

# DEMODULACIÓN AM

## · INTRODUCCIÓN

Ahora que estamos más familiarizados con el software GNU Radio, en esta práctica realizaremos una demodulación AM. Vamos a recordar primero que la modulación AM, consiste básicamente en variar la amplitud de una señal sinusoidal, denominada portadora, de frecuencia relativamente alta de acuerdo con la amplitud de una señal moduladora.

La modulación en amplitud es de tipo lineal, esto quiere decir que utiliza la amplitud instantánea de la señal modulada como medio para transportar la información. Este tipo de modulación se usa en radiodifusión comercial y en algunas bandas de ciudadana, abarcando desde 535 a 1605 kHz.

Estas señales rebotan en la ionosfera debido a su gran longitud de onda, pudiendo viajar grandes distancias. Esto requiere una alta potencia para ser emitidas, ya que dicha modulación consiste en variar la amplitud.

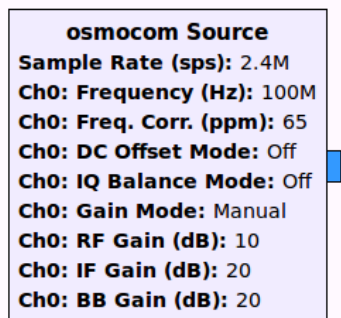
## · DEMODULACIÓN CON GNURADIO Y HACKRF

Cuando demodulamos una onda AM, la portadora y la porción de la envolvente que lleva la información se trasladan del espectro de radio frecuencia a la fuente original de información. Dicha demodulación la podemos realizar de dos métodos:

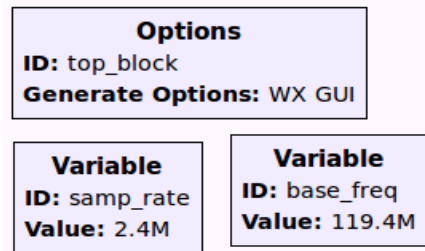
- El *detector de envolvente* es un método muy simple de demodulación que consta de un rectificador y un filtro paso bajo.

- Hay señales AM en las que la portadora se reduce o suprime totalmente. Dichas señales requieren una demodulación coherente. Para ello se utiliza un detector de fase PLL.

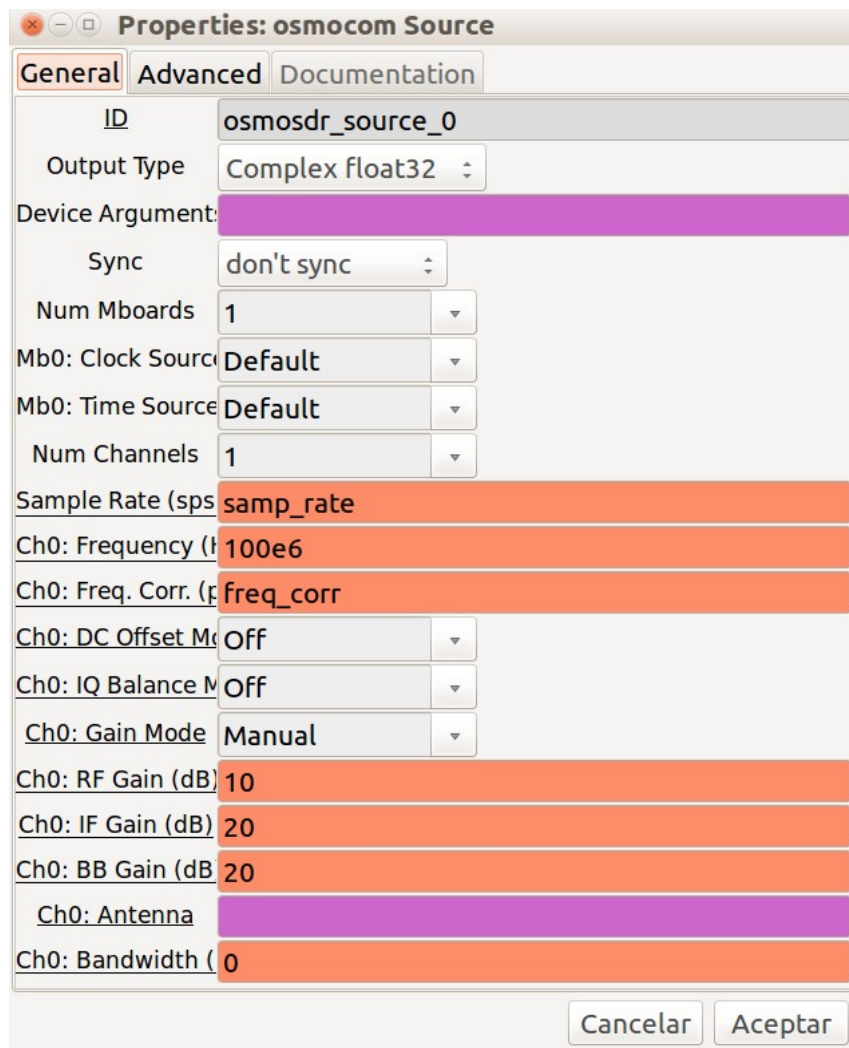
Respecto a la implementación de la demodulación AM, lo primero que necesitamos hacer en GNU Radio es crear el bloque Osmocom Source para poder recibir la señal que nos envía el HackRF:



Creamos una variable denominada *samp\_rate* (frecuencia de muestreo) para asignar un valor definido al span visualizado en la FFT, además de la variable estática *base\_freq* (frecuencia base).



De este modo el bloque *Osmocom Source* queda configurado de la siguiente manera:



Creemos un slider para la variable `freq_corr` definida en el bloque Osmocom. Esto nos permitirá movernos con mayor precisión en la banda de frecuencias AM.

**WX GUI Slider**

**ID:** `freq_corr`  
**Label:** `freq correction (ppm)`  
**Default Value:** 65  
**Minimum:** -127  
**Maximum:** 127  
**Converter:** Integer

**Properties: WX GUI Slider**

General | Advanced | Documentation

ID: `freq_corr`  
Label: `freq correction (ppm)`  
Default Value: 65  
Minimum: -127  
Maximum: 127  
Num Steps: 20  
Style: Horizontal  
Converter: Integer  
Grid Position:  
Notebook:

Cancelar Aceptar

Ahora vamos a utilizar el bloque *WX GUI FFT SINK* para observar lo que está ocurriendo. Este bloque lo unimos con nuestra fuente, así veremos la señal que captamos y podremos modificar los parámetros en caso necesario.

**WX GUI FFT Sink**

**Title:** FFT Plot  
**Sample Rate:** 2.4M  
**Baseband Freq:** 0  
**Y per Div:** 10 dB  
**Y Divs:** 10  
**Ref Level (dB):** 0  
**Ref Scale (p2p):** 2  
**FFT Size:** 512  
**Refresh Rate:** 5  
**Freq Set Varname:** None

**Properties: WX GUI FFT Sink**

Parameters:

ID: `wxgui_fftsink2_0`  
Type: Complex  
Title: `FFT Plot`  
Sample Rate: `samp_rate`  
Baseband Freq: 0  
Y per Div: 10 dB  
Y Divs: 10  
Ref Level (dB): 0  
Ref Scale (p2p): 2.0  
FFT Size: 512  
Refresh Rate: 5

Cancelar Aceptar

Si recordamos lo visto en clase, ahora necesitamos filtrar la señal paso bajo para quedarnos con la parte que nos interesa. En GNU Radio tenemos el acceso a diferentes bloques de filtros que vienen predefinidos, pero en este caso haremos uso del bloque *Frequency Xlating FIR Filter* que nos permite diseñar cualquier tipo de filtro con los parámetros deseados, junto a un bloque de control de nivel *AGC2*.

Sabemos que el HackRF recoge la señal y la modula en banda base, y que las emisoras AM ocupan un ancho de banda de 10 kHz. Esto implica que tenemos un espectro de 5 kHz en frecuencias positivas y 5 kHz en frecuencias negativas, por lo que en nuestro caso, escogeremos una frecuencia de corte de aproximadamente 25 kHz para captar toda la información y tener una mejor calidad de la señal, ya que en AM las emisoras se encuentran a una distancia elevada.

Para saber que parámetros definir a la hora de diseñar nuestro filtro paso bajo, utilizamos la siguiente guía donde nos especifica los diferentes tipos de filtros que podemos implementar:

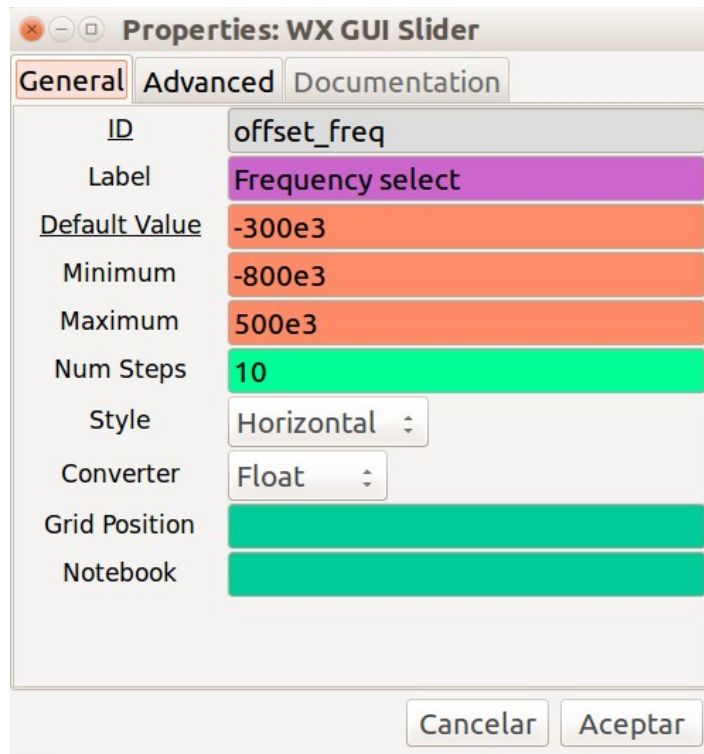
[http://gnuradio.org/doc/doxygen/classgr\\_1\\_1filter\\_1\\_1firdes.html](http://gnuradio.org/doc/doxygen/classgr_1_1filter_1_1firdes.html)

Vamos a utilizar el filtro paso bajo *low\_pass 2* definido por los siguientes parámetros; ganancia, variable *sample\_rate* declarada al inicio, frecuencia de corte, ancho de banda y la atenuación de banda.

Para poder movernos a través de las emisoras AM vamos a definir un nuevo bloque denominado *WX GUI SLIDER* que nos permitirá desplazarnos a lo largo del ancho de banda de AM. Dicho bloque lo definiremos con una variable denominada *offset\_freq* evaluada entre un mínimo y un máximo (total del ancho de banda), donde la colocaremos en el parámetro Center Frequency dentro de los parámetros del filtro.

<b>WX GUI Slider</b> <b>ID:</b> offset_freq <b>Label:</b> Frequency select <b>Default Value:</b> -300k <b>Minimum:</b> -800k <b>Maximum:</b> 500k <b>Converter:</b> Float
---

Propiedades del Slider que nos permitirá desplazarnos a través de las emisoras AM:



Por tanto el filtro queda definido de la siguiente manera:



**Properties: Frequency Xlating FIR Filter**

**Parameters:**

ID	freq_xlating_fir_filter_xxx_0
Type	Complex->Complex (Complex Taps) ▾
Decimation	50
Taps	firdes.low_pass_2(1,samp_rate,25e3,10e3,40)
Center Frequency	offset_freq
Sample Rate	samp_rate
Core Affinity	
Min Output Buffer	0
Max Output Buffer	0

**Error Messages:**

Sink - in(0):

Cancelar    Aceptar

**Properties: AGC2**

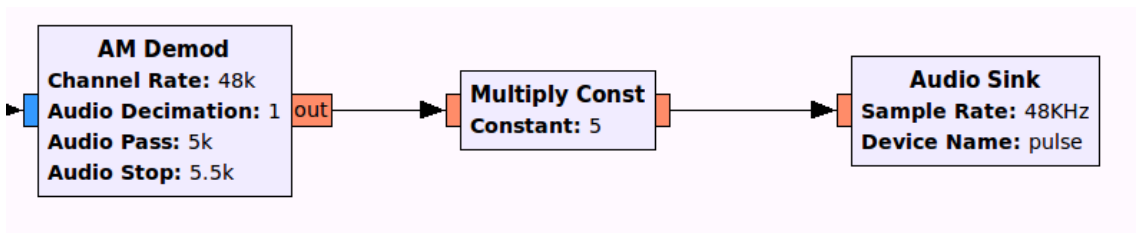
General    Advanced    Documentation

ID	analog_agc2_xx_0
Type	Complex ▾
Attack Rate	100e-3
Decay Rate	10e-6
Reference	1.0
Gain	0
Max Gain	5

