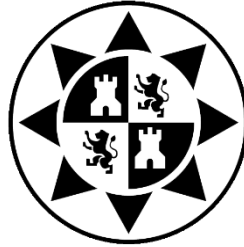


ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Trabajo Fin de Grado

Implementación de modulaciones analógicas y
digitales en GNU Radio



AUTOR: José Luis Sanz Ruiz
DIRECTOR: Leandro Juan Llácer

Autor:	José Luis Sanz Ruiz
E-mail del Autor:	Joseluis90m@gmail.com
Directores:	Leandro Juan Llácer
E-mail de los Directores:	leandro.juan@upct.es
Título del TFG:	Implementación de modulaciones analógicas y digitales en GNU Radio.
Resumen:	<p>El desarrollo y la experimentación con sistemas de RF siempre ha implicado un gran costo en cuanto al material que se requiere para realizar implementaciones, por lo que este tipo de sistemas suelen simularse por ordenador.</p> <p>En este momento el desarrollo del software y el hardware es tal, que permite realizar sistemas usando ordenadores convencionales e interfaces equipadas con DACs, ADCs y conversores de frecuencia, basándose en el concepto de radio definida por software, o SDR (Software Defined Radio), donde mediante un software informático es posible generar cualquier señal de radio ubicada en un determinado ancho de banda y obtenerla eléctricamente a través de una interfaz de radio a la frecuencia deseada.</p> <p>El proyecto hace uso del software GNU Radio, para proponer unas implementaciones con fin educativo de las modulaciones analógicas y digitales básicas, que se estudian en los planes de ingeniería de telecomunicación. Además, se implementa un modulador AM comercial y otro de FM estéreo, utilizando el hardware HackRF One de Great Scott Gadgets. Logrando tanto el aprendizaje y manejo de la herramienta para futuros usos, como el refuerzo de los conceptos teóricos vistos en teoría de la comunicación, y entre los que se incluyen teoría de la señal y conceptos sobre tratamiento digital de la señal.</p> <p>El modelado de sistemas se realiza utilizando la aplicación GNU Radio Companion, que es un editor de diagramas de bloques típico. La ventaja de usar GNU Radio frente a otros, es que es gratuito, de uso libre y adaptable. Además está plenamente enfocado al desarrollo de sistemas RF, por lo que este tipo de tareas se realizan casi de forma inmediata.</p>
Titulación:	Grado en Ingeniería en Sistemas de Telecomunicación
Departamento:	Tecnologías de la Información y las Comunicaciones
Fecha de Presentación:	Diciembre 2020

Agradecimientos

A mi director de trabajo, Leandro Juan, por brindarme la oportunidad de desarrollar este trabajo.

A mis padres, por recordarme siempre la importancia de la formación y la educación. Y por ayudarme a conseguirlo.

A toda mi familia, mi hermana, primos, tíos y a la memoria de mis abuelos, por formar parte de mi vida.

También a mis amigos Elena, Claudia, Inés, Laura, Andrés y Miguel, por estar siempre en lo bueno y en lo malo.

A mis compañeros y amigos Irene, José Antonio y Ricardo.

A Fabian Castillo y Daniel Argilés.

A mis amigos Natalia, Sergio, Juan Carlos, David, Carlos y Judith. Por las tardes tan divertidas que solíamos pasar después de clase.

A todos los compañeros de clase con los que en algún momento he compartido amistad durante esta etapa.

A mi amigo Emilio, por compartir conmigo sus conocimientos sobre software, escuchar mis inquietudes tecnológicas y ayudarme a buscar la inspiración.

A todos los compañeros del aula de teatro de la universidad, con quienes lo he pasado genial en las tablas, a María Juárez por su gran amistad y aprecio.

A mis compañeros de la rama de estudiantes del IEEE, por las innumerables aventuras, eventos y todo lo que hemos aprendido juntos, que también me ha servido para el desarrollo de este trabajo.

También a nuestro profesor responsable en la Universidad, José María Malgosa.

A mis amigos del servicio de documentación, por haberme ayudado y acompañado siempre durante las interminables jornadas de estudio y al demás personal de la universidad, por su trato amistoso.

Y en general a todos los amigos que están ahí.

Índice general

1. Introducción y objetivos	6
1.1. Organización del contenido del trabajo	7
2. Introducción a los sistemas de telecomunicación	8
2.1. Componentes de un sistema de comunicaciones	8
2.2. Señal, ruido, distorsión y ancho de banda	9
2.2.1. Señal	9
2.2.2. Ruido, interferencias y distorsión	9
2.2.3. Relación Señal a Ruido (SNR)	11
2.2.4. Ancho de banda	12
2.3. Concepto de modulación	12
2.3.1. Señal moduladora	13
2.3.2. Señal portadora	13
2.3.3. Señal modulada	13
2.4. Espectro radioeléctrico	13
2.4.1. Bandas de radio UIT (ITU)	14
3. Técnicas de modulación analógicas y digitales	15
3.1. Modulaciones de amplitud	16
3.1.1. Amplitud modulada (AM)	16
3.1.2. Doble banda lateral con portadora suprimida (DSB-SC)	18
3.1.3. Banda lateral única (SSB)	20
3.1.4. Desplazamiento de amplitud (ASK)	21
3.2. Modulaciones de fase	22
3.2.1. Desplazamiento de fase (PSK)	23
3.3. Modulaciones de frecuencia	24
3.3.1. Frecuencia modulada (FM)	24
4. Radio definida por software (SDR)	27
4.1. Introducción a la Radio Definida por Software	27
4.2. Modelado de SDR mediante GNU Radio (GR)	29
4.2.1. Introducción a GNU Radio Companion (GRC)	30
4.2.2. Entorno de trabajo	31
4.2.3. Bloques de entrada o source	32
4.2.4. Bloques de procesado	32
4.2.5. Bloques de salida o sink	32
4.3. Dispositivo para SDR, HackRF One	32
4.3.1. Características técnicas del HackRF One	32
4.3.2. Instalación del software de HackRF en Debian	34

