

Decodificación de señales RBDS/RDS con GNURADIO

Emmanuel Ortiz López

Abstract

Desarrollo de una aplicación Software Defined Radio basada en GNURADIO para sintonizar estaciones FM y desplegar la información RDS utilizando un receptor RTL-SDR

Keywords: SDR, GNURADIO, RDS

1. Software Defined Radio

Radio Definido por Software (SDR por sus siglas en inglés) es una de las tendencias actuales en educación y diseño de sistemas de comunicaciones, la idea es usar un sistema mínimo de hardware (un receptor genérico) solo para capturar las señales de radio frecuencia (RF), y procesar la señal mediante software es decir modular, demodular, decodificar, etc. es decir sustituye la mayoría del hardware de los sistemas de comunicaciones.

1.1. RTL-SDR

Este es el nombre de los receptores de RF, existen varios modelos, en específico para este proyecto utilizamos RTL2832 este trabaja en las frecuencias de 24MHz hasta 1850MHz, funciona con todos los sistemas Operativos.²



Figure 1: RTL2832

1.2. GNURADIO

GNURADIO es un proyecto Open Source de GNURADIO Companion que ofrece un entorno gráfico de programación a bloques. Estos bloques están basados en C++, Volk, qgrx, xml y Python. Tiene 2 modalidades: front y back end a través del GUI de bloques, y creación back end. Existe la posibilidad de crear tus propios bloques para el entorno gráfico.

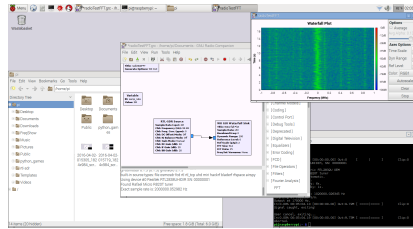


Figure 2: Ejemplo de GNURADIO

2. Radio Data System

Es un esquema de modulación para transmitir un estándar de datos utilizando las transmisiones FM, RDS por sus siglas en inglés fue creado por la Unión Europea de Radiodifusión. Estos datos son inaudibles pero contienen información como nombre de la estación, nombre de la canción hasta el informe de tráfico. Esta información viene en una soportadora de 57 KHz que está en el ancho de banda de la transmisión FM, es una señal digital codificada en “Manchester”

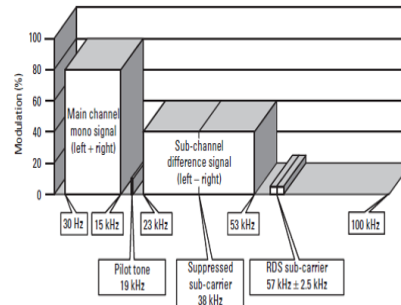


Figure 3: FFT de la señal FM-RDS

En específico la información que contiene son 104 bits, dividido en paquetes de 26 bits, 16 de estos son información el resto son bits usados en la corrección de errores.

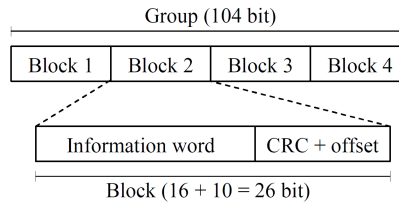


Figure 4: Paquetes de información RDS

3. Resultados

En esta sección describiré como obtuve la señal RDS usando GNURADIO, RTL2832 y Ubuntu.

3.1. Instalación RTL

Lo primero que se hizo es los sistemas operativos basados en debian (Ubuntu y Raspian) pudieran utilizar el RTL2832 por medio de terminal.

```
sudo apt-get install git cmake libusb-1.0-0-dev build-essential
cd ~
git clone git://git.osmocom.org/rtl-sdr.git
cd rtl-sdr
mkdir build
cd build
cmake ../ -DINSTALL_UDEV_RULES=ON -DDETACH_KERNEL_DRIVER=ON
make
sudo make install
sudo ldconfig
sudo cp ./rtl-sdr/rtl-sdr.rules /etc/udev/rules.d/
cd ~
sudo nano /etc/modprobe.d/rtl-blacklist.conf
```

