

Desarrollo de sistemas receptores de AM, FM y ADS-B utilizando radio definida por software, hardware y software libre

Santiago Romero¹, Christian Tipantuña², José Antonio Estrada², Jorge Carvajal²

¹ CELEC EP COCA CODO SINCLAIR, Av. 6 de Diciembre N31-110 y Whymper, Quito, Ecuador, 170122.

² Departamento de Electrónica, Telecomunicaciones y Redes de Información, Facultad de Ingeniería Eléctrica y Electrónica, Escuela Politécnica Nacional, Ladrón de Guevara E11-253, Quito, Ecuador, 17-01-2759.

Autores para correspondencia: santiago.romero@celec.gob.ec, christian.tipantuna@epn.edu.ec, jose.estrada@epn.edu.ec, jorge.carvajal@epn.edu.ec

Fecha de recepción: 28 de septiembre 2015 - Fecha de aceptación: 12 de octubre 2015

RESUMEN

En el presente artículo se muestra en aplicaciones reales el potencial y la versatilidad de la tecnología de Radio Definido por Software (RDS), para ello, se presenta el desarrollo de diferentes sistemas de comunicaciones de bajo costo utilizando plataformas de hardware y software libre. Los receptores de AM, FM y ADS-B descritos, han sido desarrollados utilizando un ordenador de placa reducida Raspberry Pi y un módulo RTL-SDR como plataforma de hardware; como plataforma de software se han empleado la distribución Raspbian, la herramienta GNU Radio y el software Dump 1090.

Palabras clave: RDS, Raspberry Pi, RTL-SDR, GNU Radio, Dump 1090, AM, FM, ADS-B.

ABSTRACT

This paper describes in real applications the potential and flexibility of the Software Defined Radio (SDR) technology, for this purpose the development of different low-cost communications systems using open source hardware and software is shown. The AM, FM and ADS-B receivers described in this paper have been developed using a credit card size computer Raspberry Pi and a module RTL-SDR as hardware platform; on the other hand as software platform is used the distribution Raspbian, GNU Radio toolkit and the software Dump 1090.

Keywords: SDR, Raspberry Pi, RTL-SDR, GNU Radio, Dump 1090, AM, FM, ADS-B.

1. INTRODUCCIÓN

Los sistemas de comunicaciones en los últimos años han experimentado una enorme evolución, pasando de ser sistemas para funciones específicas a ser sistemas reconfigurables y reprogramables. La tecnología de RDS ha aportado en gran medida a esta evolución ya que permite que componentes típicamente implementados en hardware sean implementados en software (Tuttlebee, 2003).

Hoy en día la utilización de software de tipo *open source* es muy común en un sinnúmero de servicios y aplicaciones, constituye una filosofía bastante cimentada y cuenta con el respaldo de un gran número de empresas y comunidades a nivel mundial; así también, un tema que paralelamente ha ido tomando fuerza en los últimos años, es el hardware de tipo *open source* o hardware libre, un tipo de hardware en el que se aplica la filosofía del software libre y, por ende, constituye un tópico de interés a nivel investigativo, académico e industrial (Pearce, 2012). Conjugando adecuadamente las tecnologías de RDS, hardware y software libre, es posible desarrollar gran variedad de aplicaciones, las cuales presentan una gran flexibilidad, versatilidad y pueden ser implementadas reduciendo

notablemente los costos, por ende, promueven un espacio propicio para la innovación, investigación y desarrollo de nuevos estándares y sistemas de comunicaciones. Al ser soluciones de bajo costo las planteadas en este artículo, son plenamente asequibles a los investigadores, docentes y estudiantes, y propician el conocimiento en detalle de los procesos involucrados en diferentes sistemas de comunicaciones.

En el presente artículo se presenta el desarrollo de sistemas receptores de radio comercial AM (Amplitude Modulation, 550 a 1600 KHz) y FM (Frequency Modulation, 88 a 108 MHz); así como un sistema de ADS-B (Automatic Dependent Surveillance – Broadcasting), el cual es un sistema abierto de vigilancia de vuelos comerciales que funciona a 1090 MHz (McCallie *et al.*, 2011). Todos los sistemas utilizan el mismo hardware, la placa Raspberry-Pi y el módulo RTL-SDR; los cambios radican en la configuración por software y el Front End mismo, que depende del tipo de sistema.

