PROPOSTA DE PROTÓTIPO DE RÁDIO DE COMUNICAÇÃO EM 250 MHZ COM
TECNOLOGIA DE RÁDIO DEFINIDO POR SOFTWARE PARA INTERNET DAS
COISAS**

Monografia apresentada ao Curso de Engenharia de Computação da Universidade Estadual do Maranhão, como registro para obtenção do grau de Bacharel em Engenharia de Computação.

Trabalho aprovado. São Luís – MA, ____ de _____ de 2019.

Prof. Dr. Carlos Henrique Rodrigues de Oliveira

Orientador

Esp. Ana Paula Ferreira Costa

Primeiro Membro

Esp. Luiz Carlos Chaves Lima Junior

Segundo Membro

AGRADECIMENTOS

A Deus, pelo dom soberano da vida, pela infinita bondade e misericórdia e pelo cuidado durante os diversos períodos de formação acadêmica até o presente momento em que teço estas linhas.

À minha família por sempre me apoiar em todas as coisas, até mesmo nos momentos mais críticos, sempre fazendo o possível e o impossível para meu progresso como ser humano, cidadão e acadêmico.

À pessoa do Prof. Dr. Carlos Henrique, exemplo de tutor e mestre, que me guiou na construção deste projeto desde sua concepção.

Ao Prof. Msc. Denner Araujo pela ajuda no aprendizado das plataformas utilizadas e por fornecer os materiais necessários para a realização do sistema, meu muito obrigado.

“Porque Deus é o que opera em vós tanto o querer como o efetuar, segundo a sua boa vontade.”

Apóstolo Paulo

RESUMO

O presente trabalho propõe uma solução de um sistema de comunicação via rádio para Internet das Coisas, tendo como foco um alcance médio de cobertura, baixas taxas de transmissão de dados, baixo custo e baixo consumo de energia utilizando a tecnologia de Rádio Definido por Software na faixa de 250 MHz, tendo como ponto de partida a Resolução n° 555 de 20 de dezembro de 2010 da Agência Nacional de Telecomunicações (ANATEL). Apresenta-se, portanto, a constituição de sua arquitetura ideal e a construção de um protótipo experimental, que consiste em um sistema *Wide Band Frequency Modulation* (WBFM) a partir de uma comunicação em linha de visada direta utilizando dois tipos de rádio, sendo um para transmissão (LimeSDR-Mini) e outro para recepção do sinal (RTL2832U) aliados ao uso de dois softwares de desenvolvimento para configuração dos mesmos, o *GNU Radio* (utilizado no transmissor) e o *SDRSharp* (utilizado no receptor). O protótipo é configurado de maneira que permita a operação tanto na faixa de 250 MHz, como no espectro convencional de rádio (sendo utilizada para o experimento a frequência de 105,6 MHz), permitindo a possibilidade de mudança de faixa da portadora em tempo real. Os resultados obtidos provam sua viabilidade e flexibilidade de operação, assim como seu apelo de baixo custo e praticidade de programação e configuração.

Palavras-chave: Rádio Definido por Software; Indústria 4.0; Internet das Coisas; Interoperabilidade.

ABSTRACT

The present work proposes a solution of radio communication system for Internet of Things, focusing on a medium coverage range, low data rates, low cost and low power consumption using the Software Defined Radio technology in the 250 MHz band, having as its starting point the Resolution n° 555 from December 20, 2010 by the National Telecommunications Agency (ANATEL). Therefore, the constitution of its ideal architecture and the construction of an experimental prototype is presented, which consists of a Wide Band Frequency Modulation (WBFM) system from a direct line of sight communication using two types of radio, one of which is for transmission (LimeSDR-Mini) and another for signal reception (RTL2832U) combined with the use of two development software to configure them, GNU Radio (used in the transmitter) and SDRSharp (used in the receiver). The prototype is configured to allow operation in both the 250 MHz band and the conventional radio spectrum (105.6 MHz frequency is used for the experiment), allowing the possibility of real-time carrier band switching. The obtained results prove its viability and flexibility of operation, as well as its low cost appeal and practicality of programming and configuration.

Keywords: Software Defined Radio; Industry 4.0; Internet of Things; Interoperability.

LISTA DE FIGURAS

Figura 1 - Comparativo entre redes LPWA e demais redes tradicionais	19
Figura 2 - Cinco cenários típicos de aplicações LPWA M2M	20
Figura 3 - Largura de banda de um canal Sigfox ultra-narrowband	21
Figura 4 - Exemplo de tamanho de <i>payload</i> SigFox	22
Figura 5 - Arquitetura de rede Sigfox	23
Figura 6 - Especificações LoRa regionais	24
Figura 7 - Arquitetura da Rede LoRa	25
Figura 8 - Casos de uso da tecnologia Wi-SUN	28
Figura 9 - Visão sistêmica de um SDR ideal	31
Figura 10 - Arquitetura básica de um SDR	32
Figura 11 - LimeSDR USB com plug tipo A	35
Figura 12 - LimeSDR-Mini v1.2	36
Figura 13 - Parte superior da placa com os componentes principais	37
Figura 14 - Diagrama de blocos da arquitetura do LimeSDR-Mini	38
Figura 15 - Circuito integrado LMS7002M	39
Figura 16 - Ambiente de testes do GNU Radio	41
Figura 17 - Diagrama de blocos da arquitetura	43
Figura 18 - Diagrama de blocos do transmissor WBFM	46
Figura 19 - Bloco Options	47
Figura 20 - Bloco Variable	48
Figura 21 - Bloco WX GUI Chooser	49
Figura 22 - Bloco Audio Source	49
Figura 23 - Fonte de monitoração de som do computador	50
Figura 24 - Arquivo .asoundrc	51
Figura 25 - Bloco WX GUI FFT Sink	51

Figura 26 - Bloco WX GUI Waterfall Sink	52
Figura 27 - Número de série da placa LimeSDR utilizada	53
Figura 28 - Bloco LimeSuite Sink (TX)	53
Figura 29 - Funções secundárias do bloco LimeSuite Sink (TX)	54
Figura 30 - Bloco WBFM Transmit	55
Figura 31 - Bloco Rational Resampler	56
Figura 32 - RTL2832U	57
Figura 33 - Instalação de Drivers via Zadig	58
Figura 34 - Interface SDR#	58
Figura 35 - Seleção da placa RTL-SDR para uso no SDR#	59
Figura 36 - Desktop utilizado para o Tx	60
Figura 37 - LimeSDR-Mini utilizado	61
Figura 38 - Notebook utilizado para o Rx	61
Figura 39 - RTL2832U utilizado	60
Figura 40 - Gráficos do Tx em 105,6 MHz	62
Figura 41 - Gráficos do Rx em 105,6 MHz	62
Figura 42 - Gráficos do Tx em 250 MHz	63
Figura 43 - Gráficos do Rx em 250 MHz	64

LISTA DE TABELAS

Tabela 1 - Sistemas Multicanais com canalização de 1,25 MHz	17
Tabela 2 - Principais componentes do LimeSDR-Mini	36
Tabela 3 - Estimativa de custo do projeto	59

LISTA DE ABREVIATURAS E SIGLAS

3GPP – *Third Generation Partnership Project*
ADC – *Analog to Digital Converter*
ALSA – *Advanced Linux Sound Architecture*
AMI – *Advanced Metering Infrastructure*
ANATEL - *Agência Nacional de Telecomunicações*
ANSI – *American National Standards Institute*
API – *Application Programming Interface*
ARIB – *Association of Radio Industries and Businesses*
bps – bits por segundo
BPSK – *Binary Phase Shift Keying*
Cmake – *Cross-Platform Makefile Generator*
CMOS – *Complementary Metal Oxide Semiconductor*
CSS – *Chip Spread Spectrum*
DAC – *Digital to Analog Converter*
DSP – *Digital Signal Processor*
DVB – *Digital Video Broadcast*
EDGE – *Enhanced Data rates for GSM Evolution*
ETSI – *European Telecommunications Standards Institute*
FAN – *Field Area Network*
FCC – *Federal Communications Commission*
FDD – *Frequency Division Duplex*
FFT – *Fourier Fast Transform*
FI – *Frequência Intermediária*
FIR – *Finite Impulse Response*
FPGA - *Field Programmable Gate Array*
FPRF – *Field-programmable Radiofrequency*
FSK – *Frequency Shift Keying*
GPIO – *General Purpose Input/Output*
GPRS – *General Packet Radio Service*
GRC – *GNU Radio Companion*
GSMA – *Global System for Mobile Communications Association*
GUI – *Graphical User Interface*

IEEE – *Institute of Electrical and Electronics Engineers*
IED – *Intelligent Electronic Device*
IETF – *Internet Engineering Task Force*
IoT – *Internet of Things*
IP – *Internet Protocol*
ISM – *Industrial, Scientific and Medical*
LNA – *Low Noise Amplifier*
LPWA – *Low Power Wide Area*
LTE – *Long Term Evolution*
M2M – *Machine to Machine*
MAC – *Media Access Control*
MIMO – *Multiple Input Multiple Output*
PA – *Power Amplifier*
QoS – *Quality of Service*
RF – *Radiofrequência*
Rx – *Receptor*
RFID – *Radio-Frequency Identification*
SDR – *Software Defined Radio*
SISO – *Single Input Single Output*
SNO – *Sigfox Network Operators*
SNR – *Signal-to-Noise Ratio*
SUN – *Smart Utility Network*
SWIG – *Simplified Wrapper and Interface Generator*
TDD – *Time Division Duplex*
TIA – *Telecommunications Industry Association*
Tx – *Transmissor*
UART – *Universal Asynchronous Receiver/Transmitter*
UNB – *Ultra Narrowband*
UMTS – *Universal Mobile Telecommunications System*
USB – *Universal Serial Bus*
WAN – *Wide Area Network*
WBFM – *Wide Band Frequency Modulation*
WSN – *Wireless Sensor Network*

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivo Geral	14
1.2	Objetivos Específicos	14
1.3	Metodologia	14
1.4	Estrutura do Documento	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Regulamentação Vigente	16
2.1.1	Faixa de Frequência de Operação	16
2.1.2	Agregação de Canais	16
2.2	Redes Low Power Wide Area (LPWA)	18
2.3	Sigfox	20
2.4	LoRa	23
2.5	Wi-SUN Alliance Field Area Network	26
2.6	Rádio Definido por Software	29
2.6.1	Sistemas reconfiguráveis	30
2.6.2	Arquitetura Básica de um SDR	31
2.6.3	Vantagens e Desvantagens	33
2.7	Exemplo de arquitetura SDR: LimeSDR	34
2.7.1	LimeSDR-Mini	36
2.7.2	LMS7002M	38
2.8	GNU Radio	40
3	DESENVOLVIMENTO	42
3.1	Arquitetura Ideal do Protótipo	42
3.2	Instalação de Software	43
3.3	Fluxograma do <i>GNU Radio</i>	46
3.3.1	Blocos Hierárquicos	47
3.3.2	Blocos Fonte e Sumidouro	49
3.3.3	Outros Blocos	54
3.4	Receptor RTL-SDR	56
3.5	Construção do Experimento e Resultados	59
4	CONCLUSÕES E TRABALHOS FUTUROS	65
	REFERÊNCIAS	68

1 INTRODUÇÃO

Através dos últimos anos, a Internet das Coisas, ou *Internet of Things* (IoT) tem se tornado um termo onisciente. Essa tecnologia tem rapidamente avançado de uma simples expectativa para uma realidade já palpável. Até 2021, a estimativa é de que já existirão cerca de 3,3 bilhões de dispositivos *Machine to Machine* (M2M) conectados em rede. Dentre esses, 31% (cerca de 930 milhões) consistem em módulos LPWA e os demais em redes de telefonia celular tradicionais.¹

As razões para isso são simples: área de cobertura, tempo de vida de bateria e custo. Sabe-se que a área de cobertura é inversamente proporcional à taxa de transmissão de dados. Isso significa que, ao trabalhar-se com uma média/longa área de cobertura, devemos transmitir dados com uma taxa menor, o que se torna extremamente viável tendo em conta os tipos de aplicação em IoT. Essas variáveis são de total importância para que o tempo de vida da bateria do dispositivo seja alto. Além disso, o custo não é tão expensivo pelo fato de o sistema requerer baixa velocidade, com uma transmissão de dados razoavelmente confiável.

Para que se possa construir um sistema de comunicação de forma legítima, é necessário que se esteja totalmente de acordo com as normas regulamentadoras que certificam este sistema a operar de forma correta, dentro da legislação vigente. Para isso, toma-se como norte a Resolução nº 555 de 20 de dezembro de 2010 da Agência Nacional de Telecomunicações (ANATEL), que é a responsável por regulamentar a canalização e condições de uso de radiofrequências na faixa de 225 MHz a 270 MHz.²

Dentre todas as outras características do sistema, o coração da arquitetura consiste na tecnologia de Rádio Definido por Software, ou como indica o termo usado internacionalmente, *Software Defined Radio* (SDR). O SDR consiste basicamente em um sistema de comunicação de rádio, onde os componentes que eram tipicamente implementados em hardware (como por exemplo, filtros, amplificadores, moduladores/demoduladores, detectores etc.) são agora implementados em software utilizando um computador pessoal ou um sistema embarcado [1]. Sua função é a mesma de um transceptor e é o maior responsável pela convergência entre hardware e software na infraestrutura do sistema, o qual é controlado por um *Field Programmable Gate Array* (FPGA) interno. Existem diversos SDR's de diferentes marcas no

¹ https://www.cisco.com/c/dam/m/en_us/network-intelligence/service-provider/digital-transformation/knowledge-network-webinars/pdfs/0221-mobility-ckn.pdf.

² <https://www.anatel.gov.br/legislacao/resolucoes/2010/18-resolucao-555>.

mercado, de modelos mais simples que oferecem apenas funções básicas, até modelos mais robustos e avançados. Felizmente, os SDR mais básicos ainda assim podem oferecer grande custo-benefício e fornecer uma gama de possibilidades.

Para que a convergência possa ser completa, é necessário que tenhamos um software de desenvolvimento que permita a programação e configuração do sistema. Para isso, utilizaremos o *GNU Radio*, que consiste em um kit de ferramentas de desenvolvimento de software gratuito e de código aberto que fornece blocos de processamento de sinal para implementar softwares de rádios e sua interface gráfica, o *GNU Radio Companion* (GRC) que fornece um ambiente onde esses blocos podem ser facilmente visualizados e configurados através de fluxogramas. O *GNU Radio* pode ser utilizado juntamente com um hardware de radiofrequência (RF) externo de baixo custo prontamente disponível para criar rádios definidos por software ou sem hardware em um ambiente de simulação. É amplamente utilizado em ambientes de pesquisa, indústria, academia, entre outros, para apoiar tanto a pesquisa em comunicações sem fio quanto os sistemas de rádio do mundo real.³

1.1 Objetivo Geral

Propõe-se a construção de um protótipo de rádio via tecnologia SDR na faixa de 250 MHz, que atenda às requisições da Norma N° 555 da ANATEL quanto a canalização e frequência de operação.

1.2 Objetivos Específicos

- Interfaceamento entre as diferentes tecnologias de hardware e software;
- Construção de uma arquitetura ideal de rádio partindo de um modelo teórico e tendo em vista um produto final;
- Realizar um protótipo experimental de rádio via SDR com parâmetros ajustáveis;
- Estabelecer enlace na comunicação.

1.3 Metodologia

Fez-se necessária a utilização de pesquisas bibliográficas e testes de laboratório, reunindo materiais necessários para adquirir conhecimento e habilidades para execução correta das tarefas. As pesquisas bibliográficas basearam-se em publicações científicas e livros

³ <https://www.gnuradio.org/about/>.

relacionados a rádio definido por software, tecnologias de comunicação sem fio e softwares de desenvolvimento de rádio, assim como a implantação dessas tecnologias via SDR. Já os testes de laboratório foram realizados com a mínima infraestrutura necessária para a construção do protótipo.

1.4 Estrutura do Documento

Este relatório está estruturado da forma como segue. No Capítulo 2 é apresentada a fundamentação teórica. Uma breve descrição a respeito da legislação, um resumo de algumas tecnologias em alta no mercado atualmente e toda a documentação básica sobre os diversos componentes utilizados na constituição da arquitetura do sistema. O Capítulo 3 apresenta o desenvolvimento do trabalho que consiste na construção do protótipo, seus testes e resultados. Finalmente, no Capítulo 4 são apresentadas as conclusões e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Regulamentação Vigente

Antes de mais nada, é necessário que se tenha em mente a parte principal na construção de qualquer sistema de telecomunicações, a saber, a legislação que dá legalidade para a realização e execução do mesmo.

A resolução nº 555 de 20 de dezembro de 2010 da ANATEL é a que tomaremos como regra para algumas das premissas iniciais do sistema. Ela é a responsável por certificar o regulamento sobre a canalização e as devidas condições de uso de radiofrequências na faixa de 225 MHz a 270 MHz. Essa é uma norma relativamente recente e poucos têm utilizado ela para construção de sistemas e notado as vantagens que a mesma pode trazer. Ela converge totalmente com a nossa proposta de sistema, como veremos adiante.

2.1.1 Faixa de Frequência de Operação

A faixa de 250 MHz é uma vantagem para otimização do sistema, tendo em vista o fato de ser uma frequência licenciada. Isso significa, em termos gerais, vantagens técnicas no sentido de imunidade legal à interferências de outros rádios, a disponibilidade do enlace não será alterada com o passar do tempo, a utilização de técnica de duplexação por divisão de frequência (FDD), ou seja, transmissor e receptor em frequências diferentes simultaneamente (*full-duplex* nativo) como veremos a seguir, assim também como vantagens no ponto de vista econômico, contribuindo para uma visão sistêmica séria em relação ao projeto, aumentando o índice de qualidade e confiabilidade. Por ser uma frequência relativamente baixa, a penetração de sinal e o raio de cobertura são altos, não sendo necessários gastos expensivos nesse quesito.

2.1.2 Agregação de Canais

A norma prevê a opção de uso multicanal com largura de banda de 1,25 MHz ou monocanal com largura de banda de 25 kHz, além da possibilidade de agregação de canais. De acordo com o capítulo II, § 1º da norma:

"O uso da canalização [...] poderá ser autorizada de forma individual ou agregada, sendo a agregação máxima permitida de 5 (cinco) canais, de forma a

construir blocos inteiros de 2,5 MHz, 3,75 MHz e 6,25 MHz, desde que de forma a proporcionar uso mais eficiente da faixa de radiofrequências."⁴

É nítida a vantagem da utilização de canalização agregada, pois ela permite trabalhar com FDD e alta largura de banda de canal, possibilitando a escolha entre uma variedade de tecnologias de banda base que podem vir a se adequar nessas faixas. Na Tabela 1, temos o anexo que lista a numeração dos canais e a faixa de cada enlace, sendo o *uplink* o canal de ida (ou enlace reverso) e o canal de volta o *downlink* (ou enlace direto).

Tabela 1 - Sistemas Multicanais com canalização de 1,25 MHz⁵

Canal n°	Ida (MHz)	Volta (MHz)
1	226,25	248,75
2	227,50	250,00
3	228,75	251,25
4	230,00	252,50
5	231,25	253,75
6	232,50	255,00
7	235,00	262,50
8	236,25	263,75
9	237,50	265,00
10	238,75	266,25
11	240,00	267,50
12	241,25	268,75

⁴ <https://www.anatel.gov.br/legislacao/resolucoes/2010/18-resolucao-555>.

