Creating a Software Radio Spectrum Analyzer for IoT Healthcare Applications

Namarig Mohemed Taha[1,2], Iyad Dayoub[1,3], Elhadj Dogheche[1,3] Amin Babiker Abdelnabi[2] and Ahmed.M.Alhassan[2]

[1]Universite. Polytechnique Hauts-de-France, Institut d’Electronique de Microlectronique et de Nanotechnologie (IEMN), CNRS, UMR 8520, ISEN, Centrale Lille, University of Lille, 59313 Valenciennes, France.
[2]Department of Communication Engineering, Faculty of Engineering, Al-Neelain University, Sudan.
[3]INSA Hauts de France, F-59313 Valenciennes, France.

Abstract

This study investigates a spectrum analyzer simulation prototype using software-defined radio (SDR) technology. This technique continues to grow rapidly and has motivated researchers to develop it. The experiment aims to determine whether the SDR can pick up a signal at a frequency within the 2.4 GHz range to ensure the right channel and then employ this measurement system for IoT healthcare applications. In addition, we programmed all the capabilities of the spectrum analyzer

Key words: SDR, Spectrum Analyzer, GUI.

Introduction

Software Defined Radio (SDR) is a wireless communication system that uses software to process different signals. The SDR design objective is to produce a radio capable of transmitting and receiving a new protocol form once the program runs [1]. In the future, it is expected that SDR technology will dominate radio communications. Software-defined radios are handy for mobile phone services, serving a wide variety of radio protocols that change in real-time

SDR Hardware Components

This hardware comprises mixers, amplifiers, modulators, demodulators, and filters. SDR uses only an ADC and DAC with antennas without requiring multiple hardware components to convert the signal from analogue to digital and digital to analogue. This, in turn, makes SDR more flexible in resolving problems since most of the processing is performed on the software rather than on the hardware. The software may operate on a PC, an integrated or embedded system. Figure 1 and figure 2 show the SDR transmitter and receiver

SDR Transmitter

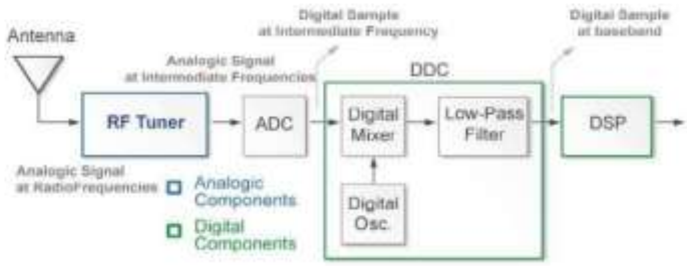

Figure1: SDR Transmitter

SDR Receiver

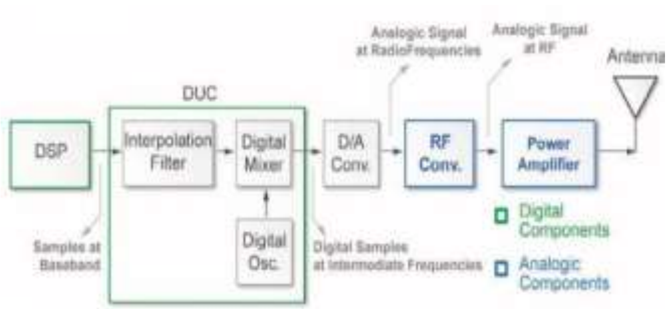

Figure 2: SDR Receiver

SDR APPLICATIONS

SDR is widely used for a various medical application, including the seamless interoperability of medical devices in a common platform, cognitive radio for BANs, and WSN for medical surveillance [2].
Different applications carried out using SDR, such as:

- Listen to EMS/Ambulance/Fire communications
- A police radio scanner.
- Scan wireless cable conversations
- Aircraft location tracking is like a radar with ADSB decoding [3].

Advantages of SDR Technologies

Flexibility: different communication systems can be implemented on the same device.

