Low cost spectrum monitoring system based on Dragonboard 410C and RTL-SDR 2832U dongle

Héctor Iván Reyes Moncayo, Leidy Alexandra Sarmiento Ortiz, Víctor Manuel Martínez Cuervo
Ingeniería eléctrica, Universidad de los Llanos, Villavicencio, Colombia
Correo-e: hreyes@unillanos.edu.co

***Resumen*— Mientras el espectro de radio se hace más escaso, monitorear sus niveles y patrones de uso se convierte en una herramienta importante para implementar medidas tecnológicas y regulatorias con el fin de gestionar más eficientemente este recurso. Estudios de ocupación del espectro han sido conducidos por periodos limitados de tiempo en diferentes lugares alrededor del mundo. El alto costo de los equipos usados para conducir dichos estudios ha dificultado la realización de mediciones de espectro en más lugares de una manera permanente. Este artículo propone un sistema de monitoreo de espectro de bajo costo basado en la tarjeta Dragonboard 410C, una plataforma embebida desarrollada por Qualcomm®;un módulo RTL-SDR 2832U, cuyo propósito original es ser un receptor de televisión digital; y Python. El prototipo funcionó como se esperaba durante mediciones de espectro tomadas en diferentes frecuencias.**

***Palabras clave*— Porcentaje de ocupación de canal, Python, Radio definida por software, Sensado de espectro.**

***Abstract*— As the radio spectrum gets scarcer, monitoring its usage levels and patterns becomes an important tool to implement technological and regulatory measures to manage this resource more efficiently. Spectrum occupancy surveys have been conducted for limited periods at different locations all over the world. The high cost of the equipment used to conduct those surveys has made it difficult to implement spectrum measurements in more places in a permanent fashion. This paper presents a low cost spectrum monitoring system based on a Dragonboard 410C, an embedded platform developed by Qalcomm®, an RTL-SDR 2832U, whose original purpose is being a digital TV receiver, and Python. The prototype performed as expected during spectrum measurements taken at different frequency bands.**

***Key Word* — Channel occupancy percentage, Python Software defined radio, spectrum sensing.**

I. INTRODUCTION

We are in the wireless age. We want to free ourselves from cables that limit our mobility. Satisfying this desire comes with a price: more radio spectrum. As the demand for radio spectrum increases every day, it becomes more critical to look for more efficient ways to administer this resource. Spectrum monitoring is a tool for gaining awareness about the spectrum utilization. This spectrum awareness is important to propose more efficient ways to administer this resource. Spectrum monitoring consists in taking measurements from the environment through an antenna to determine the occupancy level of specific portions of the spectrum. This information can be useful not only for regulators, but also for radio devices that access dynamically the spectrum. Knowing the spectrum utilization patterns, the agencies that regulate the use of this resource can propose new applications and services in underutilized portions of the spectrum. In the case of the devices that access the spectrum dynamically, known as cognitive radios, being aware about the spectrum utilization enables them to discover spectrum holes to exploit them opportunistically.

Most of the spectrum monitoring campaigns reported in the literature use expensive equipment and logistics. Those spectrum surveys were in specific locations and time windows to prove that the spectrum underutilization is a fact, and to justify the search for more efficient spectrum management frameworks. In order to exploit the spectrum usage information effectively, it needs to be collected permanently and in more places. Employing the same type equipment used in the aforementioned spectrum surveys would be very expensive; therefore, low cost solutions are necessary to monitor the spectrum constantly in more places.

In this paper, we present a low cost spectrum sensor based on software defined radio (SDR) technology. The SDR is implemented with a Dragonboard 410 C, Python, and a RTL-SDR dongle. The proposed approach can also work on other embedded systems able to run Python. It also has connectivity

Fecha de Recepción: 25 de septiembre de 2017
Fecha de Aceptación: 24 de febrero de 2018

to internet, so the collected data can be sent to a database for real time or historical analysis.

This paper is organized as follows. In section 2 we talk about work related to spectrum sensing, spectrum monitoring, and open source solutions for spectrum monitoring. In section 3 we provide details about the prototype proposed in this paper. In section 4 we explain the experiments and tests that we used to validate the prototype. In section 5 we present and discuss the test results. Finally, in section 6 we provide conclusions and mention the future work.