5. Simulación de moduladores en GNU Radio	35
5.1. Amplitud modulada (AM)	35
5.1.1. Desarrollo del esquema del modulador	36
5.1.2. Interfaz de control del modulador	38
5.1.3. Interfaz de visualización del modulador	41
5.1.4. Prueba del simulador de AM	43
5.2. Doble banda lateral con portadora suprimida (DSB-SC)	45
5.2.1. Transformación del modulador AM a DSB-SC	45
5.2.2. Prueba del simulador de DSB-SC	46
5.3. Banda lateral superior (USB)	47
5.3.1. Transformación del modulador DSB-SC a USB	47
5.3.2. Prueba del simulador de USB	49
5.4. Banda lateral inferior (LSB)	50
5.4.1. Transformación del modulador USB a LSB	50
5.4.2. Prueba del simulador de LSB	51
5.5. Desplazamiento de amplitud (ASK)	51
5.5.1. Generación de secuencias binarias a partir de un entero	52
5.5.2. Implementación de la modulación ASK	53
5.5.3. Prueba del simulador ASK	54
5.6. Desplazamiento de fase (PSK)	54
5.6.1. Transformación del modulador ASK a PSK	54
5.6.2. Prueba del simulador PSK	55
5.7. Frecuencia modulada (FM)	55
5.7.1. Desarrollo del esquema del modulador	55
5.7.2. Prueba del simulador FM	57
6. Sistemas de RF con HackRF One	61
6.1. Introducción	61
6.1.1. Obtención y adecuación del fichero de sonido para modular	61
6.1.2. Receptores de radio comercial para pruebas	62
6.2. Radiodifusión AM comercial en onda media	63
6.2.1. Adecuación de la señal de audio al modulador	63
6.2.2. Etapa de modulación	64
6.2.3. Prueba del modulador AM comercial	67
6.3. Radiodifusión FM estéreo comercial	69
6.3.1. Tratamiento previo de la señal estereofónica	70
6.3.2. Generación del multiplex de FM estéreo	71
6.3.3. Modulación FM de la señal multiplex de FM	72
6.3.4. Alternancia de las pestañas de la interfaz gráfica	72
6.3.5. Prueba del modulador FM estéreo comercial	74
7. Conclusiones y futuras líneas	79
7.1. Conclusiones	79
7.2. Futuras líneas	79
A. Imágenes de los esquemas realizados	81
Bibliografía	88

Capítulo 1

Introducción y objetivos

El presente trabajo busca la incorporación del proyecto GNU Radio a la formación académica, se trata de realizar una serie de prácticas en el software GNU Radio Companion, que demuestren que la herramienta es útil a la hora de implementar modulaciones sencillas. Por lo tanto, durante el trabajo vamos a plantear diferentes moduladores sencillos. El cometido no es exprimir las capacidades del software o el hardware, sino de lograr un procedimiento por el cual, implementar dichas modulaciones de una forma didáctica, por lo que no será considerado en cualquier caso el derroche de recursos o procedimientos que no sean verdaderamente óptimos desde el punto de vista del procesamiento digital.

Cometido

La necesidad del uso de software para la formación en conceptos como el de modulación y sistemas de RF, nace de la dificultad existente a la hora de llevar a cabo implementaciones en laboratorios con sistemas reales. Durante el periodo académico, se realizan esta clase de experiencias y prácticas empleando una gran diversidad de software, donde el alumno casi siempre es responsable de programar sus funciones en un lenguaje de programación. Si bien programar es un buen ejercicio, muchas veces puede significar que el alumno centre su atención en programar, y no en explorar los conceptos que se persiguen con suficiente profundidad.

Además de este objetivo académico, está el de demostrar que GNU Radio es útil para implementar sistemas de RF funcionales, empleando hardware SDR y obteniendo señales de radiofrecuencia de naturaleza eléctrica, lo cual permitirá la interacción con otros instrumentos del laboratorio, como osciloscopios, analizadores de espectro, amplificadores, y cualquier equipo de radiofrecuencia, siendo posible realizar un producto con propósito comercial.

La cantidad de proyectos de SDR que existen alrededor de GNU Radio son un motivo más para aprender a utilizarlo:

- Software para implementación de comunicaciones móviles (GSM, LTE, 5G, Wi-Fi, etc.)
- Software para demodular el espectro: Por ejemplo Gqrx. Figura 1.1.
- Software para recepción de datos e imágenes por satélite.
- Software para transmisión y recepción de televisión digital.
- Software para protocolos de telemetría tipo Zigbee, Lo-Ra, etc.

Todas estas herramientas tienen algo en común, y es que todas son software experimental, es decir, no son programas para el uso común, sino que en muchos casos requieren de conocimientos, y el uso que se hace es responsabilidad del usuario, ya que como es obvio GNU Radio y los demás proyectos y dispositivos SDR son un acceso potencial a los sistemas inalámbricos y pueden recibir usos malintencionados. Es importante respetar la legislación de la regulación de las telecomunicaciones en cada país.

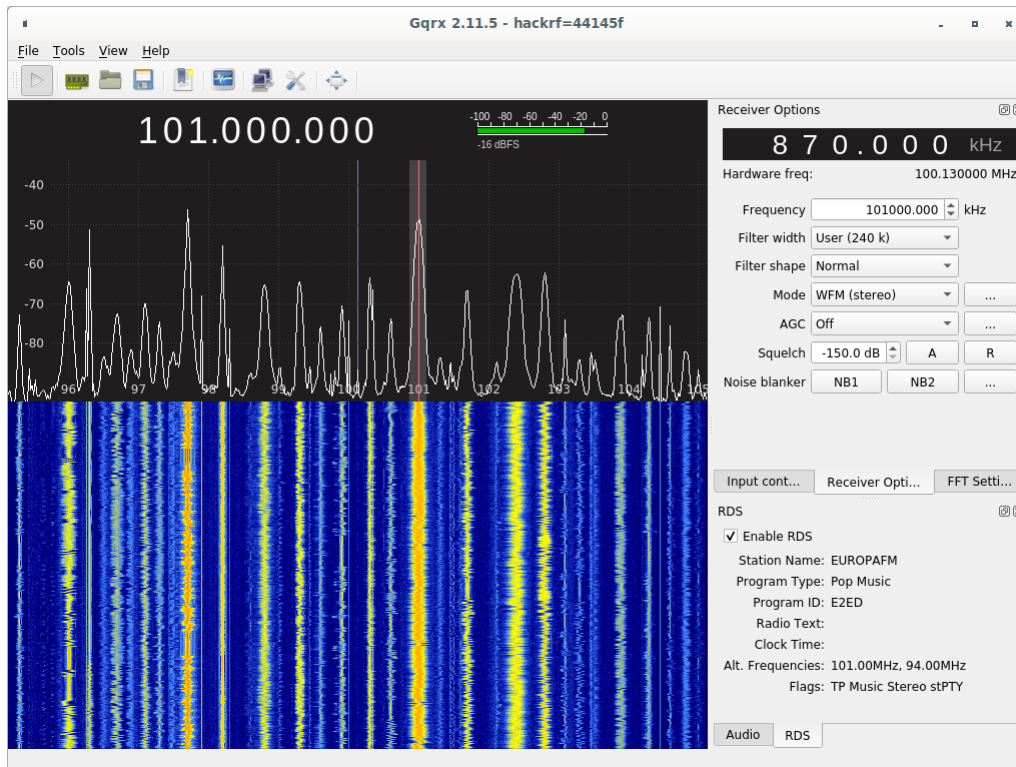


Figura 1.1: Gqrx, software de recepción SDR basado en GNU Radio mostrando el espectro de radio FM y demodulando una estación.

