

# Plataforma baseada em SDR com Baixo Custo e Comunicação Remota via TCP

Helder E. Sales, André A. Mariano, Evelio M. G. Fernández

**Resumo**—Neste artigo uma plataforma de desenvolvimento sem fio é proposta baseada no protocolo TCP (*Transport Control Protocol*) para realizar experimentos com comunicação via rádio utilizando um equipamento SDR (*Software Defined Radio*), um SBC (*Single Board Computer*) Raspberry Pi e um RTL-SDR. O objetivo da plataforma é deixar o sistema simples e fácil de usar através da rede internet, tornando mais acessível a implantação para a realização de experimentos. Um estudo de caso foi realizado com dois experimentos implementados no GNU Radio. Foram implementados com sucesso dois tipos de comunicação, utilizando modulações NBFM e AM para enviar e receber sinais de áudio utilizando a plataforma desenvolvida.

**Palavras-Chave**—GNU Radio, RTL-SDR, TCP, Comunicação Sem Fio, Rádio Definido Por Software, SDR

**Abstract**—In this paper, a wireless development platform is proposed based on TCP protocol to conduct experiments with radio communication using SDR equipment, a Raspberry Pi SBC (single board computer) and an RTL-SDR. The objective of the platform is to make the system simple and easy to use through the internet, making the deployment more accessible to perform experiments. A study case was done with two experiments implemented in GNU Radio. It was successfully implemented two types of communication, using NBFM and AM modulations to transmit and receive audio signals through the developed platform.

**Keywords**—GNU Radio, RTL-SDR, TCP, Wireless Communication, Software Defined Radio, SDR

## I. INTRODUÇÃO

As comunicações sem fio estão em constante evolução, com novos desafios a serem solucionados e muitas novas oportunidades de desenvolver e testar novos tipos de tecnologias. Com a implantação dos sistemas 5G e a ascensão da Internet das Coisas Industrial (IIoT) [1], é possível desenvolver várias aplicações como: manutenção e monitoramento da agricultura [2]; comunicações veículo-para-tudo (V2X) para controle de tráfego [3]; iluminação e segurança em cidades inteligentes [4]; e comunicações M2M para automação, monitoramento entre outras aplicações. Devido aos vários padrões de comunicações sem fio, experimentos tendem a ficar custosos pelo fato de os equipamentos de rádio serem feitos para um propósito específico, sendo necessário a aquisição de vários equipamentos para poder operar em várias frequências ou modulações diferentes.

Os avanços tecnológicos atuais estão melhorando as formas de realizar experimentos com dispositivos sem fio, acompanhado do desenvolvimento do rádio definido por software (SDR), hardwares especializados em que é possível modificar

vários parâmetros como frequência, taxa de amostragem, tipos de modulação, entre outros [5]. Também houve progressos na forma de interagir com o hardware, através de softwares especializados que permitem construir sistemas de diagrama de blocos para especificar como que os dispositivos de rádio devem se comportar, além de simplificar e tornar mais intuitivo o processamento de sinais, sendo o GNU Radio um exemplo de software [6].

Existem vários trabalhos que usam o dispositivo de rádio definido por software USRP (*Universal Software Radio Peripheral*) para analisar sinais de radiofrequência (RF), como em [7] para detectar o espectro de energia emulando transceptores 16-QAM e 64-QAM e analisar os diferentes impactos de métodos de janelamento de filtros; analisar e gerar sinais de geolocalização por satélite (GNSS) de forma a validar a aquisição de sinais de posicionamento [8][9]; obter energia através de sinais RF para minimizar a dependência de baterias em dispositivos de baixo consumo [10]; detecção de padrões de respiração para ajudar em aplicações como a previsão do COVID-19, muito importante nesses tempos de pandemia [11]. Apesar da flexibilidade e utilidade do USRP em testar novos conceitos e tecnologias, estes equipamentos de rádio são caros [5], dificultando o acesso ao público geral.

Com a necessidade de ter um hardware específico para cada aplicação e faixa de frequências de operação, pesquisadores conduziram estudos para desenvolver plataformas SDR de baixo custo, dos quais alguns trabalhos utilizam o RTL-SDR ou derivados como o principal hardware RF [5][12][13][14][15][16], mas com restrições tal como a incapacidade de transmitir dados. Outros autores fazem uso de SDRs que têm capacidade de transmissão, sendo controlados por um Raspberry Pi de maneira a facilitar a configuração do sistema [17]. Entretanto, em todos os casos há a necessidade da plataforma ser operada localmente sem ter um meio de acesso remoto.