⁵ Ibid.

2.2 Redes Low Power Wide Area (LPWA)

As redes LPWA têm atraído bastante atenção no mercado de IoT devido a sua habilidade de oferecer uma conexão de baixa potência bastante acessível para dispositivos distribuídos sobre grandes áreas geográficas, muitas vezes remotos, a uma baixa taxa de transmissão de dados, alta latência, menor uso de bateria e menor custo, atendendo assim boa parte dos requisitos necessários na construção de um sistema M2M. Atualmente, as conexões feitas para altas distâncias na maioria dos casos operam utilizando redes celulares de diversas tecnologias, as mais comuns sendo a *General Packet Radio Service (GPRS)*, *Enhanced Data rates for GSM Evolution (EDGE)*, 3G e 4G.⁶ Não obstante, segundo a *Global System for Mobile Communications Association (GSMA)*, até 2022 as conexões LPWA irão ultrapassar as redes 2G, 3G e 4G, tornando-se a tecnologia líder em sistemas para Internet das Coisas com cerca de 1,4 bilhão de conexões.⁷

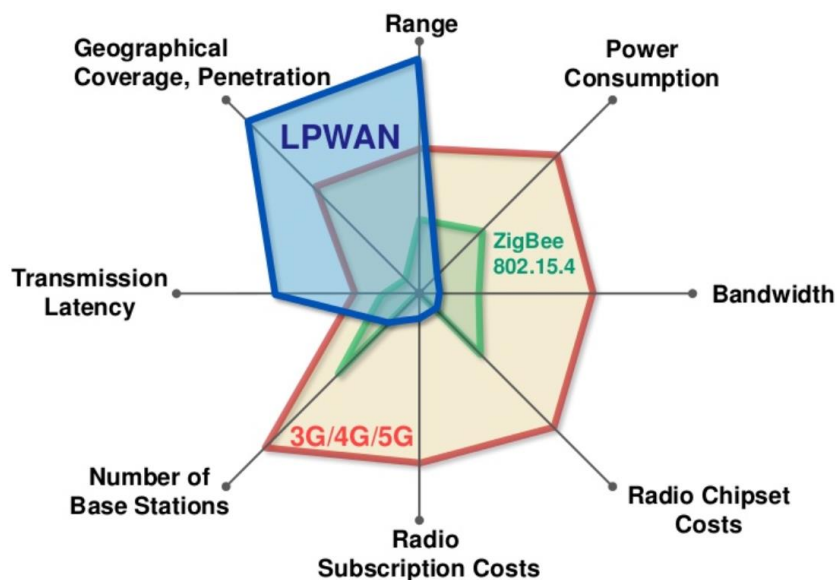
Redes LPWA possuem qualidades diferenciadas em comparação as tecnologias de rede sem fio mais tradicionais, tais como *ZigBee*, *Bluetooth*, *Wi-Fi*, *Radio-Frequency Identification (RFID)* e redes celulares como a *Global System for Mobile Communications (GSM)*, *Long Term Evolution (LTE)* etc. Essas tecnologias, apesar de suficientes para determinados projetos em IoT, se tornam limitadas quando se trata de sistemas M2M, onde o alcance elevado e baixo consumo são requisitos fundamentais. Por um lado, o alcance de tecnologias como *ZigBee*, *Bluetooth* e *RFID* é limitado apenas a algumas centenas de metros. Isso significa que os dispositivos conectados não podem ser movidos de forma arbitrária, o que acaba sendo um requisito importante em aplicações para cidades inteligentes, logística e *health-care* [2]. Já por outro lado, uma cobertura de área maior pode ser fornecida por redes celulares, uma das razões pelas quais vários sistemas M2M utilizam-se de tecnologia 2G e 3G. Entretanto, esse tipo de tecnologia não possui eficiência energética suficiente para oferecer anos de vida de bateria, além de oferecer determinada complexidade sistêmica, o que resulta em um aumento de custos desnecessário para dispositivos de baixa potência, justamente onde o foco é a redução de custos adicionais. Em contrapartida, a necessidade de aprimoramento em relação a quantidade de dispositivos que podem ser conectados simultaneamente nessas redes ainda é um ponto iminente e até mesmo um desafio para os provedores desta tecnologia, pois não é uma tarefa simples prover uma escalabilidade segura com recursos limitados e de baixo custo num ponto de vista sistêmico.

⁶ <http://www.teleco.com.br/lpwa.asp>.

⁷ <https://www.businesswire.com/news/home/20170201005644/pt/>.

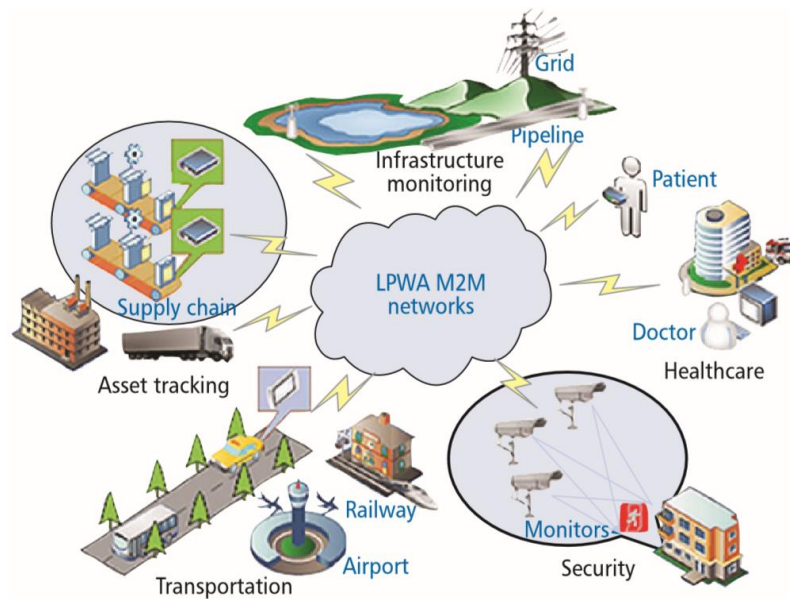
A interoperabilidade entre diferentes protocolos LPWA também é uma dificuldade bastante enfrentada, devido à falta de padronização. Além desses, destacam-se também as questões de segurança e autenticação, aprimoramento de mobilidade, *roaming* e possíveis causadores de interferência na rede, como por exemplo, outras redes sem fio coexistindo em um mesmo ambiente. Atualmente, existem diversas tecnologias LPWA competindo no mercado, cada uma empregando várias técnicas para obter longo alcance, operação em baixa potência e escalabilidade. Atrelado a isso, há também o esforço por parte de várias organizações desenvolvedoras de padrões como o *Third Generation Partnership Project (3GPP)*, *Institute of Electrical and Electronics Engineers (IEEE)*, entre outras, todas trabalhando em prol da padronização aberta para tecnologias LPWA. Dentre os tipos de tecnologia proprietárias mais conhecidas no mercado estão a *Sigfox* e a *LoRa* as quais nos restringiremos a abordar neste trabalho. Na Figura 1, tem-se um breve comparativo entre as tecnologias celulares tradicionais e novas (3G, 4G e 5G), além do *ZigBee* (ou protocolo IEEE 802.15.4) e a tecnologia LPWA, no que diz respeito às questões básicas como alcance geográfico, consumo de energia, largura de banda, entre outros. Já na Figura 2, são mostrados alguns tipos de aplicações onde podem ser utilizadas as redes LPWA.

Figura 1 - Comparativo entre redes LPWA e demais redes tradicionais⁸



⁸ <https://www.slideshare.net/PeterREgli/lpwan>.

Figura 2 - Cinco cenários típicos de aplicações LPWA M2M [2]



2.3 Sigfox

A rede *Sigfox* é uma tecnologia que utiliza a técnica de banda estreita (*narrowband* ou *ultranarrowband*). Existente desde 2009, a empresa proprietária (de mesmo nome) possui o controle dos servidores, abrindo espaço para que os *Sigfox Network Operators* (SNO), forma como são chamados os usuários da tecnologia, desenvolvam os seus dispositivos *end-points* e os implementem com as estações rádio base da tecnologia, as quais já vem equipadas com softwares de rádio cognitivo, que por sua vez, são conectadas aos servidores através da rede *Internet Protocol* (IP).⁹ Basicamente, os *end-points* são conectados às estações rádio base via modulação *Binary Phase Shift-Keying* (BPSK) [3] para transmissão de mensagens. Cada mensagem possui 100Hz de largura de banda nas regiões da Europa, a qual é regida segundo o *European Telecommunications Standards Institute* (ETSI) e do Japão, que segue o padrão da *Association of Radio Industries and Businesses* (ARIB) e de 600 Hz nos Estados Unidos, padronizada segundo a *Federal Communications Commission* (FCC), sendo transmitida a uma taxa de dados de 100 a 600 bits por segundo (bps), dependendo da região.¹⁰

Na Europa a banda do espectro utilizada é 868 MHz e 902 MHz nos Estados Unidos.¹¹ Essa banda utilizada para transmissão é caracterizada como SUB-GHz ISM¹², que é a faixa

⁹ <https://www.disk91.com/wp-content/uploads/2017/05/4967675830228422064.pdf>.

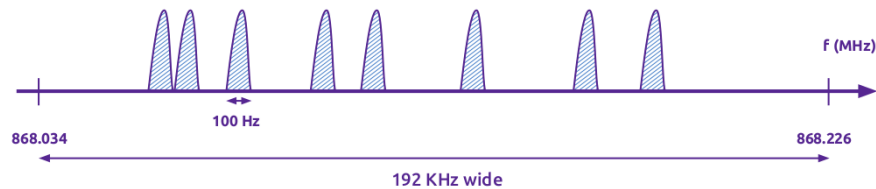
¹⁰ *Ibid.*, p. 8.

¹¹ *Ibid.*, p. 8.

¹² Industrial, Scientific and Medical. Bandas reservadas internacionalmente para uso industrial, médico e científico. https://www.teleco.com.br/tutoriais/tutorialredeswifi1/pagina_5.asp.

separada internacionalmente para uso industrial, científico e médico. A figura 3 mostra um exemplo de canal *Sigfox*.

Figura 3 - Largura de banda de um canal *Sigfox ultra-narrowband*¹³



A vantagem da utilização de frequência *ultra narrowband* (UNB) é uma alta seletividade no receptor, o que permite a eficácia de utilização de largura de banda, além de possuir uma baixa taxa de interferência de ruído, permitindo uma relação sinal-ruído (SNR) aceitável mesmo com um sinal relativamente fraco sendo recebido. Tudo isso resulta em um baixo consumo de potência, um alcance de transmissão maior e um *design* de antena mais simplificado [3].

A *Sigfox*, em seus primórdios, funcionava com suporte apenas para comunicação *uplink*, mas posteriormente evoluiu para uma tecnologia bidirecional, porém ainda com assimetria na comunicação dos links. Dessa forma, a confiabilidade da comunicação é melhorada através do uso de transmissões redundantes e de uma utilização variada do tempo e na frequência. Uma única mensagem de um dispositivo final (que pode ser um sensor, atuador ou qualquer outro tipo de dispositivo que não necessite de uma conexão de rede de alta velocidade) pode ser transmitida diversas vezes sobre canais com frequências diferentes. Como as estações rádio base têm a possibilidade de vasculhar cada canal para decodificar as mensagens, os dispositivos finais podem escolher um canal na frequência desejada para transmitir suas mensagens. Sendo assim, uma mensagem é transmitida diversas vezes para que se aumente a probabilidade de sucesso na recepção [3].

No que diz respeito ao tamanho das mensagens, é necessário que as restrições de autonomia da bateria e de custo dos dispositivos sejam atendidas. O tamanho das mensagens pode ir de 0 até 12 *bytes*. Alguns exemplos de tamanho de *payload* são listados na Figura 4. O

¹³ <https://www.disk91.com/wp-content/uploads/2017/05/4967675830228422064.pdf>.

tamanho, portanto, é suficiente para transferir dados de sensores, status de um determinado evento, coordenadas de GPS e afins, os quais são tarefas comuns em sistemas IoT.¹⁴

Figura 4 - Exemplo de tamanho de *payload Sigfox*¹⁵



A rede *Sigfox* consiste em dispositivos finais, estações rádio base ou *gateways* que são conectados à nuvem *Sigfox* e posteriormente aos servidores de aplicação. Os módulos são conectados ao *gateway* utilizando uma topologia em estrela. A partir daí, há enlaces ponto a ponto entre os *gateways* e a nuvem, a qual possui servidores usando diferentes tipos de protocolos.¹⁶

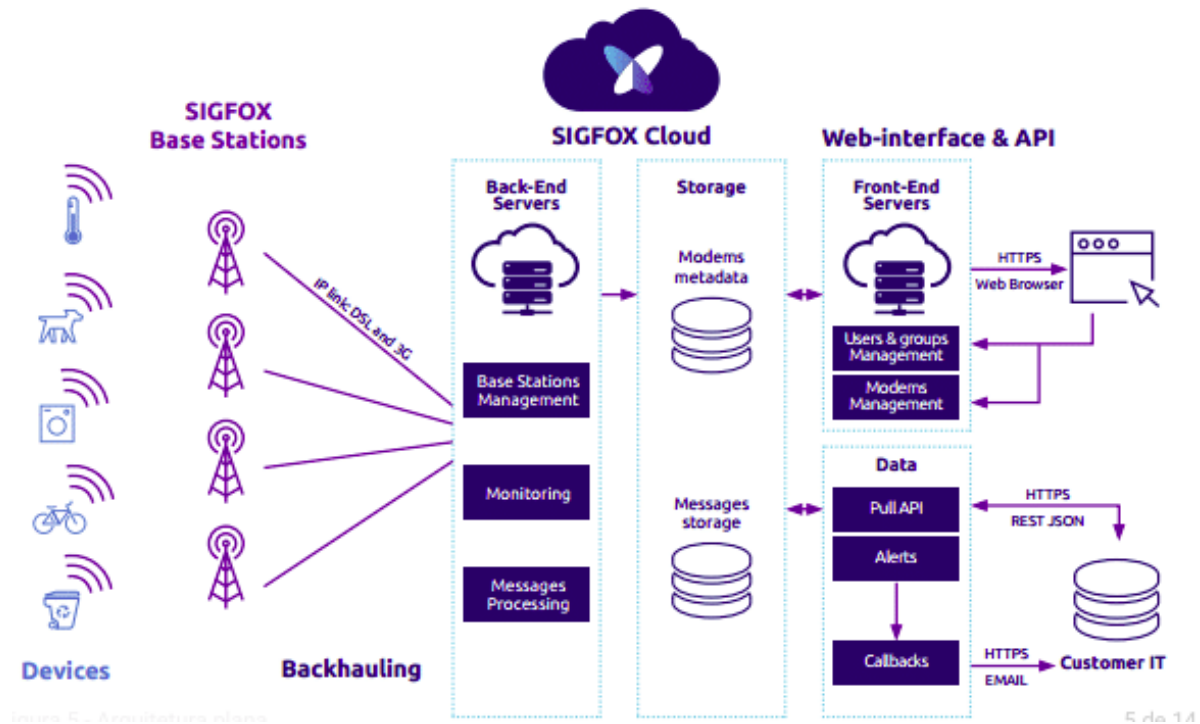
Essa arquitetura, a qual é mais amplamente esquematizada na Figura 5, diminui em grande parte os custos de operação e investimento, já que o rádio definido por software (SDR) da *Sigfox* ajuda a reduzir os custos relacionados a estações rádio base e aumenta significativamente sua flexibilidade, permitindo que apenas um hardware seja utilizado no mundo inteiro. A operadora *Sigfox* (a qual varia de região para região) entrega para o cliente todas as camadas de rede intermediárias já prontas para utilização, deixando a cargo do desenvolvedor apenas a escolha dos dispositivos que pretende usar e a criação de sua aplicação na nuvem, descartando assim toda a preocupação com a compra de estações rádio base/*gateways*, planejamento de RF, *backhaul*, entre outros, sendo a *Sigfox* já responsável por fornecer todos esses serviços.¹⁷

¹⁴ <https://www.disk91.com/wp-content/uploads/2017/05/4967675830228422064.pdf>.

¹⁵ *Ibid.*, p. 10.

¹⁶ <http://www.rfwireless-world.com/Tutorials/Sigfox-tutorial.html>.

¹⁷ *Ibid.*

Figura 5 - Arquitetura de rede Sigfox¹⁸

2.4 LoRa

A LoRa (termo para “*Long Range*”) é uma tecnologia desenvolvida e comercializada pela *Semtech Corporation*. Promovida pela *LoRa Alliance*, sua arquitetura possui uma comunicação bidirecional que consiste basicamente em duas camadas: a primeira é uma camada física utilizando a técnica de modulação por espalhamento espectral *Chirp Spread Spectrum* (CSS) [3], o qual espalha o sinal de entrada a uma largura de banda de canal bem maior do que a mínima necessária para sua transmissão, seguido de um código que independe da informação. Isso resulta em um sinal “mergulhado” em ruído, tornando-o mais difícil de ser detectado por terceiros, a menos que possuam a sequência de código necessária no receptor para decodificar o sinal recebido [4]. A segunda trata-se de um protocolo da camada *Media Access Control* (MAC) chamado LoRaWAN, que permite que vários dispositivos finais estabeleçam comunicação com o *gateway* utilizando a modulação da LoRa. Embora a camada física seja de tecnologia proprietária, a LoRaWAN possui padrão aberto, sendo desenvolvida pela própria *LoRa Alliance*.¹⁹

¹⁸ <https://www.disk91.com/wp-content/uploads/2017/05/4967675830228422064.pdf>.

¹⁹ <https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf>.

As faixas de frequência de operação utilizadas para LoRa, brevemente indicadas na Figura 6, onde se tem uma ideia geral das camadas que compõem a arquitetura, são as faixas de 867-869 MHz e 433 MHz na Europa, utilizando dez canais, oito deles possuindo uma taxa de dados que varia de 250 bits por segundo (bps) a 5,5 kbps, um canal de taxa única de 11 kbps e outro canal com modulação por chaveamento de frequência, ou *Frequency Shift Keying* (FSK) de 50 kbps.²⁰ A faixa utilizada nos Estados Unidos é a de 902-928 MHz, onde são definidos 64 canais *uplink* de 125 kHz que vão de 902,3 a 914,9 MHz em incrementos de 200 kHz. Além disso, há 8 canais adicionais de *uplink* com 500 kHz de largura de banda com incrementos de 1,6 MHz iniciando de 903 MHz a 914,9 MHz. Há também 8 canais de *downlink* com 500 kHz de largura de banda indo de 923,3 a 927,5 MHz. Em outras regiões o uso de banda varia de acordo com suas respectivas regulamentações e diferentes alocações do espectro, pois embora as especificações da LoRaWAN já estejam definidas, os comitês técnicos de outros países ainda estão neste processo juntamente com a LoRa Alliance.

Figura 6 - Especificações LoRa regionais²¹

Application				
LoRa® MAC				
MAC options				
Class A (Baseline)	Class B (Baseline)	Class C (Continuous)		
LoRa® Modulation				
Regional ISM band				
EU 868	EU 433	US 915	AS 430	—

A arquitetura de rede LoRa é formada por uma topologia em estrela estendida, sendo constituída por vários dispositivos, como mostra a Figura 7. Os dispositivos finais estabelecem comunicação com os *gateways* via LoRa com a LoRaWAN. Estes recebem os quadros do dispositivo e os enviam ao servidor de rede através de uma interface de *backhaul* (Ethernet ou 3G). Ou seja, os *gateways* funcionam apenas como dispositivos bidirecionais conversores de protocolo, sendo o servidor responsável pela decodificação dos pacotes que foram enviados

²⁰ Ibid., p. 13.

