

DEMODULACIÓN FM

1. INTRODUCCION

Si recordamos lo visto en clase, FM se define como una modulación angular, concretamente una modulación en frecuencia; esto es, mediante variaciones en la frecuencia de la portadora se puede codificar la señal analógica para transmitir la información.

La modulación FM es común en altas frecuencias por la alta fidelidad de su difusión, por lo que la radio o el propio sonido de la televisión es difundida por FM. Presenta mejor relación señal a ruido (SNR) que la AM y mayor resistencia a las interferencias, tan presentes en AM.

Dentro de las aplicaciones de FM se encuentra la radio, en donde los receptores emplean un detector de FM y el sintonizador es capaz de recibir la más fuerte de las señales que transmiten en una frecuencia. Veamos cómo podemos crear nuestro propio receptor de radio FM.

2. DEMODULACION CON GNURADIO Y HACKRF

Las modulaciones FM pueden modelarse del mismo modo que las de PM, pero con un integrador a la entrada de la señal moduladora; o con un VCO realimentado (PLL), lo que minimiza el error pero vuelve el circuito más caro. No obstante, nos interesa saber cómo demodular, pues de la difusión de radio nos llega ya la señal modulada en FM.

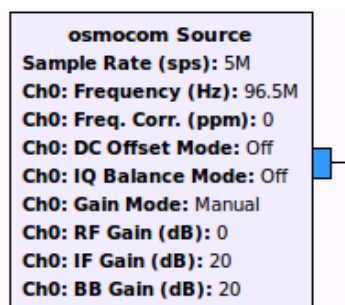
Para demodular una señal FM tenemos dos opciones:

- **Discriminador reactivo:** conseguimos que la señal de salida aparezca modulada en amplitud (AM) y aplicamos un detector de envolvente.
- **Detector de fase con PLL.**

Primeramente debemos conocer el espectro de la radio FM, comprendido entre los 85 MHz y los 107 MHz aproximadamente. Cada uno de los canales de radio FM suelen tener un ancho de banda alrededor de los 200 KHz para evitar interferencias entre ellas (cada emisora no suele ocupar más ancho de banda que 80 KHz).

A continuación presentaremos cómo implementar nuestra demodulación FM en GNU Radio. Cabe destacar que todos los parámetros que aparecen en las figuras son orientativos, se deben calcular además de todos los resultados que queramos obtener y justificarlos.

El primer paso, como hicimos en la práctica anterior, será crear nuestro bloque Osmocom Source para poder captar la señal recibida por HackRF.:



Options

ID: top_block

Generate Options: WX GUI

Variable

ID: samp_rate

Value: 5M

Variable

ID: center_freq

Value: 96.5M

Variable

ID: channel_freq

Value: 96.5M

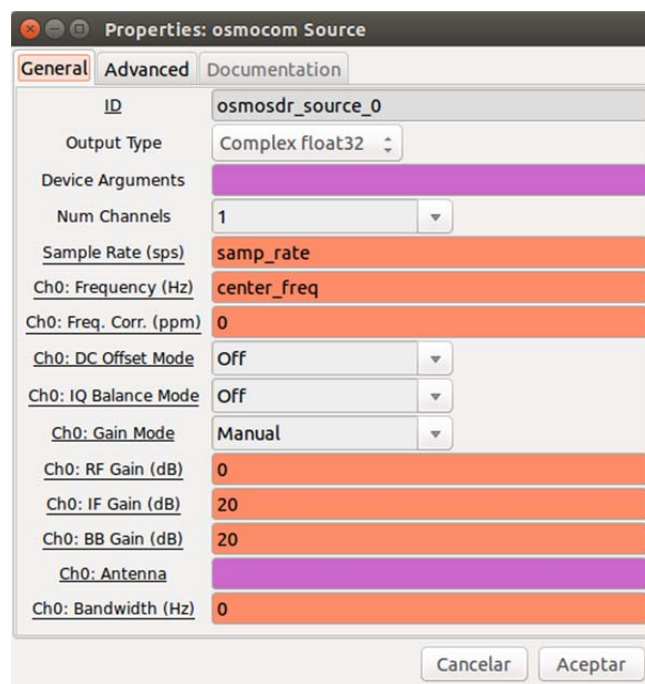
Variable

ID: channel_width

Value: 200k

Aparte de nuestra variable *samp_rate* (ya conocida), creada para dar un valor definido al span visualizado en la FFT, ahora creamos otras tres variables para las frecuencias que nos permitirán movernos con facilidad por la banda FM (85-107 MHz).