Assim, com base no exposto relativo às limitações em ter que operar as plataformas SDR localmente, sem que haja um meio para acessá-la via rede, além do alto custo de equipamentos especializados, neste trabalho é proposto uma configuração de uma plataforma SDR de baixo custo que permite operar o sistema remotamente via protocolo de controle de transmissão (TCP). Ao usar um Raspberry Pi em conjunto com um RTL-SDR para a transmissão e recepção dos dados, também diminui-se o custo total, tornando a configuração mais acessível. Com uma rede local ou através da internet (supondo que o protocolo de internet (IP) esteja disponível externamente), pesquisadores podem experimentar com o SDR virtualmente em qualquer lugar, mesmo sem ter o hardware em mãos. Portanto, o problema do alto custo relacionado à

utilização do USRP é mitigado com o uso do Raspberry Pi em conjunto com o RTL-SDR e pode-se operar à distância através da utilização do protocolo de rede TCP.

O restante do artigo está organizado como segue. A Seção II apresenta os componentes de hardware e software usados para testar e desenvolver a plataforma proposta, provendo uma visão geral das funcionalidades básicas e das peças-chave relevantes. Na Seção III, dois casos de teste são apresentados utilizando modulação de frequência de banda estreita (NBFM) e modulação de amplitude (AM) para transmitir o sinal via TCP no Raspberry Pi, recebendo estes no RTL-SDR através do GNU Radio, também por TCP. Na Seção IV são apresentadas as conclusões obtidas neste trabalho.

## II. DETALHAMENTO DA PLATAFORMA

A configuração proposta está explicitada na Figura 1, cujo sistema consiste num Raspberry Pi com um RTL-SDR conectado de forma a receber sinais RF e manipulá-los através do GNU Radio, em cujo software é possível analisar e processar uma gama diversa de frequências e modulações [6]. Além disso, o Raspberry Pi em si é capaz de transmitir sinais de RF através de seus pinos de entrada/saída (GPIOs) emulando um SDR [18]. Consequentemente, a plataforma pode ser tratada como um transceptor que com a ajuda do protocolo TCP [19] pode ser utilizada como uma plataforma SDR móvel e portátil.

Um jumper é conectado ao Raspberry Pi atuando como uma antena conectada ao GPIO 4, suportando transmissões entre 5 kHz e 1500 MHz aproveitando as capacidades de modulação de largura de pulso (PWM) desse dispositivo. Adicionalmente, um RTL-SDR conectado a uma das portas USB do Raspberry Pi possibilita a recepção de sinais entre 500 kHz e 1766 MHz.

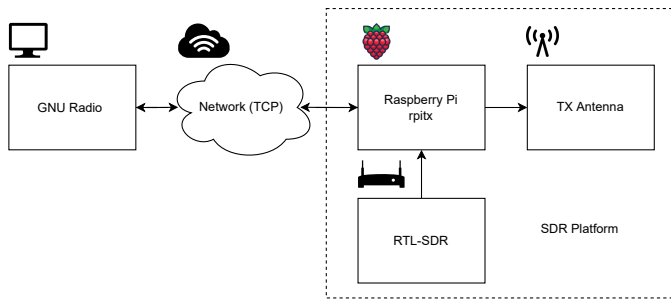


Fig. 1

DIAGRAMA DE BLOCOS DA CONFIGURAÇÃO PROPOSTA.

### A. Hardware

1) *Cliente*: O cliente é o computador principal por trás da interação entre o GNU Radio e o Raspberry Pi/RTL-SDR, que executa o sistema operacional Ubuntu. A escolha da distribuição Ubuntu do Linux dá-se ao fato de ser uma plataforma estável e com suporte de longo termo, além de uma grande comunidade de apoio [20].

2) *RTL-SDR*: O RTL-SDR é um dispositivo de rádio definido por software de baixo custo que pode receber sinais entre 500 kHz e 1766 MHz com uma taxa de amostragem de até 2,4 MSps. É composto de um conversor analógico-digital Realtek RTL2832U e um sintonizador Rafael Micro RT820T2.