2. TRABAJOS RELACIONADOS

La evolución de la tecnología digital, en especial el desarrollo y evolución de las FPGA, ha permitido la implementación práctica de sistemas de comunicaciones que antes eran únicamente abordados de manera teórica; diferentes sistemas de comunicaciones basados en RDS y que, principalmente, utilizan FPGAs, han sido desarrollados en la última década. Los dispositivos de RDS, comercialmente más conocidos, son las plataformas USRP (Universal Software Radio Peripheral) de la empresa Ettus ResearchTM. Con los dispositivos USRP se han desarrollado varios sistemas de comunicaciones experimentales, entre los que se pueden mencionar: analizadores de espectro (Costa *et al.*, 2009), sistemas MIMO (Chen *et al.*, 2014), sistemas modernos de comunicaciones analógicas y digitales (Gandhiraj & Soman, 2014), entre otros.

Si bien la plataforma USRP es uno de los sistemas de RDS más conocidos, es robusta, potente y flexible, el gran inconveniente es su elevado costo, solo su versión más económica tiene un costo de varios miles de dólares (Ettus Research, 2015). En el mercado se disponen de otras soluciones de SDR más económicas y compatibles con GNU Radio, entre las que se pueden mencionar: Fairwaves UmTRX, HackRF, Funcube Dongle, Nuand BladeRF, Microtelecom Perseus y RTL-SDR; siendo la plataforma RTL-SDR la más económica de todas. Considerando que la mayoría de sistemas desarrollados utilizan una plataforma USRP, en el presente trabajo se presenta una solución *open source* de muy bajo costo, para ello, toda la configuración y programación se ha realizado en la plataforma Raspberry Pi y el Front End está constituido principalmente por el módulo RTL-SDR, en las siguientes secciones se indicará, en detalle, el desarrollo de los diferentes sistemas receptores.

3. RADIO DEFINIDO POR SOFTWARE

El término *Software Radio* fue adoptado por Joseph Mitola III, a inicios de los noventa, al referirse a los tipos de radio reconfigurables o reprogramables (Mitola III, 1993). Las primeras aplicaciones de la tecnología RDS fueron de tipo militar y, conforme al avance de los años, la tecnología fue adoptada al campo de los sistemas y redes de comunicaciones.

Según el Wireless Innovation Forum: “Un radio definido por software es un radio en el que algunas o todas las funciones de capa física son definidas en software” (Wireless Innovation Forum, 2014). El término definido en software implica que diferentes formas de onda pueden ser soportadas mediante el cambio en software pero no alterando el hardware. Al estar toda la funcionalidad de radio de banda base implementada en software, las opciones de diseño y los módulos de radio disponibles para la plataforma RDS pueden ser guardados y utilizados cuando sean necesarios, estos módulos podrían ser un esquema de modulación en específico, algún código corrector de errores u otro bloque funcional; adicionalmente, dichos bloques podrían ser cambiados en tiempo real y los parámetros de operación de los mismos podrían ser ajustados, ya sea por un operador o proceso automatizado. Las características principales de RDS son:

- Facilidad de actualización: Actualización del sistema mediante actualización de software.
- Interoperabilidad: Un RDS puede comunicarse con muchos estándares de comunicaciones.
- Reutilización oportuna de frecuencia: Un RDS puede tomar ventaja del espectro subutilizado.
- Bajo costo: Un SDR tiene un reducido costo debido a su múltiple aplicación.
- Desarrollo e investigación: Implementación de muchas y distintas formas de onda.

4. PLATAFORMA DE HARDWARE

4.1. Dispositivo RTL-SDR

Dispositivo de RDS que forma parte activa en todos los sistemas de comunicaciones presentados, el dispositivo usa tanto el chip RTL2832U como el chip R820T. En resumen, el funcionamiento del dispositivo se muestra en la Figura 1.

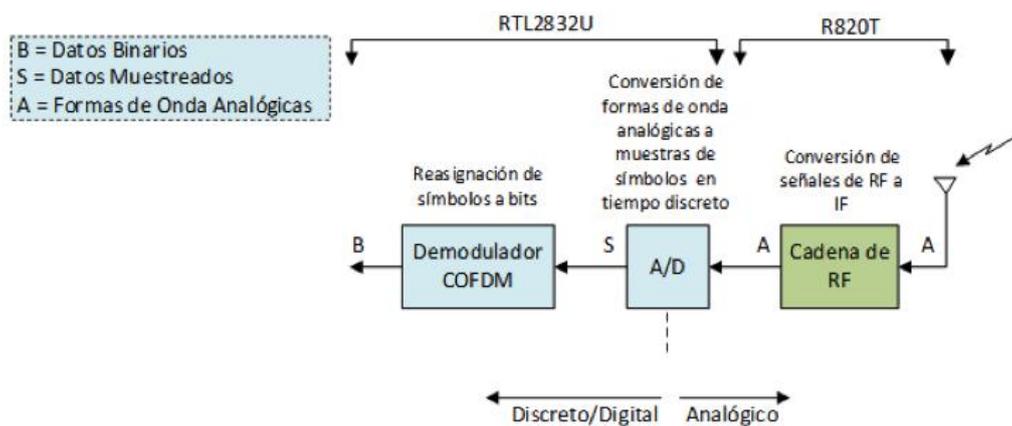


Figura 1. Funcionamiento dispositivo RTL-SDR.

