



# UNIVERSIDAD NACIONAL DE LA PLATA

COMUNICACIONES - AÑO 2019

Laboratorio: Demodulación de señales con Dongle SDR1

Pedro Nicolas Abba - 1257/5

## Objetivos

El laboratorio a desarrollar consiste en la utilización de un Dongle USB para el entendimiento de lo que se conoce como Software-Defined Radio (SDR) o Radio Definida por Software. En una primera instancia se hará uso de un software SDR y un dongle proveído por la cátedra para analizar y entender algunas características de este tipo de sistemas de radiocomunicaciones mediante la sintonización de radios FM comercial.

En una segunda instancia se utilizará este dongle y una librería también proveída por la cátedra para la toma de muestras de una estación FM comercial a elección, dichas muestras serán utilizadas para realizar una demodulación mediante el uso de la herramienta Matlab.

## Introducción

En esta sección se hará un introductorio en lo que se conoce como Software-Defined Radio mediante el uso del dongle y el software SDR#.

Antes de comenzar con nuestro laboratorio es de vital importancia entender qué es este nuevo concepto de Software-Defined Radio. El software definido por radio es la implementación en software mediante el uso de un computador o sistema embebido de los componentes de hardware utilizados normalmente en los sistemas de radiocomunicaciones.

Por otro lado se hará un pequeño detalle acerca del dongle que utilizaremos en este trabajo. Dicho dispositivo se encuentra compuesto por dos integrados principales, donde el primero nos permite la sintonización de un rango de frecuencias de 24 MHz a 900 MHz, con una figura de ruido de 3.5 dB y consume alrededor de 180 mA. El segundo integrado que compone al dongle es un ADC y un DSP que realiza la conversión de frecuencia intermedia a banda base a través de mezcladores en fase y cuadratura (I/Q), el filtrado pasa bajos, el re-muestreo en fase y cuadratura y el envío de dichas muestras por el puerto USB.

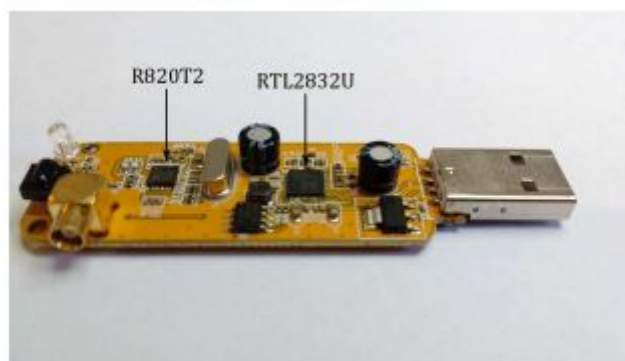


Figura 1 - Circuito Impreso del Dongle

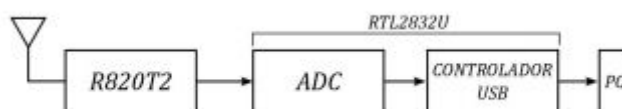


Figura 2 - Diagrama en bloques del Dongle

Una vez instalada los drivers correspondientes para el uso del dongle, me dirijo a abrir el SDR# para ver su funcionamiento (Figura 3).

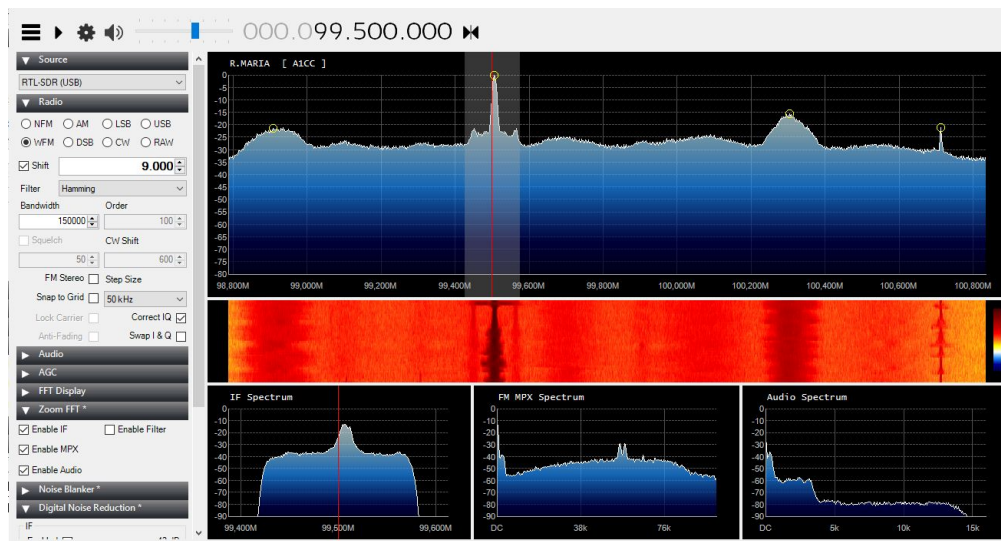


Figura 3 - SDR#

Dicho programa lo utilizaré para la sintonización y demodulación mediante las muestras tomadas a través del dongle de las radio FM comercial para tener un primer acercamiento a lo que es SDR. Al abrir dicho programa teniendo la opción seleccionada WFM (Wide FM) se puede sintonizar las diferentes radios FM comercial que van desde 87,5 MHz a los 108 MHz. (Figura 4)

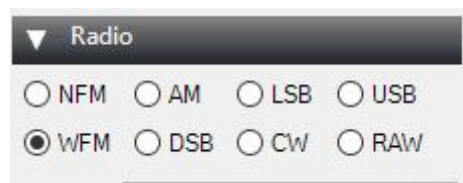


Figura 4 - Wide FM

No detallaré a fondo dicho programa ya que no es el fin de este laboratorio pero haré hincapié en la primera gráfica donde se puede ver el espectro de frecuencia que está sintonizando el dongle (Figura 5). En dicho espectro se puede detectar rápidamente aquellas frecuencias donde se encuentran las diferentes radios comerciales debido a los picos de frecuencia que se puede notar en todo el espectro de frecuencias que se visualiza. Rápidamente se puede observar que en algunas radios no se tiene buena señal y estas se confunden con el ruido blanco lo que produce que no se escuchen claramente.