### B. Servidor

O servidor consiste num Raspberry Pi 3B+, que é um computador de placa única projetado principalmente para ajudar pessoas ao redor do mundo aprenderem computação e criações digitais [21]. Possui um processador de quatro núcleos ARM Cortex-A53 system-on-chip (SoC) rodando a 1.4 GHz (Broadcom BCM2837B0) e 1 GB de memória RAM LPDDR3. A interface de usuário é feita através dos pinos de entrada/saída e os vários periféricos presentes na placa. Também, o Raspberry Pi possui uma rede WiFi de banda dupla (2.4 GHz and 5 GHz) e uma conexão Bluetooth 4.2 ou BLE (Bluetooth Low-Energy) para lidar com diversas aplicações sem fio.

O sistema operacional principal usado neste computador é o Raspberry Pi OS (anteriormente Raspbian), uma distribuição Linux baseada em Debian otimizada para o hardware com arquitetura ARM presente no dispositivo.

### C. Software

1) *GNU Radio*: O GNU Radio é um software de código livre desenhado para construir diagramas de bloco e experimentar com processamento de sinais. O usuário cria e simula diferentes tipos de sistemas de comunicação e os integra com dispositivos SDR ou hardwares externos, como o RTL-SDR e o Raspberry Pi [6].

2) *rpitx*: O rpitx [18] é um software feito para fins educacionais que transforma o Raspberry Pi em um SDR transmissor portátil (mostrado na Figura 1), capaz de alcançar uma faixa de 5 kHz à 1.5 GHz modulando os sinais pelo GPIO [18]. Ao ser iniciado, o software rpitx aguarda o recebimento de amostras não processadas dos sinais de fase e quadratura ( $I/Q$ ) que podem ser geradas por diferentes métodos ou programas (como por exemplo sinais gerados pelo GNU Radio), de forma que o Raspberry Pi realize a transmissão RF dos dados.

3) *rtl\_tcp*: O rtl\_tcp é um pacote do Linux que permite enviar dados do RTL-SDR para qualquer dispositivos através de soquetes TCP [22], podendo ser utilizado o GNU Radio para receber os dados.

4) *gr-gnnet*: Este pacote é um bloco do GNU Radio que provê ferramentas de rede tais como TCP e UDP [23]. É um portal para trafegar os sinais gerados pelos blocos do GNU Radio até a rede e também obtê-los desta. Exemplos de funcionamento são mostrados na Subseção III-A e Subseção III-B.

O bloco TCP padrão embutido no GNU Radio 3.8 não é estável o suficiente, por isto fez-se necessário o uso deste bloco (também chamado de *out-of-tree* (OOT)) que é mantido separadamente no GitHub [22].

## III. ESTUDOS DE CASO

Nesta seção serão apresentados dois estudos de caso em que um sinal de áudio será enviado através de uma portadora centrada em 48 MHz utilizando modulações NBFM e AM, respectivamente. Os sinais serão recebidos no próprio transmissor através de uma interface de *loopback* com o RTL-SDR.

A lógica do transmissor é executada no sistema operacional Ubuntu no cliente, enviando os dados para a rede através do TCP, enquanto que o Raspberry Pi recebe estes dados e os

processa localmente. Para isto, no Raspberry Pi executa-se o seguinte comando no terminal do Linux:

```
$ nc -l <porta> | sudo rpitx -i- -m IQFLOAT \
-f <frequencia_khz>
```

que estabelecerá uma conexão TCP com o IP do servidor com a porta especificada no campo <porta> (de 0 a 65535), a qual irá receber as amostras I/Q e as modulará na frequência estabelecida pelo campo <frequencia\_khz>, que aceita entradas de 5 à 1500000 kHz.

Em seguida, usando um jumper conectado ao GPIO 4 como antena, o Raspberry Pi transmite o sinal resultante da modulação dessas amostras, recebe este mesmo sinal através do RTL-SDR em um loop e o envia via TCP de volta para o Ubuntu.

Para receber os sinais de rádio e enviá-los novamente para o cliente, abre-se uma nova instância do terminal no Raspberry Pi e insere-se o seguinte comando:

```
$ rtl_tcp -a <ip> -p <porta>
```

A partir destes dois comandos, basta executar o GNU Radio no computador cliente e utilizar o bloco *TCP Sink* para enviar as amostras I/Q para o Raspberry Pi pela rede e utilizar o bloco *RTL-SDR Source* para recebê-las, passando o endereço IP com a porta TCP no parâmetro *Device Arguments*.

Uma fotografia da configuração contendo o Raspberry Pi e o RTL-SDR é mostrada na Figura 2.