Una característica inherente de este chip es el soporte para estándares de radio como FM. El dispositivo SDR a usar es de la marca NooElec, tiene una interfaz USB 2.0, el conector para la antena tiene una impedancia de 75Ω y es del tipo micro coaxial (MCX) hembra.

4.2. Conversor Ham It Up

La función de este dispositivo es convertir las señales de las bandas Medium Frequency (MF) y High Frequency (HF) a señales en la banda Very High Frequency (VHF). En el presente trabajo, este conversor permite que sea posible la implementación del sistema receptor de AM, ya que el dispositivo RTL-SDR, específicamente el chip R820T (24 MHz - 1766 MHz) por sí mismo, no es capaz de captar señales comprendidas en el rango de AM comercial (550 KHz - 1600 KHz). El conversor es de la marca NooElec y su frecuencia intermedia de salida es de 125 MHz.

4.3. Raspberry Pi

Es una computadora de placa reducida y de muy bajo costo desarrollada en el Reino Unido por la Fundación Raspberry Pi (Richardson & Wallace, 2012). Existen varios modelos de esta placa, el que se ha utilizado en este caso es el Raspberry-Pi 2 modelo B, lanzado al mercado en febrero de 2015 y que cuenta con las siguientes características: CPU ARM Cortex-A7 quad-core de 900MHz, 1GB de RAM, 4 puertos USB, 40 pines de entrada y salida, puerto Full HDMI, puerto Ethernet, jack de 3.5 mm de audio y video compuesto, interfaz para cámara (CSI), ranura para tarjeta Micro SD y núcleo para gráficos 3D VideoCore IV, adicionalmente este modelo es compatible con sistemas operativos Linux y Windows.

En el presente trabajo, esta plataforma es utilizada para la recepción y procesamiento de las señales captadas por el dispositivo RTL-SDR. Para el caso de los receptores AM y FM, mediante el procesamiento de las señales correspondientes, es posible visualizar e incluso escuchar las señales procesadas en banda base.

5. PLATAFORMA DE SOFTWARE

5.1. *Raspbian*

Raspbian es un sistema operativo libre basado en Debian y optimizado para arquitecturas ARM (Harrington, 2015). El sistema operativo que se ha utilizado en la plataforma Raspberry Pi es Raspbian Debian Wheezy, específicamente la versión: 2015-01-31.

5.2. *GNU Radio*

GNU Radio es una herramienta de desarrollo de código libre y abierto que permite implementar radios en software mediante bloques de procesamiento de señal (Blossom, 2004), y puede ser usado en conjunto con hardware de radio frecuencia de bajo costo como el dispositivo RTL-SDR. Este software posee una gran variedad de herramientas y programas, una de las más importante es GNU Radio Companion (GRC), la cual ofrece un ambiente de desarrollo gráfico para el procesamiento de señales, en donde los bloques que procesan la señal están escritos en C++, mientras que la creación de gráficos de flujo y la conexión de estos bloques de señal utiliza el lenguaje de programación Python. Para la implementación de sistemas receptores se utilizará la fuente RTL-SDR (osmocom source), esta fuente permite establecer la comunicación con el dispositivo RTL-SDR mediante el ingreso de la dirección IP del Raspberry-Pi y el puerto en el campo *Device Arguments*.

5.3. *Software RTL-SDR*

Este software contiene la propia librería **librtlsdr** y una serie de herramientas de línea de comando como: **rtl_test**, **rtl_sdr**, **rtl_tcp** y **rtl_fm**. Estas herramientas utilizan la librería para asegurarse que existen dispositivos RTL-SDR conectados y para realizar funciones básicas de transferencia de datos hacia y desde el dispositivo. Una de las ventajas de este software es que permite varias opciones para comunicarse con el hardware, como por ejemplo: el programa **rtl_fm**, el cual es una herramienta de línea de comando que inicializa el dispositivo, sintoniza una frecuencia determinada, guarda el sonido recibido en un archivo o lo canaliza para que se reproduzca por los parlantes. Para verificar que está funcionando el software y hardware RTL-SDR, se puede utilizar la herramienta **rtl_test**.