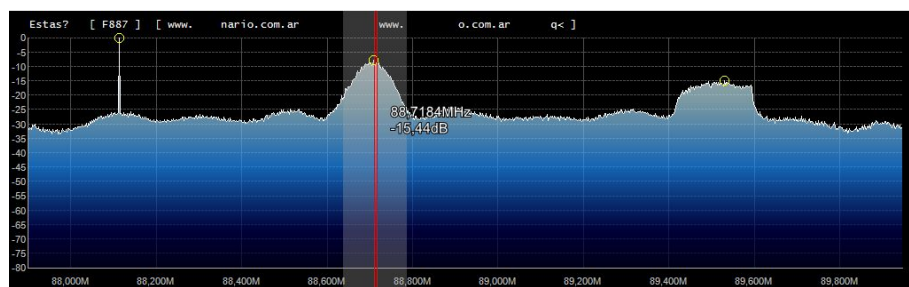


Figura 5 - Espectro de frecuencias

Mediante la visualización del espectro de frecuencias podré seleccionar alguna de las radios FM comercial de las cuales se tiene una buena potencia de señal recibida para el muestreo y su posterior demodulación mediante Matlab.

Además el programa permite variar el ancho de banda del filtro utilizado, siendo este de 250 KHZ por defecto (Figura 6). Esto será de gran utilidad más adelante para corroborar si el ancho de banda que elegiré para el primer filtro en la demodulación es el correcto o no, escuchando dicha radio mediante el mismo valor de ancho de banda.

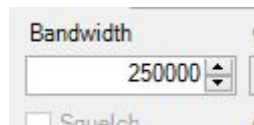


Figura 6 - Valor actual del ancho de banda del filtro

Por otro lado también se tiene la posibilidad de cambiar manualmente los valores de ganancia del SDR, desmarcando la opción Tuner AGC y cambiando el RF Gain (Figura 7).

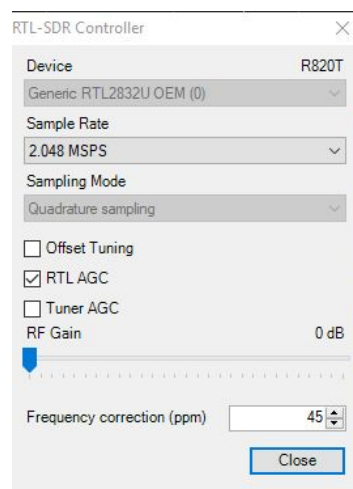


Figura 7 - Configuraciones del SDR

Si se define un valor de ganancia de 20 db se puede apreciar la aparición de otros picos de frecuencias en el espectro producto de esta acción, y si se intenta sintonizar en las frecuencias donde se encuentran estos nuevos picos lo único que se puede escuchar es ruido con algunas interferencias de dos estaciones, o directamente puramente ruido (Figura 8). Por otro lado si se aumenta bastante la ganancia, estos picos desaparece ya que además de aumentar la potencia de estas señales también se aumenta la potencia del ruido por lo que se termina difuminando con este mismo, y por otro lado disminuye la relación señal a ruido (Figura 9).

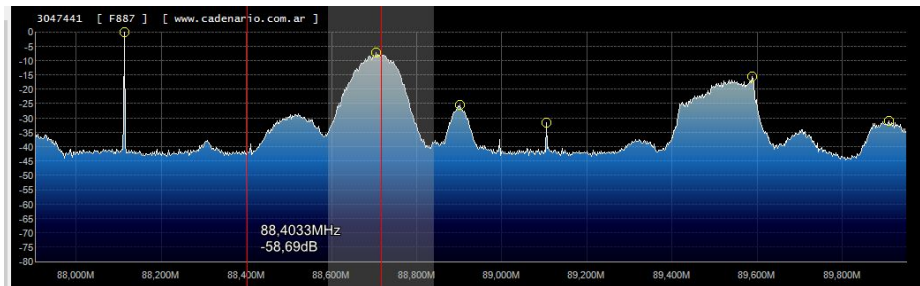


Figura 8 - Espectro con ganancia 20db

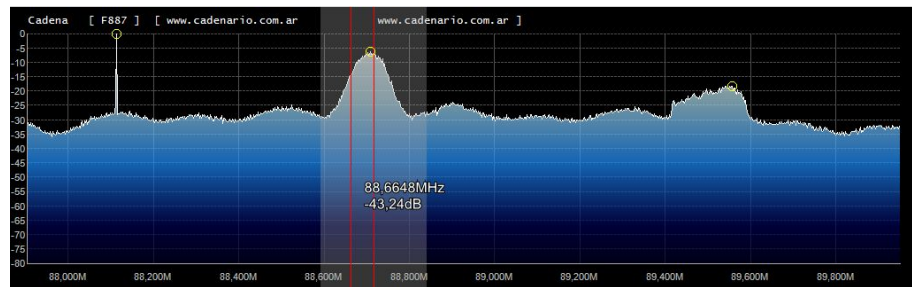


Figura 9 - Espectro con máxima ganancia posible

## Laboratorio

Para el muestreo y demodulación que realizaré en este trabajo seleccioné la radio 88.7, la misma que se puede ver en el espectro de frecuencia mostradas en las imágenes anteriores, dado que dicha radio posee una muy buena potencia de señal recibida.

Comenzaré con el uso de la librería rtl-sdr-release que se encarga de tomar las muestras en fase y cuadratura de la radio seleccionada (Figura 10). Al ejecutar este comando se comenzará a muestrear a una frecuencia de 2.048 MHz (la cual corresponde a la frecuencia de muestreo por defecto), generando  $21 \times 10^6$  muestras en un archivo con nombre a elección nuestra (para este caso muestras-88.7).

```
rtl_sdr.exe -f 88.7e6 -n 21e6 muestras-88.7
```

Figura 10 - Uso de la librería para tomar muestras de la radio 88.7

En este laboratorio haré uso de dos funciones en matlab que fue proveída por parte de la cátedras las cuales una permite cargar el archivo de muestras al programa y la otra permite dibujar la densidad espectral de potencia de la señal que deseo (Figura 11 y Figura 12).

```
1 function y = loadFile(filename)
2 % y = loadFile('nombreadarchivo')
3 %
4 % Lee el archivo generado por rtl_sdr.exe y entrega un vector
5 % complejo (I y Q)
6 fid = fopen(filename,'rb');
7 y = fread(fid,'uint8=>double');
8 y = y-127;
9 y = y(1:2:end) + i*y(2:2:end);
10 end
```