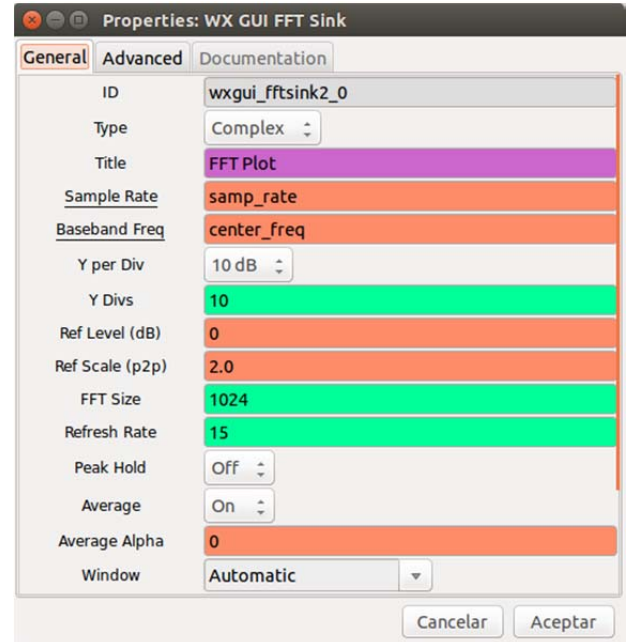
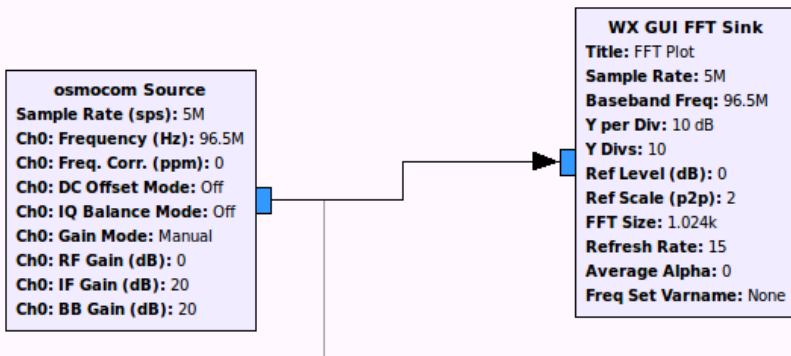
De este modo, Osmocom Source queda configurado de manera similar a:



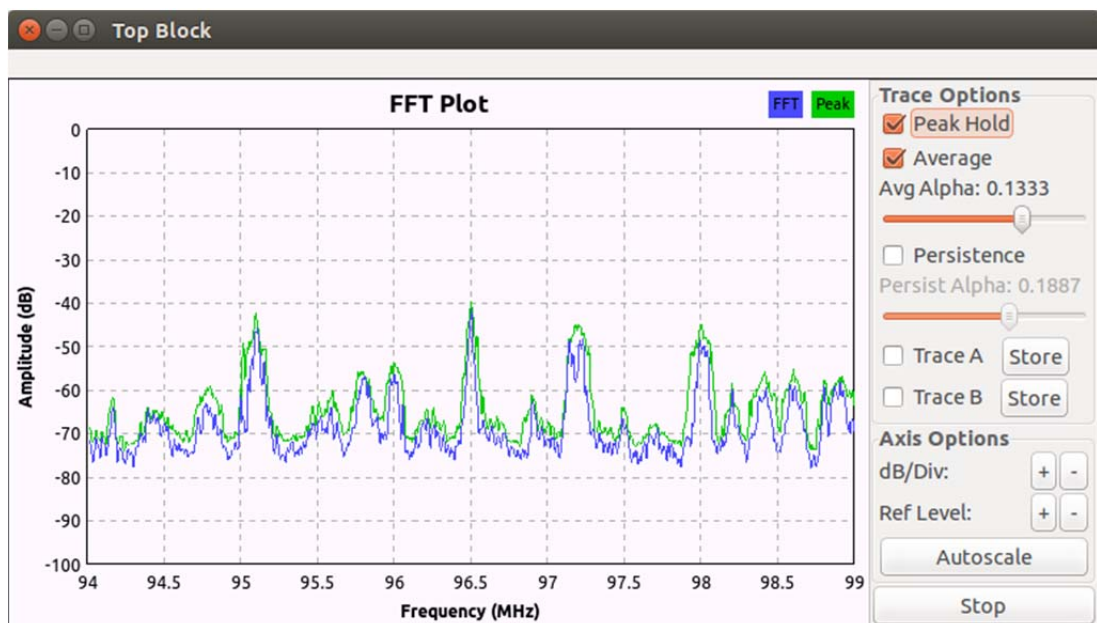
Realmente podríamos haber dado cualquier valor entre 85 y 107 MHz a la variable *center_freq*, simplemente hemos escogido este valor por estar en el centro de la banda FM. Sin embargo, es interesante introducir otros valores y observar qué sucede.

Las otras dos variables las utilizaremos más adelante.

Para estudiar lo que ocurre de forma visual, debemos observar el espectro de la señal. Para ello, hacemos uso del bloque WX GUI FFT SINK y lo unimos con nuestra fuente para ver qué estamos captando y cambiar los parámetros si hiciera falta.



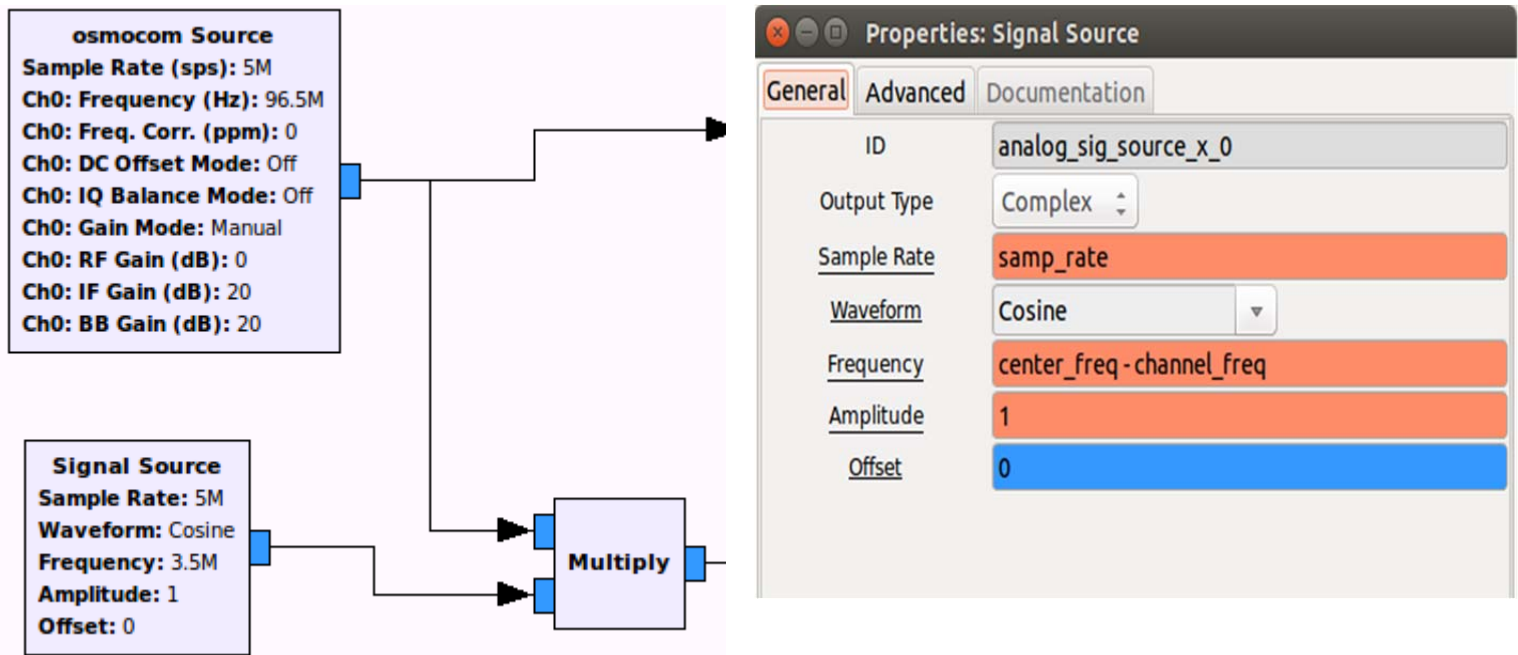
Vemos que las propiedades del WX GUI FFT SINK deben estar acordes con el Osmocom Source. Si representamos la señal veremos:



Podemos observar con claridad una serie de deltas (portadoras) centradas en unas frecuencias determinadas. Cada una de estas deltas corresponde a las emisoras de radio que conocemos. Nuestro objetivo es demodularlas y conseguir escucharlas.

Podemos cambiar el *span* del plot (la variable *samp_rate*) a más de 5MHz, pero esto puede ocasionar problemas de *overflow* (O). Debe notarse que este parámetro hay que ajustarlo en función del procesador del ordenador. Si la *samp_rate* es muy elevada para la capacidad de procesamiento de nuestra CPU, la señal se mostrará muy lentamente.

Para poder movernos a través de la banda FM, haremos uso de trigonometría básica. Una manera útil de desplazar una señal en frecuencia es multiplicarla por un coseno:

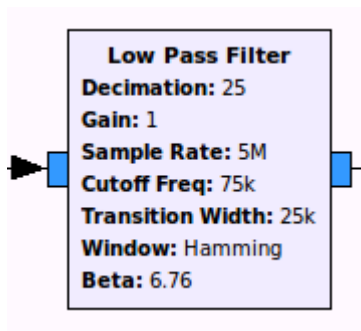


$$\cos(a) * \cos(b) = \frac{1}{2} \cos(a - b) + \frac{1}{2} \cos(a + b)$$

Recordando lo visto en clase, es necesario filtrar la señal paso baja y quedarnos con la parte que nos interesa. En GNU Radio hay bloques de filtros predefinidos, así que será tarea sencilla encontrar un LPF.

Una emisora en FM no ocupa un ancho de banda más allá de unos 80 KHz, aunque la separación entre cada canal es de 200 KHz. Por tanto, hay que prestar atención a la frecuencia de corte del filtro.

Como explicamos en la anterior práctica, HackRF recoge la señal y la modula en banda base. Esto, sumado a que la separación entre las emisoras en FM es de 200 KHz, hace que tengamos un espectro de 100KHz en frecuencias positivas y otros 100KHz en frecuencias negativas. Por ello, una frecuencia de corte de entre unos 75-100 KHz bastará para filtrar cada una de las emisoras, por lo general. Debe tenerse en cuenta que no estamos trabajando con filtros ideales, sino que tienen una transición de caída. Además, si ponemos un filtro demasiado grande, se crearán interferencias con otras emisoras, pues se captará también su espectro.