5.4. *Software Dump 1090*

El software DUMP1090 hace uso del software RTL-SDR previamente instalado para comunicarse con el dispositivo RTL-SDR. Este software permite mostrar todo el tráfico aéreo capturado de forma interactiva, es decir, hace posible que se muestre un mapa con el tráfico existente en un explorador web, tanto del Raspberry-Pi como de cualquier otro dispositivo conectado a la red, y junto a este mapa se visualiza una lista de las principales características de cada aeronave como número de vuelo, altitud y velocidad.

6. IMPLEMENTACIÓN Y RESULTADOS

6.1. *Sistema receptor de AM*

El diagrama de bloques en GNU Radio de este receptor es el que se muestra en la Figura 2.

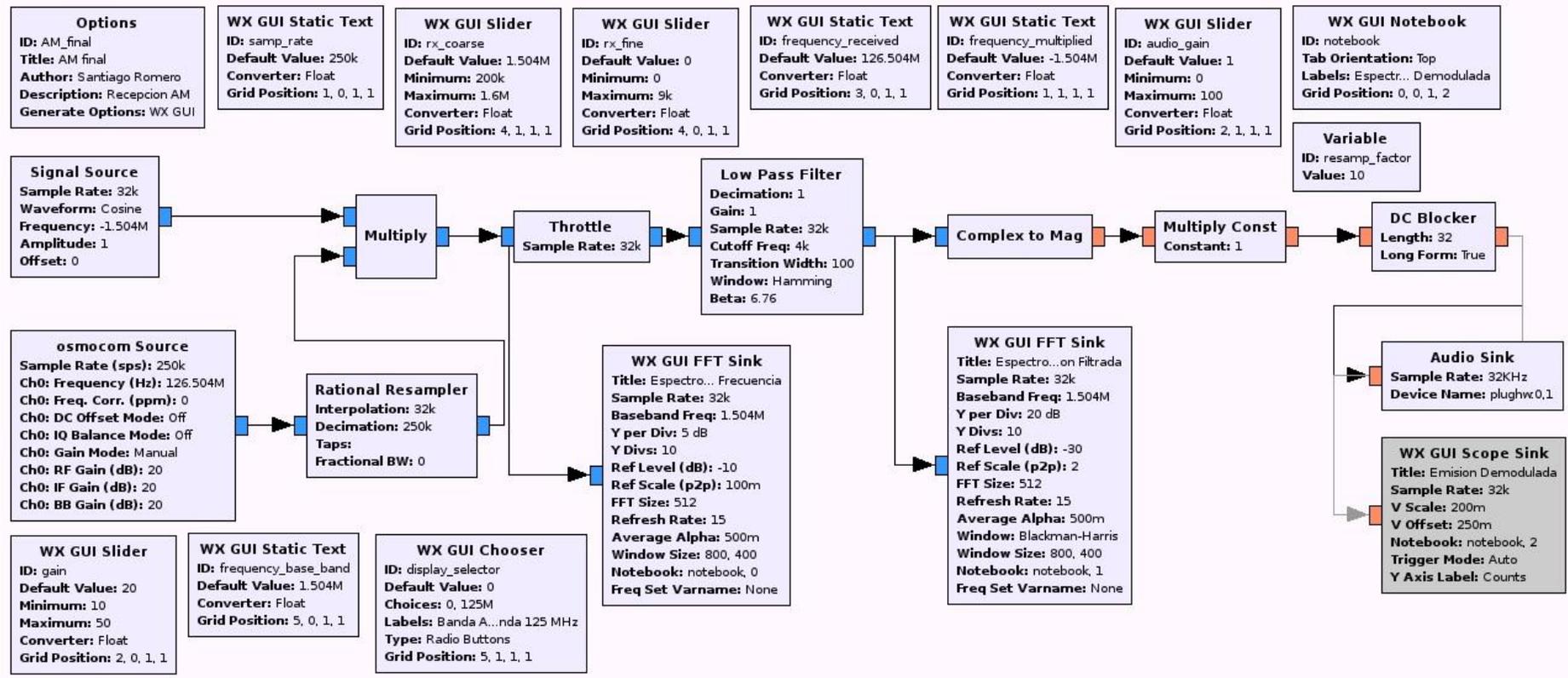


Figura 2. Esquema receptor AM en GNU Radio.