Fig. 2  
FOTOGRAFIA DA CONFIGURAÇÃO UTILIZADA.

Existem limitações quanto à capacidade de transmissão do dispositivo, pois o Raspberry Pi não possui nenhum *front-end* dedicado à transmissão de sinais RF, que inclui filtros, amplificadores e pinos analógicos, sendo assim esperado que os sinais possam conter harmônicas e problemas de atenuação. Somente a recepção dos sinais envolvendo o RTL-SDR passa

por alguma filtragem e amplificação, pelo fato do RTL-SDR ser desenvolvido especificamente para este fim.

A. Modulação de Frequência de Banda Estreita (NBFM)

A modulação FM de banda estreita foi realizada de acordo com o diagrama de blocos mostrado na Figura 3. A fonte de sinal utilizada é um arquivo de áudio (wav) com uma taxa de transmissão de 48 kHz que é modulado em frequência pelo bloco transmissor NBFM. Em seguida, o sinal (complexo) modulado passa pelo bloco TCP, responsável por encaminhar as amostras I/Q pela rede até o Raspberry Pi.

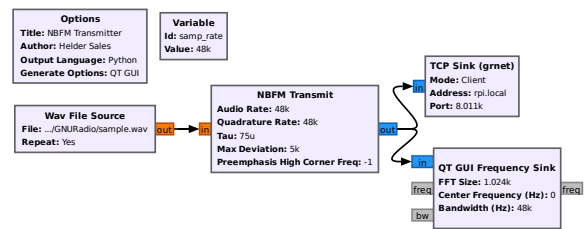


Fig. 3  
FLUXOGRAMA DO TRANSMISSOR NBFM NO GNU RADIO.

A Figura 4 mostra o espectro de frequências do sinal NBFM que está sendo enviado ao Raspberry Pi. Este por sua vez processa os sinais I/Q transmitindo o sinal de RF resultante na frequência de 48 MHz (escolhida arbitrariamente, através do campo <frequencia\_khz> conforme mostrado no primeiro comando da Seção III).

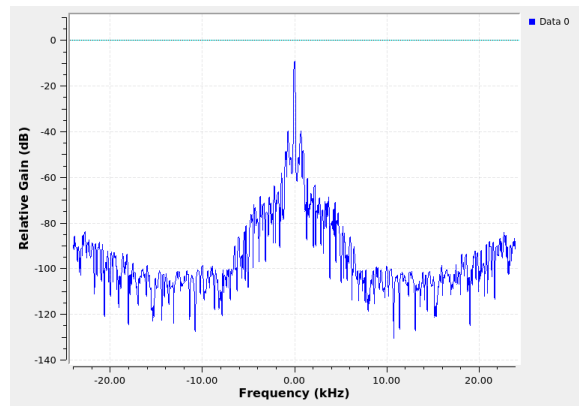


Fig. 4  
ESPECTRO DE FREQUÊNCIAS DO SINAL TRANSMITIDO.

No lado do receptor, o GNU Radio processa os dados recebidos do Raspberry Pi de acordo com o diagrama em blocos mostrado na Figura 5, onde o bloco *RTL-SDR Source* está sintonizado na frequência de 48 MHz.

Para demodular e reproduzir o áudio enviado pelo transmissor, os dados brutos obtidos a partir do RTL-SDR são processado pelo filtro passa-baixas, decodificado e multiplicado por uma constante para controlar o volume. Finalmente, o áudio é enviado para a placa de som do computador a uma taxa de amostragem de 48 kHz para ser reproduzido pelo sistema.

A Figura 6 mostra o sinal NBFM recebido depois do filtro passa-baixas. A transmissão foi recebida conforme esperado, com o sinal sonoro estando audível e sem muitos ruídos perceptíveis, demonstrando a capacidade do Raspberry Pi de transmitir os sinais através de seus pinos de entrada/saída.