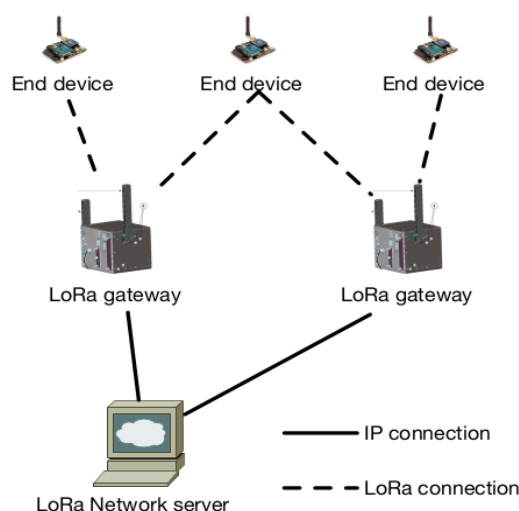
²¹ Ibid., p. 7.

pelos dispositivos assim como a geração e envio dos pacotes que devem ser retornados pelo caminho reverso [5].

Existem três tipos de dispositivos finais que podem ser utilizados em redes LoRa dependendo da forma de aplicação a ser realizada, os quais podem ser classificados da seguinte forma [5]:

- Dispositivos de Classe A: consistem basicamente em sensores. Estes podem realizar uma transmissão *uplink* baseada nas suas próprias necessidades, com um *jitter*²² pequeno. Permite comunicação bidirecional, através da qual cada transmissão *uplink* é seguida por duas curtas janelas de recepção *downlink*. Estes dispositivos são os que oferecem o menor consumo de energia, entretanto, possuem menor flexibilidade em transmissões *downlink*:
- Dispositivos de Classe B: é onde os atuadores entram em cena. Também possuem comunicação bidirecional e permitem que janelas de recepção extra sejam realizadas no schedule de tempo. Para que isso aconteça, é necessário que o *gateway* envie quadros *beacon*²³ sincronizados, para que o servidor de rede saiba que o dispositivo final esteja ouvindo;
- Dispositivos de Classe C: possuem janelas de recepção contínuas, ou seja, o dispositivo final estará sempre apto a receber dados de seu *gateway*. Como consequência, o nível de energia que consomem é máximo.

Figura 7 - Arquitetura da Rede LoRa [5]



²² Variação no tempo de entrega das informações devido à atrasos na rede (latência). https://www.teleco.com.br/tutoriais/tutorialqosqoe/pagina_2.asp.

²³ Quadro enviado pelo ponto de acesso que tem como propósito informar as características de conexão. <https://web.archive.org/web/20180805075033/http://www.wi-fiplanet.com/tutorials/print.php/1492071>.

Deve-se destacar que a LoRaWAN não permite a comunicação de dispositivo para dispositivo. É possível somente a comunicação do dispositivo para o servidor e vice-versa.

2.5 Wi-SUN Alliance Field Area Network

Ao decorrer dos últimos anos, a demanda para o sistema de comunicação *Wireless Smart Utility Network* (Wi-SUN) tem crescido de forma vertiginosa. Embora não sendo originalmente uma rede LPWA, definitivamente vai de encontro aos critérios necessários para caracterizá-la como tal, possuindo os mesmos casos de uso e podendo até mesmo ser considerada como forte alternativa a tecnologias como *Sigfox*, LoRa, entre outras as quais são as mais primordiais quando pensamos em LPWA.

Cada vez mais as concessionárias de energia elétrica estão implantando dispositivos eletrônicos inteligentes, mais conhecido como *Intelligent Electronic Device* (IED) e outros aparelhos inteligentes em subestações e ao longo de alimentadores de distribuição, bem como equipando os trabalhadores de campo com computadores portáteis e tablets. Essas tecnologias costumam ser chamadas de redes de energia elétrica inteligentes, ou *Smart Grids*. Esses dispositivos permitem aplicações valiosas, de importância crítica em sistemas desse tipo, atuando com infraestrutura de medição avançada, a chamada *Advanced Metering Infrastructure* (AMI). Além da AMI, outras funções são executadas, como a automação de subestação, automação da rede de distribuição de energia elétrica e gerenciamento de interrupções.

A comunicação bidirecional em *Smart Grids* conecta as pessoas e os dispositivos localizados no campo com o software nas subestações e no centro de operações da concessionária, permitindo que os aplicativos ofereçam grandes melhorias em eficiência, segurança e confiabilidade. Estender a comunicação da central de controle da subestação até o pátio da subestação e ao longo dos alimentadores de distribuição é melhor realizado usando redes *Field Area Network* (FAN). As redes FAN são redes IPv6 em topologia de malha que possuem suporte a nível de segurança empresarial.²⁴ Em uma rede em malha, os dispositivos se comunicam com dispositivos vizinhos em sua rede, cada um destes servindo como uma rota para a estação rádio base. Isso permite múltiplas conexões redundantes, possibilitando assim a melhoria de escalabilidade, confiabilidade e performance à medida que novos caminhos de comunicação são estabelecidos na rede. Seu propósito é oferecer alta confiabilidade através de

²⁴ <https://www.cisco.com/c/en/us/solutions/industries/energy/external-utilities-smart-grid/field-area-network.html>.

alta redundância (devido ao nível crítico de operação), boa escalabilidade (tendo em vista a flexibilidade da configuração em malha), o que a permite ocupar grandes áreas geográficas e boa resistência a interferências.²⁵

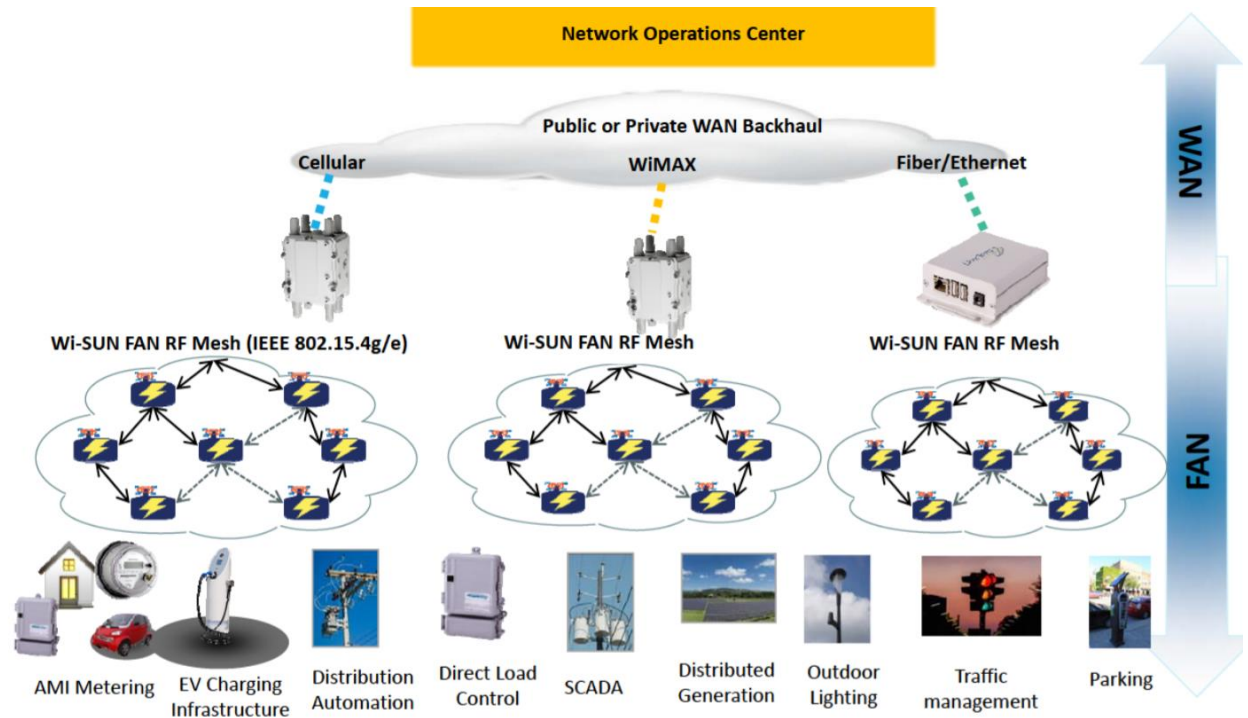
A Wi-SUN é uma tecnologia desenvolvida para *Utilities*, *Smart Cities* e IoT baseada em vários padrões como IEEE, *Internet Engineering Task Force* (IETF), *American National Standards Institute* (ANSI) e a *Telecommunications Industry Association* (TIA),²⁶ contidos na sua pilha de protocolos. É apoiada pela Wi-SUN Alliance, uma aliança que conta com parcerias internacionais que promovem as especificações Wi-SUN, tendo como objetivos primários desenvolver testes para certificar equipamentos de IoT baseados em IEEE 802.15.4g que sejam interoperáveis com outros dispositivos certificados²⁷ e a busca por concordância em padrões IPv6 para sistemas de medição elétrica eficiente (*Smart Grids*) no mundo. Entre as parcerias mais importantes, estão a Cisco e a Itron.

Usualmente utilizados em sistemas de energia elétrica, gás e água, os medidores equipados com os rádios com tecnologia *Smart Utility Network* (SUN) podem retransmitir automaticamente de forma eficaz os dados de medição para a coleção destes pela estação rádio base através de uma operação multissalto. Posteriormente, esses dados são transmitidos para a nuvem ou servidor por meio de redes de longa distância (WAN) tais como sistemas de comunicação móvel ou redes cabeadas e então os dados coletados são dissecados e os resultados são transmitidos novamente ao sistema, que pode ser, por exemplo, um medidor de energia elétrica inteligente, que possui como objetivo controlar o consumo de energia de uma determinada residência. A Figura 8 mostra uma visão vertical da arquitetura básica do Wi-SUN e algumas variações no que diz respeito às tecnologias utilizadas para sua construção.

²⁵ <https://tools.ietf.org/id/draft-heile-lpwan-wisun-overview-00.html#draft-wisun-use-cases>.

²⁶ Ibid.

²⁷ Ibid.

Figura 8 - Casos de uso da tecnologia Wi-SUN²⁸

Dentre as diversas características da tecnologia Wi-SUN, podemos citar algumas que possuem alta relevância para aplicações com redes LPWA, tais como²⁹:

- Cobertura: o alcance de uma rede Wi-SUN está em torno de 2 a 3 km em linha de visada direta. É possível criar uma extensão com o uso de múltiplos saltos;
- Alta largura de banda e baixa latência: o Wi-SUN pode suportar larguras de banda relativamente altas, as quais transitam em torno de 300 kbps, além de permitir a atualização e melhoria dos dispositivos para que novas aplicações possam ser executadas e o tempo de vida seja estendido;
- Baixo consumo de energia: dispositivos FAN gastam cerca de 2 μ A em período de descanso e 8 mA quando em funcionamento. Tais dispositivos podem, portanto, manter um longo período de vida mesmo que frequentemente usados;
- Escalabilidade: Cerca de 91 milhões de dispositivos Wi-SUN estão espalhados pelo mundo³⁰, sendo implantados em ambientes urbanos, suburbanos e rurais, com algumas dessas aplicações contendo milhões de dispositivos.

²⁸ <https://www.tele.soumu.go.jp/resource/j/equ/mra/pdf/29/e/12.pdf>.

²⁹ Ibid.

³⁰ <https://www.wi-sun.org/>.

2.6 Rádio Definido por Software

Em meio as constantes evoluções tecnológicas, os sistemas de telecomunicações galgaram passos significativos, a ponto de requerer que os profissionais da área desenvolvessem diversas habilidades e ferramentas para solucionar os emergentes problemas técnicos atuais. Este não era o caso há décadas atrás, onde em uma equipe de desenvolvimento cada indivíduo possuía habilidades diferentes restritas apenas a uma determinada área e complementando uns aos outros, para construir um sistema. Era necessário que vários engenheiros estivessem encarregados de tarefas individuais (tais como desenvolver um algoritmo para o transceptor, analisar a performance do sistema ponto a ponto, implementar um *front-end* de RF etc.) de tal forma, que cada engenheiro possuía grande gama de conhecimentos sobre sua área, porém não era obrigado a tomar conhecimento sobre as outras áreas concorrentemente. Logicamente, com o avanço do tempo, todas essas áreas começaram a se conectar umas com as outras, sendo atualmente, quase que puramente integradas. Na conjuntura atual, se espera que um engenheiro saiba como prover soluções em diferentes níveis e áreas em um mesmo projeto, por exemplo, desenvolvendo um algoritmo de um sistema de comunicação e também implementando-o de forma satisfatória em um FPGA ou um sistema embarcado qualquer.

Além disso, também temos presenciado a convergência constante entre as redes IP e tecnologias móveis de terceira e quarta geração da rede celular, fornecendo acesso à alta largura de banda e alta taxa de dados. Neste ambiente de composição, onde ainda existem diversos esquemas padronizados de sistemas rádio legados, uma possível solução a médio prazo seria desenvolver redes reconfiguráveis e sistemas que permitam uma escolha dinâmica entre esquemas de rádio para fornecer uma solução flexível, que atenda às necessidades do usuário em termos de qualidade de serviço e de custo [1].

O conceito de rádio definido por software permite que a implementação tradicional utilizada nos hardwares em sistemas de comunicação seja substituída por uma implementação mais flexível, a qual utiliza dispositivos programáveis que podem ser controlados via software, no caso, um computador pessoal ou um processador embarcado [6]. Por ser uma tecnologia relativamente recente, diferentes conceitos são trazidos à tona. De acordo com o *Wireless Innovation Forum*, o SDR é definido como “radio no qual algumas ou todas as funções da camada física são definidas por software.”³¹ Basicamente, um SDR é um transceptor onde os parâmetros de operação podem ser configurados via software e em tempo de execução, podendo ser reconfigurado com facilidade de acordo com o tipo de aplicação que o usuário deseja [6].

³¹ https://www.wirelessinnovation.org/Introduction_to_SDR

Este conceito mais tradicional implica em uma noção de reconfiguração de terminais mais simplificada e flexível, substituindo rádios que são completamente implementados em hardware por rádios que são configuráveis ou programáveis quase que completamente em software. Isso inclui, por exemplo, reconfiguração de antena, e mudança de banda base.

A combinação do processamento de sinais a engenharia de RF sempre foi um fator primário em sistemas de comunicação. Nos sistemas modernos, o processamento de sinais progrediu de tal forma, que a maioria das funcionalidades de banda base está sendo implementada em software. A flexibilidade no hardware de RF a ser reconfigurado é tal que apenas um rádio de *front-end* é necessário em alguns sistemas de RF. Esta combinação de *front-end* de RF flexível e processamento de sinal conduziu ao nascimento do SDR [7].

2.6.1 Sistemas reconfiguráveis

O conceito de sistema reconfigurável está diretamente atrelado à essência de um SDR. A reconfiguração é sinônimo de múltiplas funcionalidades e implica no fornecimento de suporte a diferentes tipos de propriedades do sistema. Em um sistema de comunicação móvel reconfigurável, o objetivo é prover uma plataforma comum para múltiplas interfaces aéreas, múltiplos protocolos, e múltiplas aplicações, aumentando assim a capacidade e versatilidade na rede realizando as modificações via software. Devido ao espaço de mercado que as *Application Programming Interfaces* (API) de padrão aberto têm conquistado, softwares de diversos fabricantes podem ser executados em tecnologias de hardware proprietárias.

De acordo com Arslan [8], um sistema reconfigurável possui as seguintes características:

- **Multibanda:** ou seja, o sistema permite suporte a mais de uma banda de frequência de operação específica, atendendo diversos tipos de padrões. O *front-end* de RF precisa, portanto, ser capaz de cumprir com as especificações necessárias e ser ajustável sobre uma variedade de bandas.
- **Multifuncional:** As plataformas reconfiguráveis devem ser independentes da aplicação, provendo um ambiente multitarefa para os processos de interface aérea, protocolos de camadas superiores e aplicações.

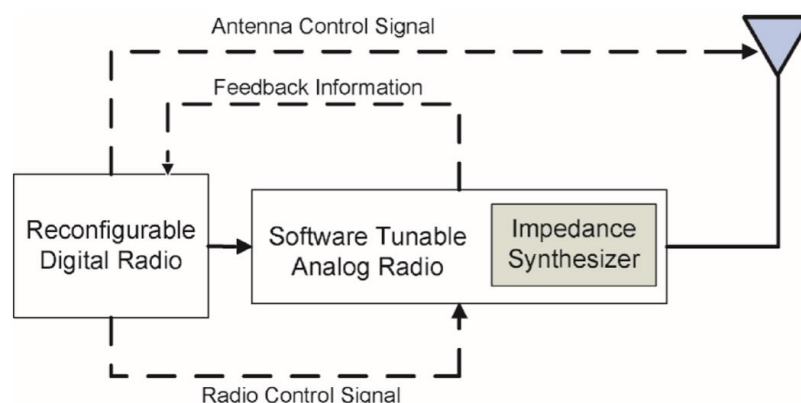
2.6.2 Arquitetura Básica de um SDR

Primeiramente, devemos ter em mente uma arquitetura ideal para um rádio definido por software, que consiste em três componentes fundamentais, os quais podem ser melhor visualizados na Figura 9. São estes [8]:

- Rádio digital reconfigurável: responsável pela geração de diferentes formas de onda, algoritmos de otimização e controle das outras unidades;
- Rádio analógico ajustável via software: limitado aos componentes que não são de formato digital, tais como filtros de RF, amplificadores de potência (PA), amplificadores de baixo ruído (LNA) e conversores de dados. É composto por um sintetizador de impedância, que ajuda na sua otimização de performance e atua também no ajuste de antena;
- Sistema de antenas inteligente: reagem de forma dinâmica às mudanças do ambiente, tendo como objetivo fornecer um sinal de maior qualidade e gerar um melhor aproveitamento das faixas de frequência na comunicação.

Devido a diversas questões tais quais o custo, tamanho e performance, sendo limitações fundamentais dentro das condições de contorno de um SDR, uma arquitetura ideal seria atualmente inviável. O mercado tecnológico oferece diversas opções de SDR's e deste modo, espera-se que as plataformas disponíveis atualmente evoluam de forma significativa no futuro para que as presentes limitações sejam removidas.