Para este tipo de receptor se ha empleado una demodulación de tipo síncrona, la cual consiste en multiplicar la señal modulada por una señal de naturaleza similar, posteriormente, con un filtro pasa bajos se procede a recuperar la señal en banda base, este tipo de demodulación permite que el receptor pueda demodular señales moduladas DSB-LC, DSB-SC y SSB-SC.

El diseño del receptor AM principalmente consiste de: una fuente RTL-SDR **Osmocom Source**, una fuente que genera la señal de la portadora **Signal Source**, un bloque multiplicador que permite mezclar las señales **Multiply**, un filtro pasa bajo **Low Pass Filter** que opera con una frecuencia de corte de 8 KHz y con una ventana de Hamming, y un sumidero de audio **Audio Sink** el cual permite que la señal demodulada pueda ser escuchada.

El diagrama de conexión para el receptor AM es el que se muestra en la Figura 3.

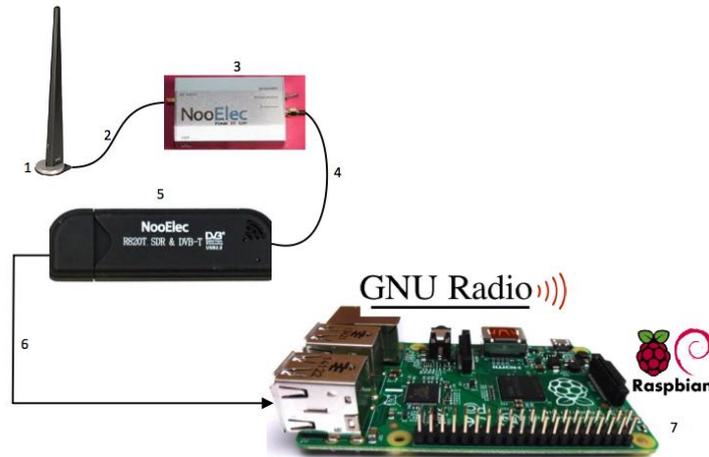


Figura 3. Diagrama de conexión para el receptor de AM.

Los elementos que forman parte de la implementación del receptor AM son: 1) Antena AM Terk para recepción en interiores, 2) Cable bifilar con conector SMA macho, 3) Conversor Ham It Up, 4) Cable adaptador SMA macho - MCX macho, 5) Dispositivo RTL-SDR, 6) Cable USB, y 7) Raspberry-Pi con sistema operativo Raspbian y software RTL-SDR.

En la Figura 4 en la parte izquierda se observa el espectro de una emisora cuya frecuencia central es de 717 KHz, en el centro se observa el espectro demodulado centrado a 125 MHz que es la frecuencia intermedia de salida del conversor Ham It Up, y en el lado derecho de la figura se puede apreciar la señal demodulada en el dominio del tiempo.



Figura 4. Señal AM en frecuencia y tiempo.

6.2. Sistema receptor de FM

El receptor implementado es de banda ancha WBFM (Wide Band FM) mismo que, principalmente, consiste en: una fuente RTL-SDR **Osmocom Source**, un filtro pasa bajo **Low Pass Filter** que opera a una frecuencia de corte de 100 KHz, valor que resulta igual a la mitad del ancho de banda comercial de una emisión de una estación de radio FM. Adicionalmente, el receptor consta de un módulo demodulador WBFM **WBFM Receive** que es en sí un demodulador por cuadratura, un filtro pasa bajo **Low Pass Filter** para el audio y que trabaja una frecuencia de corte de 7.5 KHz, una anchura de transición de 1 KHz (banda de rechazo) y una ventana de Hamming, un módulo de re-muestreo