También es aplicable al hardware SDR, que en muchos casos las soluciones que utilizamos no cumplen las características técnicas según las normativas de los equipos de radio, por lo tanto tiene que quedar claro que el uso que se hace de estos sistemas es en entornos cerrados, y que en ningún caso vamos a usar los dispositivos en situaciones que puedan poner en riesgo o interrumpir otras comunicaciones. En caso de estar interesados en la experimentación real y ampliar el rango de acción, podríamos solicitar una autorización de radioaficionado, con esta dispondríamos de unas condiciones para realizar nuestras experiencias de radiofrecuencia dentro de la legalidad, utilizando un sinfín de bandas de radio.

1.1. Organización del contenido del trabajo

El primer capítulo es una introducción y los motivos por los que hemos realizado este trabajo. El segundo capítulo es una pequeña introducción a algunos conceptos sobre comunicaciones que ayudan a poner en contexto el trabajo. El tercer capítulo habla sobre las técnicas de modulación y explica algunos conceptos teóricos sobre estas que pueden ser útiles a la hora de las implementaciones futuras. En el cuarto capítulo explicamos lo que es la radio definida por software, damos una pequeña introducción a GNU Radio y el hardware HackRF One usado más adelante. En el quinto capítulo explicamos el desarrollo de determinados moduladores analógicos y digitales a modo de simulación, que sirven para estudiar las modulaciones y poder observar los principios. En el sexto capítulo hemos creado un par de experimentos usando HackRF One, se trata de un modulador de AM comercial y otro de FM estéreo, mostrando el proceso de implementación y prueba de estos en receptores comunes.

Capítulo 2

Introducción a los sistemas de telecomunicación

Los sistemas de telecomunicación, son sistemas con la capacidad de transmitir información a distancia. Se trata de sistemas compuestos por dispositivos electrónicos, que desempeñan las distintas funciones requeridas para la comunicación. El diseño de estos dispositivos se lleva a cabo mediante técnicas y desarrollos, adquiridos por el ingeniero de telecomunicación.

En esta primera introducción veremos los componentes que forman un sistema de comunicación, de forma genérica, explicaremos el concepto de *señal*, *ruido*, *distorsión*, *ancho de banda*, *modulación*, y *espectro radioeléctrico*, que son conceptos que debemos conocer y recordar para el desarrollo de este trabajo.

2.1. Componentes de un sistema de comunicaciones

Los elementos que componen el sistema de comunicaciones son, *la fuente*, *el transmisor*, *el canal* de comunicación, *el receptor*, y *el destino*, como se observa en la figura 2.1. Además está presente *el mensaje*, y otros como *el ruido y las interferencias*.

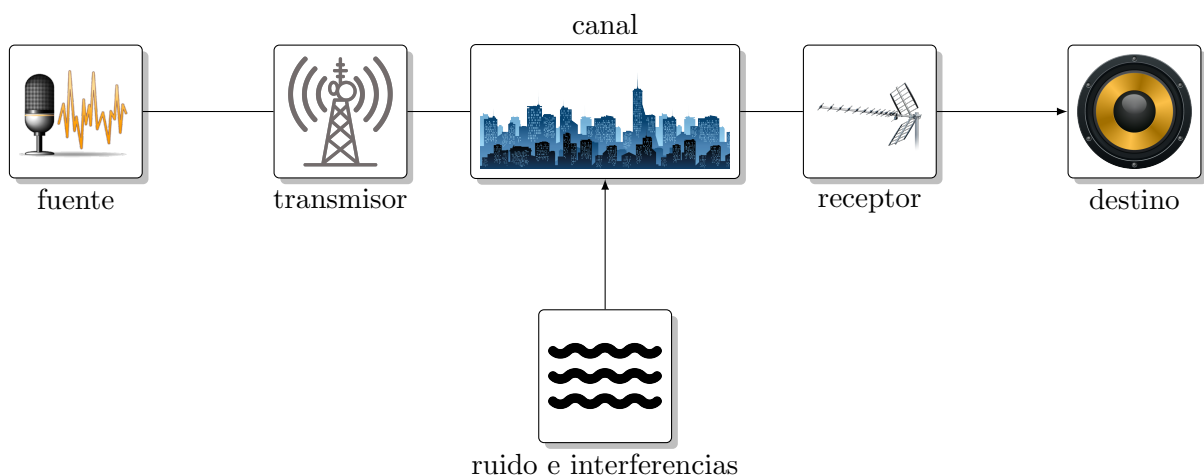


Figura 2.1: Esquema de un sistema de comunicaciones básico.

El mensaje: supone la *información* que queremos transmitir a través del sistema de comunicación, por ejemplo imágenes, sonidos, texto o fenómenos que se pueden medir mediante sensores.

La fuente: es el punto de *entrada* del sistema de comunicación, comprende el transductor y los demás elementos que adecúan el mensaje, de forma que se pueda transmitir empleando un transmisor. Existe una gran diversidad de fuentes atendiendo a la naturaleza del mensaje a transmitir.

El transmisor: es el elemento que permite que la información adecuada por la fuente, se pueda transmitir a través del canal. En el transmisor se encuentra el modulador, encargado de la *modulación*, por lo tanto, en este se encuentra ubicado el contexto de este trabajo.

El canal: es el *medio* a través del cual se realiza la comunicación entre el transmisor y el receptor. Ese medio puede ser *guiado*, como en el caso de los cables, las fibras ópticas y las guías de onda. O *no guiado*, como las comunicaciones por ondas de radio, sonoras o luz infrarroja.

El ruido, las interferencias y la distorsión: son señales y fenómenos no deseados que se añaden y producen durante la transmisión en el canal, haciendo más complicada la tarea de recepción, obligando a emplear en el transmisor y el receptor técnicas que permitan realizar correctamente la recuperación de los mensajes en el destino.

El receptor: recibe la transmisión a través del canal, con distorsiones, y junto con ruido e interferencias. Realiza la tarea de *demodulación*, que es inversa a la de modulación realizada en el transmisor, para recuperar la información introducida por la fuente.

El destino: es la *salida* del sistema, utiliza la información recuperada por el receptor, para recrear de nuevo el mensaje que se introdujo en la fuente. En el destino se reproduce una copia del mensaje de entrada, la cual se pretende que sea lo más fiel posible a la original.

2.2. Señal, ruido, distorsión y ancho de banda

2.2.1. Señal

Cuando hablamos de señales nos referimos a señales de naturaleza *analógica* y *digital*, estas señales son funciones matemáticas reales, de variables reales, en el caso de las digitales, las señales son discretas, lo que significa que toman valores discretos, en intervalos de tiempo también discretos. La figura 2.2 representa dos ejemplos.