Despues editar el archivon rtl-blacklist.conf

```
blacklist dvb_usb_rtl28xxu
blacklist rtl2832
```

Para probar que se instaló correctamente escribimos ahora en terminal

```
rtl_test -t
```

y nos debe de mostrar:

```
Found 1 device(s):
 0:  Generic , RTL2832U, SN: 77771111153705700
```

```
Using device 0: Generic RTL2832U
```

```
Found Rafael Micro R820T tuner
```

```
Supported gain values (29): 0.0 0.9 1.4 2.7 3.7 7.7 8.7 12.5 14.4 15.7
16.6 19.7 20.7 22.9 25.4 28.0 29.7 32.8 33.8 36.4 37.2 38.6 40.2 42.1
43.4 43.9 44.5 48.0 49.6
```

```
Sampling at 2048000 S/s.
```

Info: This tool will continuously read from the device, and report if samples get lost. If you observe no further output, everything is fine.

Reading samples in async mode...

Para sintonizar una estación de FM vía terminal:

```
sudo apt-get install sox
sudo rtl_fm -M wbfm -f 99.3M | play -r 32k -t raw -e s -b 16
-c 1 -V1 -
```

3.2. Instalación GNURadio en Ubuntu

Este paso aunque a primera vista algo trivial no lo es, como es un proyecto Open Source tiene muchas formas de instalación que según se instale puede o no funcionar correctamente. Para solo utilizar módulos básicos y proyectos menores la instalación sería

```
sudo apt-get install gnuradio gnuradio-dev
```

Pero esta versión es algo inestable en especial para la generación de gráficos en WX o QT sin mencionar que utiliza una versión distinta de Volk por lo que agregar módulos tiene muchos errores. Después de probar otros métodos como Pybombs para la instalación decidí construir y compilar todo desde los archivos fuentes. Esto dio buenos resultados y es lo más estable para creación de módulos y manejo de hardware.

```

pi@raspberrypi: ~
File Edit Tabs Help
Found 1 device(s):
.: (raw)
  Encoding: Signed PCM
  Channels: 1 @ 16-bit
  Samplerate: 32000Hz
  Replaygain: off
  Duration: unknown
In:0.00% 00:00:00.00 [00:00:00.00] Out:0 [ | ] Clip:0
0: Realtek, RTL2838UHDIR, SN: 00000001
Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
Tuner gain set to automatic.
Tuned to 99571000 Hz.
Oversampling input by: 6x.
Oversampling output by: 1x.
Buffer size: 8.03ms
Exact sample rate is: 1020000.026345 Hz
Sampling at 1020000 S/s.
Output at 170000 Hz.
In:0.00% 00:00:24.83 [00:00:00.00] Out:786k [ -====|===== ] Clip:0

```

Figure 5: FM en Terminal

3.3. RDS

Esto es lo central del proyecto, el primer paso fue sintonizar alguna estación de FM, para esto se usa bloque RTL-SDR donde se configure la frecuencia FM, el ancho de banda de esta y la ganancia de nuestro RTL2832, para mi caso en particular dada las condiciones de ruido RF (por la cercanía a antenas celulares, redes WIFI, dispositivos Bluetooth y muros) solo dos estaciones FM eran posible sintonizar siempre que se usara una ganancia superior a 40dB. Las estaciones fueron 95.9 y 99.3.

El siguiente paso fue escuchar la estación FM, primero propuse algunos parámetros para el muestro, decimación, ancho de banda y frecuencia de corte, como los utilizo en más de un bloque los use como variables 6 y para controlar la ganancia, volumen y estación (Frecuencia) use controles GUI 7.