- Ability to adapt to the requirements.
- The ability to achieve high performance.
- Able to reconfigure radios using updating software.
- Reduce the deployment cost.

Disadvantages of SDR Technologies

- For simple radios, the primary platform may be expensive.
- SDR development requires both software and hardware skills

Experiment

The objectives of this experiment were to see if the SDR could pick up a signal and program all the spectrum analyzer's capabilities.

Figure 3: Experiment Setup

Software Environment and Discussion

Software List

-GNU Radio version 3.7.5
-WINDOWS 10 V20H2
-UHD version3.8.2

SDR Environment setup involves installing the firmware and appropriate software and hardware. The GNU is installed here as a programming software tool [4]. The hardware used in this experiment to transmit the signals is two USRP B200 and antennas from Ettus Research. GNU Radio and UHD are installed on Windows 10, but it works best when the system is running on Linux directly.

Hardware Environment and Discussion

HARDWARE LIST

- 2 USRP NI B200
- 2 Laptops Del
- 2 USB 3.0 cables
- 2 VERT 2450 Antennas from Ettus Research

Figure 4: USRP B200 Figure 5 : USB 3.0 cable Figure 6 : VERT 2450 Antenna

The frequency ranges of USRP B200 from 70 MHz to 6 GHz, and the programming languages are Python and C++. USB 3.0 is capable of bandwidth up to 56 MHz, with a 12-bit ADC/DAC and data rate of 64.11 MS/s [5].

System Functionality

In this system, we used GNU radio which uses GNU Radio Companion as the graphical user interface (GUI) to simulate and process the signal. In GNU Radio, there are signal processing blocks in a logical order called flow graphs. Each block represents different signal processes, such as BPSK and QPSK modulators. Blocks are coded using Python or C++. Custom blocks are coded and imported into GNU Companion. There may be multiple sinks and sources, but most programs are used one source and multiple sinks

Spectrum Analyzer Flow Graph

Figure 7 shows the flow graph for the spectrum analyzer into the SDR created using USRP B200.

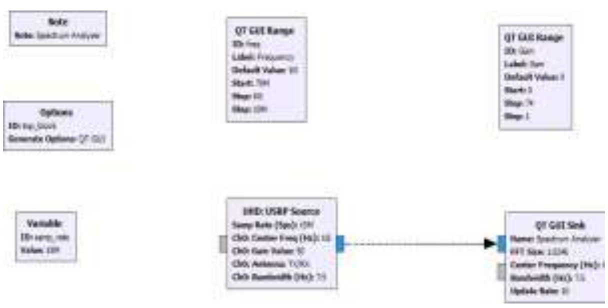

Figure 7**:** Spectrum Analyzer Flow graph

The flow graph has one source block and one sink. The source block receives data from USRP following the specified values. The sampling frequency is determined based on a variable sample frequency

The center of frequency is defined in the QTGUI box Range. This range is determined based on the start and stop values. In this experiment, the sampling rate is 15MHz, the central frequency is 1GHz, and the bandwidth is 7.8MHz, which means that the received frequency range will vary from 1.0078GHz to 0.9928GHz, where this range is within the Internet of Things domains. As a result, we can use SDR for IoT in the healthcare domain.

Parameters Setting

The Figures 8, 9 represent the parameters used in the experiment.