Figura 9 - Visão sistêmica de um SDR ideal [8]

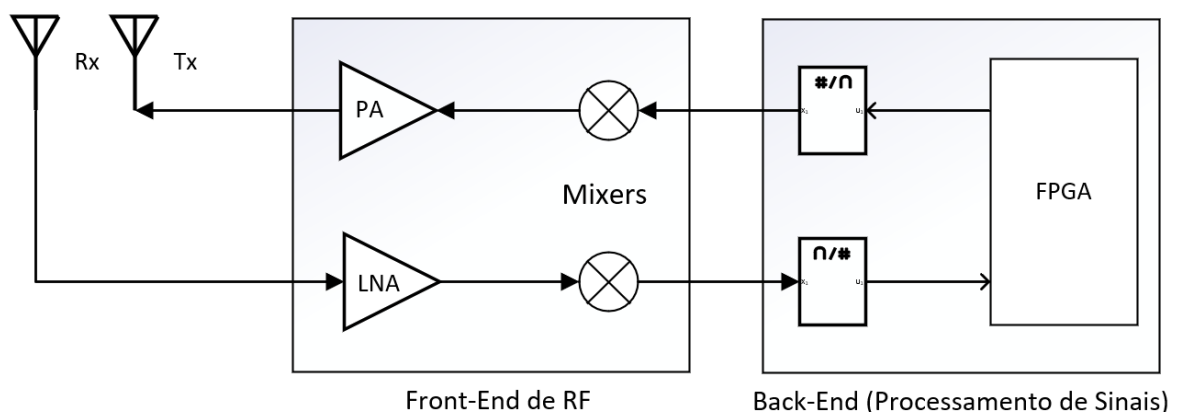


Já uma arquitetura real/prática de um SDR pode ser dividida em duas partes predominantes [9]:

- *Front-end* de rádio: responsável pelos módulos de RF, onde o sinal analógico em frequência intermediária (FI) é transformado em sinal de alta frequência e então preparado para ser transmitido;
- *Back-end* de rádio: responsável pelos módulos de processamento digitais, onde está presente o sinal em FI e que será transformado em sinal banda-base para que haja o processamento.

Na parte de *front-end* de RF, existe um subsistema de antenas ao qual se pode incluir variados tipos de tecnologia, devido à característica já inerente do sistema em dar suporte a variados tipos de frequências, sendo na sua maioria, antenas inteligentes. Este também é responsável pelo controle de ganho, filtragem, amplificação de sinal e mudança de frequência de operação. Existe também um outro subsistema que é responsável por converter uma ou mais bandas de frequências de interesse em sinal de FI, podendo incorporar múltiplos *front-ends* de RF operando em paralelo para suportar os requisitos determinados. Em seguida há um subsistema seletor/combinador de canais que permite o ajuste de frequência de banda passante e execução do processo de amostragem. Esta funcionalidade é geralmente realizada por um FPGA e por conversores analógico-digitais (ADC) e digital-analógicos (DAC). Por fim, tem-se a fase de processamento, onde há o uso de processadores digitais de sinal (DSP) que fornecem a codificação e decodificação do sinal em banda base [9]. A Figura 10 fornece um entendimento visual em relação a essa arquitetura.

Figura 10 - Arquitetura básica de um SDR [10]



A conversão de frequências entre os processos é um dos tópicos mais fundamentais na arquitetura do SDR, podendo ser classificada em duas formas: a arquitetura de conversão direta, que é a mais utilizada, e a arquitetura de conversão múltipla [9]. A primeira permite identificar

somente um estágio de conversão. A frequência intermediária é transladada diretamente para o oscilador/misturador, realizando assim a conversão. Já a segunda possui um sistema de translação do sinal em múltiplas etapas, onde é possível analisar o sinal em cada uma delas. Esta possui um custo de complexidade maior devido à quantidade aumentada de filtros e osciladores a serem implementados [9].

2.6.3 Vantagens e Desvantagens

Ainda que o uso de um SDR ofereça alguns recursos mais apurados no estado da arte em sistemas de comunicação sem fio e métodos, os quais em comparação aos já tradicionais esquemas de desenvolvimento de placas de RF agregam altíssimo valor, a tecnologia ainda está evoluindo para uma localização mais permanente em termos de consolidação. Isso traz como consequência algumas desvantagens, que dependendo do projeto, podem ser cruciais. Algumas vantagens e desvantagens de um SDR são listadas a seguir. [9]

- Vantagem: dispositivos com tendência evolutiva. Ou seja, que permitem a possibilidade do equipamento receber novas atualizações de funcionalidades e correção de erros mesmo após sua comercialização, tendo em vista que a implementação destes dispositivos é em software, tornando mais fácil a manutenção e permitindo flexibilidade;
- Vantagem: reuso de hardware. Os sistemas de comunicação sem fio atuais operam de forma diferente dependendo da região onde estão situados. Desta forma, diversas empresas necessitam criar diversas versões de um mesmo equipamento apenas para atender a legislação local de cada país. Logicamente isso causa um aumento no custo de produção do sistema, uma vez que vários componentes são fabricados. Entretanto, com a utilização de hardware programável, as especificações de cada região seriam aplicáveis em software, tendo como consequência uma única plataforma de hardware que seria modificada conforme a região;
- Vantagem: utilização de níveis de abstração mais altos. Aqui entram em cena as API's de hardware programável. As API's fornecem um nível de abstração da aplicação mais intuitivo, através de linguagens de programação. Um nível de abstração maior permite que compiladores, sistemas operacionais e *middlewares*

possam executar de forma melhor o código e de efetuar métodos de otimização com maior rapidez e confiabilidade;

- Desvantagem: desempenho inferior. Sistemas implementados apenas com circuitos integrados fornecem desempenho superior a sistemas baseados em software, pelo simples fato de serem fabricados para cumprir tarefas específicas. Sistemas que são implementados em software possuem alto nível de abstração e fornecem um hardware mais genérico (como um FPGA). Devido a possibilidade de compartilhar tarefas em um dispositivo programável, os atrasos são recorrentes e a capacidade de processamento é diminuída devido ao custo adicional, o que implica no desempenho final;
- Desvantagem: problemas de interoperabilidade. Devido à grande quantidade de produtos e implementações fabricadas e proliferadas no mercado e também as diversas customizações de rede, a dificuldade em produzir estes dispositivos com redes interoperáveis também aumenta, resultando em casos onde os pontos de acesso trabalham com pacotes desconhecidos;
- Desvantagem: No tocante à segurança, ainda existem diversos transceptores cujas funcionalidades são implementadas em hardware, o que torna impossível sua modificação sem que haja o acesso físico ao mesmo. Isso torna o sistema mais vulnerável em níveis abaixo dos protocolos, o que gera como resultado uma alta propensão a invasões remotas.

2.7 Exemplo de arquitetura SDR: LimeSDR

A LimeSDR é uma família de placas de desenvolvimento de rádio definido por software de baixo custo desenvolvido pela *Myriad RF*³², uma empresa que trabalha na criação de hardware e software em padrão de código aberto para sistemas de comunicação sem fio. Possui a capacidade de transmitir e receber diversas tecnologias de banda base conhecidas, tais como a *Universal Mobile Telecommunications System* (UMTS), LTE, GSM, LoRa, *Bluetooth*, *ZigBee*, entre outras. Enquanto grande parte dos fabricantes de SDR permanecem limitados ao domínio dos segmentos de RF e protocolos, a LimeSDR pode ser útil a qualquer um que possuir a noção do conceito de *app-store*, sendo o primeiro SDR a integrar com o *Snappy Ubuntu Core*³³, versão mais compacta do sistema operacional Ubuntu criada para sistemas IoT. Isso

³² <https://myriadrf.org/>

³³ <https://limemicro.com.>

significa que qualquer usuário da plataforma LimeSDR pode baixar aplicativos novos feitos por desenvolvedores espalhados pelo mundo, assim como permitir o compartilhamento por parte dos desenvolvedores.

A *Myriad RF* trabalha em conjunto com a *Lime Microsystems*, empresa responsável pela fabricação dos transceptores de RF das placas, conhecidos como *Field-programmable Radiofrequency* (FPRF).³⁴ A LimeSDR conta também com uma coleção de softwares de suporte às plataformas de hardware, com drivers para o transceptor e outras ferramentas, chamada de *LimeSuite*.³⁵ Ela oferece suporte também a diversas outras placas, contendo uma variedade de interfaces gráficas de usuário.

As placas mais difundidas no mercado, sendo uma delas a utilizada para a confecção do experimento, são mostradas nas Figuras 11 e 12. A família LimeSDR é composta por cinco tipos de placas.³⁶:

- LimeSDR: placa SDR com uma interface *Universal Serial Bus* (USB) 3;
- LimeSDR-Mini: placa SDR em formato reduzido, também possuindo uma interface USB3;
- LimeSDR-PCIe: placa SDR com quatro interfaces PCI Express 1.0;
- LimeSDR-QPCIe: placa SDR com quatro interfaces PCI Express 1.0 e dois transceptores LMS7002M;
- LimeSDR GPIO *Board*: placa de expansão com entrada e saída bidirecional ajustáveis para FPGA.

Figura 11 - LimeSDR USB com plug tipo A³⁷



³⁴ Ibid.

³⁵ <https://wiki.myriardf.org/LimeSDR>.

³⁶ Ibid.

³⁷ <https://wiki.myriardf.org/LimeSDR-USB>.

Figura 12 - LimeSDR-Mini v1.2³⁸

Neste trabalho nos restringiremos apenas à LimeSDR-Mini, a qual será a placa utilizada na realização do experimento.

2.7.1 LimeSDR-Mini

A placa de desenvolvimento LimeSDR-Mini é uma versão menor e de baixo custo da LimeSDR original. No entanto, ainda conta com diversas funcionalidades interessantes, pois as duas placas possuem o mesmo transceptor de RF. Provê uma plataforma de hardware para prototipagem de alta performance com lógica digital intensa tendo o Intel MAX 10 como seu FPGA. As faixas de frequência em que a placa permite operação vão de 10 MHz até 3,5 GHz e a largura de banda pode ser programável de 100 kHz a 108 MHz.³⁹ Os componentes principais da placa são relacionados na Tabela 2.

Tabela 2 - Principais componentes do LimeSDR-Mini⁴⁰

Componente	Descrição
Transceptor de RF	Lime Microsystems LMS7002M
FPGA	Intel MAX 10 (10M16SAU169C8G 169-UBGA)
Controlador USB 3.0	FTDI FT601

³⁸ <https://wiki.myriadrf.org/LimeSDR-Mini>.

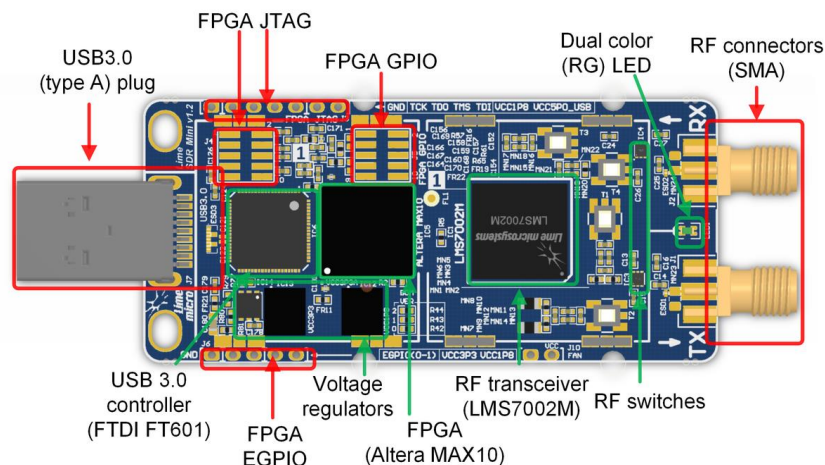
³⁹ <https://wiki.myriadrf.org/LimeSDR-Mini>.

⁴⁰ Ibid.

Oscilador	Rakon RTX5032A @40.00MHz
Conector de RF	SMA (2x)
Indicadores de status	LED's programáveis
Dimensões	69mm x 31.4mm

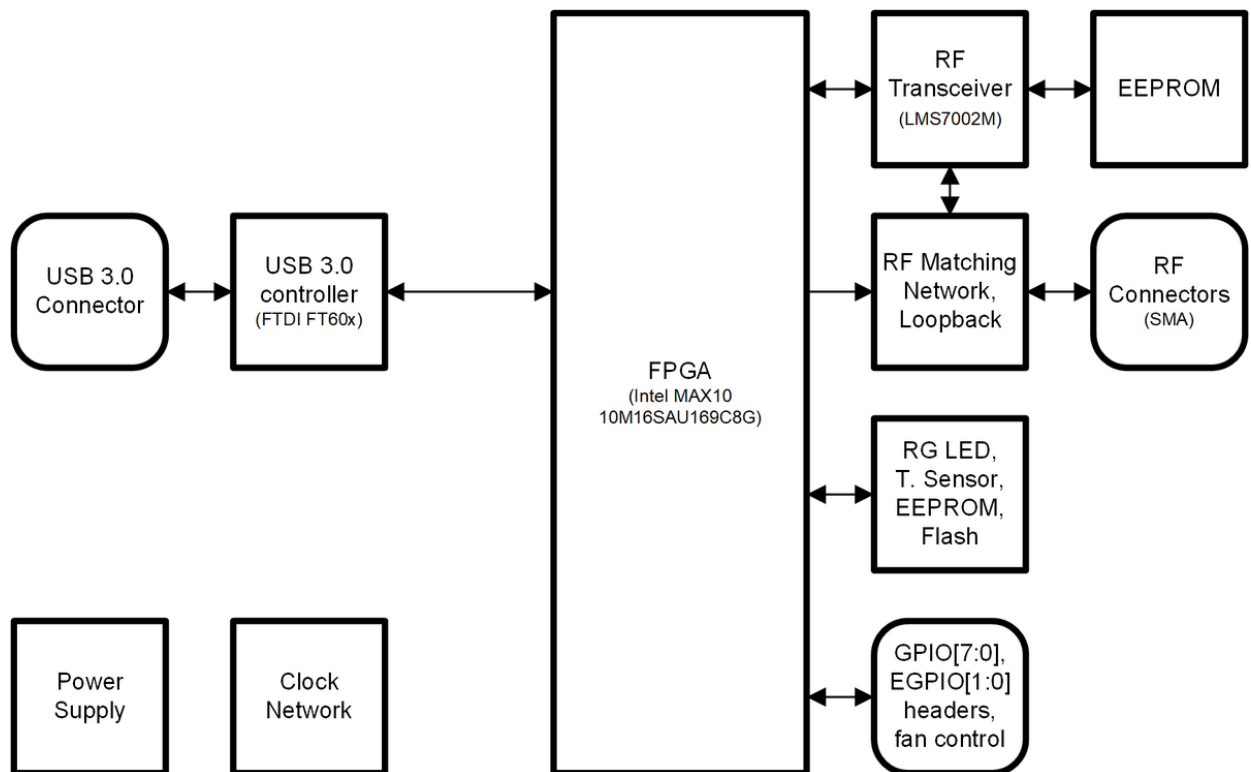
Pode-se dizer que o coração da placa reside em seu FPGA Intel MAX 10, que possui como principal função transferir os dados digitais para o computador através da interface USB 3.0. Mesmo sendo um dispositivo lógico programável de baixo custo, ocupa um papel importantíssimo no processamento, trazendo também eficiência energética. Permite flexibilidade e integração de recursos, suporte a processamento digital de sinais, DAC's e ADC's, configuração dual de memória flash e um ciclo de vida de bateria de aproximadamente 20 anos.⁴¹ As figuras 13 e 14 dão uma descrição em termos de estrutura física a respeito da arquitetura integrada da placa.

Figura 13 - Parte superior da placa com os componentes principais⁴²



⁴¹ https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/max-10/m10_overview.pdf.

⁴² https://wiki.myriadrf.org/LimeSDR-Mini_v1.2_hardware_description.

Figura 14 - Diagrama de blocos da arquitetura do LimeSDR-Mini⁴³

2.7.2 LMS7002M

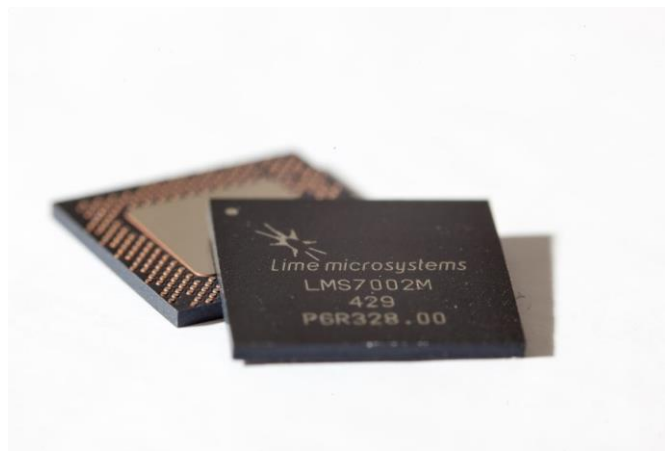
O LMS7002M é um circuito integrado de transceptor de RF multibanda, multipadrão totalmente integrado e altamente programável, possuindo funcionalidades estendidas para cobrir um grande leque de frequências e aplicações. Possui uma combinação de amplificadores de baixo ruído, amplificadores de potência, misturadores de transmissão e recepção, filtros, sintetizadores, ganho de controle de receptor, controle de potência de transmissor, ADC's e DAC's, de tal forma que seu *design* permite que poucos componentes exteriores sejam utilizados, devido seu poder de integração.⁴⁴ O dispositivo detém o estado da arte em *design* de transceptores com tecnologia *Complementary Metal Oxide Semiconductor* (CMOS) de baixo custo, com o objetivo de oferecer o mínimo de consumo de energia possível. É integrada a este uma arquitetura de transceptor dupla, para permitir suporte a operação *Multiple Input Multiple Output* (MIMO) 2x2 com funções adicionais de processamento digital de sinais, provendo assim dois percursos de recepção/transmissão, permitindo uma operação dupla em um único

⁴³ Ibid.

⁴⁴ <https://limemicro.com/app/uploads/2017/07/LMS7002M-Data-Sheet-v3.1r00.pdf>.

chip. A vantagem desta integração permite que o processamento digital de sinais seja combinado aos filtros analógicos para que a performance seja aprimorada, evitando a necessidade ou diminuindo a quantidade de filtros externos para tal.⁴⁵ O chip, mostrado na Figura 15, conta com malhas de captura de fase que geram a frequência a ser modulada ou demodulada no transmissor e no receptor, respectivamente. Há também a opção de *loopback* de RF, a qual permite que o sinal a ser transmitido seja realimentado no próprio sistema para calibração e possíveis testes.

Figura 15 - Circuito integrado LMS7002M⁴⁶



O LMS7002M oferece um leque de espectro de frequências que podem ir de 100 kHz a 3.8 GHz, com largura de banda de canal operante até 120 MHz e permite a configuração de diversas tecnologias banda-base e padrões de segunda, terceira e quarta gerações, podendo operar tanto com duplexação por divisão no tempo (TDD), quanto por divisão na frequência em modo *full-duplex*. Dentre alguns sistemas onde o LMS7002M pode ser aplicado estão⁴⁷:

- Comunicação sem fio em banda larga;
- GSM, CDMA2000, TD-SCDMA, WCDMA/HSPA, LTE;
- Quaisquer rádios com padrão IEEE (xxx.xxx);
- Rádio Cognitivo;
- Veículo aéreo não tripulado (drones).

⁴⁵ Ibid.

⁴⁶ <https://limemicro.com/technology/lms7002m/>.

⁴⁷ Ibid.