Figura 10 - Función en matlab que lee un archivo de muestras y genera un vector

```

function [f, Sxx] = DEP(x,fs,signal_name)
% [f, Sxx] = DEP(x,fs,signal_name)
% INPUTS:
% x = se~nal de entrada
% fs = frecuencia de muestreo
% signal_name = cadena de caracteres para el titulo. Ej. 'Se~nal antenna'
% OUTPUTS:
% f = vector de frecuencias donde se evalu'o la DEP
% Sxx = Estimacion de la DEP
[Sxx,f] = pwelch(x,ones(1,8192),0,[],fs,'twosided');
Sxx = fftshift(Sxx); f = fftshift(f);
f(1:floor(length(f)/2)) = f(1:floor(length(f)/2)) - fs;
handle = figure('units','normalized','outerposition',[0 0 1 1]);
plot(f,Sxx); xlim([-fs/2 fs/2]);
set(gca, 'FontSize', 18);
legend('S_{XX}(f)', 'Location', 'NorthEast'); grid on;
xlabel('f [Hz]', 'Interpreter', 'Latex', 'FontSize', 20);
title(['DEP de ' signal_name], 'FontSize', 20);
grid on;
end

```

Figura 11 - Función en matlab que recibe una señal, su frecuencia de muestreo y un título y gráfica y entrega su densidad espectral de potencia

Como primer paso grafiqué la densidad espectral de potencia de la señal muestreada (Figura 12), en dicha gráfica se puede apreciar además de nuestra señal centrada en cero, el espectro de otras señales que se metieron también en nuestras muestras las cuales debería filtrar mediante un primer filtro de un cierto ancho banda que a continuación comentaré.

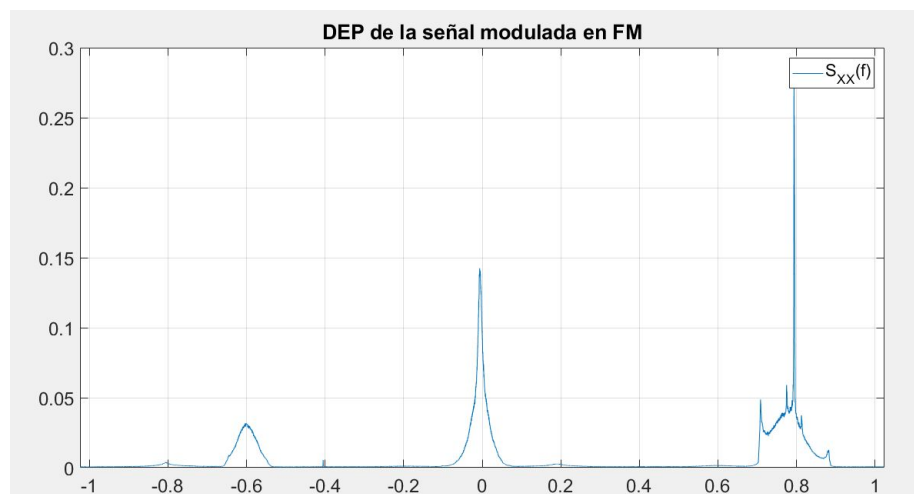


Figura 12 - Densidad espectral de potencia de la señal muestreada modulada en FM

Realicé una búsqueda para conocer el ancho de banda que tiene cada canal de emisión en FM comercial donde me encontré que cada canal ocupa un ancho de banda de 200 KHz teniendo además 25 KHz de ancho de banda para separación entre las demás estaciones de FM comercial (Figura 13). Dicho esto se puede concluir que nuestro ancho de banda útil termina siendo de 150 KHz, esto me permitirá definir nuestra frecuencia de corte para el primer filtrado que realizaré mediante los comandos ya conocidos butter y filter con una frecuencia de corte a 75 KHz.



Figura 13 - Ancho de banda de una estación de FM comercial

Para corroborar lo dicho anteriormente realicé un acercamiento de la gráfica de la densidad espectral de potencia de la señal y pude corroborar a simple vista que dicho ancho de banda me permitirá quedar con lo necesario para demodular la señal de audio (Figura 14). Por otro lado también hice uso del software SDR comentado anteriormente para corroborar que dicho ancho de banda me permite escuchar correctamente la estación de radio elegida y definitivamente así lo es.

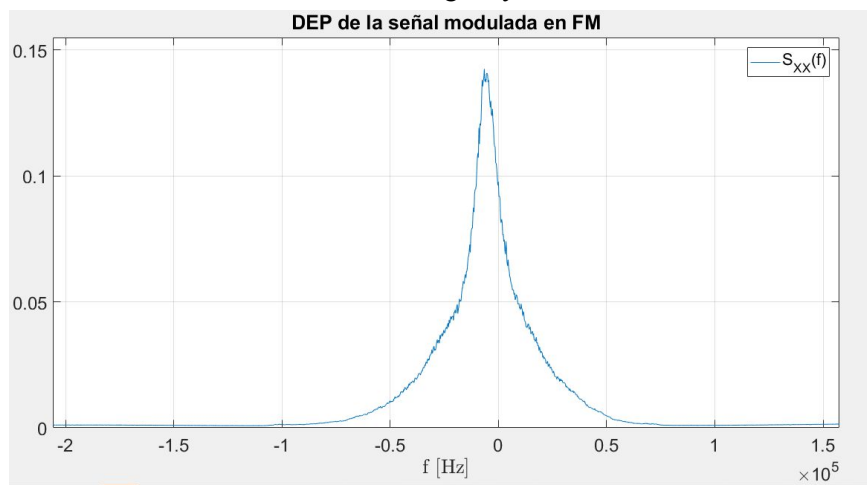


Figura 14 - Acercamiento de la gráfica de la densidad espectral de potencia de la señal de audio

Una vez realizado el filtrado anteriormente dicho obtengo una nueva señal donde tengo el ancho de banda necesario para demodular la señal en FM (Figura 15).