II. RELATED WORK

Spectrum monitoring has the purpose of scanning a predefined set of frequencies with the objective of knowing the channel occupancy rate, percentage of time the channel is occupied. The spectrum occupancy campaigns reported in the literature are based on measuring the power level in the channels of interest. Those power levels are compared with pre-established thresholds to detect whether the channel is either free or occupied; method known as energy detection (E.D) spectrum sensing. The equipment used during those campaigns includes spectrum analyzers connected to computers to record and analyze the measurements. This was the common denominator in spectrum surveys performed in the U.S. [1-3] , Germany [4], New Zealand [5], Singapore [6], and Spain [7].

The literature also reports other spectrum sensing techniques that unlike the E.D. method do not rely on measuring the power. The other types of detectors mentioned in the literature are the cyclostationary detector, the matched filter detector, and the covariance or autocorrelation based detector. Figure 1 compares these techniques in terms of complexity and accuracy.

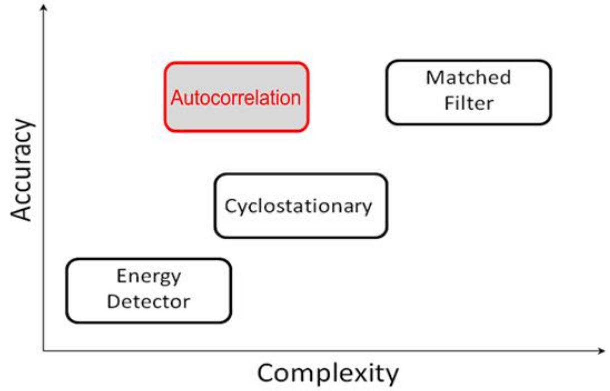


Figure 1. Spectrum sensing methods comparison [8]

The energy detector is the simplest but also the least accurate .This sensor does not require any previous knowledge of the signal. The signal is detected by comparing the output of the energy detector with a threshold which depends on the noise floor [9]. The problem with this type of detector is that it needs to know the noise level in order to establish the threshold. Additionally it performs poorly at low signal to noise ratio (SNR).

The cyclostationarity detector takes advantage of the fact that the statistical parameters of practical communication signals vary periodically [10]. Examples of these signals include sinusoidal carriers in amplitude, phase, and frequency modulation systems, and periodic keying of the amplitude, phase, or frequency in digital modulation systems [11]. The cyclostationarity can be extracted by the spectral-correlation density (SCD) function [11-13]. The SCD function of modulated signals takes nonzero values at some nonzero cyclic frequencies. On the other hand, since noise is not cyclostationary, its SCD has zero values at all non-zero cyclic frequencies. Therefore, by analyzing the SCD function it is possible to differentiate a particular signal from noise. In addition, the SCD provides a way to distinguish the signal type because different signals have different non-zero cyclic frequencies [10]. Although cyclostationarity sensing performs well under low SNR conditions and uncertainty in the propagation channel, it has some drawbacks such as: (1) the sampling rate needs to be high, (2) the computation of SCD requires a large amount of samples, (3) the sampling time error and frequency offset could affect the cyclic frequencies [10] and (4) it is more complex compared with energy detection.

The matched filter detector is the most optimum method for spectrum sensing when the transmitted signal is known. Its main advantage is that it takes less time than the other techniques to achieve a determined probability of false alarm or misdetection [14]. Nevertheless, matched filter sensing requires the sensor to demodulate received signals. This condition means that the receiver needs to know signal features of the signals such as bandwidth, operating frequency, modulation type, pulse shaping, and frame format [15]. That implies that the spectrum sensor needs a different receiver for each type of signal it expects to detect. Out of these three methods, ED is perhaps the simplest and most popular method; however it requires knowledge of the noise power. MF and CSD need information about the signal prior to receive it. Inaccuracy in the knowledge of the noise power causes mistakes in the detection of signals. Similarly, the need of prior information about the signal increases complexity.