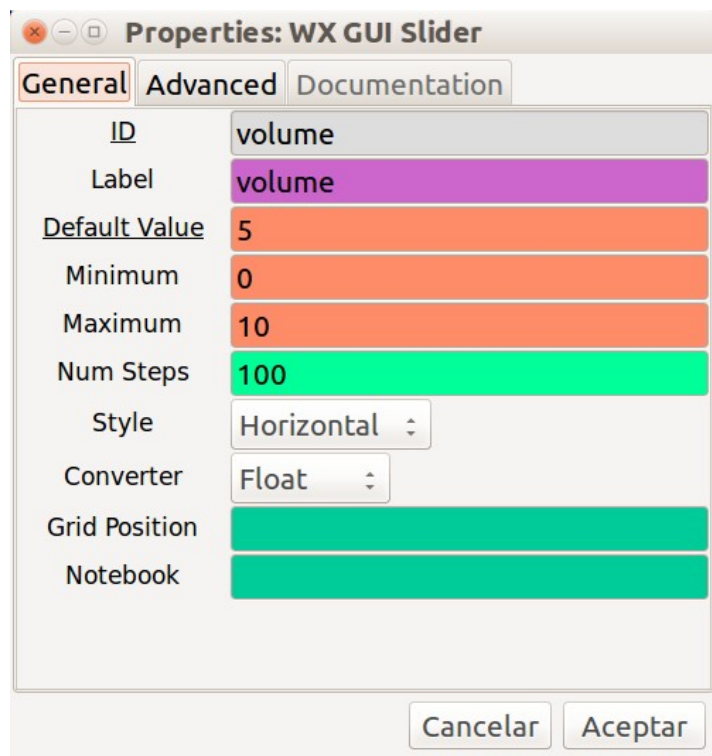
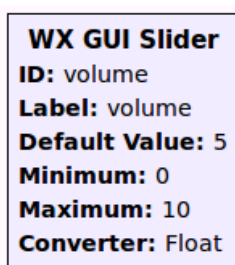
Cancelar    Aceptar

Resulta útil hacer uso de las variables creadas al inicio para definir los parámetros de cada bloque. Esto nos servirá si nos equivocamos o queremos modificar una determinada frecuencia, únicamente habrá que cambiar el valor de la variable y de este modo no tendremos que ir bloque por bloque ajustándolo.

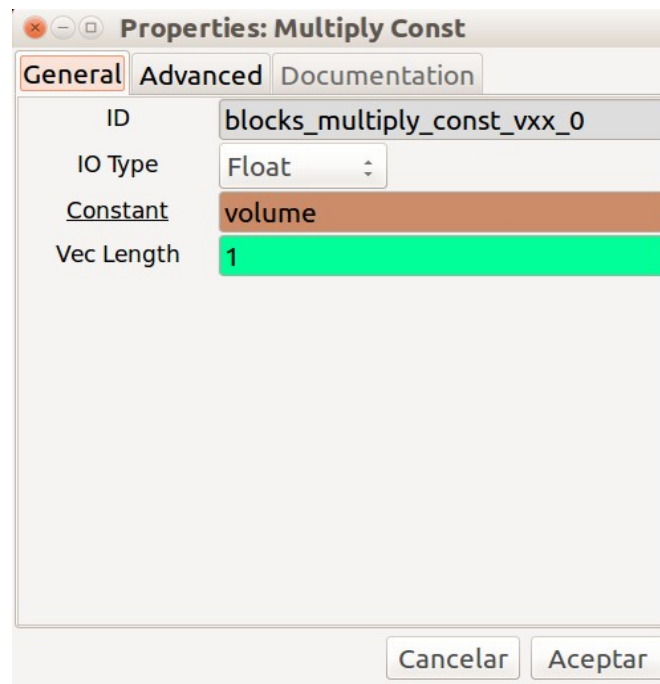
Por último, vamos a utilizar una serie de bloques que nos servirán para demodular nuestra señal. Usaremos el bloque *AM Demod* que incorpora un filtro paso bajo de audio, seguido de los bloques *Multiply Const* y *Audio Sink*.



El bloque *Multiply Const* nos aplica una cierta ganancia debido a la atenuación de la señal que recibimos. Por ello, definimos una variable llamada *volume* que nos permitirá subir y bajar el volumen de nuestra radio. Para poder variar el volumen crearemos otro bloque *WX GUI SLIDER*, donde indicaremos como ID *volume* para que corresponda con el parámetro del bloque *Multiply Const*.



*Multiply Const* quedará configurado de la siguiente manera:



El bloque que nos queda, *Audio Sink* nos permitirá escuchar si hemos sintonizado correctamente las emisoras de AM. Muchas tarjetas de sonido requieren un sampleo de 44.1 kHz como mínimo, por lo que poniéndolo a 48 kHz evitaremos problemas.