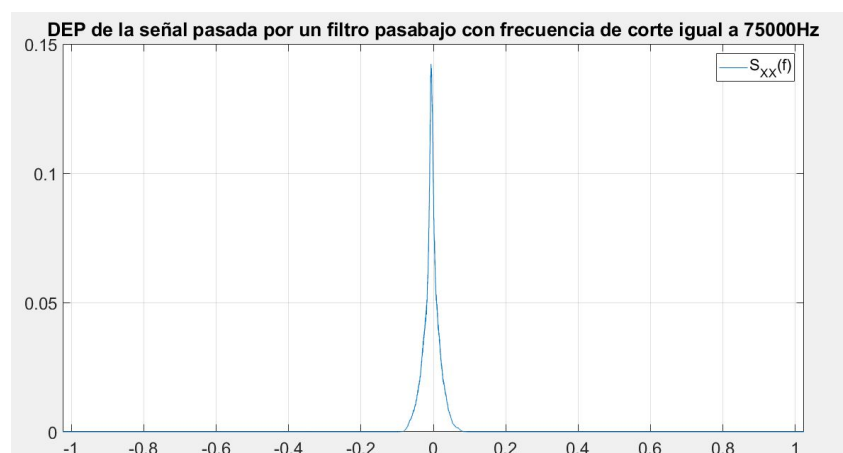


Figura 15 - Densidad espectral de potencia de la señal pasada por un filtro pasabajo con frecuencia de corte a 75 KHz

Para ahorrar operaciones y tiempo de ejecución voy a realizar un diezmado siguiendo la fórmula de Nyquist con el nuevo ancho de banda que tengo, y la frecuencia de muestreo conocida para así calcular el valor N1 que utilizaré para diezmado.

$$\frac{F_s}{N1} > 2 \times B1$$

Despejando dicha fórmula y utilizando los valores conocidos, N1 termina dando 13,6533, dado que este valor debe ser entero redondeo hacia abajo (debido a que la desigualdad es menor) obteniendo un valor igual a 13.

Una vez obtenido el valor de veces a diezmado procedo con la operación haciendo uso de la de la función en Matlab decimate y así obtengo la nueva señal con la mínima información que requiero (Figura 16). A partir de esta instancia tendré una nueva frecuencia de muestreo de valor igual a la anterior dividido por el número de veces a diezmado.

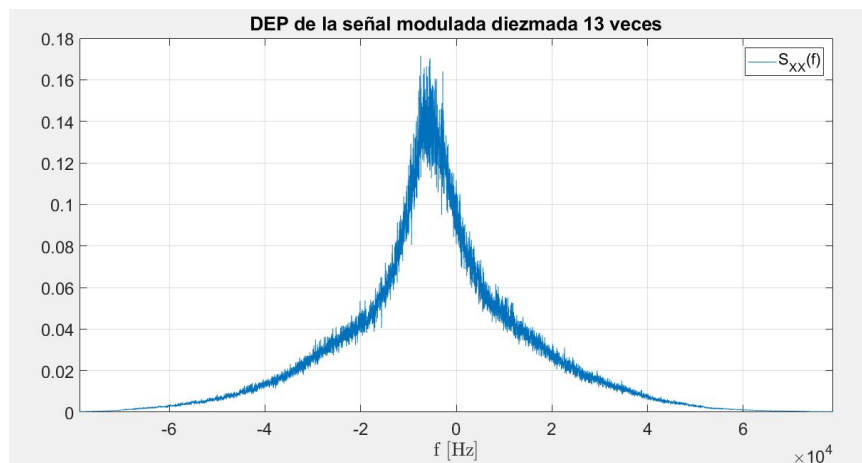


Figura 16 - Densidad espectral de potencia de la señal diezmada 13 veces

A esta altura ya tendré la mínima información necesaria de la estación de FM elegida para demodular. Para nuestro demodulador haré uso del código de un discriminador de fase que me fue otorgado en unas de las prácticas, pero a diferencia de este hay que tener en cuenta un pequeño detalle dado que las muestras se encuentran en fase y cuadratura. Esto significa que las muestras se están tomando intercaladas desfasada a 90 grados con respecto a la anterior, por lo que relacionado esto con el cos y el seno o mejor dicho los reales y complejos necesito obtener su ángulo.

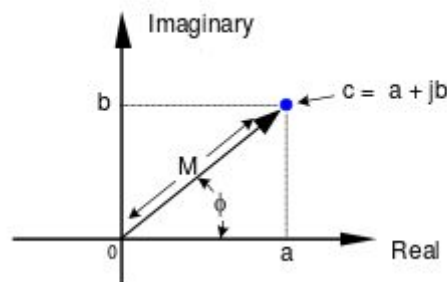




Figura 17 - Componente en fase y cuadratura (Imaginario y real).

El código del discriminador de fase que utilizaré será el mostrado en la Figura 18 donde se puede apreciar el uso de la función `angle` para obtener el ángulo de cada muestra y por último le quito el valor medio haciendo uso de la función `mean` de MatLab.

```
xd = unwrap(angle(y_N1));  
xd = [0; diff(xd).*fsN1];  
yd = xd-mean(xd);
```

Figura 18 - Código que realiza la demodulación en FM mediante un discriminador de fase

Una vez demodulada procedo a graficar la señal para ver que obtuve (Figura 19).

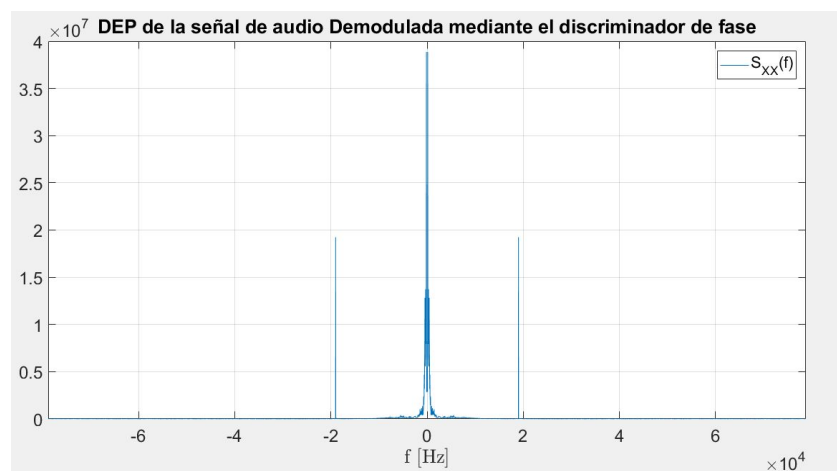


Figura 19 - Densidad espectral de potencia de la señal demodulada en FM

Lo primero que se puede notar son dos deltas aproximadamente cercanas a 20 KHz y - 20 KHz, exactamente en 19 KHz y -19KHz correspondiente al mensaje Piloto. La señal piloto nació con la necesidad de poder transmitir sonido estéreo mediante la radio por el mismo canal que se venía usando para el sonido mono de manera que las radios monofónicas que ya estaban en el mercado pudiesen recibir correctamente estas transmisiones estéreo. Lo que se hizo fue agregar un tono sinusoidal de baja potencia con la información necesaria de la frecuencia y la fase para reconstruir la señal estereofónica y desplazarla a una frecuencia inaudible, y luego que el mismo receptor la vuelva a trasladar hasta las frecuencias audibles. En la figura 20 se puede apreciar que entre 0 y 15 KHz está la señal monofónica producto de la suma de las señales sonoras correspondiente al oído derecho e izquierdo, mientras que entre 23 y 53 KHz (centradas a 38 KHz) se tiene la señal de la diferencia entre el oído izquierdo y el derecho.