2.8 GNU Radio

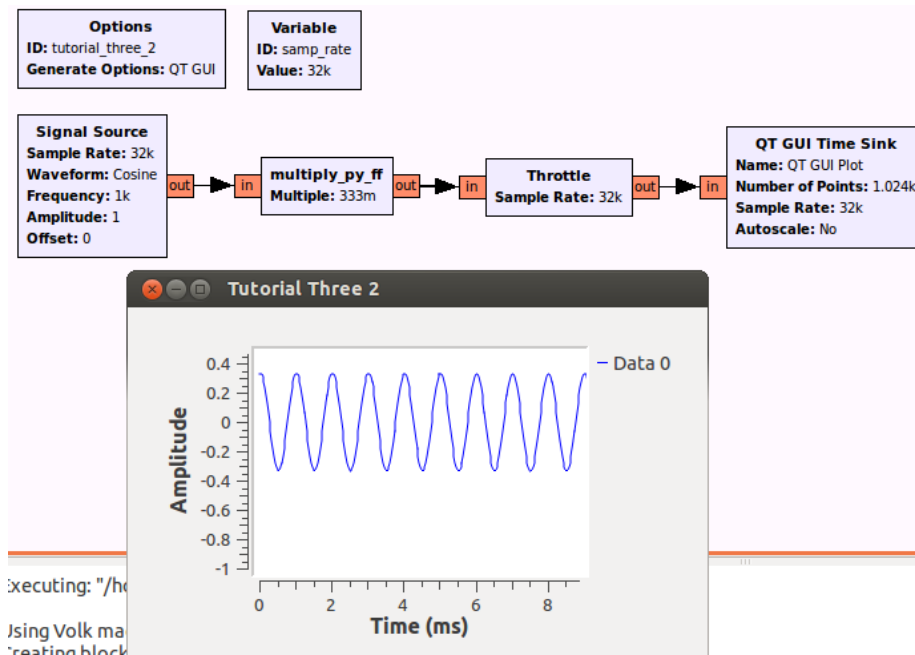
Para utilizar um SDR para criação de projetos, é necessário também definir o software para que o computador ou sistema embarcado possa controlar o hardware. O *GNU Radio*, considerado o *framework* para SDR mais ativo na comunidade *open source*⁴⁸, pode ser definido como um kit de ferramentas de desenvolvimento de software gratuito e de código aberto, que fornece blocos de processamento de sinal para implementar rádios definidos por software.⁴⁹ A plataforma *GNU Radio* aliada à um hardware de RF externo de baixo custo se torna extremamente útil no uso de rádios definidos por software, mas pode também ser utilizada sem hardware a partir de funções de *loopback*. Atualmente, o *GNU Radio* é amplamente utilizado em ambientes de pesquisa, indústria, academia, governo e hobby no que concerne o apoio a pesquisas de comunicações sem fio e sistemas de rádio do mundo real.

O software basicamente provê uma série de bibliotecas de processamento de sinais primitivas e permite interconectá-las através de uma ferramenta gráfica chamada *GNU Radio Companion* (GRC), mostrada na Figura 16, onde há a possibilidade de criar e conectar esquemáticos de diagramação de blocos, os quais, internamente contém seus respectivos códigos fonte. As bibliotecas e o software na sua maioria são constituídos em Python, sendo as primitivas de processamento de sinais implementadas em C++, o que permite ao usuário a opção de utilizar os blocos já criados disponíveis, ou editá-los através de linhas de código, além de criar novos blocos para utilização. O *GNU Radio* pode ser uma ferramenta poderosíssima nas mãos do usuário, que por sua vez, pode criar desde projetos simples até sistemas mais robustos. O *GNU Radio*, juntamente com o GRC, é o meio pelo qual se torna possível a interação com o SDR, fornecendo blocos nativos específicos para interfacear a plataforma de hardware de acordo com o tipo de SDR a ser utilizado. Para que essa interação aconteça, são usados basicamente dois tipos de blocos: o bloco “*Source*” e o bloco “*Sink*”.

⁴⁸ <https://www.linuxjournal.com/article/7319>.

⁴⁹ <https://www.gnuradio.org/about/>.

Figura 16 - Ambiente de testes do GNU Radio⁵⁰



O bloco *Source* possui a função de receber o sinal de RF advindo do hardware e em seguida é tratado de acordo com os parâmetros definidos pelo usuário. Já o bloco *Sink*, executa o processo reverso, ou seja, coleta as amostras do sinal digital e os envia para o hardware, com o propósito de transmitir o sinal de RF para o ar.

⁵⁰ https://wiki.gnuradio.org/index.php/Guided_Tutorial_GNU_Radio_in_Python.

3 DESENVOLVIMENTO

A partir do estudo e da fundamentação estabelecida até então, pode-se então introduzir a conceituação e apresentação da arquitetura do protótipo e do experimento criado, provando a sua viabilidade. Todos os temas distintos abordados anteriormente têm como propósito culminar em uma solução abrangente, descentralizando os sistemas de comunicações como são conhecidos, de forma a torná-los mais baratos e menos robustos, com uma implementação mais simples e maior praticidade de configuração.

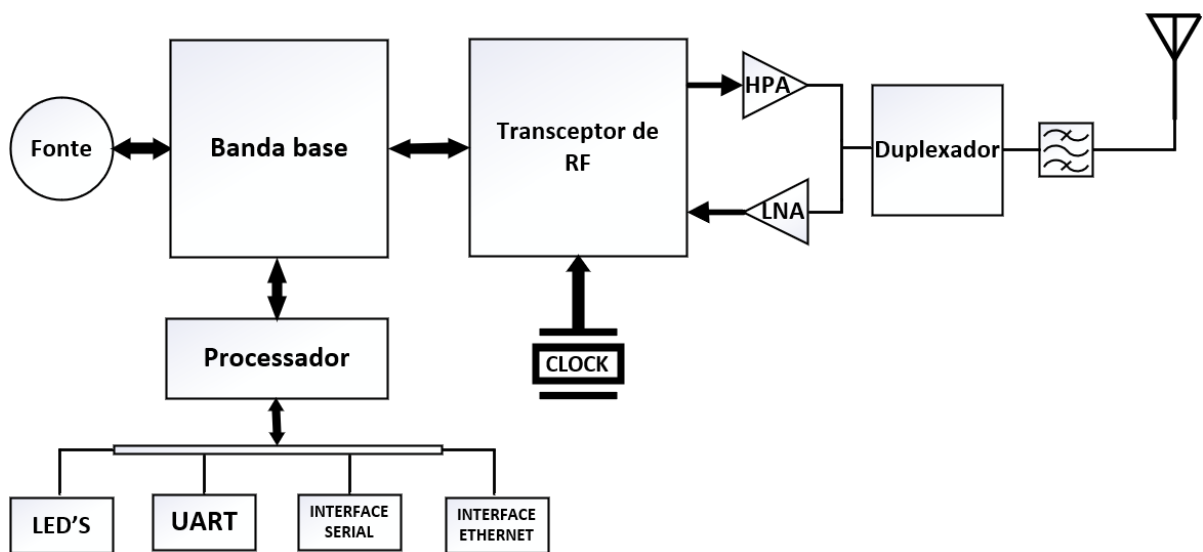
Aqui descreve-se, primeiramente, um modelo teórico ideal para a compreensão do sistema como um todo. Como qualquer sistema de comunicação básico, este protótipo é composto por vários elementos como filtros, amplificadores, moduladores, entre outros. Entretanto, boa parte do trabalho é realizado pelo SDR, que é o “coração” do sistema. É ele que torna a arquitetura mais simplificada e unifica várias funções dentro de seu escopo de atuação. Em seguida, detalha-se os processos de instalação de software (todas os repositórios necessários, assim como as bibliotecas a serem utilizadas) para que seja introduzido o experimento realizado, a saber, um transmissor WBFM com parâmetros ajustáveis, tendo como fonte do sinal de informação a própria placa de som do computador ou sistema embarcado, transmitido inicialmente na faixa de 250 MHz, com parâmetros que fazem o ajuste de frequência da portadora do sinal em tempo real para o espectro de frequência FM convencional (sendo usada como exemplo a faixa de 105,6 MHz) e, posteriormente, seus respectivos resultados.

3.1 Arquitetura Ideal do Protótipo

O protótipo de rádio ideal em sua completude é caracterizado por vários componentes que ocupam separadamente seu papel, para que haja funcionamento total do sistema, trazendo modularidade, interoperabilidade entre as partes intermediárias da comunicação e, principalmente, diminuição nos custos. A Figura 17 mostra um modelo de visão sistêmica teórico da arquitetura. O módulo SDR ocupa o papel do transceptor de radiofrequência do processamento da banda base, sendo o cristal de *clock* inerente ao próprio SDR. Cabe a sua função a geração, modulação e codificação do sinal a ser transmitido (assim como a recepção, demodulação e decodificação do sinal recebido) de acordo com a tecnologia de banda base a ser utilizada. O processador exerce um papel de intermediário dos periféricos de comunicação com a banda base, possuindo a função dupla de processador e controlador das atividades.

Posteriormente, é realizado o tratamento dos dados pelo *Universal Asynchronous Receiver/Transmitter* (UART), que possibilita a transmissão e a recepção de dados na forma serial, além de garantir a integridade dos mesmos e a sincronização do lado que transmite para que o lado que recebe as informações saiba onde começa e onde termina a respectiva transmissão, sendo os dados posteriormente extraídos por uma das interfaces disponíveis (seja a interface Serial ou a Ethernet).

Figura 17 - Diagrama de blocos da arquitetura



Devido ao apelo por uma solução de baixo custo, pode-se considerar o uso de um microcomputador (*Raspberry Pi* ou similar). Já do lado do tratamento do sinal, tem-se o HPA, que é usado para amplificar o sinal na transmissão e o LNA, que permite que o sinal de recepção – o qual geralmente chega com baixa potência – seja amplificado sem que haja mudanças em sua relação sinal-ruído, pois do contrário, informações importantes do sinal podem ser perdidas no processo. Por fim, há um filtro passa-faixa para que seja sintonizada a frequência de operação utilizada e a antena para transmissão e recepção.

3.2 Instalação de Software

Para que haja um funcionamento correto, no que diz respeito ao interfaceamento entre o LimeSDR e o *GNU Radio*, é imprescindível que sejam utilizados as dependências, repositórios e bibliotecas corretos. Assim sendo, para benefício do usuário, a própria *Myriad*

fornece os *plugins* necessários para o processo de instalação. Toda a aplicação será realizada em *kernel* Linux, utilizando o sistema operacional Ubuntu 18.04 LTS.⁵¹

Inicialmente, é necessária a instalação do *GNU Radio*. Há duas formas de instalação possíveis: a primeira é através dos arquivos binários contidos em repositório (que varia de acordo com a distribuição Linux utilizada), o qual já contém um pacote padrão para instalação. Já a segunda pode ser realizada a partir do código fonte da aplicação, que por sua vez, é mais utilizada em caso de não funcionamento do repositório, ou opção por parte do usuário de obter a última versão disponível. Opta-se pela utilização do primeiro método devido sua maior facilidade, o qual em distribuições Ubuntu pode ser instalado a partir do seguinte comando:

```
apt install gnuradio
```

Em seguida, deve-se realizar a instalação dos pacotes de drivers da *LimeSuite*. Isso pode ser feito através do repositório de arquivos da *Myriad*. Juntamente com os drivers, é necessário também a instalação do *SoapySDR*⁵², que é a API de código livre responsável por realizar a comunicação e interfaceamento entre a plataforma de software e o módulo SDR. O *SoapySDR* traz a vantagem de fornecer suporte e compatibilidade a outros tipos de dispositivos e outras plataformas disponíveis.

```
sudo add-apt-repository -y ppa:myriardf/drivers
sudo apt-get update
sudo apt-get install limesuite liblimesuite-dev limesuite-udev limesuite-images
sudo apt-get install soapysdr-tools soapysdr-module-lms7
sudo apt-get install soapysdr soapysdr-module-lms7
```

O próximo passo consiste na clonagem do repositório GitHub da *LimeSuite*.⁵³ Primeiro deve ser criado um diretório *builddir*. Em seguida, é necessário o uso do *Cross-Platform Makefile Generator* (CMake), que consiste em uma ferramenta que possibilita a geração de scripts de construção para a aplicação.⁵⁴ Como iremos lidar com projetos em C++, essa ferramenta se torna imprescindível na instalação. Consequentemente, os projetos posteriores especificarão seu processo de compilação através de listas de arquivos que são incluídas no diretório fonte. Para efetuar a compilação usa-se o comando *make*. Por fim, é dado o comando

⁵¹ <https://ubuntu.com/>.

⁵² <https://github.com/pothosware/SoapySDR/wiki>.

⁵³ <https://github.com/myriardf/LimeSuite>.

⁵⁴ <https://cmake.org/>.

ldconfig, que é responsável pela construção de um cache, em outras palavras, um link que aponta para os arquivos das bibliotecas compartilhadas que são instaladas no processo, para que em caso de uma futura atualização das mesmas seja mais fácil de identificar os parâmetros das respectivas bibliotecas.

```
git clone https://github.com/myriadr/f/LimeSuite.git
cd LimeSuite
git checkout stable
mkdir builddir && cd builddir
cmake ..
make
sudo make install
sudo ldconfig
```

Para que as rotinas em C++ e Python possam ser executadas corretamente, é fundamental o papel de duas bibliotecas: a primeira é a biblioteca Boost, que é responsável por prover suporte no que diz respeito à tarefas que demandam uso de álgebra linear, geração de números, expressões regulares, processamento de imagens, entre outras funções.⁵⁵ A segunda é a biblioteca Simplified Wrapper and Interface Generator (SWIG), que é uma ferramenta utilizada para permitir que linguagens de script (como é o caso do Python) possam invocar ou conectar bibliotecas em C++ de forma indireta.⁵⁶ Para instalá-las deve ser dado o comando:

```
sudo apt-get install libboost-all-dev swig
```

Por ser uma plataforma de código livre, o *GNU Radio* além das bibliotecas padrão e blocos básicos, oferece também a possibilidade da criação de blocos por outros usuários. Consequentemente, a própria *Myriad* criou os *plugins* (em forma de blocos) necessários para o funcionamento das placas LimeSDR no *GNU Radio* e disponibiliza para uso um repositório GitHub⁵⁷, o qual permite a clonagem dos arquivos do projeto com os respectivos pacotes, importando-os para a máquina do usuário. Pode-se realizar a cópia do repositório por meio do comando:

```
git clone https://github.com/myriadr/gr-limesdr
```

⁵⁵ <https://www.osetc.com/en/how-to-install-boost-on-ubuntu-16-04-18-04-linux.html#what-is-boost>.

⁵⁶ <https://www.dev2qa.com/how-to-install-swig-on-macos-linux-and-windows/>.

⁵⁷ <https://github.com/myriadr/gr-limesdr>.

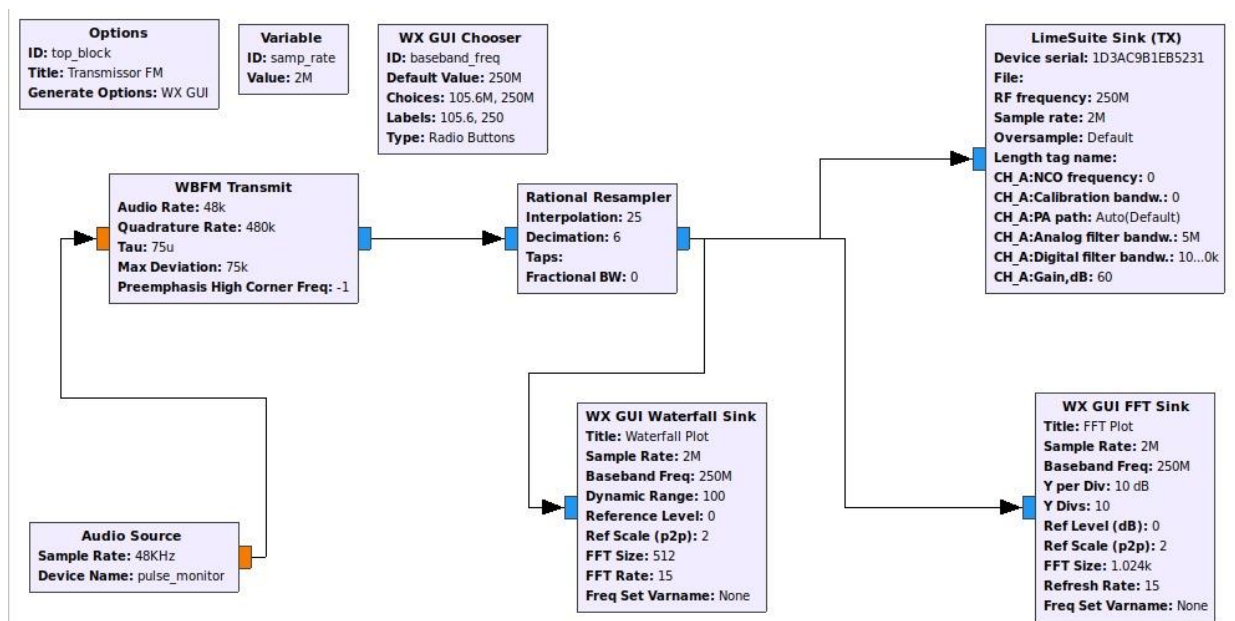
Feita a cópia do repositório, um diretório chamado *gr-limesdr* deve ser criado para realizar o *build* e executar o projeto. Pode-se então efetuar a instalação e compilação dos arquivos.

```
cd gr-limesdr
mkdir build
cd build
cmake ..
make
sudo make install
sudo ldconfig
```

3.3 Fluxograma do GNU Radio

Após a instalação das respectivas bibliotecas, pode-se realizar a construção da parte lógica que consiste na diagramação e conexão dos blocos responsáveis pela geração e tratamento do sinal e posteriormente, sua comunicação com o LimeSDR. Dessa forma, tendo como base um dos diagramas exemplo que são disponibilizados pela *Myriad* (os quais explicam em termos básicos como os blocos que foram especificamente criados para efetuar a conexão com a placa transceptora funcionam) nos arquivos de instalação, foi construído o fluxograma como indica a Figura 18.

Figura 18 - Diagrama de blocos do transmissor WBFM



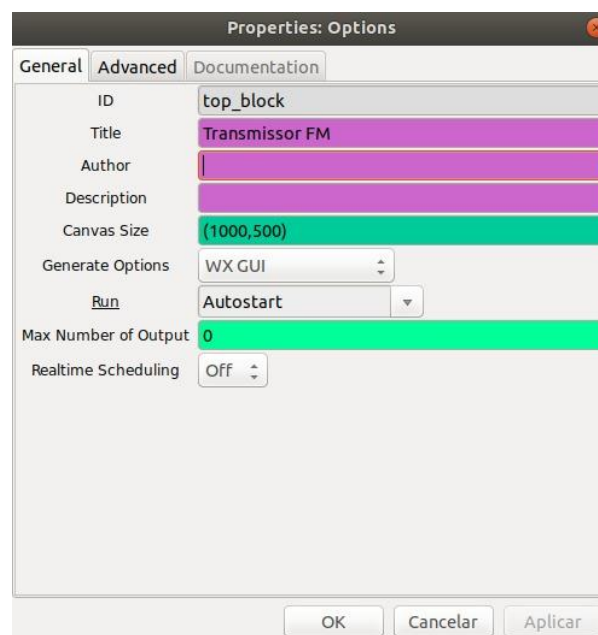
Cada bloco possui uma função diferente e parâmetros que podem ser próprios ou hierárquicos, ou seja, que podem exercer alguma influência em outros blocos. As configurações variam conforme a necessidade do usuário.