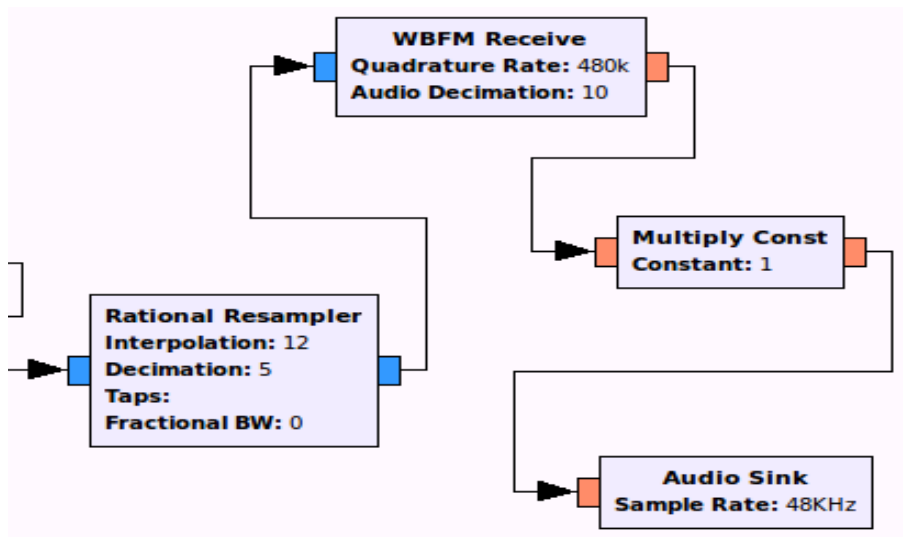
Properties: Low Pass Filter

General | Advanced | Documentation

ID	low_pass_filter_0
FIR Type	Complex->Complex (Decimating)
Decimation	int(samp_rate/channel_width)
Gain	1
Sample Rate	samp_rate
Cutoff Freq	75e3
Transition Width	25e3
Window	Hamming
Beta	6.76

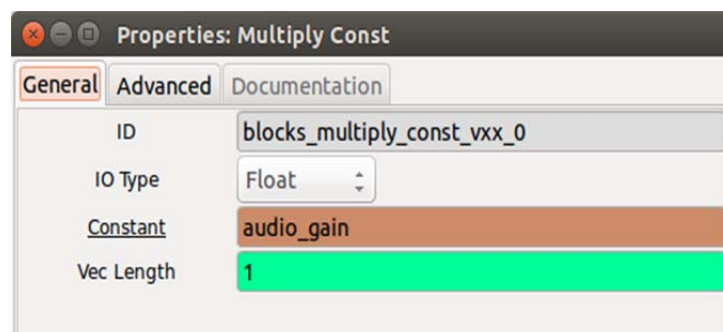
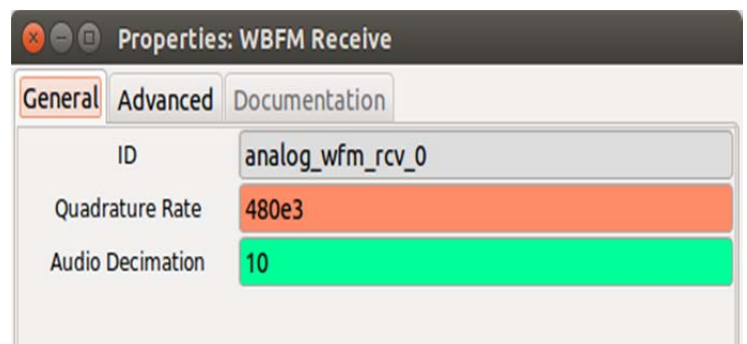
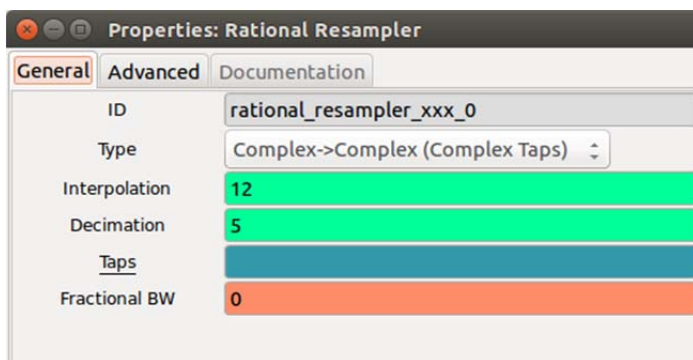
Los parámetros y el tipo de eventanado puede modificarse según creamos conveniente. Recordamos que es útil hacer uso de las variables creadas al principio para definir los parámetros de cada bloque. De este modo, si nos equivocamos o queremos modificar una determinada frecuencia, bastará con cambiar el valor de la variable y no tener que ir bloque por bloque ajustándolo.