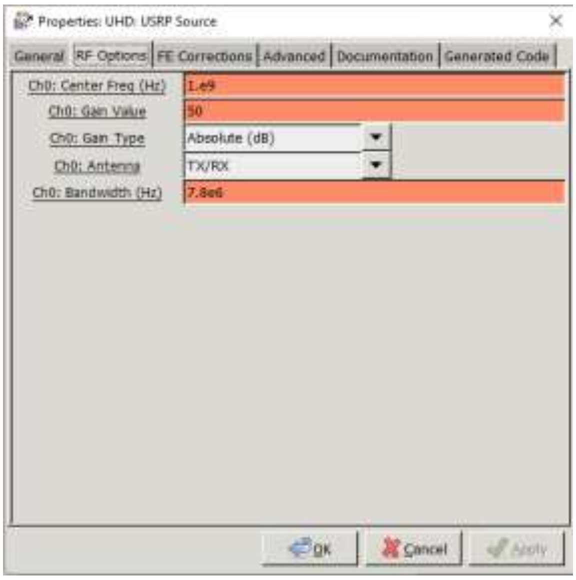

Figure 8: Radio frequency parameters

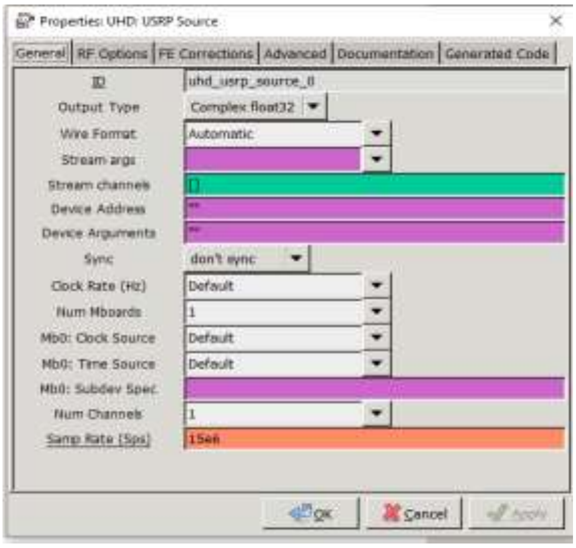

Figure 9: USRP source parameters

Table I: Setting of Parameters

The parameters	**value**
Center frequency	1G(Hz)
Bandwidth	7.8(Hz)
Sample Rate	15MHz

Spectrum Analyzer Results and Discussion

Figure 10 shows the Spectrum Analyzer GUI. The SDR is tuned at 1GHz. SDR gain has been set at zero, which means no further amplification. This chart shows the power (relative gain) to frequency. The signal picked up around -105 dB. By increasing the gain of USRP, the receiving sensitivity also increases, and we can sense if there are signals.

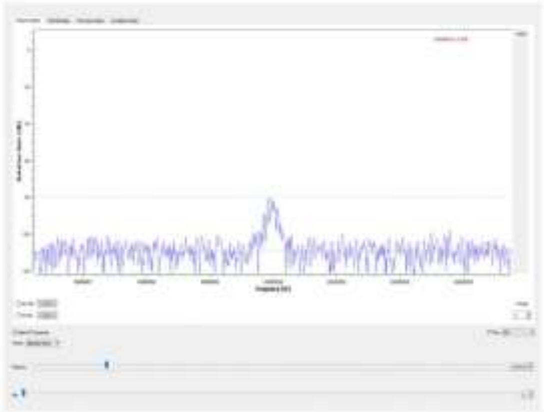

Figure 10**:** Spectrum Analyzer GUI Result

Conclusion

In this article, we determined different applications of SDR. Two main benefits were outlined, cognitive radio and device interoperability. We were implementing the SDR sensing of signals to know the interoperability of different medical devices. The result showed that SDR could sense the signal in the range of 2.4 GHz, which includes the IoT frequency range. This means that we can use SDR in hospital environments.

Acknowledgment

This research is supported by the Ministry of Higher Education and Research of the Republic of Sudan, Al-Neelain University, and the French government scholarship.

References

[1] Machado-Fernández, JoséRaúl."Software defined radio: Basic principles and applications. " Revista Facultad de Ingeniería 24, no. 38 (2015): 79-96.

[2] Chávez-Santiago R, Mateska A, Chomu K, Gavrilovska L, Balasingham I. Applications of software-defined radio (SDR) technology in hospital environments. In2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) 2013 Jul 3 (pp. 1266-1269). IEEE.

[3] Garg, Vijay K. "An overview of wireless systems." Wireless Communications Networking, Morgan Kaufmann, Ist edition. ISBN 123735807 (2007).

[4] http://sdrforum.org

[5]https://www.trendmicrocom/vinfo/pl/security/definition/sdr