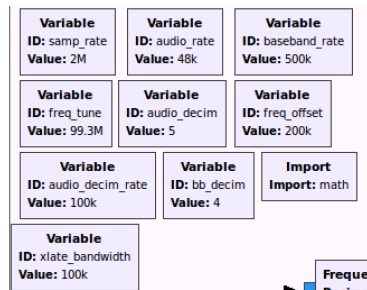


Figure 6: Variables

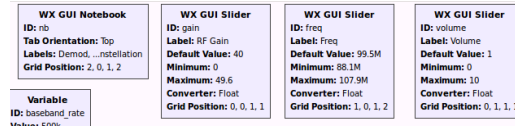


Figure 7: Controles del programa

Ahora para sintonizar la estación utilice el bloque RTL-SDR donde se configura las variables antes mencionadas, a esta señal hay que aplicarle filtro pasabajas para solo tomar el ancho de banda de la señal que necesitamos como para limpiar la señal de cualquier ruido, para esto utilice un filtro FIR pasabajas. Para la demodulación de FM utilizamos el bloque WBFM (FM de banda ancha el ideal para sistemas RDS) esto nos entrega una señal donde podemos observar, la señal FM monoaural, tono piloto, L-R,L+R (señales de estéreo) y la señal RDS centrada en los 57KHz.

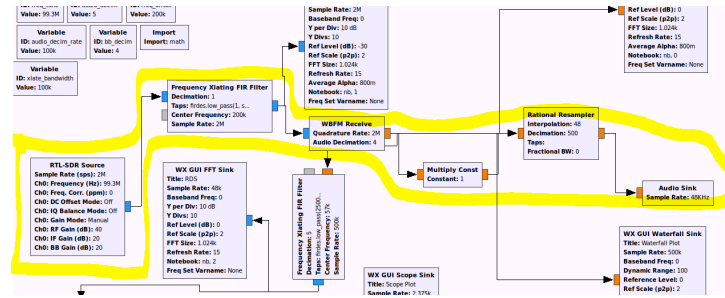


Figure 8: Etapa de Radio FM

El siguiente paso es extraer la portadora RDS de la señal FM demodulada, por lo que se aplica otro filtro FIR pasabajas centrado en 57 KHz, para demodularla a una señal digital. Para esto se usa una técnica llamada "Matched Filter".

El Matched filter es necesario para obtener una señal cuadrada, primero usamos el metodo "root-raised-cosine filter" para sincronizar estos pulsos hay varias técnicas para hacer algo llamado "Costas loop" para esta aplicación se usó un demodulador MPSK, ya en este punto podemos observar la constelación de la modulación. Posteriormente procedemos a hacer un threshold para crear una señal binaria, que se va acumulando en paquetes de bits. Estos son decodificados con dos bloques el primero utiliza código Manchester y va tomando los 16 bits de información de cada bloque del protocolo RDS. A continuación toma las palabras decodificadas y hace una búsqueda en LUTs

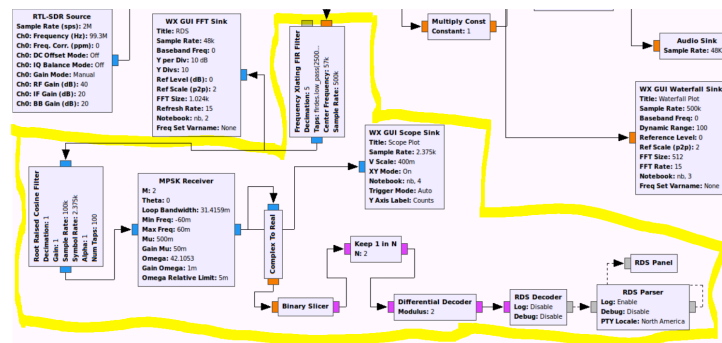


Figure 9: Etapa de procesamiento RDS

de su significado estos dos bloques están C++ creados para otra plataforma de SDR, pero utilizables en GNURADIO y por ultimo van a un bloque de exhibición creado en Python y WX, en las siguientes figuras muestro capturas de pantalla del GUI, las estaciones locales solo transmiten el nombre de estas, pero la plantilla del programa permite extraer y desplegar más información.