3.3.1 Blocos Hierárquicos

De acordo com a *GNU Radio Wiki*, os blocos hierárquicos são blocos que podem ser instanciados dentro de outros blocos num mesmo diagrama.⁵⁸ Esses blocos facilitam a organização do programa, principalmente em casos onde há uma grande quantidade de blocos que precisam ser configurados com um mesmo parâmetro. No diagrama construído, os blocos hierárquicos utilizados são o *Options*, *Variable* e *WX GUI Chooser*,

Como indica a Figura 19, o bloco *Options* possui como parâmetros principais o *ID* do projeto (o qual deve ser exclusivo entre a biblioteca de blocos), que define o nome do módulo, o *Title*, que será o nome de exibição da tela do projeto quando compilado, o qual foi definido como “Transmissor FM” e o *Generate Options*, que é responsável pela escolha de *Graphic User Interface* (GUI) a ser utilizado. O *GNU Radio* em sua versão 3.7.0, é constituído por dois tipos de GUI que oferecem kits de desenvolvimento para a ferramenta: o *WX GUI* e o *QT GUI*, ambos derivados de *frameworks* Python.⁵⁹ Optamos pelo uso do *WX GUI*, embora seja também possível a construção do mesmo fluxograma utilizando o *QT GUI*.

Figura 19 - Bloco Options



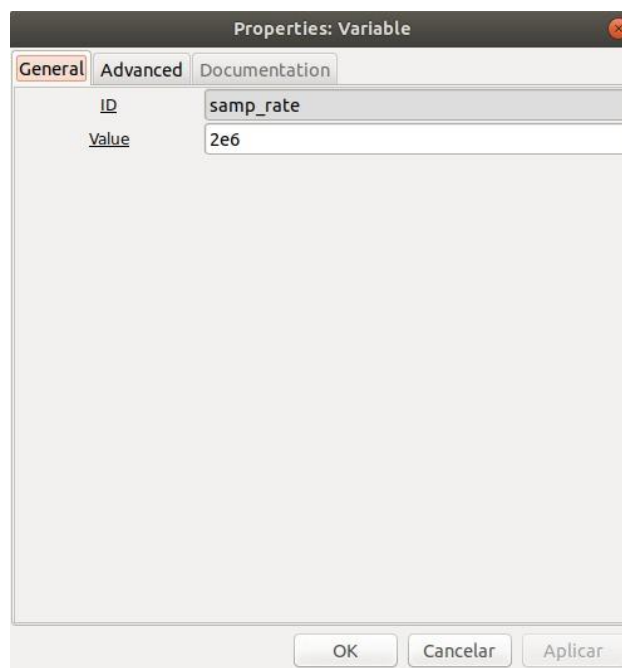
⁵⁸ <https://wiki.gnuradio.org/index.php/GNURadioCompanion>.

⁵⁹ <https://opensource.com/article/17/4/pyqt-versus-wxpython>.

O bloco *Variable* é a forma mais simples de utilizar uma variável no *GNU Radio*. Deve-se definir o ID da variável e seu respectivo valor. Para utilizá-la em outros blocos, é necessário colocar o seu nome simbólico no parâmetro desejado. A Figura 20 mostra a lista de parâmetros de configuração do bloco.

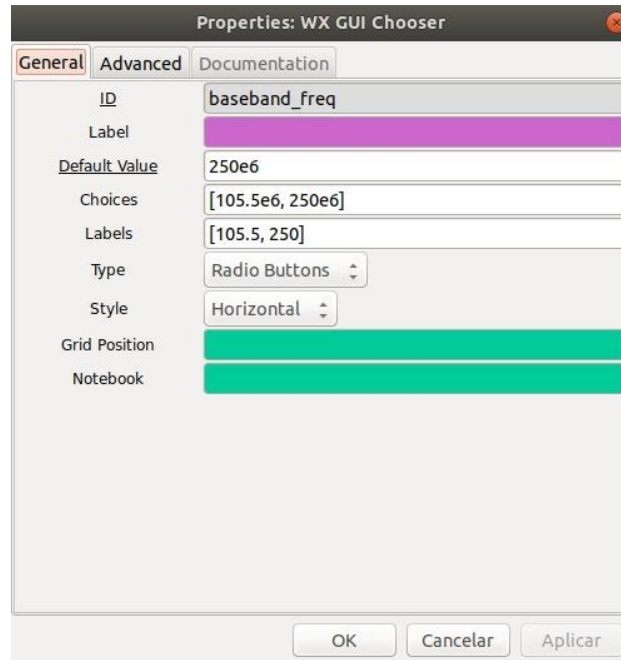
Usa-se o *WX GUI Chooser* como um bloco de controle variável, o qual permite que parâmetros possam ser alterados durante a execução do fluxograma. No caso desse bloco, deve-se determinar a posição das opções a serem selecionadas, seus respectivos valores e a forma como são dispostos na janela de execução (horizontal/vertical). Os controles de variação em geral (sejam por blocos seletores de opções ou que fazem um ajuste de variação linear) no GRC usam variáveis em combinação com métodos de retorno de chamada para modificar esses parâmetros.

Figura 20 - Bloco Variable



Na Figura 21, são dados os parâmetros da frequência de operação do sistema. O bloco, portanto, é configurado de forma a possuir um vetor com duas opções de faixas (embora haja a possibilidade de criar-se um vetor com mais valores), a saber, a faixa de 250 MHz, a qual é a frequência do sistema e a faixa de 105,6 MHz, que está dentro da banda disponível para rádios FM convencionais.

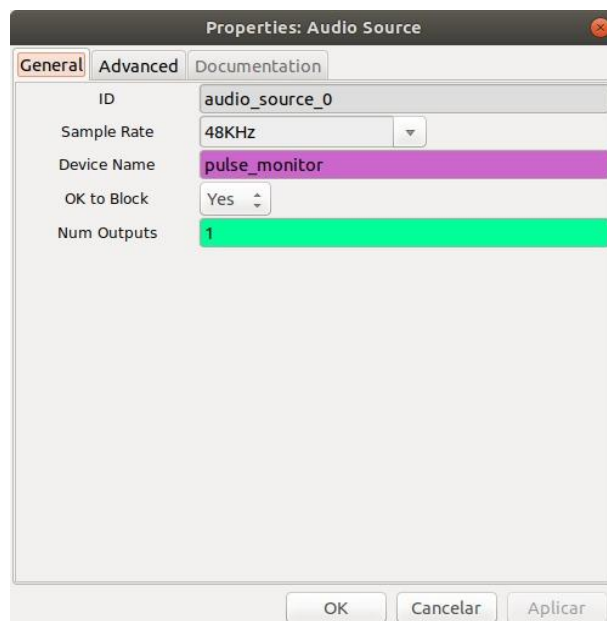
Figura 21 - Bloco WX GUI Chooser



3.3.2 Blocos Fonte e Sumidouro

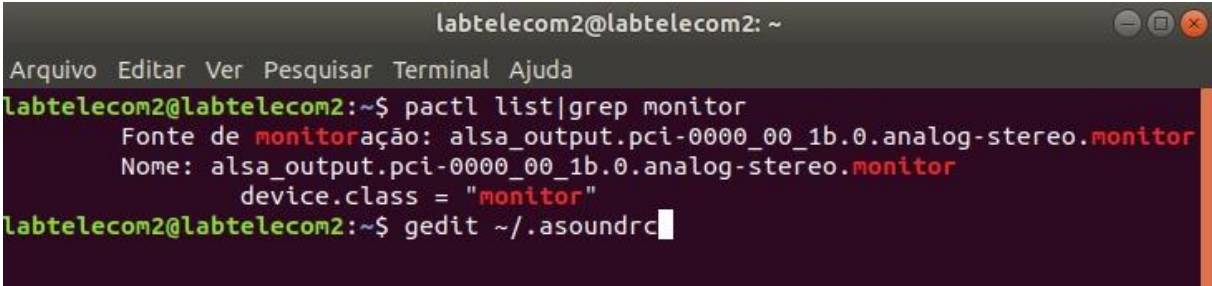
Os blocos que possuem característica de fonte e sumidouro são responsáveis por criar as entradas e saídas para os blocos hierárquicos. Neste projeto são utilizados um bloco fonte e três blocos sumidouros, os quais veremos em seguida. O bloco *Audio Source* possui um papel fundamental na constituição do fluxograma, pois é por meio dele que o sinal da informação é gerado.

Figura 22 - Bloco Audio Source



Os parâmetros de interesse no que diz respeito à configuração do bloco de acordo com a Figura 22 são, em primeiro lugar, a taxa de amostragem do sinal (*Sample Rate*), que depende da qualidade da placa de som do computador ou dispositivo embarcado. Se a placa possuir alta qualidade, maiores taxas são permitidas. Usamos a taxa de 48 MHz, que se caracteriza como uma taxa intermediária, o que permite que se tenha um som com qualidade normal, sem distorções. Em segundo lugar, o nome do arquivo ou dispositivo pelo qual o sinal será obtido pelo bloco. É imprescindível aqui, o uso da biblioteca *ALSA/PulseAudio*, oriunda do servidor de som *PulseAudio*, e dos drivers de placa de som através de controle de kernel da *Advanced Linux Sound Architecture (ALSA)* os quais são bastante utilizados em sistemas Linux. Por ser também um software baseado em código livre, o *PulseAudio* torna seu uso muito útil para a aplicação, além de atuar sobre os termos da *GNU General Public License*.⁶⁰ A biblioteca permite que o som oriundo diretamente da placa de som do computador possa ser obtido pelo *GNU Radio* e, posteriormente, processado da maneira desejada com uma qualidade de áudio significativa, dependendo da taxa de amostragem utilizada. Para que sua instalação seja feita, é necessário que seja localizada a fonte de monitoração de som do computador, que pode ser localizado a partir do procedimento indicado na Figura 23.

Figura 23 - Fonte de monitoração de som do computador



```

labtelecom2@labtelecom2: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
labtelecom2@labtelecom2:~$ pactl list|grep monitor
Fonte de monitoração: alsa_output.pci-0000_00_1b.0.analog-stereo.monitor
Nome: alsa_output.pci-0000_00_1b.0.analog-stereo.monitor
device.class = "monitor"
labtelecom2@labtelecom2:~$ gedit ~/.asoundrc

```

O *pactl* é um comando específico para realizar tarefas de controle sob o servidor *PulseAudio*. Como neste caso o objetivo é encontrar o módulo fonte carregado, usa-se o comando *list* que mostra uma lista com todos os dispositivos fonte e sumidouro, módulos disponíveis, etc.⁶¹ Como o objetivo é encontrar o módulo de monitoração que está em utilização pelo computador, o comando *grep* é utilizado como filtro de busca.

Posteriormente, é necessário que seja criado um arquivo *.asoundrc* para adicionar o módulo utilizado às funcionalidades do *ALSA*, criando uma configuração personalizada para a

⁶⁰ <https://wiki.archlinux.org/index.php/PulseAudio>.

⁶¹ <http://manpages.ubuntu.com/manpages/xenial/man1/pactl.1.html>.

placa de som e permitindo também a conversão da taxa de amostragem do sinal posteriormente.⁶² A Figura 24 mostra o conteúdo do arquivo criado no computador utilizado para o experimento.

Figura 24 - Arquivo .asoundrc



```

pcm.pulse_monitor {
    type pulse
    device alsa_output.pci-0000_00_1b.0.analog-stereo.monitor
}

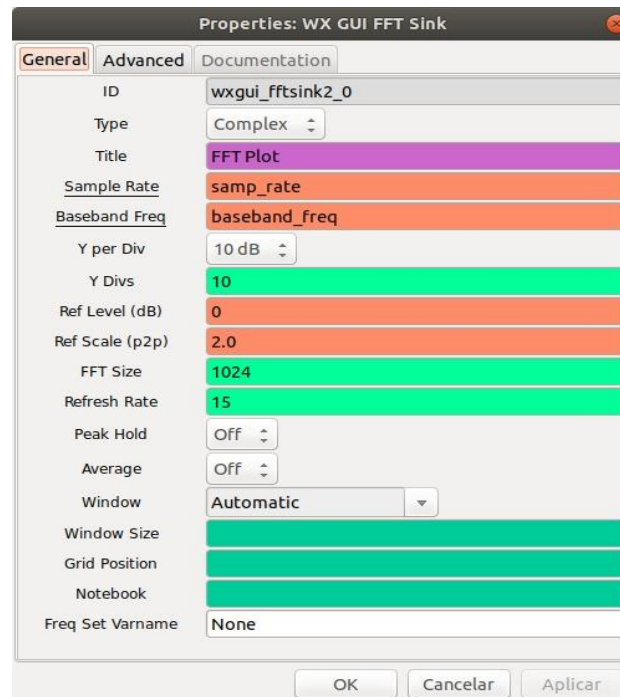
ctl.pulse_monitor {
    type pulse
    device alsa_output.pci-0000_00_1b.0.analog-stereo.monitor
}

```

Após a criação do arquivo, os seguintes parâmetros devem ser colocados para a criação da variável *pulse_monitor*, que será a variável a ser utilizada pelo *GNU Radio* para realizar a chamada de função. Dentro de seu argumento é colocado o respectivo módulo encontrado pelo passo anterior.

O bloco *WX GUI FFT Sink* possui a função de executar o gráfico do sinal a ser transmitido no domínio da frequência por meio da Transformada Rápida de Fourier (FFT).

Figura 25 - Bloco WX GUI FFT Sink

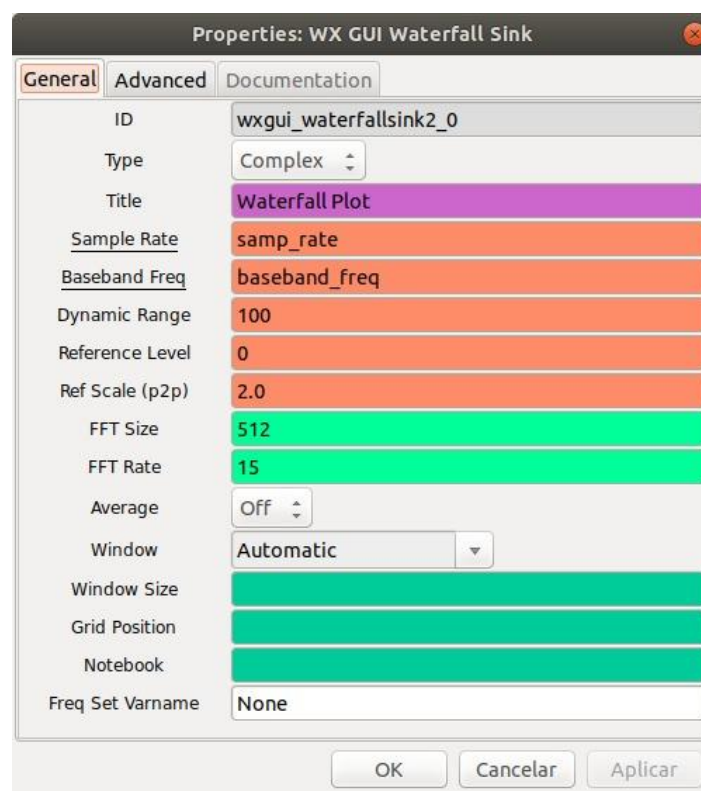


⁶² <https://alsa.opensrc.org/Asoundrc>.

Dentre os diversos parâmetros desse bloco, indicados na Figura 25, os principais a serem utilizados no mesmo são a taxa de amostragem e a frequência de operação (*Baseband Frequency*), os quais são preenchidos com suas respectivas variáveis já criadas, a *samp_rate* e a *baseband_freq*. Os valores de outros parâmetros como o tamanho e taxa da FFT, por exemplo, são originados por padrão na criação do bloco.

Semelhantemente ao bloco anterior, o *WX GUI Waterfall Sink* também executa os resultados gráficos da FFT, porém dispondo-os em forma de cascata. Os parâmetros aqui são configurados de forma análoga e ilustrados na Figura 26.

Figura 26 - Bloco WX GUI Waterfall Sink



Por último, tem-se o bloco responsável pelo controle da placa LimeSDR. O *LimeSuite Sink (TX)* recebe diversos parâmetros para que o sinal seja tratado pelo transceptor LMS7002M e seus filtros internos e, por fim, ser transmitido para o ar. Para configurá-lo, primeiramente deve-se inserir o número de série da placa a ser utilizada. Ao efetuar o comando⁶³ mostrado na Figura 27, é dado na tela do terminal Ubuntu o número de série do dispositivo usado para este trabalho.

⁶³ https://wiki.myriadrf.org/Gr-limesdr_Plugin_for_GNURadio.

Figura 27 - Número de série da placa LimeSDR utilizada

```

labtelecom2@labtelecom2: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
labtelecom2@labtelecom2:~$ sudo LimeUtil --find
[sudo] senha para labtelecom2:
* [LimeSDR Mini, media=USB 2.0, module=FT601, addr=24607:1027, serial=1D3AC9B1EB5231]
labtelecom2@labtelecom2:~$ █

```

Em seguida, o número de série é colocado no seu respectivo parâmetro do bloco para que no momento da comunicação a placa possa ser identificada pelo computador, juntamente com as outras configurações relacionadas na Figura 28. No entanto, o usuário tem a opção de ajustar outros parâmetros que exercem funções que podem impactar na característica final do sinal. O parâmetro *Channel* possibilita a configuração de um canal *Single Input Single Output* (SISO) ou MIMO. Na placa LimeSDR-Mini, não há a possibilidade de uso de canal MIMO, o qual é restrito a outras placas (Como a LimeSDR em tamanho padrão). Entretanto, existem dois canais SISO disponíveis (A e B), sendo o canal A o escolhido para a aplicação.

Na segunda aba referente ao bloco, indicada na Figura 29, pode-se efetuar configurações no que diz respeito ao tratamento do sinal. Os parâmetros de importância para o projeto são os referentes aos filtros analógico e digital internos à placa e o ganho de transmissão do canal, o qual pode ser ajustado até um máximo de 60 dB.