Covariance and autocorrelation spectrum sensing can overcome these problems, since they capitalize on the fact that the covariance matrix and autocorrelation of noise and signals behave differently; therefore they require no information about either the noise power or the signal [16]. The covariance matrix of the samples collected by the receiver contains information exploitable for the purpose of spectrum sensing. Zeng and Liang [16, 17] have proposed two methods that extract information out of the covariance matrix of the samples. In [16] these authors introduce two statistics: the sum of the matrix elements that are not in the main diagonal, and the sum of the elements that are in the main diagonal. Comparing the ratio of these statistics with a threshold can tell the presence of either signal or noise. In [17] the authors use two metrics: the maximum to minimum eigenvalue (MME) ratio and the average received power to minimum eigenvalue ratio, also called energy with minimum eigenvalue (EME) detection. As in [16], the comparison of these ratios with a threshold can differentiate between noise and signal. Reyes *et al*. [18] proposed a spectrum sensing method based on the Euclidean distance between a reference line and the autocorrelation of the received samples.

The literature reports the use of open source software for spectrum monitoring. Matamoros *et al*. presented a prototype for assessing the spectrum usage in the 500 MHz to 686 MHz band in Pichincha, an Ecuadorian province. They used a Raspberry Pi along with an RF Explorer WSUB 1G, a low cost open hardware spectrum analyzer, a Global Sat BU-353-S4 for geo-localization, and a Python program. The used the energy detection sensing method to detect TV white spaces. O'Shea *et al*. [19] and Giatsos el al. [20] have proposed the use of GNU Radio Software and the Universal Software Radio Peripheral (USRP®) for spectrum monitoring. GNU Radio is an open source library that contains a set of assorted signal processing blocks that has become the most known SDR platform in the research and academic realms. Something similar has occurred channel busy is represented with 1. To analyze the samples the sensor calculates the autocorrelation of N samples and finds the Euclidean distance between this autocorrelation and a reference line. If that distance is lower that a certain threshold, the channel is considered to be busy, otherwise the channel is considered empty.

The N samples are stored in an 1XN array; therefore, the autocorrelation is also an 1XN array. The reference line is also represented as a 1XN array. The autocorrelation of the samples and the reference line can be considered as vectors, which allows for the calculation of the Euclidean distance between them.

Figure 2 shows the autocorrelation of the received samples when only Gaussian noise in present and when a signal plus Gaussian noise are present. The reference line acts as a limit for the autocorrelation; no matter how high the SNR is, the autocorrelation will never go further than that line. The distance between the autocorrelation and the reference line is a metric for analyzing the samples and determining the state of the channel. The spectrum sensor yields 1 if it finds the channel busy and 0 if it finds the channel empty.

The reference line $R(n)$ is defined in Eq. (1). N is the number of samples, and n is the sample index.

$$\mathrm{R(n)} = \frac{-n}{N} + 1 \qquad (1)$$

The autocorrelation $\lambda(n)$ is calculated through this formula (2)

with USRP, although its high price makes it difficult for researchers in developing to get access to it.

III. PROPOSED SOLUTION

We propose a low cost spectrum monitoring system, the objective of which is to scan a list of channels to determine the channel occupancy percentage of each channel. The system consists of a spectrum sensor, a tuner, and an internet connectivity block. The tuner controls the channel to be sensed. The spectrum sensor takes samples from the channel, analyzes those samples to determine the state of the channel, empty or occupied; and stores the decision in a file along with the date, time, central frequency of the channel, and the coordinates of the place where the spectrum sensor is installed. The connectivity block allows the system to send the information to a database for further analysis. The next sections explain each one of the system components in more detail.