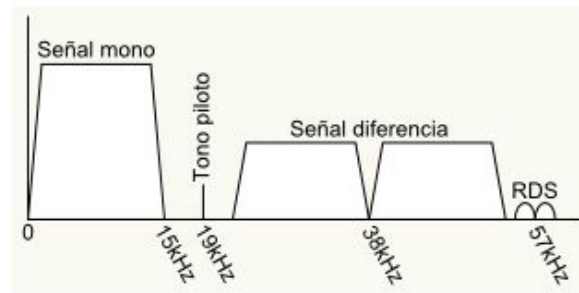


Figura 20 - Representación de una señal estereofónica

Por lo que si L y R fueran las señales correspondientes al oído izquierdo y derecho, tendría M siendo la suma de L y R y por otro lado tendría D siendo su diferencia. Dicho esto, si quisiéra reconstruir la señal correspondiente al oído izquierdo o derecho basta con realizar su suma o resta como se muestra a continuación:

$$M + D = L + R + L - R = 2L$$

$$M - D = L + R - (L - R) = 2R.$$

Para poder restar o sumar la señal mono con la diferencia debería trasladar la señal diferencia que se encuentra a 38 KHz a 0. Pero para poder realizar esto me encuentro con dos problemas, primero sería difícil generar una senoidal que tuviera la misma frecuencia que el oscilador que desplazó la señal diferencia y por el otro que el oscilador receptor de radio y el de la emisora puedan estar fuera de fase. Aquí es cuando entra en juego la señal piloto la cual está encargada de transmitir dicha frecuencia y fase para poder lograr el traslado de la señal diferencia.

Ya explicado y entendido lo anterior procedo con la demodulación de la señal FM, lo que restaría sería realizar un segundo filtrado conociendo el ancho de banda de la señal y por último un diezmado para llevar la frecuencia de muestreo a una que pueda soportar la placa de audio.

El ancho de banda de las señales de audio en FM equivale a 15 KHz, por lo tanto tendré nuestra frecuencia de corte a 7500 Hz, dicho esto procedo a realizar el filtrado obteniendo una nueva señal (Figura 21).

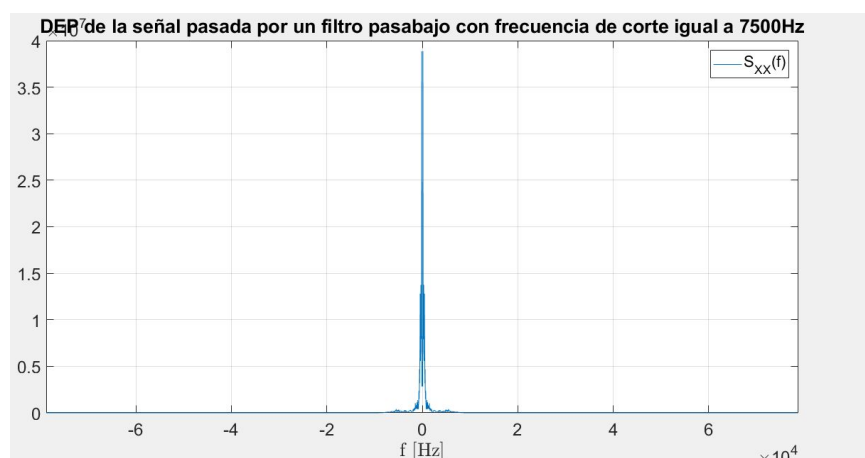


Figura 21 - Densidad espectral de potencia de la señal demodulada en FM con un filtro pasabajo con frecuencia de corte igual a 7500 Hz

Por último resta realizar un diezmado para llevar la frecuencia de muestreo a 48 KHz para poder reproducir la señal de audio sin problema, debido a que a frecuencias mayores puede que nuestra placa de audio no lo soporte. Para realizar este diezmado procedo con la misma fórmula anteriormente usada para calcular nuestro nuevo N2 y diezmar dicha señal (Figura 22).

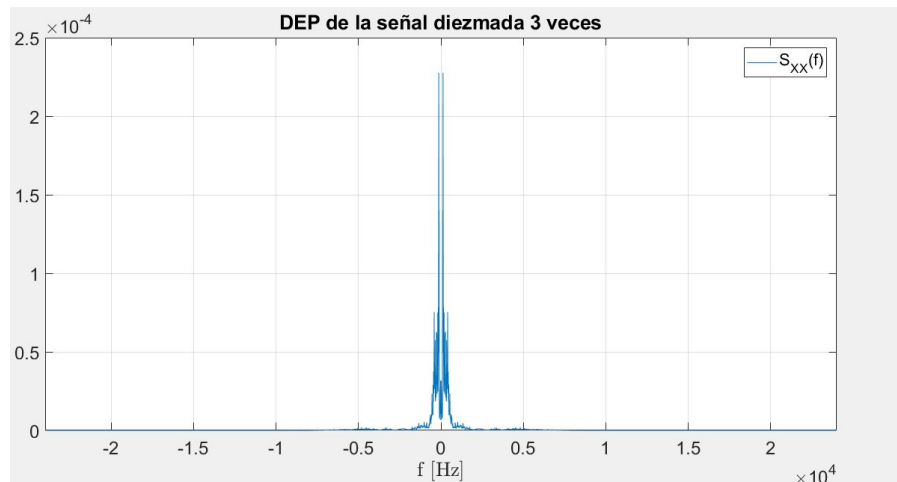


Figura 22 - Densidad espectral de potencia de la señal demodulada en FM diezmada 3 veces

Por último antes de escuchar la señal que obtuve mediante nuestra demodulación por software a partir de la función sound de Matlab, decidí normalizar la señal dividiéndola por su máximo para que no existan valores atípicos en dicha señal y su amplitud máxima sea 1 (Figura 23).

```
z_out = y_N2./max(abs(y_N2));
```

Figura 23 - Se normaliza la señal dividiendo la misma por su máximo

Efectivamente al escuchar la señal que obtuve se puede oír la transmisión de la radio que elegí donde se escucha la publicidad aparentemente de una venta de automóviles.