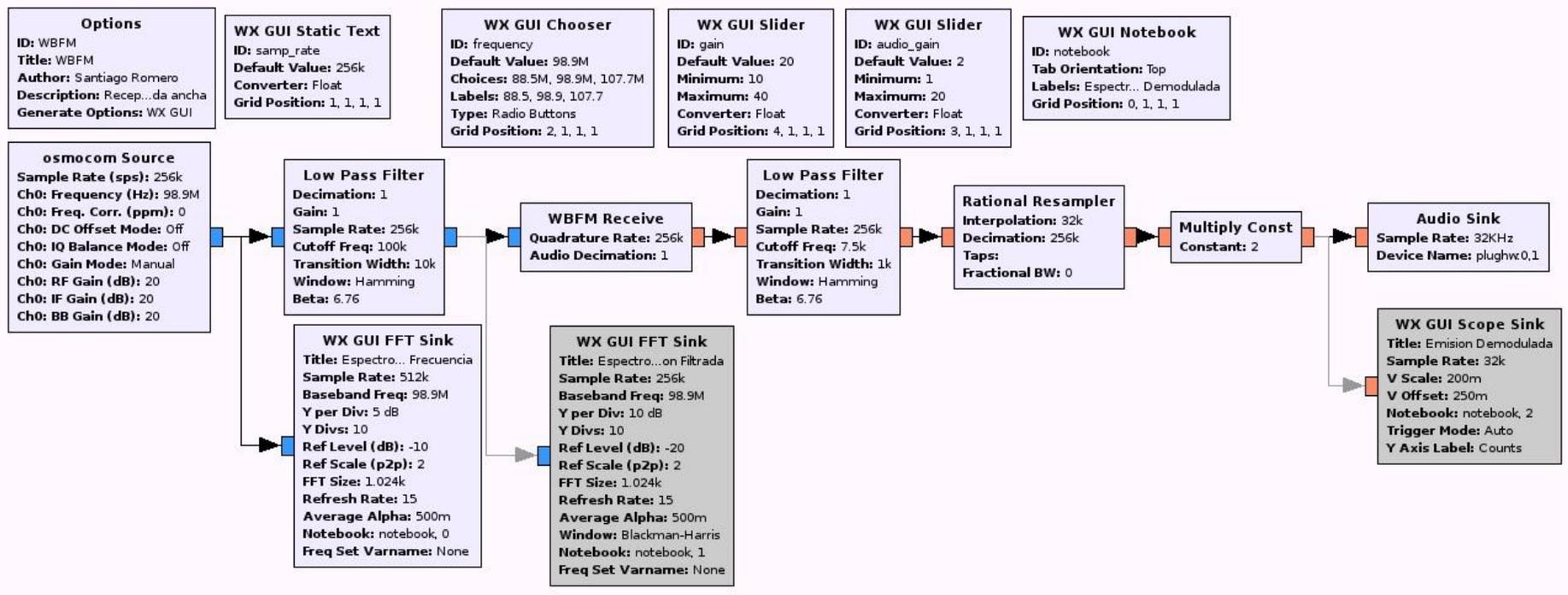


Figura 5. Esquema receptor FM en GNU Radio.

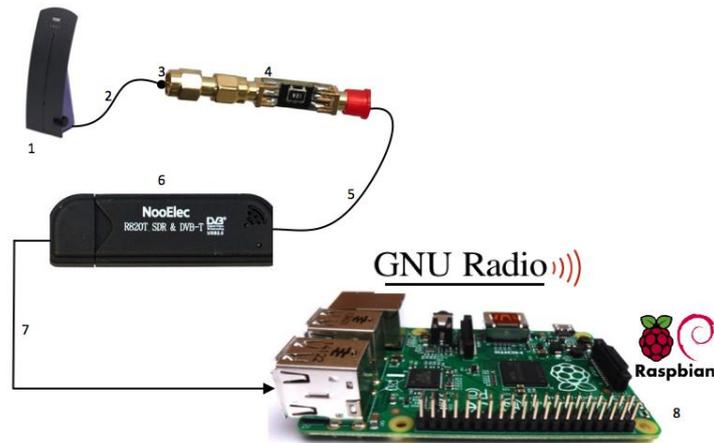


Figura 6. Diagrama de conexión para el receptor de FM

racional **Rational Resampler** para obtener una tasa de muestreo de 32 KHz a partir de 256 KHz y un sumidero de audio **Audio Sink**, a fin de que la señal demodulada sea convertida a señal analógica y fluya a través de la tarjeta de sonido. El diagrama de conexión para el receptor WBFM se muestra en la Figura 6. El diagrama de conexión en GNU Radio de este receptor es el que se muestra en la Figura 5.

Los elementos que forman parte de la implementación del receptor FM son: 1) Antena FM Terk para recepción en interiores, 2) Cable coaxial con conector F macho, 3) Adaptador F hembra – SMA hembra, 4) Filtro pasa banda para FM, 5) Cable adaptador SMA macho – MCX macho, 6) Dispositivo RTL-SDR, 7) Cable USB, y 8) Raspberry-Pi con sistema operativo Raspbian y software RTL-SDR.

El espectro de la señal de radio frecuencia, así como el espectro de la emisión filtrada pertenecientes al diseño mostrado en la Figura 6 y la señal digital demodulada que ingresa al sumidero de audio son observados en la Figura 7.