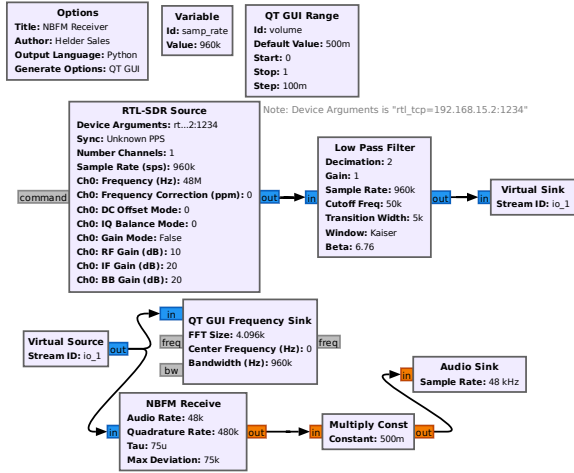


Fig. 5

FLUXOGRAMA DO RECEPTOR NBFM NO GNU RADIO.

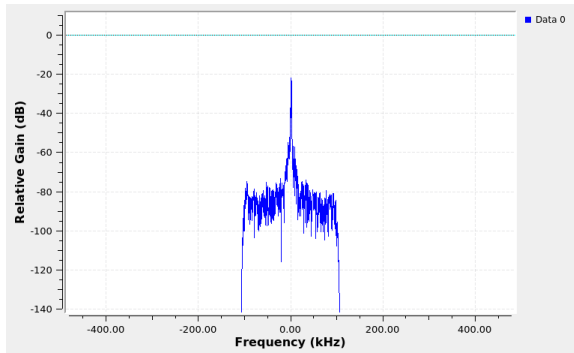


Fig. 6

ESPECTRO DE FREQUÊNCIA DO SINAL RECEBIDO.

**B. Modulação de Amplitude (AM)**

A Figura 7 mostra o diagrama em blocos implementado para o experimento com modulação de amplitude. Neste caso, o mesmo bloco fonte de áudio da Subseção III-A foi utilizado convertido para amostras complexas e processados com controle de ganho automático (AGC). O AGC garante que a intensidade de sinal não irá variar a extremos, provendo um volume de som estável. Em seguida o sinal é passado através do filtro passa-faixa e do bloco TCP para ser enviado ao Raspberry Pi, para que a transmissão do sinal AM seja realizada.

A Figura 8 mostra o espectro de frequências do equivalente em banda base do sinal AM a ser transmitido. As amostras I/Q são então encaminhadas para o Raspberry Pi via TCP para serem transmitidas pela frequência portadora de 48 MHz.

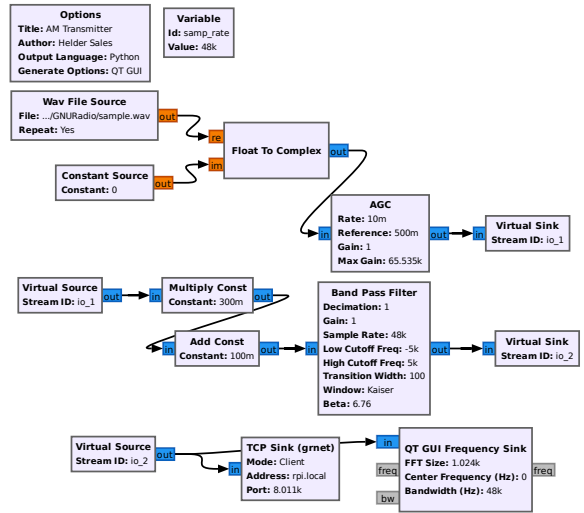


Fig. 7

FLUXOGRAMA DO TRANSMISSOR AM NO GNU RADIO.

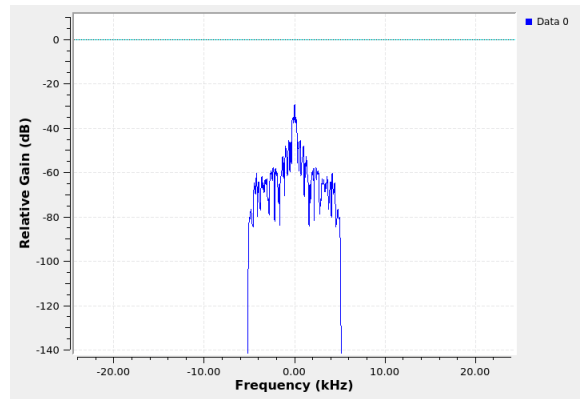


Fig. 8

ESPECTRO DE FREQUÊNCIA DO SINAL TRANSMITIDO.