Figura 28 - Bloco LimeSuite Sink (TX)

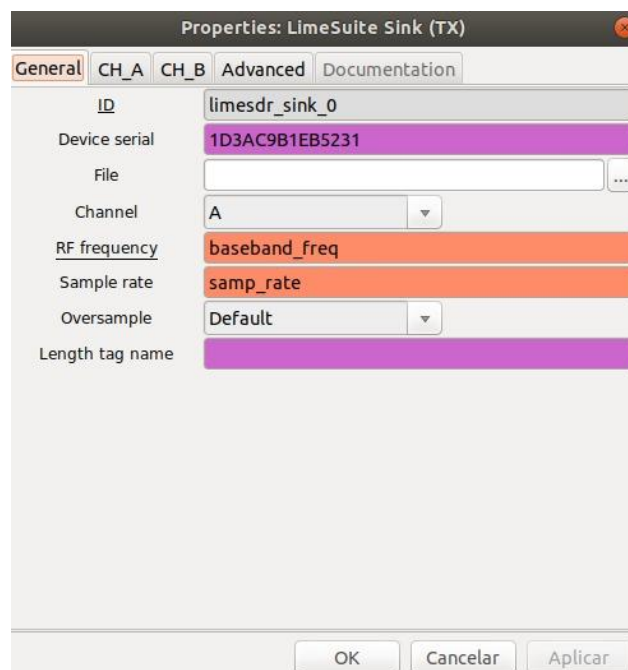
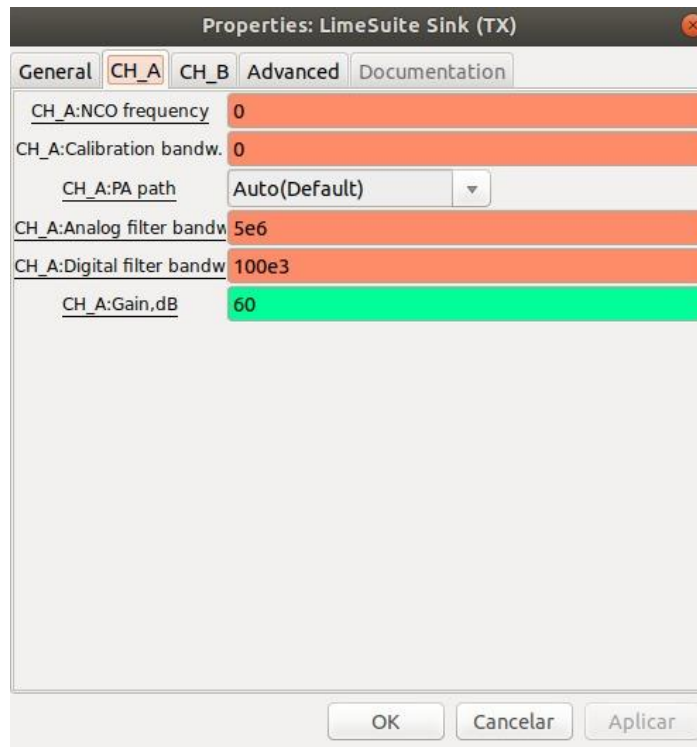


Figura 29 - Funções secundárias do bloco LimeSuite Sink (TX)



3.3.3 Outros Blocos

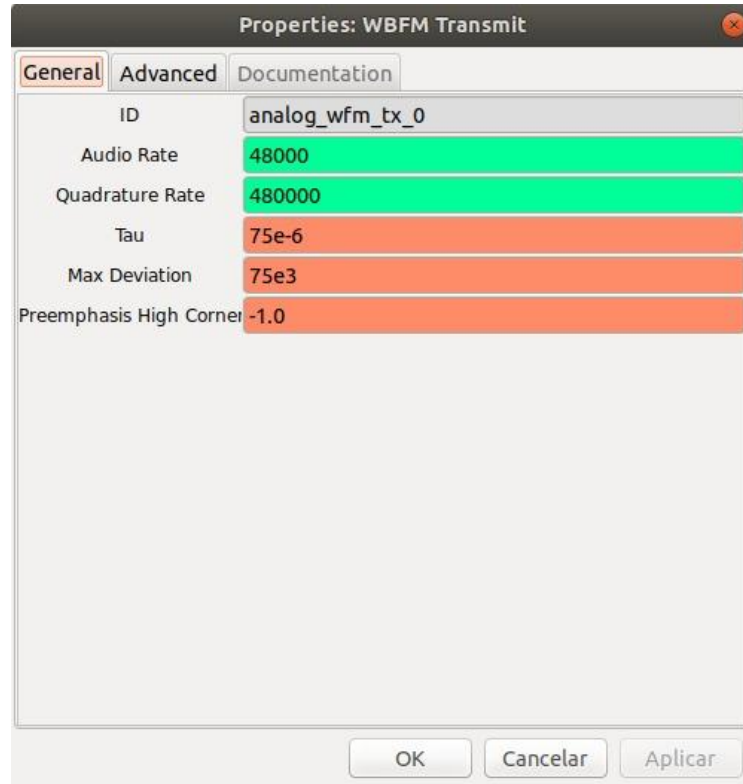
Além dos blocos citados anteriormente que fazem parte do fluxograma criado, existem também dois blocos que ocupam um papel fundamental no processamento como um todo, pois são eles os responsáveis por dar a característica final antes da transmissão do sinal. São o *WBFM Transmit* e o *Rational Resampler*.

O primeiro é responsável por atuar como um modulador FM. Os parâmetros que o constituem, segundo indica a Figura 30, são o *Audio Rate*, ou seja, a taxa de amostragem do áudio advindo da placa de som do computador; o *Quadrature Rate*, que consiste na taxa de quadratura do sinal, a qual deve ser um múltiplo inteiro da taxa de amostragem do áudio; a constante de pré-ênfase de tempo τ (τ), dada em microssegundos (μs), sendo utilizado neste bloco o valor de 75 microssegundos (μs), o qual é o mais utilizado nas regiões da América;⁶⁴ o *Maximum Deviation*, ou desvio máximo de frequência em Hz; e o *Preemphasis High Corner Freq*, que é a componente de frequência a qual a pré-ênfase se torna estabilizada. Por padrão, o *GNU Radio* adota um número menor que zero, o que significa que por padrão a frequência

⁶⁴ <https://www.electronics-notes.com/articles/audio-video/broadcast-audio/vhf-fm-pre-de-emphasis.php>.

passa pela fórmula⁶⁵ $0.925 * \text{Quadrature Rate} / 2$ que no caso da presente aplicação, a uma *Quadrature Rate* de 48000, o valor de frequência será de 22,2 kHz.

Figura 30 - Bloco WBFM Transmit

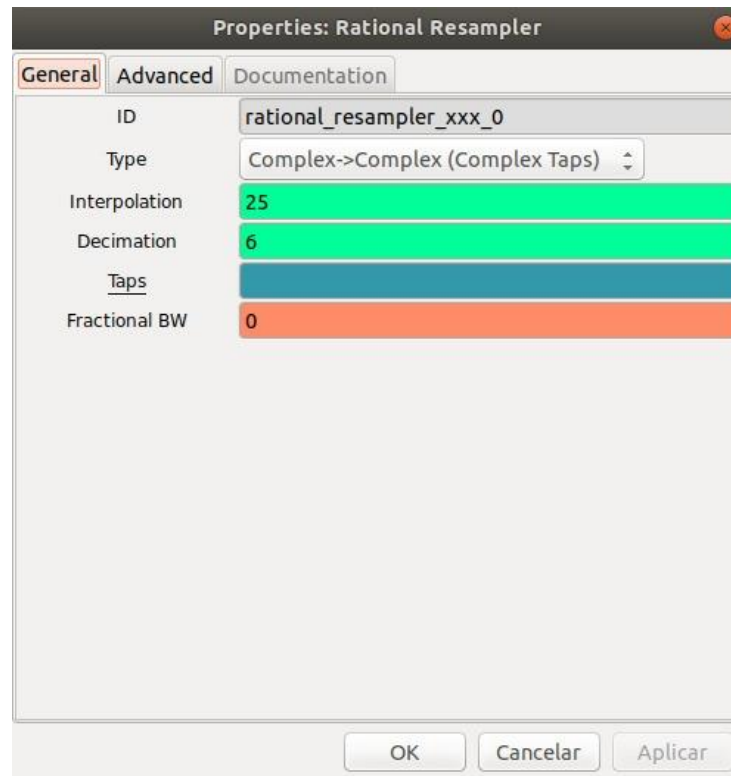


O segundo bloco, *Rational Resampler*, é constituído por um filtro de resposta ao impulso finita ou *Finite Impulse Response* (FIR), cuja função é efetuar a reamostragem do sinal que sai do bloco *WBFM Transmit* para que a taxa de amostragem de saída seja igual a taxa de amostragem de entrada dos blocos sumidouros. Para que isso seja possível, o sinal passa por um processo de decimação e interpolação para que a taxa de amostragem seja alterada. A decimação consiste na redução da amostragem do sinal e a interpolação diz respeito à expansão da amostragem do sinal obtido [11].

O bloco, cujos parâmetros são relacionados na Figura 31, executa o cálculo como sendo a taxa de amostragem do sinal de entrada multiplicado pelo fator interpolador e dividido pelo fator decimador. Sendo assim, o sinal de saída com taxa de amostragem de 480 mil amostras é multiplicado por 25 e dividido por 6, o que nos dá um total de 2 milhões de amostras, que é o valor configurado nos blocos sumidouros.

⁶⁵ https://wiki.gnuradio.org/index.php/NBFM_Transmit.

Figura 31 - Bloco Rational Resampler



3.4 Receptor RTL-SDR

O sinal enviado pelo transmissor FM pode ser facilmente detectado a partir do uso de um rádio FM comum. Todavia, como a frequência de operação real do sistema está na faixa de 250 MHz, ou seja, fora da banda padrão de atuação das rádios FM no Brasil, faz-se necessário o uso de um receptor específico para detectar o sinal em 250 MHz. Para essa finalidade, utilizamos o RTL-SDR, que é um dispositivo de interface USB de baixo custo (aproximadamente \$ 25,00 dólares) o qual consiste em uma scanner de rádio que detecta sinais de variados tipos, atuando em um *range* de 500 kHz até sinais de 1,7 GHz.⁶⁶ Através dos anos, o uso do RTL-SDR tem se tornado bastante popular pelo apelo de baixo custo que oferece, podendo ser de uso bastante útil em determinadas aplicações, principalmente em *Digital Video Broadcast* (DVB), onde seu uso tem sido mais recorrente. O modelo utilizado neste trabalho é o RTL2832U. O RTL2832U é composto pelo seu *dongle* de interface USB com entrada SMA fêmea e uma antena receptora, como realça a Figura 32.

⁶⁶ <https://www.rtl-sdr.com/about-rtl-sdr/>.

Figura 32 - RTL2832U⁶⁷

O software utilizado para o interfaceamento com o RTL-SDR é o *SDRSharp* (SDR#).⁶⁸ O SDR# é uma interface gráfica de usuário criada pela Airspy,⁶⁹ que também é uma empresa fabricante de SDR's receptores com apelo de mercado de baixo custo. Seu processo de instalação é bastante simples, além de possuir um ambiente com interface amigável para o usuário. Seu uso é gratuito e é disponibilizado o download no próprio site da empresa. Deseja-se utilizá-lo pelo Windows devido a praticidade. Para realizar a instalação correta, é necessário executar o arquivo em lotes `install-rtlsdr.bat` contido dentro da pasta de download do software. Também é necessário o uso do software *Zadig*⁷⁰, para que os drivers do RTL-SDR sejam devidamente instalados. Sua interface é de uso simples e é mostrada na Figura 33.

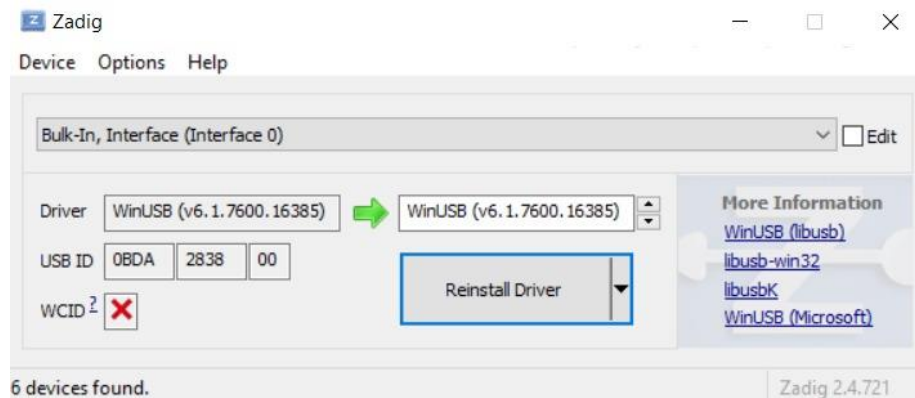
⁶⁷ <http://andrewburton.co.uk/website/ads-b-flight-tracking/>.

⁶⁸ <https://www.rtl-sdr.com/sdrsharp-users-guide/>.

⁶⁹ <https://airspy.com/>.

⁷⁰ <https://zadig.akeo.ie/>.

Figura 33 - Instalação de Drivers via Zadig



No que diz respeito à interface em si, a Figura 34 nos mostra o ambiente de configuração e visualização dos gráficos. Na parte esquerda da tela ficam as configurações da placa e do tipo de sinal a ser processado. Antes de iniciar o software, é necessário que o RTL-SDR seja selecionado no parâmetro *Source*, para identificação da placa. Há uma vasta gama de opções de SDR disponíveis para uso, como apresenta a Figura 35.

Figura 34 - Interface SDR#

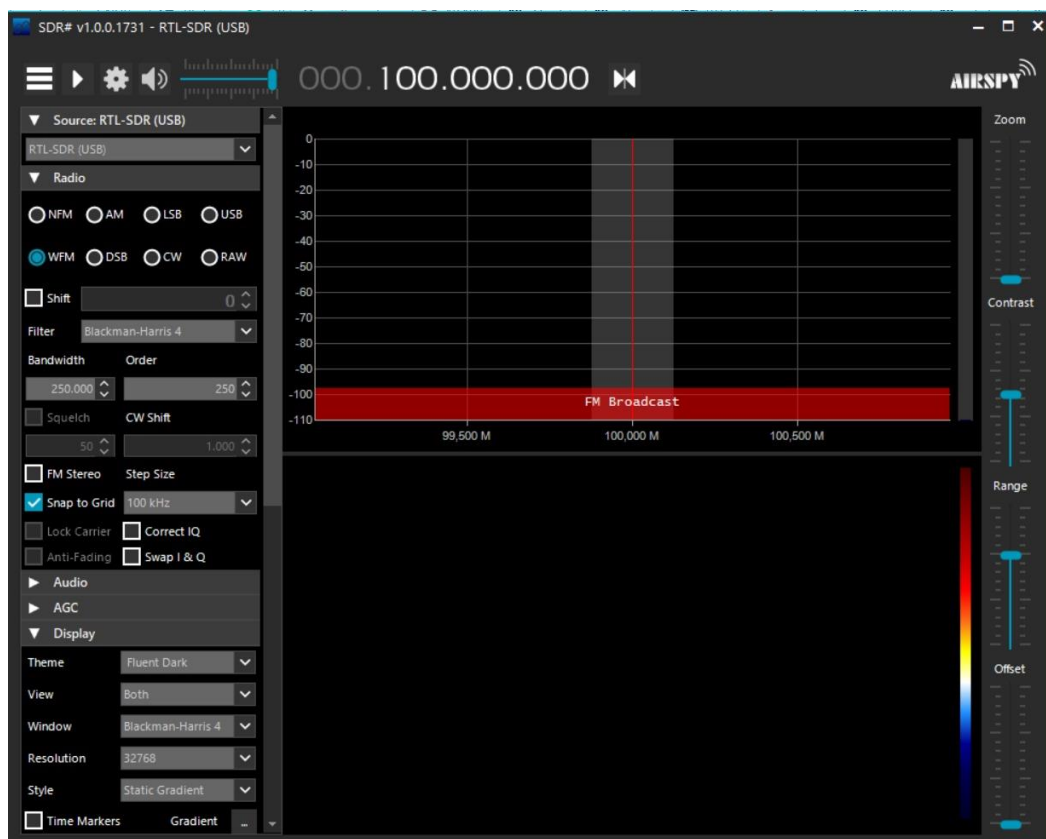
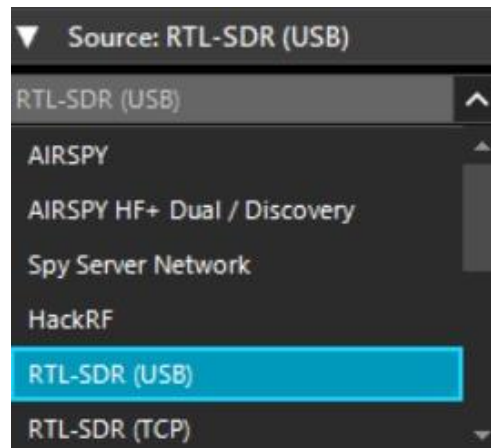


Figura 35 - Seleção da placa RTL-SDR para uso no SDR#



3.5 Construção do Experimento e Resultados

Na Tabela 3, há uma estimativa de custo unitário e o custo total dos equipamentos utilizados na montagem.⁷¹ Para a execução dos testes, tomou-se em consideração a transmissão de um sinal de banda base FM em linha de visada direta ou *Line of Sight* (LOS) através da placa LimeSDR-Mini, a qual utiliza um cabo jumper para fazer o papel de antena transmissora como indicado na Figura 36 e a recepção do mesmo sinal por meio do SDR RTL2832U mostrado na Figura 37. Foram utilizados um computador desktop com sistema operacional Ubuntu 18.04 LS do lado transmissor (Tx) e um notebook com sistema operacional Windows 10 do lado receptor (Rx) como mostrado nas Figuras 38 e 39.

Tabela 3 - Estimativa de custo do projeto

Peça	Custo (R\$)
1x LimeSDR-Mini	1.937,82
Cabo Jumper (conjunto)	5,00
1x RTLSDR RTL2832U	32,64
TOTAL	1.975,46

⁷¹ Valores pesquisados no dia 24/10/2019, todos disponíveis em <https://pt.aliexpress.com/>.

Figura 36 - LimeSDR-Mini utilizado



Figura 37 - RTL2832U utilizado



Figura 36 - Desktop utilizado para o Tx

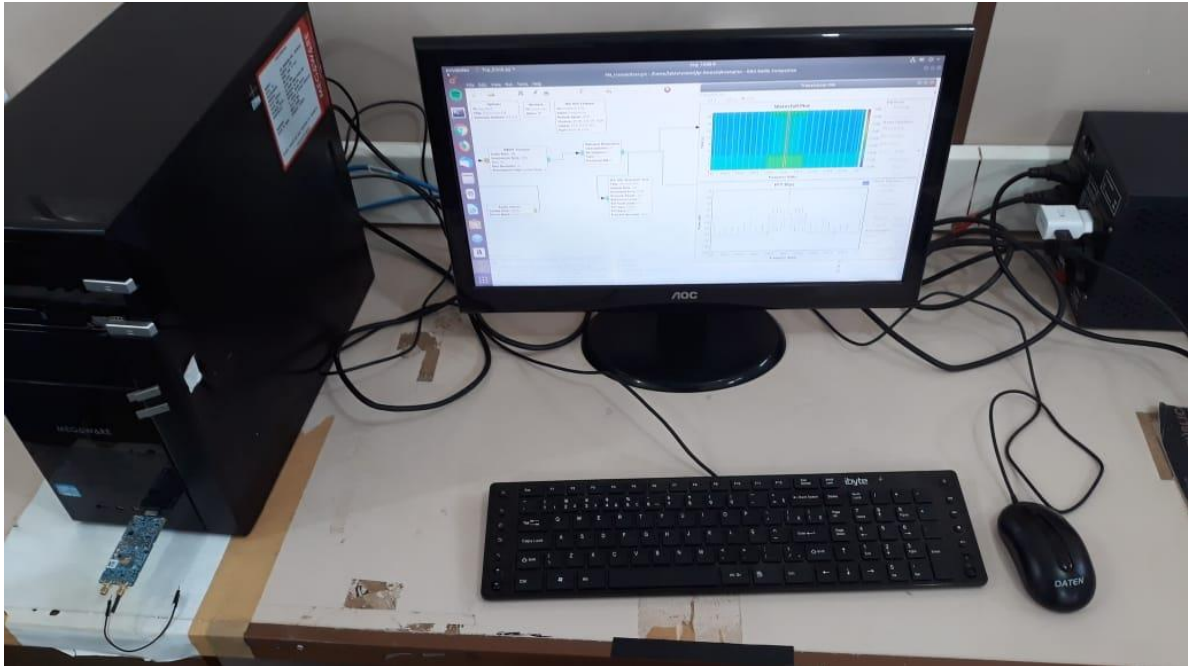


Figura 37 - Notebook utilizado para o Rx



Dois testes foram efetuados: o primeiro, utilizando a faixa de FM licenciada. De forma arbitrária, optamos pela frequência de operação de 105,6 MHz. Os gráficos das Figuras 40 e 41 mostram a forma do sinal tanto do lado da transmissão, onde são exibidos os gráficos gerados pelo *GNU Radio*, quanto da recepção, onde o *SDR#* também atua com seus respectivos gráficos.