A. Spectrum sensor

The spectrum sensor takes samples from the channel, analyzes them, and makes a binary decision: channel empty, or channel occupied. Channel empty is represented with 0, whereas

$$\lambda(n) = \frac{1}{N}\sum_{m=0}^{N-1} x(m)x(m-l), \qquad (2)$$

where $l = 0,1,\ldots,N-1$

The Euclidean distance E_d is calculated as in Eq. (3)

$$E_d = \sqrt{(\lambda(0)-R(0))^2 + (\lambda(1)-R(1))^2 + \cdots + (\lambda(N)-R(N))^2)} \qquad (3)$$

E_d is normalized to be between 0 and 1. E_d is the normalized Euclidean distance, which is compared with a threshold ψ. If $E_d < \psi$, it means that the autocorrelation $\lambda(n)$ is close enough to the reference line $R(n)$ to consider that there is a signal present, channel busy. If $E_d \geq \psi$, $\lambda(n)$ almost coincides with the autocorrelation of the noise, no signal present or channel empty. We set the threshold as $\psi = 0.94$, the same one that was used in [18], paper that originally proposed the spectrum sensing technique used in this article.

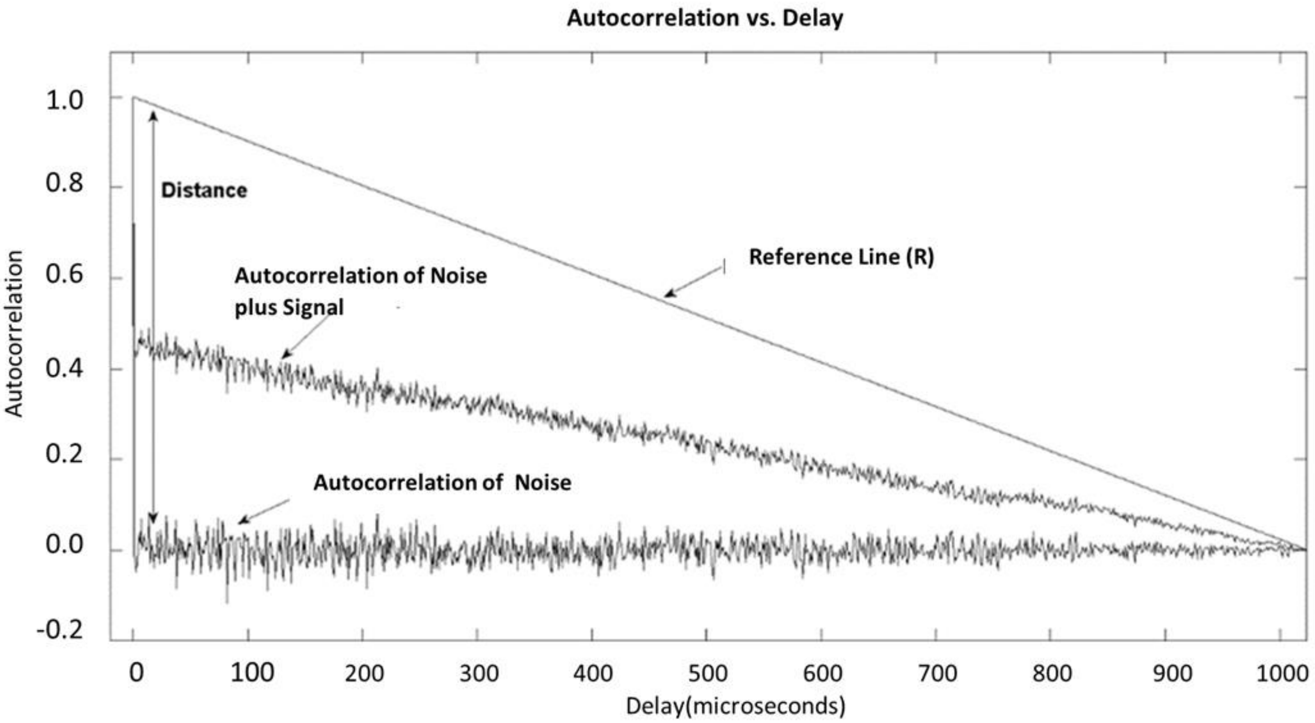


Figure 2. Autocorrelation of noise and signal plus noise.

B. RTL-SDR and Dragonboard 410C integration

Figure 3 shows the hardware components of the proposed system. The RTL-SDR acts as a radiofrequency front end (RFE). It takes samples from the radio spectrum, converts them to digital format, and transmits them to the Dragonboard 410C through a USB connection. The Dragonboard 410C3 processes and analyzes the samples to determine the state of the channel, according to the method described earlier.