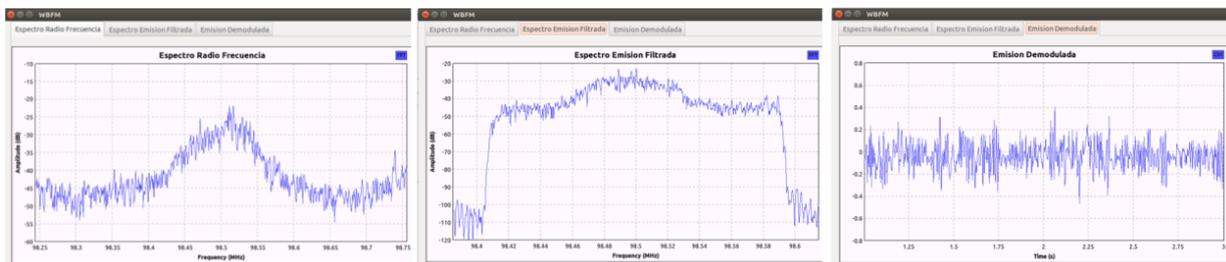


Figura 7. Señal FM en frecuencia y tiempo

6.3. Sistema receptor ADS-B

El diagrama de conexión de este receptor es el que se muestra en la Figura 8.

Los elementos que forman parte de la implementación del receptor ADS-B son: 1) Antena ADS-B para exteriores, 2) Cable adaptador BNC macho - SMA hembra, 3) Filtro pasa banda, 4) Cable adaptador SMA macho - MCX macho, 5) Dispositivo RTL-SDR, 6) Cable USB y 7) Raspberry-Pi con sistema operativo Raspbian y software RTL-SDR.

Hay que mencionar que para la implementación de este sistema no se utiliza GNU Radio, en cambio, se utiliza el software *open source* DUMP1090, el cual se ejecuta directamente en el Raspberry Pi. Entre las consideraciones necesarias para el funcionamiento del sistema ADSB, se tienen: conexión a Internet con el fin de cargar y actualizar el mapa en el cual se visualizará la ruta de vuelo de los aviones a detectar, considerando que la antena es el único elemento en tierra que recibe las señales del sistema ADS-B radiadas por las aeronaves en vuelo, la detección de todos los elementos de vuelo (número de vuelo, altitud, velocidad) depende directamente de la cobertura del elemento radiante, la

visualización de la ruta de vuelo se limita a una zona geográfica determinada (en este caso un sector de la ciudad de Quito).

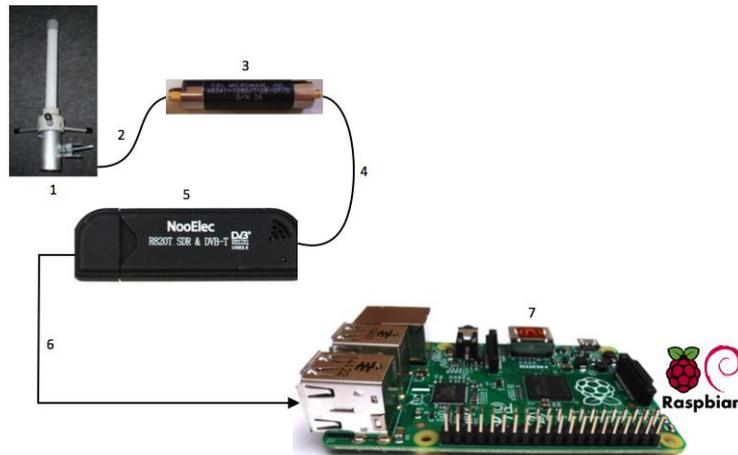


Figura 8. Diagrama de conexión para el receptor de ADS-B.

Una vez instalado el software RTL-SDR y el software DUMP1090 en el Raspberry-Pi, para ejecutar DUMP1090 es necesario ejecutar el comando: `sudo ./dump1090 --interactive --net`, luego de ejecutar este comando, en el propio terminal se muestra una tabla en la que se puede identificar el modo de operación del transpondedor del avión que se está rastreando, al igual que el número de vuelo, altitud, velocidad, latitud y longitud. Paralelamente, la ejecución del comando, hace posible que se muestre un mapa con el tráfico existente en un explorador web tanto del Raspberry Pi como de cualquier dispositivo conectado a red local, para ello, en el explorador web, es necesario ingresar la dirección IP del Raspberry Pi utilizando el puerto 8080. Junto al mapa se visualiza una lista de las principales características de cada aeronave como: número de vuelo, altitud y velocidad; ver en la Figura 9.