2.2.2. Ruido, interferencias y distorsión

El ruido es son componentes no deseadas que se añaden a la señal eléctrica. Diferenciamos dos tipos de ruido, el *ruido inherente*, y el *ruido de interferencias*. [4]

El ruido inherente aparece en los dispositivos electrónicos debido a la naturaleza física de estos, algunos factores relevantes que introducen el ruido inherente en los sistemas son:

- La agitación térmica de las moléculas en los materiales (ruido térmico).
- El movimiento de los electrones (ruido de tránsito).
- El ruido de disparo en semiconductores (ruido shoot).

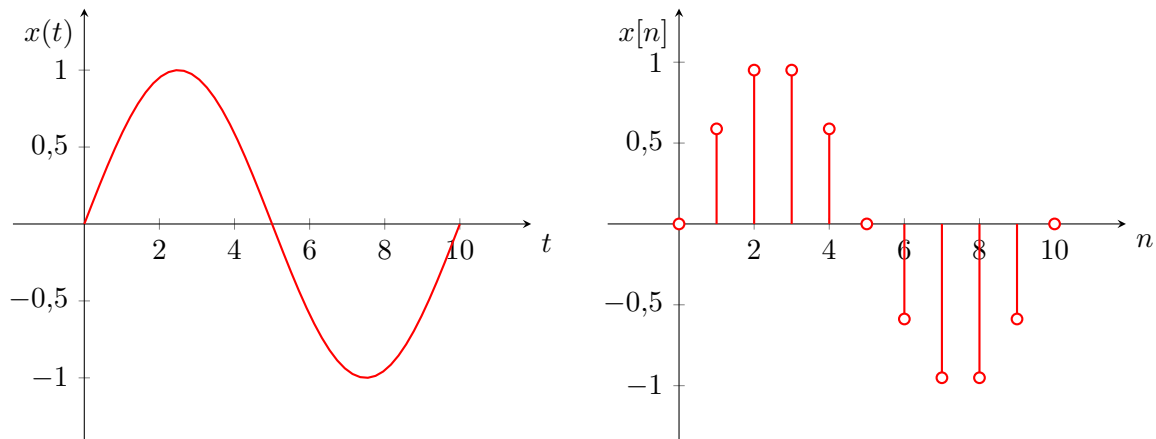


Figura 2.2: Señal analógica $x(t) = \sin(2\pi \cdot t)$ y señal digital $x[n] = \sin\left(2\pi \cdot \frac{n}{10}\right)$.

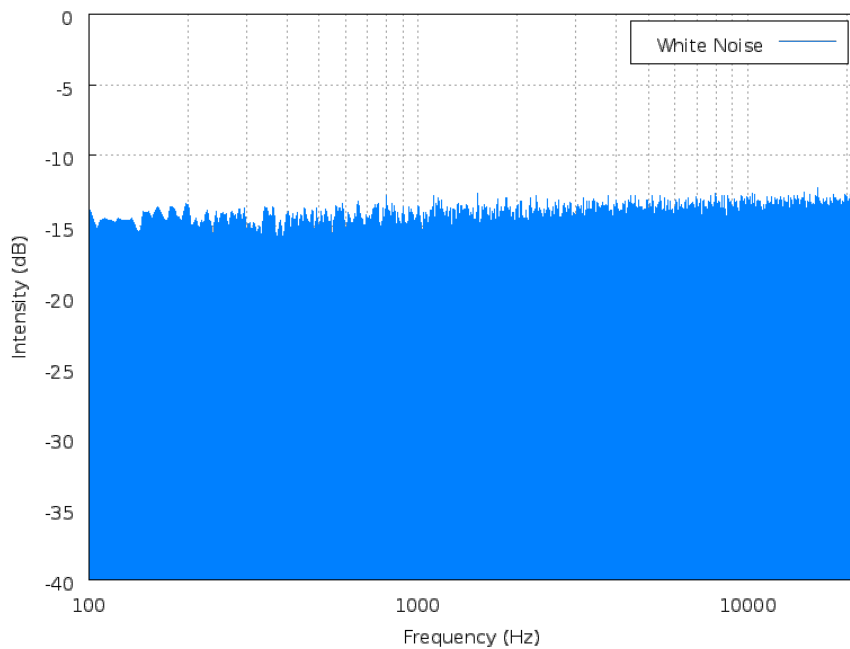


Figura 2.3: Ejemplo de distribución espectral de potencia (PSD) de ruido blanco. [3]

El ruido inherente es de carácter *aleatorio*, como muestra el gráfico de la figura 2.3. El ruido blanco se asemeja bastante al ruido térmico. [3]

El ruido de interferencias resulta del acoplamiento eléctrico y magnético del sistema, es decir, proviene del exterior. Si el acoplamiento es con otros sistemas, se trata de *ruido artificial*, en cambio también podría deberse a *causas naturales*:

- Ruido de fenómenos como tormentas eléctricas o tormentas solares (ruido atmosférico).
- Ruido de radiación de fondo de microondas (ruido cósmico).
- Ruido proveniente de otros sistemas (ruido industrial).

A diferencia del ruido inherente, el ruido de interferencias puede ser *periódico*, *intermitente* o *aleatorio*.

La distorsión La distorsión es un fenómeno no deseado que transforma las señales, produciendo la degradación de estas. [2]

Generalmente se distingue entre *distorsión lineal* y *distorsión no lineal*, por el parámetro afectado de la señal.

La *distorsión lineal* se produce cuando la señal sufre alteraciones en la amplitud o la fase de sus componentes frecuenciales. Estas distorsiones se producen normalmente cuando la señal atraviesa sistemas o canales cuya respuesta en frecuencia no es plana, o las diferentes componentes de la señal sufren diferentes retardos, generalmente debidos a la propagación de la señal en un canal con dispersión. Esta distorsión se caracteriza por no generar nuevas componentes espectrales en la señal afectada. Los ecualizadores permiten corregir la distorsión lineal.

La *distorsión no lineal* se produce cuando la señal produce saturación en amplitud en algún punto del sistema, viéndose recortada, o debido al efecto de intermodulación. Este tipo de distorsión es difícilmente recuperable, y puede dar lugar a nuevas componentes espectrales.

El estudio del ruido y la distorsión es muy importante en los sistemas de telecomunicación, pero en este trabajo no vamos a profundizar más en él.

2.2.3. Relación Señal a Ruido (SNR)

La *Relación Señal a Ruido*, o *Signal-to-Noise Ratio*, *SNR* mide la proporción entre la potencia de la señal y la potencia de ruido. [2]

La calidad de la señal en el sistema será superior cuando la potencia de la señal supere a la potencia de ruido, en el mayor orden posible, ya que esta condición supone una menor degradación de la señal por el efecto del ruido.

Al nivel de ruido en ausencia de señal se le llama *umbral de ruido*, al bajar el nivel de potencia de la señal por debajo del umbral de ruido, esta desaparece enmascarada por este.