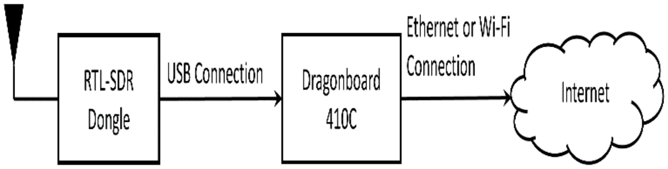


Figure 3. Spectrum sensing hardware components.

Figure 4 shows the software components of the proposed system. The python interface pyrtlsdr turns the RTL2832U dongle, whose original purpose is being a DVB-T receiver, into an RFE, the hardware part of an SDR system. This interface, pyrtlsdr, allows for the creation of a python object, which communicates with the main python program. Through this object we modify the operating parameters of the RTL2832U dongle, such as the central frequency (F_C), sampling frequency (F_S), and the number of samples to be read (N). The main function of this object, called *sampler*, is to take samples from the radio spectrum according with the parameters F_C , F_S , and N, and pass them to the *detector*, a python function that calculates the autocorrelation of the samples, and run the algorithm explained in the previous section to determine whether the channel is empty or occupied. The *detector* receives from the main program the parameters F_C , F_S , N, and *Thr*, the threshold used to decide on the state of the channel as indicated previously. A web page embedded in the Dragonboard 410C provides a friendly interface to configure these parameters, Figure 5. Similarly, the list of the channels to be scanned is inserted through either the web page or a text file. The *detector* function yields 1 or 0 to the main program, which inserts this result into an array along with the time and date, the coordinates of the location, and the frequency at which the detection was made. The main python program inserts the content of this array into MySql database. This database is installed in a remote server accessed through internet via IP V4 or IP V6.

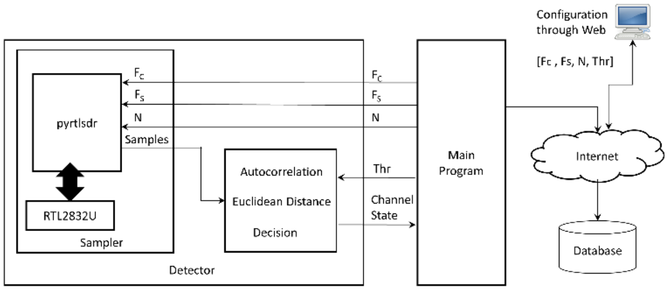
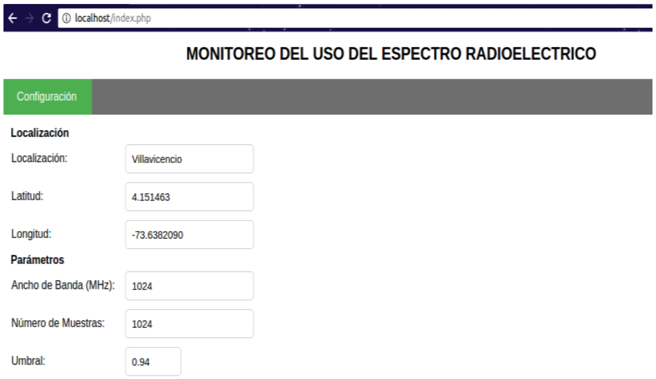


Figure 4. Software components.



localhost/index.php

MONITOREO DEL USO DEL ESPECTRO RADIOELECTRICO

Configuración

Localización

Localización:	Villavicencio
Latitud:	4.151463
Longitud:	-73.6382090

Parámetros

Ancho de Banda (MHz):	1024
Número de Muestras:	1024
Umbral:	0.94

Figure 5. Web page to configure the parameters F_C , F_S , *N*, and *Thr*

C. Experiments

We performed a series of experiments to test the functionality of the prototype. In the first experiment we used an RF signal generator to radiate low power (under 1mW) signals at several frequencies (149.1 MHz, 181.25MHz, and 193.25 MHz) ; we set the carrier frequency of the sensor at the same frequencies to see if it was able to detect the signals. Figure 6 shows the experimental setup used in this test.