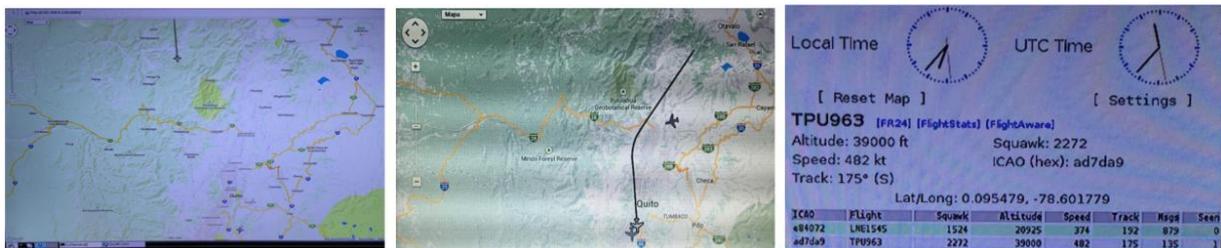


Figura 9. Mapa del tráfico aéreo visualizado en explorador web del Raspberry Pi.

7. CONCLUSIONES

La tecnología de RDS permite que diferentes formas de onda sean soportadas mediante cambios en software a través del tiempo, sin ninguna alteración en hardware, es decir, usando la misma plataforma es posible implementar nuevas funcionalidades que nunca fueron previstas.

El ambiente de desarrollo gráfico GNU Radio Companion (GRC) es un conjunto de herramientas muy potente, provee una variedad de bloques de procesamiento digital de señal para una gran variedad de aplicaciones, el entorno gráfico, sin duda alguna, permite implementar sistemas de comunicaciones de manera fácil y rápida.

Para AM y para FM, las señales demoduladas pudieron ser observadas tanto en tiempo como en frecuencia, se comprobó su correcta demodulación; adicionalmente, las transmisiones pudieron ser escuchadas percibiendo una muy buena calidad del audio en las mismas.

Mediante el sistema ADS-B se pudo observar en un terminal del Raspberry-Pi datos claves para la detección y vigilancia de aeronaves como: identificador único, número de transmisiones, número de vuelo, altitud, velocidad, latitud y longitud. La trayectoria descrita por la aeronave puede ser visualizada en un mapa en tiempo real, por tanto, es posible verificar si dicho vuelo en verdad está en curso.

REFERENCIAS

- Blossom, E., 2004. *GNU radio: tools for exploring the radio frequency spectrum*. Linux Journal, 122, 4. Disponible en <http://www.linuxjournal.com/article/7319>.
- Chen, X., B.P. Einarsson, P.S. Kildal, 2014. Improved MIMO throughput with inverse power allocation - Study using USRP measurement in reverberation chamber. *Antennas and Wireless Propagation Letters, IEEE*, 13, 1494-1496.
- Costa, A., J. Lima, L. Antunes, N.B. de Carvalho, 2009. *Spectrum analyzer with USRP, GNU Radio and MATLAB*. In: 7th Conference on Telecommunication, Portugal. Disponible en <http://www.av.it.pt/conftele2009/papers/114.pdf>, 5 pp.
- Ettus Research, 2015. *A National Instruments Company, USRP Networked Series*. Disponible en <http://www.ettus.com/product/category/USRP-Networked-Series>.
- Gandhiraj, R., K.P. Soman, 2014. Modern analog and digital communication systems development using GNU Radio with USRP. *Telecommunication Systems*, 56(3), 367-381.
- Harrington, W., 2015. *Learning Raspbian*. Packt Publishing Ltd.
- McCallie, D., J. Butts, R. Mills, 2011. Security analysis of the ADS-B implementation in the next generation air transportation system. *International Journal of Critical Infrastructure Protection*, 4(2), 78-87.
- Mitola III, J., 1993. Software radios: Survey, critical evaluation and future directions. *Aerospace and Electronic Systems Magazine, IEEE*, 8(4), 25-36.
- Pearce, J.M., 2012. Building research equipment with free, open-source hardware. *Science*, 337(6100), 1303-1304.
- Richardson, M., S. Wallace, 2012. *Getting started with raspberry PI*. O'Reilly Media, Inc.
- Tuttlebee, W.H., 2003. *Software defined radio: enabling technologies*. Published online: John Wiley & Sons, Inc. Disponible en <http://onlinelibrary.wiley.com/book/10.1002/0470846003>, 397 pp.
- Wireless Innovation Forum, 2014. *What is software defined radio?* Disponible en http://www.wirelessinnovation.org/what_is_sdr.