A Figura 9 mostra o diagrama em blocos do receptor onde o RTL-SDR conectado ao Raspberry Pi está sintonizado na frequência de 48 MHz. O sinal complexo na saída do RTL-SDR é filtrado e decimado por um fator de 10 pelo filtro passa-baixas. Em seguida, o bloco demodulador AM recupera o sinal de áudio cuja taxa de amostragem será fixada em 48 kHz pelo bloco re-amostrador racional para poder ser na sequência reproduzido pela placa de som do computador.

Devido à falta de pinos analógicos no Raspberry Pi, além de filtros e amplificadores, a modulação de amplitude recebida fica sujeita a um nível um pouco maior de ruído, do qual os impactos são sentidos ao se ouvir o áudio recebido.

O Raspberry Pi possui apenas pinos que alternam entre dois estados: 0 V e 3,3 V, portanto para emular as ondas portadoras da modulação AM, faz-se o uso da modulação de largura de pulso (PWM) e varia-se seu ciclo de trabalho. Como a forma de onda de um PWM é quadrada, há componentes harmônicas que também são transmitidas, aumentando assim o nível de ruído na transmissão e tendo impacto direto na recepção.

A Figura 10 mostra o espectro de frequência equivalente em banda base do sinal recebido pelo RTL-SDR.

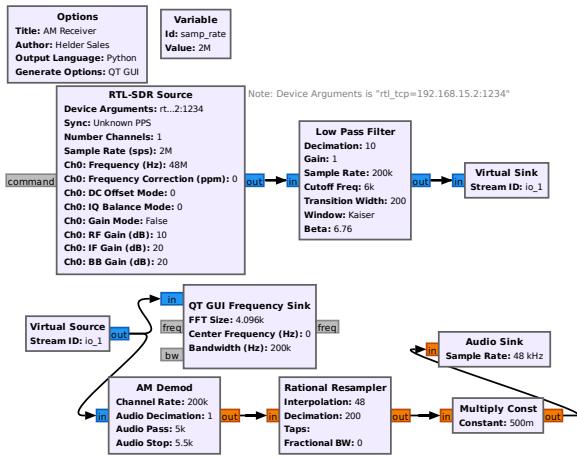


Fig. 9

FLUXOGRAMA DO RECEPTOR AM NO GNU RADIO.

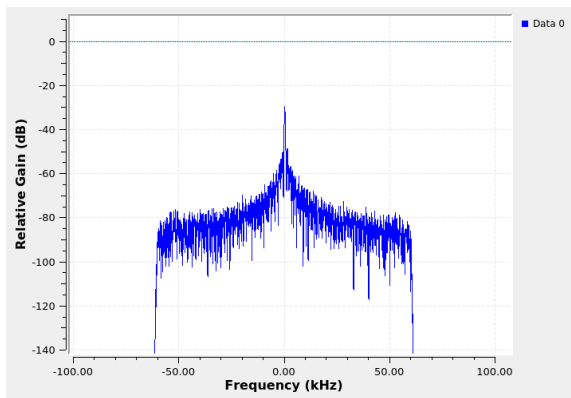


Fig. 10

ESPECTRO DE FREQUÊNCIA DO SINAL RECEBIDO.

#### IV. CONCLUSÃO

Neste trabalho foi mostrado que com ajuda do protocolo TCP, é possível fazer processamento de sinais bem como enviar/receber sinais remotamente utilizando um Raspberry Pi, um RTL-SDR, e o software GNU Radio, como mostrado através da transmissão de áudio com modulações NBFM e AM.

A modulação AM teve menor qualidade que a NBFM devido ao fato do Raspberry Pi não ter o *front-end* dedicado à transmissão de sinais de radiofrequência, além de que a forma com que é emulada a variação de amplitude da onda dá-se pela variação do ciclo de trabalho da modulação PWM, enquanto que no sinal NBFM apenas a variação da frequência é necessária, tendo um impacto menor na qualidade.

Além do baixo custo do equipamento, a possibilidade de trabalhar remotamente utilizando TCP aumenta consideravelmente a mobilidade dos usuários, em adição a experimentar

novas tecnologias sem precisar ter o hardware em mãos, sendo útil para profissionais ligados à indústria e também para universidades nas quais os alunos podem controlar e observar os resultados dos experimentos sem estar fisicamente presente na sala de aula.