Hasta ahora, hemos utilizado conceptos básicos vistos en clase para centrar nuestra señal en las frecuencias deseadas. Los bloques que vienen a continuación, propios de GNU Radio, nos servirán para demodular nuestra señal:



- Rational Resampler: adapta a la tasa de muestreo teórica de la señal mediante el factor $k = \text{interp.} / \text{decimation}$. Estos valores dependerán de lo que necesitemos. Esto es necesario, en este caso, porque el muestreo del *Osmocom Source* no es el mismo que el del *WBFM Receive* ni que el del *Audio Sink*.

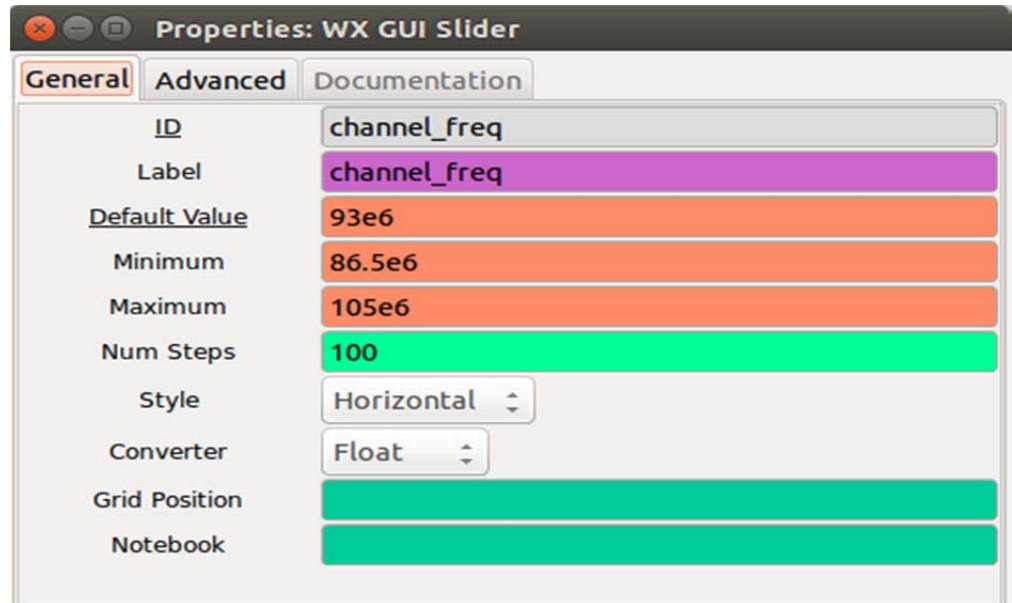
- WBFM Receive (Wide Band FM): Es un receptor digital de FM definido en el software GNU Radio. Debe notarse el diezmado que aplicamos a la señal de entrada para adecuarlo al *Audio Sink*. Por otro lado, existe otro widget llamado *NBFM Receive (Narrow Band FM)*, que se diferencia principalmente en el filtro interno que llevan asociado. Es interesante hacerlo también con esta opción y ver qué sucede.
- Multiply Const: Las señales llegan atenuadas por varios procesos durante su transmisión. Es necesario aplicarles una ganancia. Para ello, en el siguiente paso, nos crearemos otra variable llamada *audio_gain* que permitirá subir y bajar el volumen de nuestra radio particular.
- Audio Sink: Gracias a este bloque de GNU Radio, podremos escuchar si hemos sintonizado correctamente las emisoras de FM. Muchas tarjetas de sonido requieren un muestreo de 44.1 KHZ como mínimo, por lo que poniéndolo a 48 KHZ evitaremos problemas.