Figura 38 - Gráficos do Tx em 105,6 MHz

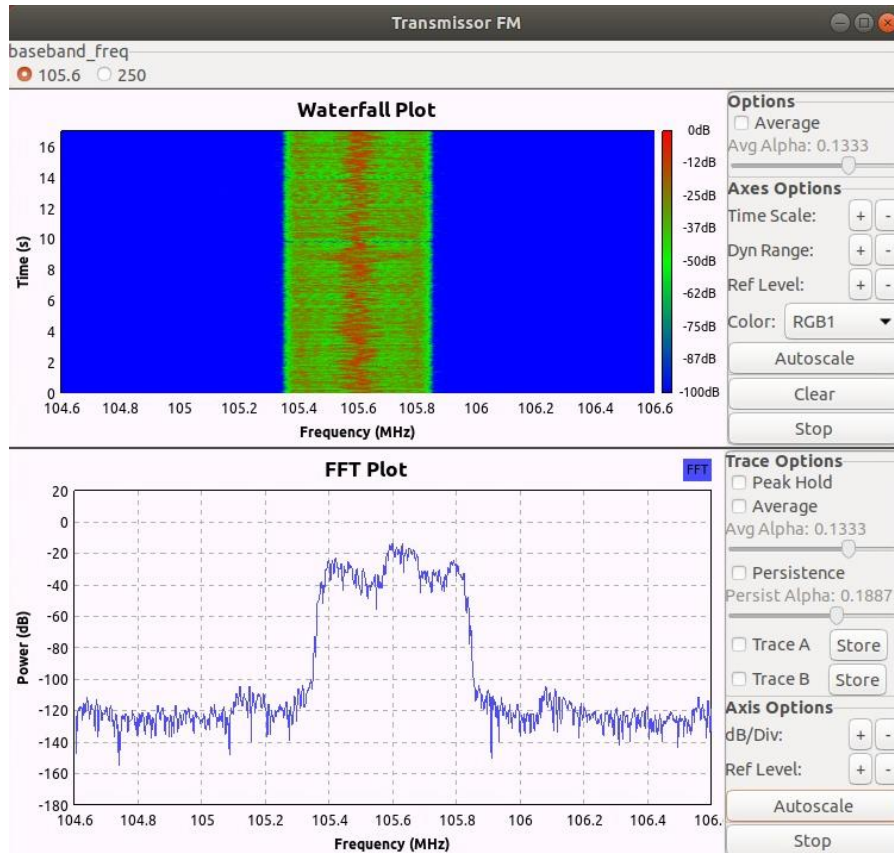
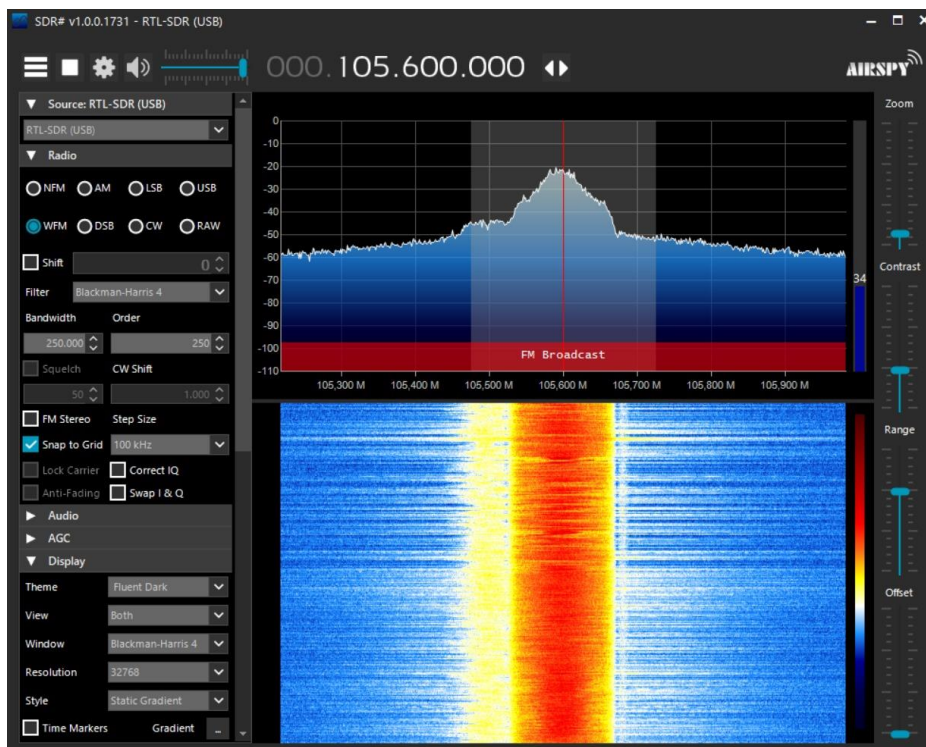


Figura 39 - Gráficos do Rx em 105,6 MHz



Vale ressaltar que esse mesmo sinal pode ser facilmente sintonizado por qualquer aparelho de rádio FM convencional, pelo fato de a faixa operante residir dentro do espectro licenciado, o qual é abrangido pelos equipamentos de rádio comuns. O sinal transmitido pelo SDR é sobreposto ao sinal operante da respectiva faixa licenciada apenas dentro da LOS, dando lugar novamente ao sinal licenciado, caso a portadora seja retirada do ar ou o receptor saia da LOS. Devido à alta sensibilidade do receptor, o valor da sintonização de frequência, embora não chegando a ser o valor exato, consiste em um valor bastante aproximado, com uma diferença decimal. É configurado também um ganho automático para caso de queda na potência do sinal, o qual na imagem indica o valor de 34 dB.

A frequência de transmissão pode ser mudada em tempo real a partir do bloco *WX GUI Chooser*, como dito anteriormente. Portanto, segue-se o segundo teste, onde a frequência de operação original do sistema é sintonizada, ou seja, a faixa de 250 MHz.

Os gráficos das Figuras 42 e 43 mostram que, mesmo sendo transmitido em uma faixa fora da banda licenciada de FM, o sinal se comporta de forma semelhante ao primeiro teste, podendo ser efetuado o enlace e recebido de forma inteligível, sem muitos ruídos prejudiciais à qualidade. Na figura 43, a indicação embaixo do gráfico de frequência indica que a banda utilizada é restrita à área militar. No entanto, diz respeito a banda utilizada nos Estados Unidos. No Brasil, como já citado anteriormente, a banda de 250 MHz possui outros propósitos [12].

Figura 40 - Gráficos do Tx em 250 MHz

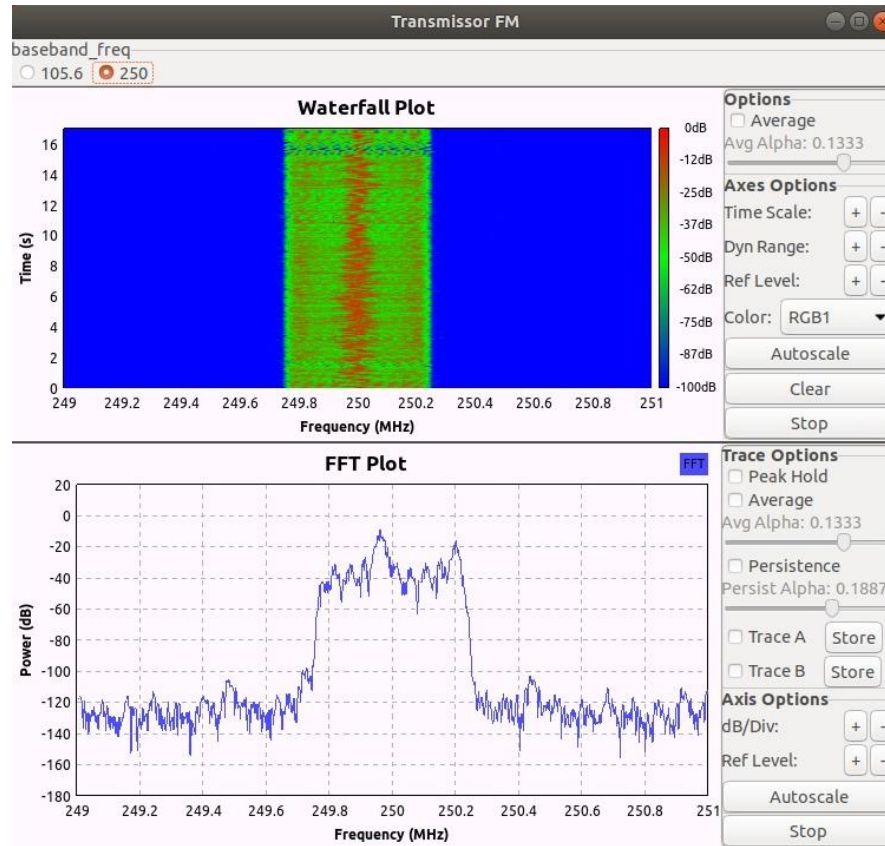
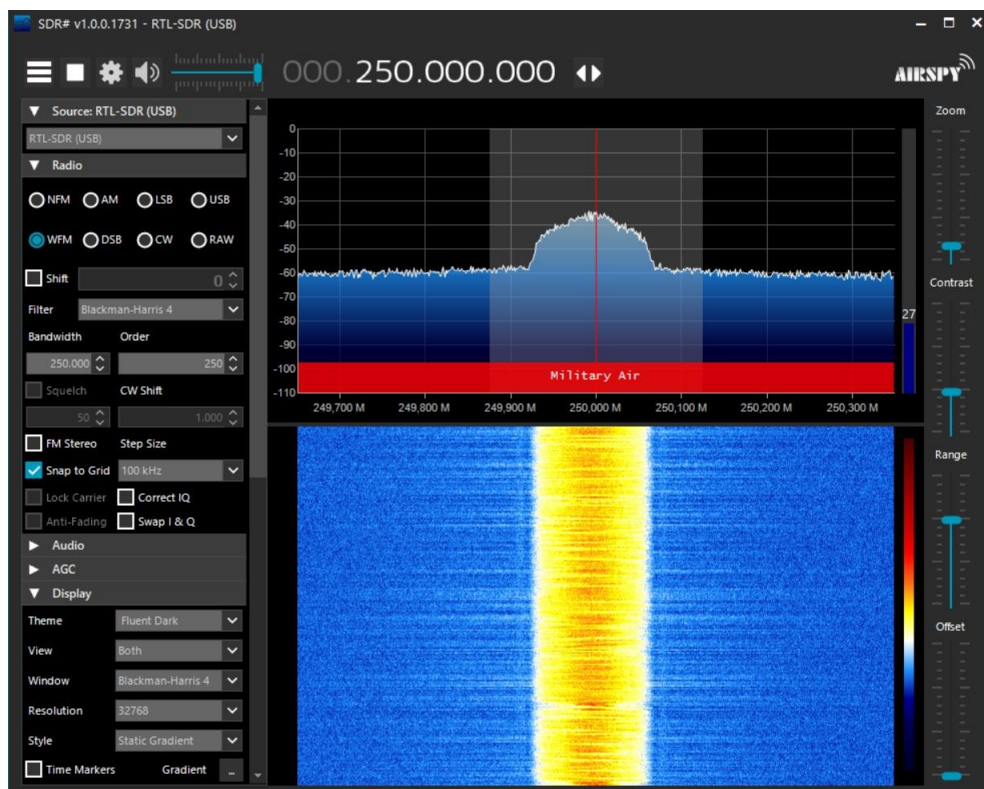


Figura 41 - Gráficos do Rx em 250 MHz



4 CONCLUSÕES E TRABALHOS FUTUROS

No presente século, é totalmente perceptível a significativa evolução não somente das Telecomunicações, mas um crescimento simultâneo no que diz respeito às demais áreas de abrangência da tecnologia. Em um curto espaço de tempo, a possibilidade de galgar passos mais longos tem se tornado plenamente concreta. A partir do advento da Indústria 4.0, percebeu-se que deve haver um crescimento síncrono entre os diversos ramos de atuação, pelo fato de não mais subsistirem por si próprios, mas também dependerem de outros para que seu desenvolvimento seja pleno. Portanto, a convergência das tecnologias é o ponto chave que garante uma promessa de longevidade no tocante ao crescimento do mercado tecnológico e a criação de novos produtos, pois permite a possibilidade de que haja maior abrangência na percepção dos problemas que enfrentamos no presente e as soluções que porventura sejam desenvolvidas. Já não é mais bem-vinda a ideia de que hardware e software podem trabalhar separadamente, mas que um complementa o outro. Dessa forma, é essencial que trabalhem em conjunto no surgimento de novas aplicações.

A demanda por serviços de Telecomunicações aumentou significativamente ao longo dessas duas décadas. Há alguns anos atrás, por exemplo, o uso da Internet cresceu de forma exponencial com o advento das novas tecnologias, de forma que no ano de 2018 estimou-se que 51,2% da população mundial (cerca de 3,9 bilhões de pessoas) já possuíam acesso à Internet;⁷² o aumento do consumo de celulares permitiu que houvesse também um aumento na taxa de assinantes de serviço de telefonia móvel banda larga, ao ponto que aproximadamente 69% da população mundial (cerca de 5,2 bilhões de pessoas) faziam parte dessa estatística até 2018.⁷³

Um dos fatores que permitiu esse aumento consiste justamente na convergência entre redes IP e telefonia móvel a partir do 3G e 4G. Semelhantemente, essa convergência tecnológica alcança o universo dos demais sistemas de telecomunicações, principalmente no que tange aos SDR's. Seria impossível a concepção de um SDR sem que outros tipos de tecnologia (como DSP's, FPGA's, além dos demais circuitos integrados e softwares para implementação de sistemas) não tivessem uma participação direta em sua constituição. A importância do desenvolvimento de cada tecnologia em particular caminha para um objetivo final de uma utilização conjunta – o que explica a necessidade urgente da compatibilidade entre diversos sistemas. Vimos também que o SDR representa o futuro das Telecomunicações, não somente

⁷² <https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx>

⁷³ Ibid.

devido à convergência que o cerca, mas pelo que já se é possível alcançar com a sua utilização. O surgimento de tecnologias emergentes como o 5G, a Internet das Coisas e as próprias redes de sensores sem fio prometem ainda levar o SDR a um patamar de alta magnitude daqui a alguns anos.

Um outro ponto de importância imprescindível é a utilização da faixa de frequência em 250 MHz no Brasil e o que concerne toda a Resolução n° 555 da ANATEL. Ao decorrer do trabalho, foram vistas as qualidades e vantagens que permitem um amplo resultado qualitativo no tocante ao uso dessa região do espectro brasileiro. Entretanto, atualmente há pouco desenvolvimento de tecnologias que ocupem essa faixa no Brasil. Portanto, é mister que haja a disseminação da importância da Resolução n° 555 da ANATEL e a utilização dessa faixa do espectro de forma que mais tecnologias possam ser desenvolvidas e, assim, não haja uma redistribuição da faixa para outros tipos de serviços, o que seria totalmente indesejável para aplicações como a deste trabalho.

Os próximos passos para que o projeto alcance a estatura do protótipo ideal giram em torno da decisão de qual tecnologia de banda base seria utilizada para atuar no protótipo. A tecnologia escolhida deve, portanto, possuir alguma forma de ser reproduzida através do software de processamento (seja o próprio *GNU Radio* ou outro tipo) para que, assim como realizado no experimento real, seja possível o interfaceamento com o dispositivo SDR. Vale ressaltar que algumas mudanças precisam ser feitas, a depender do padrão adotado pela tecnologia utilizada, como por exemplo a canalização e a largura de banda, as quais devem obedecer à Resolução 555 da ANATEL. Trabalhos semelhantes, os quais desenvolvem seus transceptores através de tecnologia *Zigbee* (IEEE 802.15.4) e *Wi-Fi* (IEEE 802.11 a/g/p) utilizando o *GNU Radio* e outros tipos de SDR nos mostraram que a possibilidade é bastante viável.⁷⁴ Entretanto, a dificuldade reside em realizar os testes a partir do LimeSDR, pois não é trivial a comunicação entre duas placas LimeSDR-Mini.

Sabe-se que toda tecnologia a ser utilizada para a constituição de um sistema de telecomunicações precisa ser devidamente licenciada para que sua operação aconteça dentro dos termos da legislação vigente. Portanto, o sistema está sujeito a mudanças no tocante à escolha de SDR a ser utilizado futuramente.

Por fim, conclui-se o alcance dos objetivos propostos com a realização deste trabalho, a saber, a apresentação de um protótipo de sistema de radiocomunicação utilizando a tecnologia SDR e que esteja sujeito à regulamentação vigente para operar na faixa de 250 MHz de acordo

⁷⁴ <https://www.wime-project.net/>

com a Resolução n° 555 da ANATEL, demonstrando teoricamente a partir de uma arquitetura e visão sistêmica ideais sua infraestrutura simplificada e a partir do experimento realizado com seus componentes mais primordiais a viabilidade, flexibilidade de operação, apelo de baixo custo e facilidade de programação via software que a arquitetura possibilita, com uma interface amigável e de uso relativamente simples e prático, permitindo um abrangente uso para sistemas IoT.

REFERÊNCIAS

- [1] DILLINGER, M.; MADANI, K.; ALONISTIOTI, N. **Software Defined Radio: architectures, systems and functions**. 1^a. ed. Chichester: John Wiley & Sons, 2003.
- [2] XIONG, X. et al. Low Power Wide Area Machine-to-Machine Networks: key techniques and prototype. **IEEE Communications Magazine**, v. 53. Setembro de 2015. 64-71.
- [3] RAZA, U. et al. Low Power Wide Area Networks: An Overview. **IEEE Communications Surveys & Tutorials**, v. 19. Janeiro de 2017. 855 – 873.
- [4] HAYKIN, S. **Sistemas de Comunicação Analógicos e Digitais**. 4^a. ed. São Paulo: Bookman, 2004.
- [5] AUGUSTIN, A. et al. A Study of Lora: Long Range & Low Power Networks for the Internet of Things. **MDPI Sensors**, v. 16. Setembro de 2016. 1466 – 1484.
- [6] SELVA, A. F. B. et al. Introduction to the Software-defined Radio Approach. **IEEE Latin America Transactions**, Janeiro de 2012. 1156 – 1161.
- [7] COLLINS, T. F. et al. **Software-Defined Radio for Engineers**. Artech House, 2018.
- [8] ARSLAN, H. **Cognitive Radio, Software Defined Radio and Adaptive Wireless Systems**. Tampa: Springer, 2007.
- [9] SOARES, J. K. A. **Rádio definido por Software: Pesquisas e verificação de testes em uma plataforma livre**. Revista Científica Multidisciplinar Núcleo do Conhecimento. Ano 1. Vol. 8. pp. 228-244. Setembro de 2016. ISSN. 2448-0959.
- [10] MAZZER, D. **Rádio Cognitivo Embarcado para Internet das Coisas**. Dissertação (Dissertação em Telecomunicações) – Inatel. Minas Gerais, p. 36. 2016.
- [11] LATHI, B. P. **Sinais e Sistemas Lineares**. 2^a. ed. São Paulo: Bookman, 2007.
- [12] OLIVEIRA, C. H. R. et al. IoT radio in 250 MHz. **ICWN'17 – The 16th International Conference on Wireless Networks**, p.76 – 80. ISBN: 1-60132-462-6.