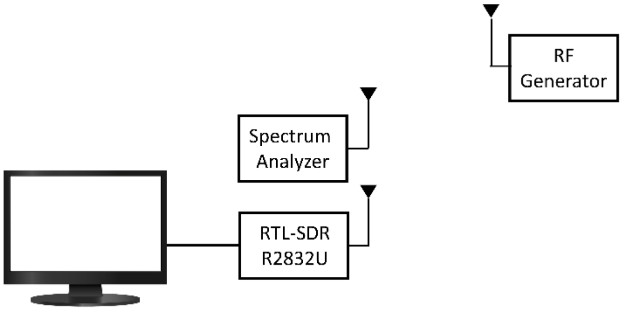


Figure 6. Experimental setup

The other experiments that we performed were aimed at verifying if the sensor was able to detect signals from TV stations, FM radio broadcasting stations, and other systems. For this experiment we consulted the list of radio and TV stations with license to operate in the area. Through one of the tabs of the configuration web page, we specified the video carrier frequencies of the TV channels that we wanted to scan: 55.25 MHz (Channel 2), 61.25 MHz (Channel 3), 67.25 MhZ (Channel 4), 77.25 MHz (Channel 5), 83.25 MHz (Channel 6), 175.25 MHz (Channel 7), 181.25 MHz (Channel 8), 187.25 MHz (Channel 9), 193.25 MHz (Channel 10), 199.25 MHz (Channel 11), 205.25 MHz (Channel 12), and 211.25 MHz (Channel 13). We also scanned several FM radio channels and other frequencies listed in the next section.

D. Results and discussion

In the first experiment, we observed that for all the frequencies the sensors yielded 1 when the signal generator was turned on, and 0 when it was turned off. We repeated this test with different levels of power, obtaining the same outcome. This test allowed us to verify that the prototype was operating as expected.

Figure 7 shows the occupancy rate of these channels in a period of 24 hours. The results agree with what we expected: channels 4, 5, 7,9,11, and 13, assigned to a TV stations, had an occupancy rate of 100%; on the other hand, channels not assigned to any TV station such as channels 2, 3, 8, 10 had an occupancy rate of 0%. Channels 6 and 12, supposed to be free, had an occupancy rate of 15 %; we think that some system could have been radiating sporadically in these frequencies.

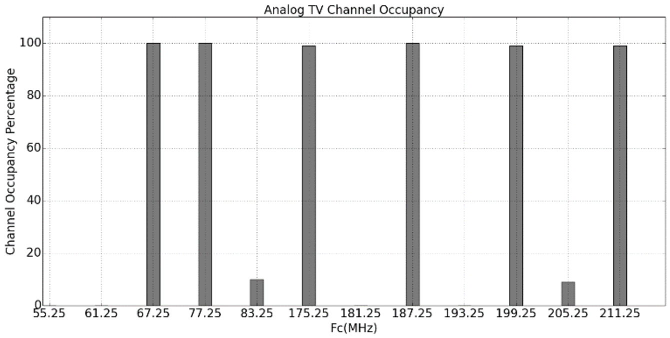


Figure 7. Results obtained when several analog TV channels were scanned.

Figure 8 shows the occupancy rates obtained when we scanned several FM radio station frequencies. The frequencies 90.3 MHz, 91.3 MHz, 92.3 MHz, 95.3 MHz, 96.3 MHz, 97. 3 MHz, 102.3 MHz, 105.3 MHz, and 107. 8 MHz exhibited an occupancy rate greater than 80 %; something we expected because these frequencies are assigned to FM radio stations supposed to cover the area of the city of Villavicencio. In the frequencies 88.3 MHZ and 88.8 MHz we obtained an occupancy rate below 10 %, something that can be explained by the fact that these frequencies belong to FM radio stations supposed to operate in the city of Acacias, a city near Villavicencio, the city where we performed the experiments. Something similar happened with the frequency 94.8 MHz, assigned to an FM radio station in the town of Restrepo, another neighbor of Villavicencio.