## Conclusión

Al realizar este laboratorio primero se encuentra con un nuevo concepto que desconocía acerca de Software-Defined Radio, lo cual me parece increíble ya que como ingeniero en computación nos abstraemos de la necesidad de obtener los componentes de hardware necesarios para realizar una demodulación en FM o en cualquier otro tipo de modulación, teniendo como único elemento algo que para cualquier persona puede ser accesible que viene a ser un computador. Además suponiendo que los componentes de hardware resulten baratos, el realizar el circuito necesario para demodular en un dado sistema de modulación conlleva más tiempo de lo que tardaría en hacerse mediante software, sumándole que este circuito será específico para cierta modulación por lo que si quisiera demodular en otro tipo debería confeccionar un nuevo circuito, lo cual con el software no pasa ya que con el mismo

computador podría crear un programa para cualquier tipo de modulación sin necesidad de cambiar el hardware.

Por último quiero destacar que me resultó muy interesante el nuevo concepto del mensaje piloto y entender cómo funciona la transmisión en estéreo y mono. Esta idea me pareció bastante ingeniosa de parte de quien la haya creado para poder transmitir estéreo sin la necesidad de cambiar los receptores mono, aquí claramente se puede ver lo que nos caracteriza principalmente como ingenieros, nuestra capacidad de adaptarnos a cualquier situación.

## Bonus

Como bonus extra se pidió intentar realizar la demodulación de la estación comercial de FM en tiempo real con algún tipo de interfaz que permita tanto variar la estación seleccionada, como el volumen de la misma u otro parámetros.

Para realizar dicho trabajo hice uso de GNURadio el cual es un kit de herramientas de desarrollo de software gratuito y de código abierto que proporciona bloques de procesamiento de señales para implementar radios de software.

En esta parte del informe no haré una explicación detallada de cómo se realiza la demodulación en FM dado a que sigue la misma lógica anteriormente usada en Matlab, con la única diferencia que fue trasladada a GNURadio. En la figura 23 se puede apreciar el diagrama en bloques del sistema de demodulación de FM en tiempo real. El primer bloque a configurar es Options donde pude definir el id, título y autor del código, además de modificar el tipo de interfaz gráfica a utilizar siendo esta WX GUI.

Como parte de la interfaz gráfica tendré la posibilidad de modificar el volumen, la ganancia y la estación a demodular además de poder visualizar el espectro de frecuencias recibido por el dongle y el espectro de frecuencia correspondiente a la salida una vez realizada la demodulación.

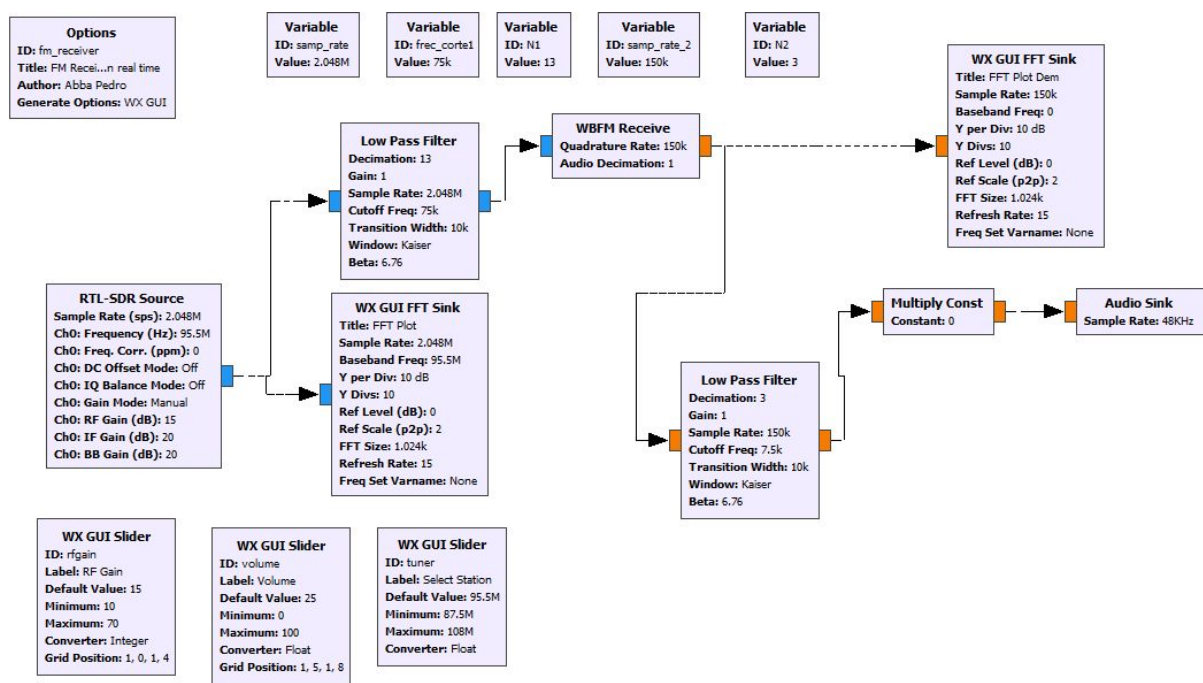


Figura 23 - Diagrama en bloques del sistema de demodulación en tiempo real

Los primeros tres bloques (Figura 24) que explicaré corresponden a la interfaz gráfica para poder modificar tanto el volumen, la ganancia y la estación seleccionada. Para estos tres casos hice uso del mismo tipo de bloque (WX GUI Slider) donde me permite elegir un valor mínimo, máximo, el valor por defecto y el número de pasos, además de su posición en la interfaz.

En el caso de la ganancia su mínimo decidí que sea 10, mientras que su máximo es 70 con un número de paso de 12 y en el caso del volumen, su mínimo es 0 y su máximo es 100 mientras que su número de paso es 100, los valores anteriormente dicho no tienen alguna lógica en sí, fueron seleccionados a preferencia mía. En el caso de las estaciones como es ya conocido su valor mínimo será de 87.5 Mhz y ira hasta 108.0 Mhz y podré ir sintonizando saltando cada 100000 Hz por lo que me da un número de paso de 205.