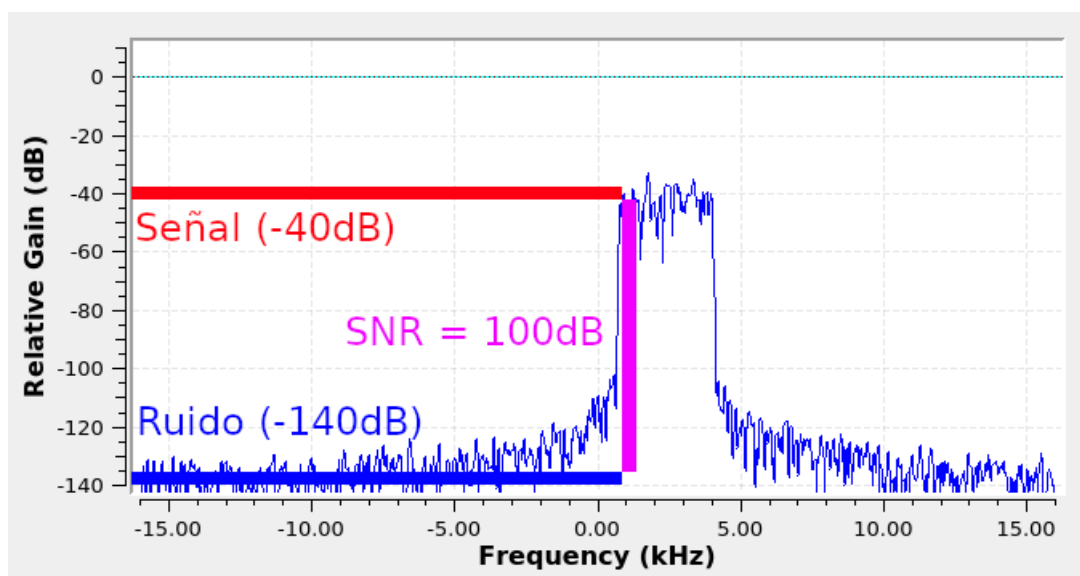
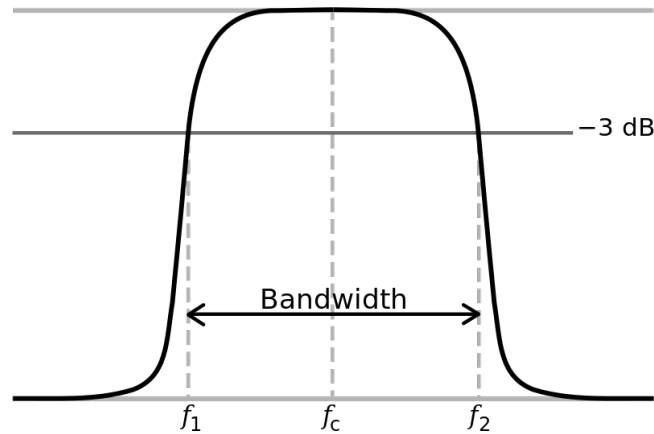


Figura 2.4: Medición de la SNR en una gráfica de densidad espectral de potencia de GNU Radio.

Figura 2.5: Ancho de banda $BW = f_2 - f_1$. [5]

La forma de calcular la SNR, tanto en lineal como en logarítmico a partir de la potencia es: [2]

$$SNR = \frac{P_{Señal}}{P_{Ruido}} \quad (2.1)$$

$$SNR_{dB} = P_{Señal,dBW} - P_{Ruido,dBW} \quad (2.2)$$

$$SNR_{dB} = 10 \cdot \log_{10} \left(\frac{P_{Señal}}{P_{Ruido}} \right) = 10 \cdot \log_{10} (SNR) \quad (2.3)$$

2.2.4. Ancho de banda

El concepto de ancho de banda puede referirse tanto al ancho de banda de una señal, como al ancho de banda de un sistema.

El *ancho de banda de una señal*, es la longitud en unidades de frecuencia, comprendida entre la frecuencia inferior y la frecuencia superior donde se concentra la mayor parte de la potencia de señal.

El *ancho de banda de un sistema*, corresponde al ancho de banda de la señal de mayor ancho de banda que es capaz de atravesarlo sin sufrir distorsión o recorte en su ancho de banda.

Para medir el ancho de banda, BW (*bandwidth*), buscamos la frecuencia a la que la potencia de la señal decae a la mitad de su valor, que es lo mismo que obtener una ganancia de -3dB, estas frecuencias reciben el nombre de *frecuencias de corte*, f_c . El gráfico de la figura 2.5 muestra la medición del ancho de banda de una señal pasa-banda.

2.3. Concepto de modulación

La modulación es el proceso por el cual se hace variar una característica en una señal portadora, mediante una señal moduladora que posee la información a transmitir.

Esta característica puede ser, *la amplitud*, *la fase* o *la frecuencia*, por lo tanto organizaremos el trabajo en estas tres técnicas de modulación básicas.

La modulación implica a tres señales fundamentales:

- La *señal moduladora* $m(t)$ que contiene la información.
- La *señal portadora* $c(t)$ generada para ser modulada por la moduladora.
- La *señal modulada* $s(t)$ que será el resultado de realizar la modulación con las dos anteriores.

2.3.1. Señal moduladora

Señal en banda base, las señales de entrada al transmisor y a la salida del receptor, se llaman señales en banda base, porque su espectro parte de continua (DC), hasta un determinado ancho de banda W , habitualmente continuo, aunque no siempre, ya que en ocasiones la señal moduladora puede ser una señal pasa-banda.

2.3.2. Señal portadora

La señal portadora o señales portadoras para el caso de sistemas multiportadora, son señales puras, generadas con el propósito de ser moduladas mediante la señal moduladora.

En el caso de los sistemas básicos que estudiamos, son funciones senoidales, que se generan con una *amplitud, frecuencia y fase* conocidas.

2.3.3. Señal modulada

La señal modulada es el resultado de aplicar la operación de modulación correspondiente en cada modulador. Veremos las diferentes señales moduladas en el capítulo donde explicamos las modulaciones que se van a realizar en este trabajo.

Banda transmitida, B_T el transmisor, al realizar la función de modulación puede aumentar el ancho de banda de la señal respecto al ancho de banda original de la señal en banda base W . El ancho de banda empleado por el modulador al transmitir es B_T , y para que no haya degradación de la señal, la banda transmitida siempre será mayor o igual que el ancho de banda de la señal en banda base, dependiendo de cada modulación.

$$W \leq B_T \tag{2.4}$$

2.4. Espectro radioeléctrico

El *espectro radioeléctrico* es una porción del *espectro electromagnético*, que sitúa la distribución energética de las ondas electromagnéticas usadas en radiocomunicaciones, según su frecuencia.

El espectro radioeléctrico está situado en el inicio del espectro electromagnético, y finaliza con el inicio del espectro infrarrojo.