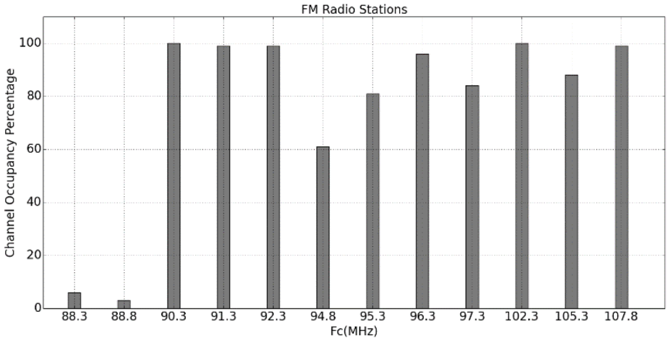


Figure 8. Results obtained when several FM radio channels were scanned

Figure 9 shows the occupancy rates of frequencies 433 MHz, 139 MHz, 137 MHz, 418 MHz, 90.3 MHz, and 77.25 MHz at four different intervals of time during a window of 24 hours. The frequencies 90.3 MHz, and 77.25 MHz had an occupancy rate of 100 % during all the four intervals; we explain this result by the fact that these frequencies belong to an FM radio station and a TV station that operate 24/7. The frequency 418 MHz had a similar behavior: 99% and 100 % occupancy rate during the 24 hour window. According to the national frequency allocation chart this frequency is assigned to mobile and fixed communications, and to space to space communication research.

The frequency 139 MHz had a fluctuating behavior in terms of its occupancy rate: around 40 % during the 9:00 PM - 3:00 AM interval; around 65 % during the 3:00 AM - 9:00 AM interval; around 80 % during from 3:00 PM to 9:00 PM; and around 90% from 9:00 AM to 3:00 PM. Let us notice that the highest occupancy rates took place during work hours and the evening; whereas, the lowest happened during the night and the early morning. According to the national frequency allocation chart this frequency is assigned to mobile and fixed communications, radio localization, and to space to earth communication research. On the other hand the frequency 137 MHz exhibited a very low occupancy rate, less than 5 %, in all the intervals. In Colombia, this frequency is assigned to aeronautical mobile communications.

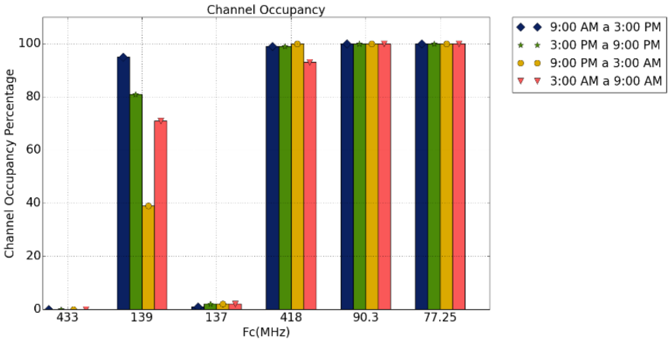


Figure 9. Occupancy rates of several channels at different intervals of time during a 24 hour window

IV. CONCLUSIONS

We have developed and tested a spectrum monitoring prototype based on the RTL2832U – SDR, the Dragonboard 410C card, and Python. The spectrum monitoring tests that we performed with the prototype have demonstrated its functionality. We have shown that it is possible to monitor the spectrum with low price tools and spectrum sensing techniques, such as the autocorrelation based sensing method, which differ from the conventional methods that use expensive spectrum analyzers and energy detection spectrum sensing. Future work includes adding a heat sink system to both the RTL2832U and the Dragonboard 410C card, developing a system for analyzing the spectrum utilization patterns, and using wide-band antennas to improve the sensitivity of the sensor.

REFERENCES

[1]. F. Sanders, “Broadband spectrum surveys in Denver, CO, San Diego, CA, and Los Angeles, CA: Methodology, analysis, and comparative results”, in Proc. IEEE International Symposium on Electromagnetic Compatibility, Denver, CO, USA, 1998, pp. 988 – 993.