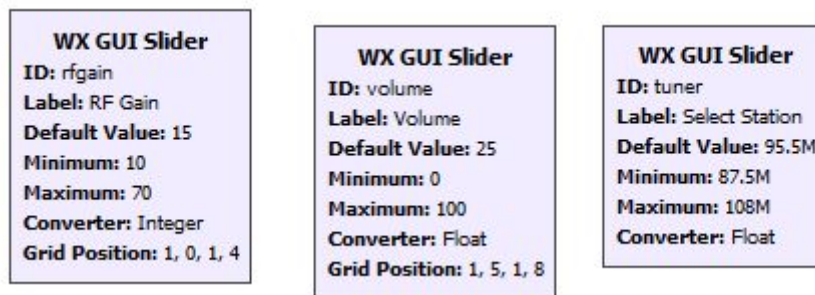


Figura 24 - Diagrama de bloques del volumen, ganancia y estación a seleccionar

El siguiente bloque a explicar es el encargado de conectarse con el dongle y captar las diferentes estaciones de FM (Figura 25), en su configuración tuve que especificar la frecuencia de muestreo siendo esta 2.048 MHZ, la frecuencia donde se centra para sintonizar correspondiente a la elegida por el usuario y la ganancia también seleccionada por el usuario, los demás valores quedaron por defecto.

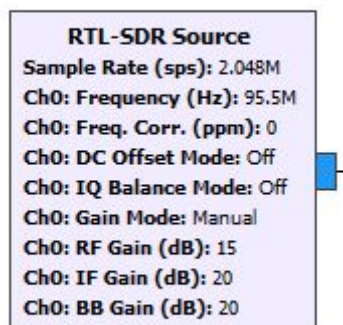


Figura 24 -Bloque RTL-SDR

Dicho bloque anteriormente explicado se conecta a un filtro pasabajo y un bloque FFT que realiza la transformada rápida de fourier y la visualiza en tiempo real (Figura 25).

El filtro pasabajo además de realizar su función principal me permite también hacer un diezmado, para esto utilizo los valores ya conocidos donde la frecuencia de corte será de 75000 y se realizará un diezmado con un valor de 13 (sale de realizar la división de la frecuencia de muestreo dividido dos veces la frecuencia de corte), por último debo indicarle la frecuencia de muestreo utilizada.

En el caso del bloque FFT lo único que tuve que indicarle es la frecuencia de muestreo y la estación seleccionada por el usuario.

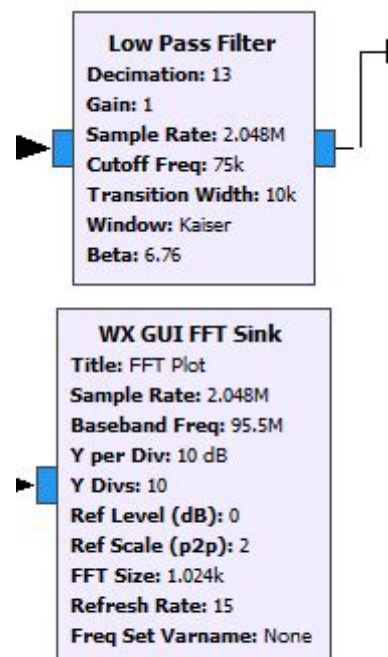


Figura 25 -Filtro Pasa Bajo y FFT

El siguiente bloque (Figura 26) es el encargado de realizar la demodulación en FM, donde lo único que hay que indicarle es la frecuencia de muestreo, dicha frecuencia sale de realizar la división de la frecuencia anterior y N1 (número del primer diezmado) siendo esta de 150000 Hz.

A la salida de dicho bloque tendré también un FFT donde la frecuencia de muestreo es la anteriormente dicha y en este caso la frecuencia en banda base es 0 debido a que la señal ya se esta en banda base. Por otro lado también se conecta a un filtro pasabajo que al igual que en Matlab realiza un filtrado con una frecuencia de corte en 7500 Hz y un valor de diezmado de 3 (el cual se obtiene para poder reproducir el audio a una frecuencia de muestreo de 48Khz).

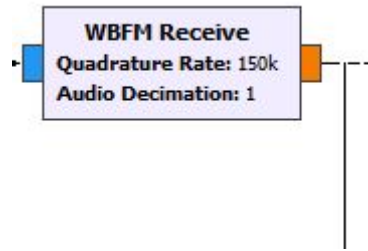


Figura 26 -Bloque Demodulador de FM

Lo único que resta es multiplicar la señal por una constante la cual se relaciona con el volumen elegido por el usuario , y a la salida de dicho bloque se encuentra el bloque encargado de reproducir dicha señal a 48 Khz (Figura 27).

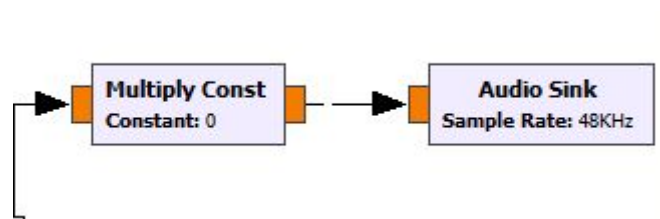


Figura 27 - Bloque encargado de modificar el volumen del audio y reproducirlo a 48 Khz.

Con todos estos bloques explicados conseguí una demodulación de FM comercial en tiempo real como se puede apreciar en la Figura 28 (Debido a que la capacidad de mi PC no me lo permite solo mostrare una FFT a la vez).

Lo primero que se puede destacar es que al igual que el SDR# se puede ver en tiempo real aquellas estaciones que tienen mayor potencia para poder sintonizar y por otro lado se puede observar también el efecto que produce la elección de la ganancia al igual que en el SDR# comparando la Figura 28 con la Figura 29 donde se aprecia un aumento tanto de la potencia de las estaciones como del ruido. Algo importante a aclarar es que en todos los bloques anterior a la demodulación se está trabajando con números complejos mientras que una vez realizada la demodulación los demás bloques trabajan con números flotantes debido a que las muestras se toman en fase y cuadratura.