La figura 2.7 representa el espectro electromagnético completo, incluyendo las aplicaciones y nomenclaturas de las distintas porciones que lo forman.

Radiofrecuencia (RF): Recibe el nombre de radiofrecuencia o RF, la porción más baja del espectro electromagnético, es decir, la que se sitúa entre 3 Hz y 300 GHz.

Nº	Banda	Frecuencia	División métrica	Abreviacion
3	ULF	300 - 3 000 Hz	Hectokilométricas	B.hkm
4	VLF	3 - 30 kHz	Miriamétricas	B.Mam
5	LF	30 - 300 kHz	Kilométricas	B.km
6	MF	300 - 3 000 kHz	Hectométricas	B.hm
7	HF	3 - 30 MHz	Decamétricas	B.dam
8	VHF	30 - 300 MHz	Métricas	B.m
9	UHF	300 - 3 000 MHz	Decimétricas	B.dm
10	SHF	3 - 30 GHz	Centimétricas	B.cm
11	EHF	30 - 300 GHz	Milimétricas	B.mm

Figura 2.6: Cuadro de bandas ITU, con la denominación de las bandas más importantes en radiofrecuencia.

Espectro de microondas: Se consideran en el espectro de microondas las señales de frecuencias situadas entre 300 MHz y 300 GHz, sin embargo, dependiendo de la bibliografía existen discrepancias a la hora de considerar la porción del espectro de microondas. Según el IEEE se sitúa entre 1 GHz y 300 GHz. No vamos a ahondar en este tema, ya que no es un asunto relevante para este trabajo, el estudio de las bandas de microondas.

2.4.1. Bandas de radio UIT (ITU)

La Unión Internacional de Telecomunicaciones (*International Telecommunication Union*) clasifica las señales de radio según su longitud de onda utilizando la nomenclatura del cuadro 2.6.

Esta forma de organizar el espectro es estrictamente matemática, se apoya en las unidades métricas y las longitudes de onda correspondientes a cada banda.

En este trabajo utilizaremos MF y VHF, es decir, la número 6 y 8 del cuadro 2.6, que albergan la radio en onda media y la radio en FM comercial.

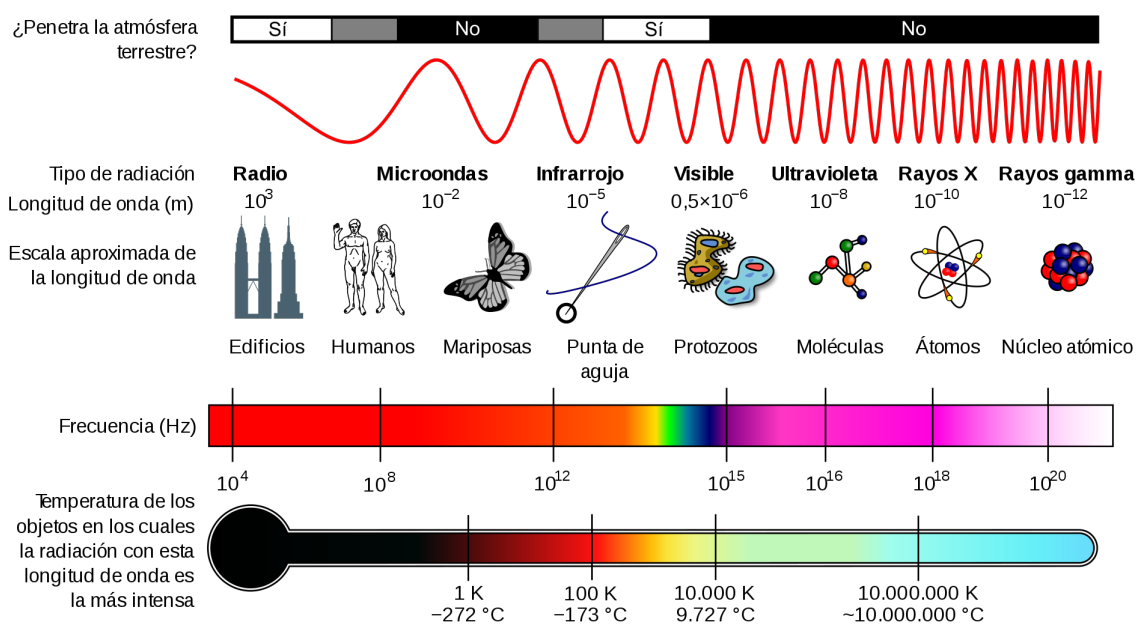


Figura 2.7: Detalle del espectro electromagnético. [6]

Capítulo 3

Técnicas de modulación analógicas y digitales

Las técnicas de modulación *analógicas* y *digitales* que vamos a emplear se pueden clasificar según el parámetro de actuación, en tres técnicas de modulación básicas: *Amplitud*, *frecuencia* y *fase*. Tal y como se muestra en la figura 3.1.

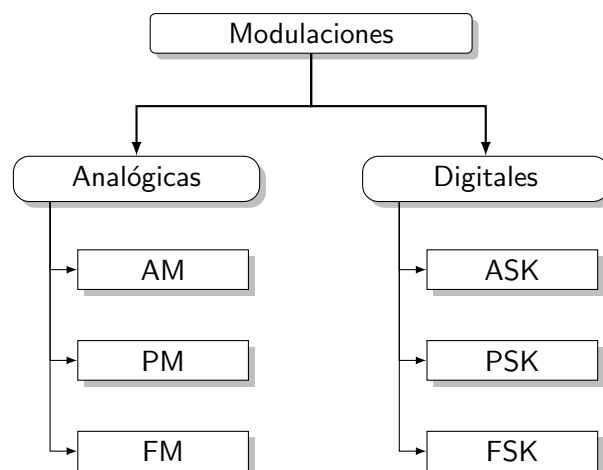


Figura 3.1: Clasificación de las técnicas de modulación analógicas y digitales.

Las tres técnicas fundamentales de modulación analógica son:

- Modulación de Amplitud (AM)
- Modulación de Fase (PM)
- Modulación de Frecuencia (FM)

Estas tres técnicas, sólo se emplean cuando la señal moduladora es una señal continua, pero cuando la información que vamos a transmitir resulta ser discreta, utilizamos una versión digital de cada una de las anteriores, más adecuada, que de forma general reciben el nombre de:

- Modulación por desplazamiento de amplitud (ASK)
- Modulación por desplazamiento de fase (PSK)
- Modulación por desplazamiento de frecuencia (FSK)

Lo normal es encontrar que los sistemas sean híbridos, es decir, emplean varias modulaciones a la vez, y mezclan tanto las técnicas de modulación analógicas como digitales, en sus diferentes subsistemas.