[2]. M. McHenry et al., “Chicago spectrum occupancy measurements & analysis and a long-term studies proposal”, in Proc. First International Workshop on Technology and Policy for Accessing Spectrum, Boston, MA, USA, 2006.

[3]. M. McHenry, “NSF spectrum occupancy measurements project summary”, Shared Spectrum Company, Viena, Switzerland,Tech. Rep. 2005.

[4]. M. Wellens, J. Wu, and P. Mahonen, “Evaluation of spectrum occupancy in indoor and outdoor scenario in the context of cognitive radio”, in Proc. Second International Conference on Cognitive Radio Oriented Wireless Networks and Communications, CrownCom, Orlando, FL, USA, 2007, pp.420 – 427.

[5]. R. Chiang, G. Rowe, and K. Sowerby, “A quantitative analysis of spectral occupancy measurements for cognitive radio”, in Proc. 65th IEEE Vehicular Technology Conference, Dublin, Ireland, 2007, pp. 3016 – 3020.

[6]. M. H. Islam et al., “Spectrum survey in Singapore: occupancy measurements and analysis”, in Proc. Third International Conference on Cognitive Radio Oriented Wireless Networks and Communications, CrownCom, Singapore, Singapore, 2008, pp. 1– 7.

[7]. M. Lopez-Benitez, A. Umbert, and F. Casadevall, "Evaluation of spectrum occupancy in Spain for

cognitive radio applications", in Proc. 69th IEEE Vehicular Technology Conference, Barcelona, Spain, 2006, pp. 1– 5.

[8]. M. Subhedar and G. Birajdar, "Spectrum sensing techniques in cognitive radio networks: A survey", International Journal of Next-Generation Networks, vol. 3, no. 2, pp. 37– 51, 2011.

[9]. H. Urkowitz, "Energy detection of unknown deterministic signals", Proceedings IEEE, vol. 55, no. 4, pp. 523– 531, 1967.

[10]. Y. Zeng et al.,"A review on spectrum sensing for cognitive radio: Challenges and solutions" EURASIP Journal on Advances in Signal Processing, 2010.

[11]. W. A. Gardner, "Exploitation of spectral redundancy in cyclostationary signals", IEEE Signal Processing Magazine, vol. 8, no. 2, pp. 14– 36. 1991.

[12]. W. Gardner, W. Brown, and C. Chen, "Spectral correlation of modulated signals: Part II--digital modulation", IEEE Transactions on Communications, vol. 35, no. 6, pp. 595– 601, 1987.

[13]. W. Gardner, W. Brown, and C. Chen, "Spectral correlation of modulated signals: Part I--analog modulation", IEEE Transactions on Communications, vol. 35, no. 6, pp. 584 – 594, 1987.

[14]. R. Tandra and A. Sahai, "Fundamental limits on detection in low SNR under noise uncertainty", in Proc. International Conference on Wireless Networks, Communications and Mobile Computing, Maui, HI, USA, 2005, pp. 464 – 469.

[15]. T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications", IEEE Communications Surveys & Tutorials, vol. 11, no.1, pp. 116 – 130, 2009.

[16]. Y. Zeng and Y. Liang, "Spectrum-sensing algorithms for cognitive radio based on statistical covariances", IEEE Transactions on Vehicular Technology, vol. 58, no. 4, pp. 1804 – 1815, 2009.

[17]. Y. Zeng and Y. Liang, "Eigenvalue-based spectrum sensing algorithms for cognitive radio", IEEE Transactions on Communications, vol. 57, no. 6, pp. 1784 – 1793, 2009.

[18]. H. Reyes, et al, "A spectrum sensing technique based on autocorrelation and euclidean distance and its comparison with energy detection for cognitive radio networks", Computers & Electrical Engineering, vol. 52, pp. 319 – 327, 2015.

[19]. T. O'shea, T. Clancy, and H. Ebeid, "Practical signal detection and classification in GNU radio", in Proc. SDR Forum Technical Conference, Denver, CO, USA, 2007, pp. 143 – 148.

[20]. D. Giatsios, et al., "Methodology and tools for measurements on wireless testbeds: the NITOS approach", Measurement Methodology and Tools, vol. 7586, pp. 61 – 80, 2013.