#### REFERÊNCIAS

- [1] P. Varga, J. Peto, A. Franko, D. Balla, D. Haja, F. Janky, G. Soos, D. Ficzer, M. Maliosz, and L. Toka, "5g support for industrial iot applications – challenges, solutions, and research gaps," *Sensors (Switzerland)*, vol. 20, 2 2020.
- [2] Y. Tang, S. Dananjayan, C. Hou, Q. Guo, S. Luo, and Y. He, "A survey on the 5g network and its impact on agriculture: Challenges and opportunities," *Computers and Electronics in Agriculture*, vol. 180, 1 2021.
- [3] C. R. Storck and F. Duarte-Figueiredo, "A 5g v2x ecosystem providing internet of vehicles," *Sensors (Switzerland)*, vol. 19, 2 2019.
- [4] M. Bauer, L. Sanchez, and J. Song, "Iot-enabled smart cities: Evolution and outlook," *Sensors*, vol. 21, 7 2021.
- [5] J. Francisco, N. Pinheiro, E. Márcio, S. D. Santana, and T. F. Maciel, "Raspbed: a low cost experimentation platform based on raspberry pi for wireless communication studies," 2019.
- [6] G. Radio. (2022) About gnu radio. [Online]. Available: <https://www.gnuradio.org/about/>
- [7] R. D. Boddu, "Experimental validation of spectrum sensing on various transceivers using software defined radio," *Wireless Personal Communications*, vol. 109, pp. 1615–1630, 12 2019.
- [8] C. Diouf, G. J. Janssen, H. Dun, T. Kazaz, and C. C. Tiberius, "A usrp-based testbed for wideband ranging and positioning signal acquisition," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, 2021.
- [9] Y. Hu, "Gnss sdr signal generator implementation based on usrp n210," vol. 1314. Institute of Physics Publishing, 11 2019.
- [10] S. Gautam, S. Kumar, S. Chatzinotas, and B. Ottersten, "Experimental evaluation of rf waveform designs for wireless power transfer using software defined radio," *IEEE Access*, vol. 9, pp. 132 609–132 622, 2021.
- [11] M. Rehman, R. A. Shah, M. B. Khan, N. A. Abuali, S. A. Shah, X. Yang, A. Alomainy, M. A. Imran, and Q. H. Abbasi, "Rf sensing based breathing patterns detection leveraging usrp devices," *Sensors*, vol. 21, 6 2021.
- [12] M. B. Perotoni, F. A. A. Silva, and M. S. Vieira, "Radio-propagation measurement based on a low-cost software defined radio," *Progress In Electromagnetics Research Letters*, vol. 101, pp. 1–10, 2021.
- [13] D. P. Wright and E. A. Ball, "Highly portable, low-cost sdr instrument for rf propagation studies," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, pp. 5446–5457, 8 2020.
- [14] I. of Electrical, E. E. B. Section, I. of Electrical, E. Engineers, Design, and M. J. I. I. of Information Technology, *Energy Detection Spectrum Sensing on RTL-SDR based IoT Platform*. Institute of Electrical and Electronics Engineers Inc., 2018.
- [15] R. Helbet, P. Bechet, V. Monda, S. Miclaus, and I. Bouleanu, "Low-cost sensor based on sdr platforms for tetra signals monitoring," *Sensors*, vol. 21, 5 2021.
- [16] R. Danyamol, T. Ajitha, and R. Gandhiraj, "Real-time communication system design using rtl-sdr and raspberry pi." Institute of Electrical and Electronics Engineers Inc., 10 2014.
- [17] C. P. M. Gowda, P. Prajwal, K. V. Bellad, V. Hima, and N. Suresh, "Design implementation of fm transceiver using raspberry pi and sdr." Institute of Electrical and Electronics Engineers Inc., 2021, pp. 156–161.
- [18] F5OEO. (2022) F5oeo/rpitx. [Online]. Available: <https://github.com/F5OEO/rpitx>
- [19] J. Postel. (1981) Transmission control protocol darpa internet program protocol specification. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc793>
- [20] Canonical. (2022). [Online]. Available: <https://ubuntu.com/>
- [21] R. P. Foundation. About us. [Online]. Available: <https://www.raspberrypi.org/about/>
- [22] Canonical. (2019) rtl\_tcp - an i/q spectrum server for rtl2832 based dvb-t receivers. [Online]. Available: [http://manpages.ubuntu.com/manpages/trusty/man1/rtl\\_tcp.1.html](http://manpages.ubuntu.com/manpages/trusty/man1/rtl_tcp.1.html)
- [23] ghostop14. (2022) ghostop14/gr-gnet. [Online]. Available: <https://github.com/ghostop14/gr-gnet>