Las técnicas de modulación donde se modula la amplitud, reciben también la denominación de *modulaciones lineales*, mientras que las que se corresponden con la modulación de la fase y la frecuencia, reciben el nombre de *modulaciones angulares*. En gran parte de la bibliografía se utiliza dicha denominación para clasificar las modulaciones a la hora de explicarlas, en este trabajo hemos optado por no realizar más grupos.

3.1. Modulaciones de amplitud

Las modulaciones de amplitud que vamos a estudiar serán:

- Amplitud modulada (AM)
- Doble banda lateral con portadora suprimida (DSB-SC)
- Banda lateral única (SSB)
- Desplazamiento de amplitud (ASK)

3.1.1. Amplitud modulada (AM)

O *amplitude modulation* en inglés. La amplitud modulada, consiste en obtener una señal modulada, cuya envolvente es la señal moduladora. Aunque se llame AM, como la denominación de todas las modulaciones de amplitud, es un tipo concreto de modulación de amplitud. Esta modulación se consigue sumando la portadora al producto realizado entre la moduladora y la portadora, como indica la figura 3.2.

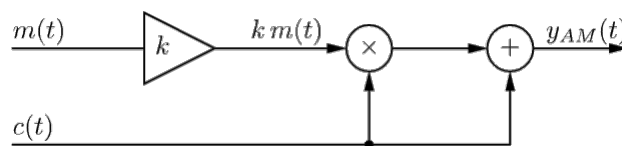


Figura 3.2: Esquema del modulador AM. [1]

Donde $m(t)$ es la *moduladora*, $c(t)$ la señal *portadora*, k es la *sensibilidad* del modulador y y_{AM} la señal *modulada*.

La portadora empleada es un tono de amplitud A_c y frecuencia f_c :

$$c(t) = A_c \cdot \cos(2\pi f_c t) \quad (3.1)$$

La envolvente de $m(t)$ se puede escribir como:

$$e(t) = A_c |1 + km(t)| \geq 0 \quad (3.2)$$

El índice de modulación m es la relación de amplitud máxima de la envolvente y la amplitud de la portadora.

$$|km(t)| \leq 1, \quad e(t) = A_c[1 + km(t)] \quad (3.3)$$

$$|km(t)| > 1, \quad \text{sobremodulación} \quad (3.4)$$

Cuando el índice de modulación es mayor que 1, se produce sobremodulación, y la modulación no es válida, ya que la envolvente superior y la inferior se entrecruzan, como veremos en el capítulo 5.

Una forma simplificada de obtener el índice de modulación en función de la sensibilidad k del modulador es mediante:

$$m = k|m(t)|_{\text{máx}} \quad (3.5)$$

Finalmente considerando la señal moduladora $m(t)$ y la señal portadora $c(t)$ como un tono, en condición de $m \leq 1$ (no sobremodulación), la expresión de la modulación AM de forma compacta se expresa como:

$$y_{AM}(t) = [1 + km(t)]A_c \cos(2\pi f_c t) \quad (3.6)$$

Modulación AM en el dominio de la frecuencia. La modulación AM en el dominio de la frecuencia, resulta en un desplazamiento de la señal a la frecuencia portadora, de forma que se crean dos bandas laterales, una superior con el espectro de la moduladora y otra inferior con el espectro de la moduladora invertido en frecuencia. La portadora se inserta en medio de las dos bandas laterales.

Mediante la realización de la *transformada de Fourier* a la expresión en el dominio del tiempo, se puede calcular la salida en el dominio frecuencial, asumiendo que la moduladora $m(t)$ es una señal limitada en banda $-W < f < W$:

$$Y_{AM}(f) = \frac{A_c}{2} \{\partial(f - f_c) + \partial(f + f_c)\} + \frac{A_c k}{2} \{M(f - f_c) + M(f + f_c)\} \quad (3.7)$$

También podemos obtener su *densidad espectral de potencia*:

$$S_{AM}(f) = \frac{A_c^2}{4} \{\partial(f - f_c) + \partial(f + f_c)\} + \frac{A_c^2 k^2}{4} \{S_m(f - f_c) + S_m(f + f_c)\} \quad (3.8)$$

Donde $S_m(f)$ es la densidad espectral de potencia de la moduladora $m(t)$. Estas funciones se pueden ver representadas en la figura 3.3.

El ancho de banda transmitido para una modulación cuya moduladora tiene un ancho de banda W es:

$$B_T = 2W \quad (3.9)$$

La ventaja de la modulación AM es, que la envolvente de la señal modulada tiene la misma forma de onda que la moduladora, facilitando la demodulación. Pero tiene otras desventajas, como la de que duplica el ancho de banda de la moduladora, y se desperdicia energía en la transmisión de la portadora, existiendo otras modulaciones donde no es necesario transmitirla, como la siguiente.

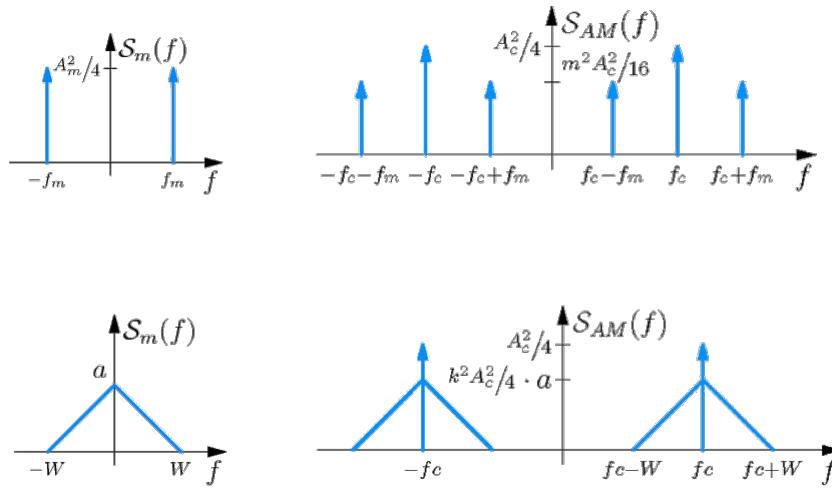


Figura 3.3: Distribución espectral de la potencia en AM. [1]

3.1.2. Doble banda lateral con portadora suprimida (DSB-SC)

O *double-sideband suppressed-carrier* en inglés y abreviada como DBL en castellano. Esta modulación consiste en realizar la multiplicación de la señal moduladora por la señal portadora, la principal diferencia de DSB-SC con AM es, que en DSB-SC no se suma la señal portadora después de la multiplicación entre moduladora y portadora, por ese motivo se añade la denominación de *portadora suprimida*, pues lo que se suprime es su reinserción respecto de AM, ya que se trata de una modulación de invención posterior, que sólo se pudo utilizar una vez que el receptor fue capaz de prescindir de la señal portadora en la transmisión, para obtener la moduladora, usando receptores de tipo regenerativo, capaces de generar el la señal portadora necesaria mediante un oscilador local, en el proceso de demodulación.