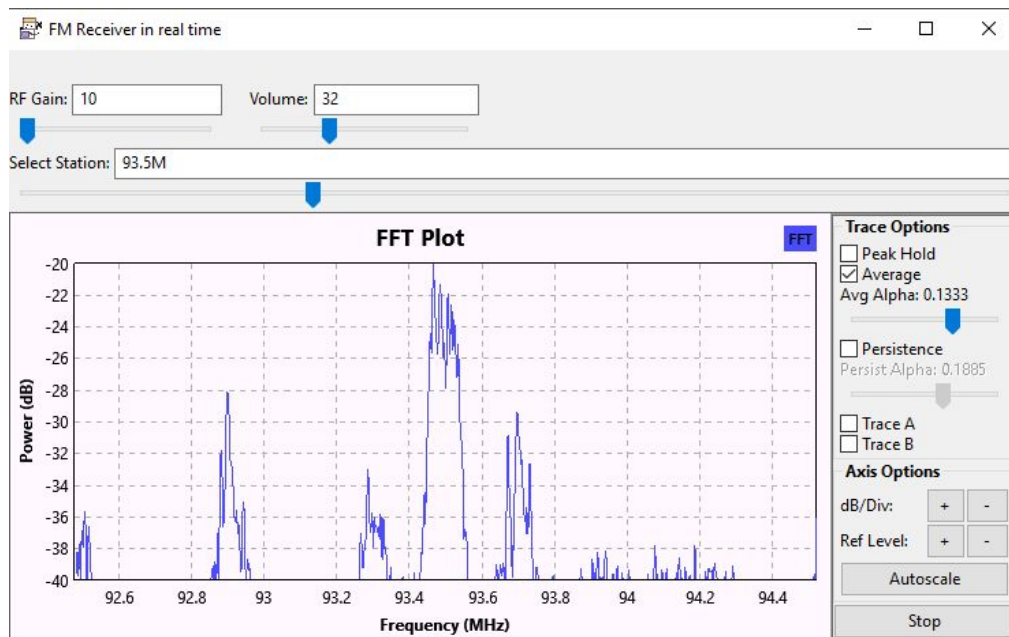


Figura 28 - Recepción de la estación 93.5 con una ganancia de 10 db

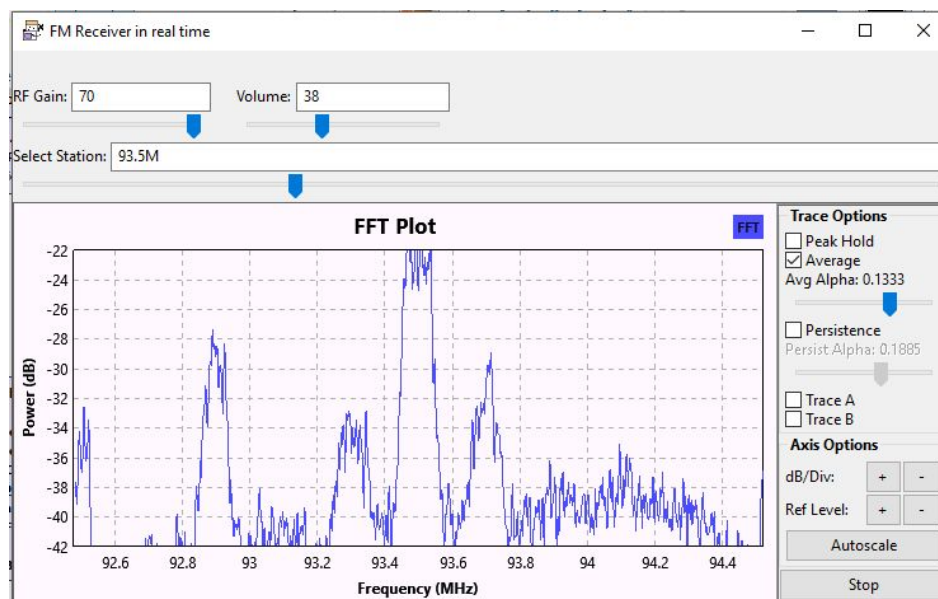


Figura 29 - Recepción de la estación 93.5 con una ganancia de 70 db

Otro punto interesante que se puede apreciar al graficar la FFT de la salida del demodulador es que al igual que en Matlab en aquellas estaciones estéreo se puede ver una primera parte correspondiente al mono y una delta en 19 KHz correspondiente al tono piloto (Figura 30).



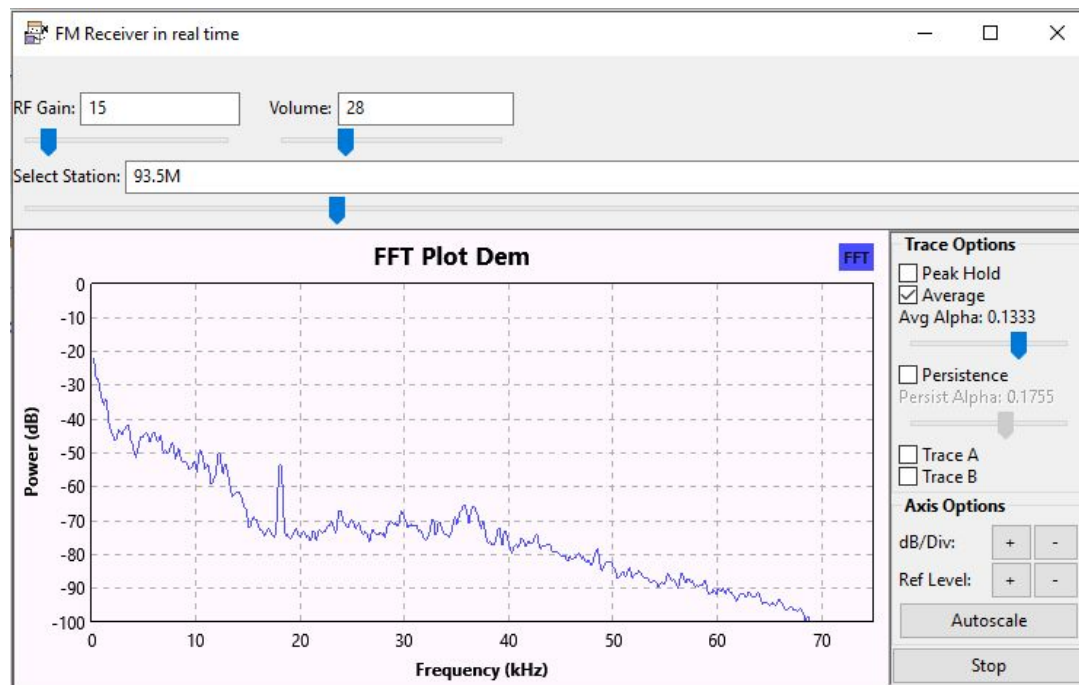


Figura 30 - Recepción de la estación 93.5 con la gráfica de la FFT de la salida

## Conclusión

Al realizar el trabajo tanto en Matlab como en GNURadio me di cuenta que dicha librería tiene un gran potencial en el procesamiento de señales, no sé si superior a Matlab pero nos otorga una gran facilidad y rapidez al momento de realizar diferentes programas. Si tuviera que elegir con qué herramienta quedarme definitivamente elegiría GNURadio dado su simplicidad y gran cantidad de herramientas que aún desconozco, pero investigando en internet vi que se puede lograr grandes cosas, además de que dicha herramienta sirve específicamente para implementar radios de software por lo que será mucho más fácil y realizable que con Matlab.