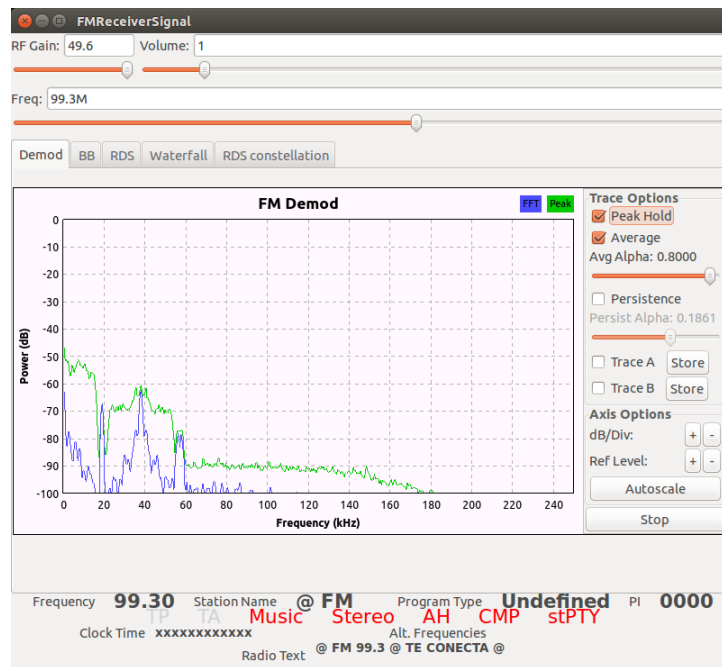


Figure 10: demodulación FM

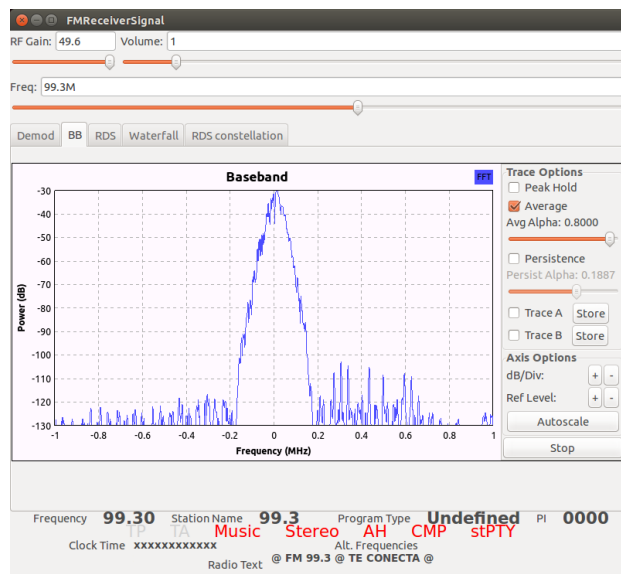


Figure 11: Banda Base FM

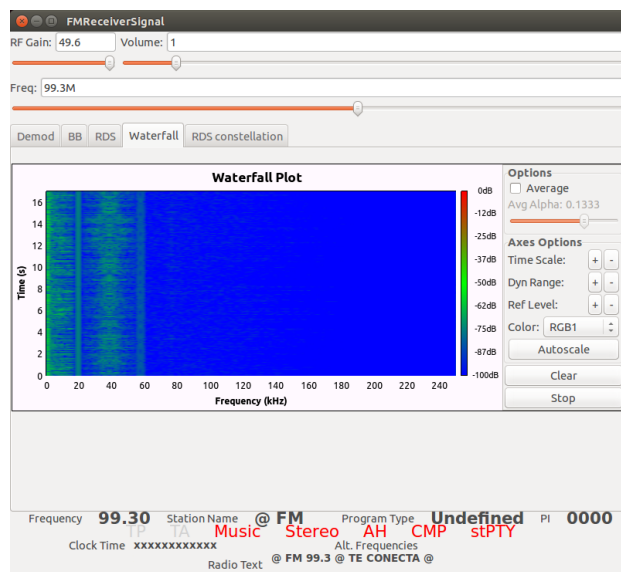


Figure 12: Grafico de Cascada FM

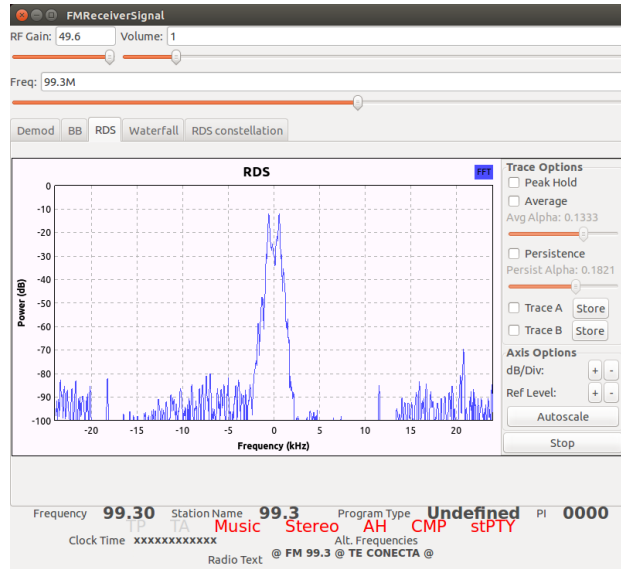


Figure 13: RDS FM

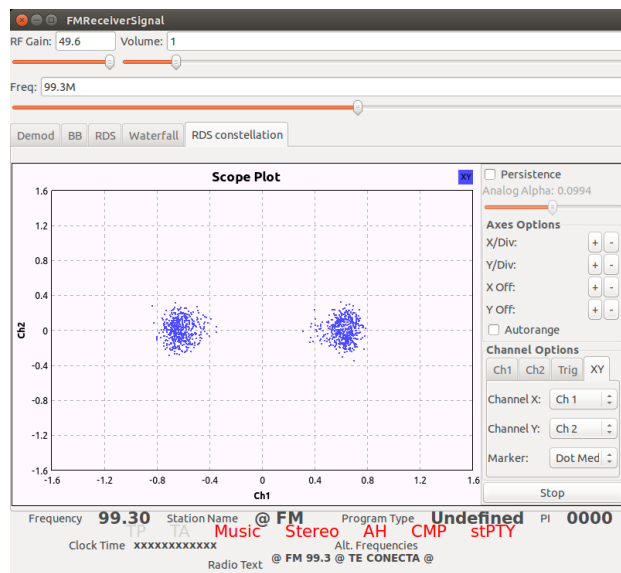


Figure 14: Constelación RDS

3.4. GNURADIO en Raspberry PI

Se intentó ejecutar estos códigos en una computadora Raspberry PI, utilice el OS Raspbian, pero es aquí donde me topé con varias dificultades que me mostraron las debilidades del proyecto GNURADIO, que van desde la instalación. Via apt-get la versión de GNURADIO es una muy anterior a la actual (3.7.5 actual 3.7.9), así que opte por construir y compilar GNURADIO en Raspberry Pi, esto toma alrededor de 48 horas de procesamiento, si aumenta la memoria SWAP al máximo (2GB) se reduce a 24 horas. También intente la instalación via Pybombs que toma otras 24 horas. Una vez instalado hice pruebas de sintonización FM si funcionaron, pero por capacidad de procesamiento solo era viable observar un GUI, así como no reproduce audio ya que hay errores con la librería gqrx, módulos como "root-raised-cosine filter" no están disponibles y tiene errores con la creación de módulos. Revisando de manera más detallada la documentación de GNURADIO Raspberry PI no es la plataforma para aplicaciones de SDR, se recomiendan plataformas como blade de HackRF pero estas plataformas tienen un costo de más de \$ 400.00 dólares. Por lo que a manera personal recomiendo plataformas basadas en PC que pueda ejecutar cualquier versión de Ubuntu superior a la 10.4.