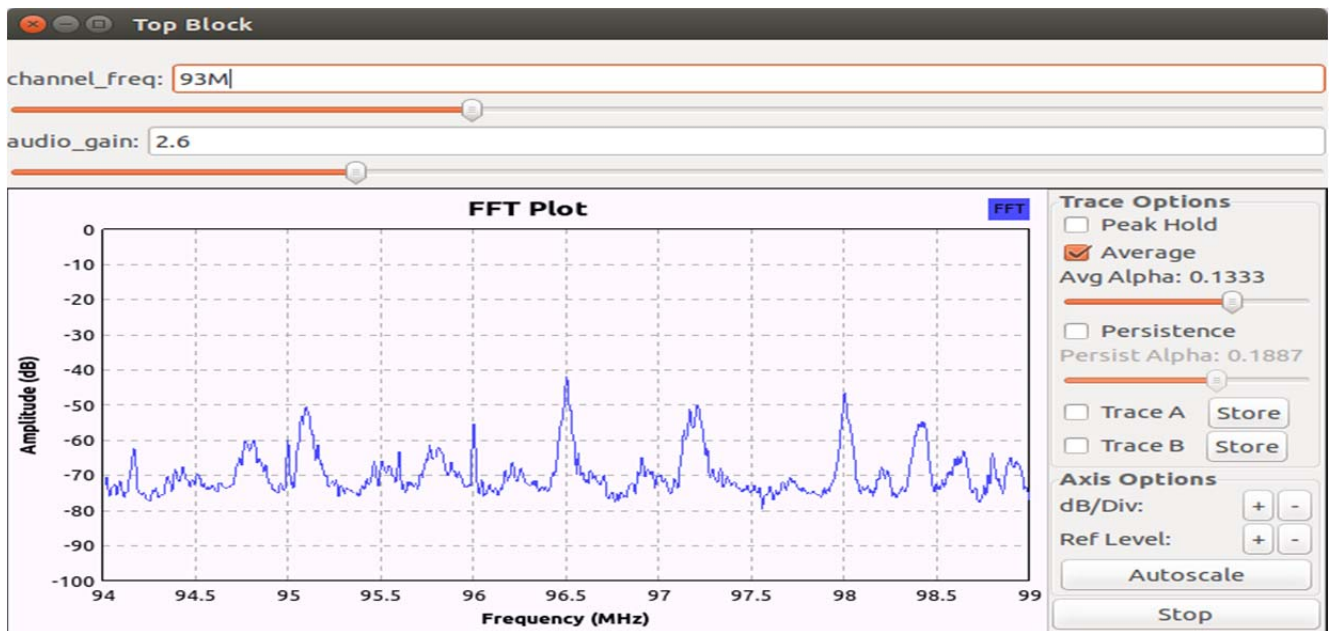
Como comentábamos, una manera fácil de crear un potenciómetro con el que subir y bajar el volumen es con el bloque *WX GUI slider* de GNU. En nuestro caso ponemos como ID *audio_gain* para que corresponda con el parámetro del bloque *multiply constant*. Seguramente, tengamos que subir y bajar el volumen dependiendo de cada estación, ya que cada emisora está a una distancia diferente de nuestro receptor y llegan más o menos atenuadas.

Del mismo modo, para completar nuestra radio finalmente, necesitamos otro “potenciómetro” que nos permita sintonizar las distintas emisoras, así que crearemos otro slider, pero esta vez para barrer todas las frecuencias:

WX GUI Slider
ID: channel_freq
Label: channel_freq
Default Value: 93M
Minimum: 86.5M
Maximum: 105M
Converter: Float



Crear este slider es una forma mucho más sencilla de movernos por la banda FM, por lo que no es necesario mantener la variable `cannel_freq` del principio (podemos desactivar el bloque). Si ponemos como ID al slider `cannel_freq`, elegiremos la frecuencia de nuestro coseno.



CUESTIONES

- Prueba a sintonizar tu radio de ambas maneras y estudia qué resultados obtienes modificando los distintos parámetros de cada bloque y coméntalos.
- *Según lo visto en clase, ¿qué componentes debería contener WBFM Receive para poder demodular la señal?*
- ¿Debemos colocar otra WX GUI FFT SINK tras demodular nuestra señal? ¿Por qué?