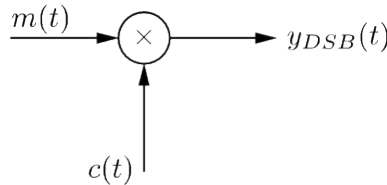


Figura 3.4: Esquema del modulador DSB-SC. [1]

En esta modulación no existe condición de sobremodulación posible, ya que como carece de reinserción de portadora, ya no hay envolvente. La amplitud de la señal modulada será el producto de las amplitudes de moduladora y portadora, por lo que cuanto más amplitud tengan las señales moduladora y portadora, mayor amplitud tendrán la modulada a la salida.

Por tanto podemos afirmar que su expresión en el dominio del tiempo, para una portadora senoidal $c(t)$, será siempre:

$$y_{DSB}(t) = m(t) \cdot c(t) \tag{3.10}$$

Donde la portadora senoidal sería:

$$c(t) = A_c \cdot \cos(2\pi f_c t) \tag{3.11}$$

Utilizamos el coseno como función senoidal de portadora, porque matemáticamente es más sencillo de desarrollar.

Luego, la expresión de un modulador DSB-SC con portadora senoidal quedaría como:

$$y_{DSB}(t) = m(t) \cdot A_c \cos(2\pi f_c t) \quad (3.12)$$

Modulación DSB-SC en el dominio de la frecuencia. La modulación DSB-SC en el dominio de la frecuencia, resulta en un desplazamiento de la señal a la frecuencia portadora, de forma que se crean dos bandas laterales, una superior con el espectro de la moduladora y otra inferior con el espectro de la moduladora invertido en frecuencia.

Mediante la realización de la *transformada de Fourier* a la expresión en el dominio del tiempo, se puede calcular la salida en el dominio frecuencial, asumiendo que la moduladora $m(t)$ es una señal limitada en banda $-W < f < W$:

$$Y_{DSB}(f) = \frac{A_c}{2} \{M(f - f_c) + M(f + f_c)\} \quad (3.13)$$

También podemos obtener su *densidad espectral de potencia*:

$$S_{DSB}(f) = \frac{A_c^2}{4} \{S_m(f - f_c) + S_m(f + f_c)\} \quad (3.14)$$

Donde $S_m(f)$ es la densidad espectral de potencia de la moduladora $m(t)$. Estas funciones se pueden ver representadas en la figura 3.5.

El ancho de banda transmitido para una modulación cuya moduladora tiene un ancho de banda W es:

$$B_T = 2W \quad (3.15)$$

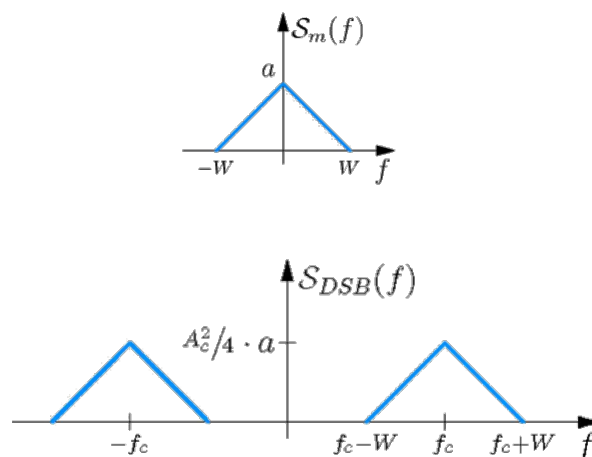


Figura 3.5: Distribución espectral de la potencia en DSB-SC. [1]

Lo más relevante es que se obtiene un ahorro de potencia al no transmitir la componente espectral de la portadora que se transmitía en AM.

3.1.3. Banda lateral única (SSB)

O *single sideband* en inglés y abreviada como BLU en castellano. La modulación en banda lateral única, consiste en eliminar una de las bandas laterales que se producen durante la modulación DSB-SC, para obtener un ahorro del ancho de banda y potencia de transmisión, pues toda la densidad de potencia espectral de la señal queda contenida dentro de un ancho de banda igual al de la moduladora.

Cuando la banda eliminada es la banda inferior, y permanece la superior, la modulación se llama banda lateral superior BLS o USB, *upper sideband* en inglés.

Cuando la banda eliminada es la banda superior, y permanece la inferior, la modulación se llama banda lateral inferior BLI o LSB, *lower sideband* en inglés.

Por tanto, esta modulación se obtiene a partir de DSB-SC y un método de discriminación que permita suprimir una de las bandas. Conocemos dos técnicas fundamentales de discriminación de bandas:

Método de discriminación de frecuencia: Consiste en aplicar un filtro pasa-banda, justo por encima o por debajo de la frecuencia central de DSB-SC, para eliminar la banda no deseada.

Este método es el más sencillo y el que suelen utilizar los equipos analógicos. La selectividad requerida en el filtro necesario para realizar la discriminación de frecuencia complica el diseño de un modulador de electrónica analógica con filtros RC, por lo que en la práctica se realiza usando filtros basados en resonadores de cristal, que son caros.

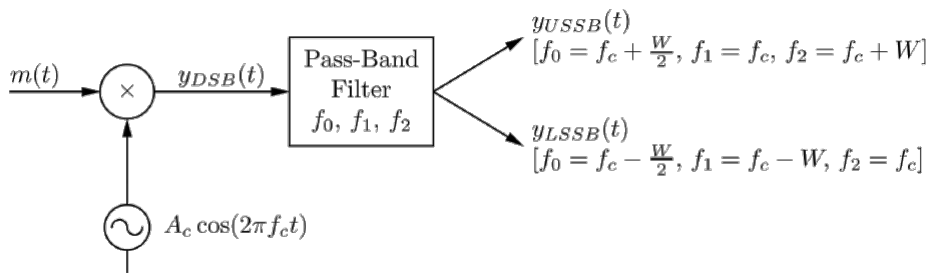


Figura 3.6: Esquema del modulador SSB mediante discriminación de frecuencia. [1]

En la figura 3.6 se muestra este método de obtención de SSB a partir de DSB-SC.

Método de discriminación de fase: Este método requiere mucha precisión por lo que no se emplea en circuitos de modulación analógicos, pero es muy interesante cuando la generación de la señal se realiza mediante SDR. Casi siempre que se trabaja en SDR se utiliza un esquema en fase y cuadratura, por lo que es ideal usar este método.

Como hemos elegido este método para generar nuestras señales SSB en el capítulo de implementación de moduladores, vamos a proceder a explicar en que consiste dicha técnica.

En primer lugar como podemos observar en la figura 3.7, se toma la moduladora $m(t)$ y se obtiene a través de un transformador de Hilbert la señal moduladora en cuadratura $m_{HT}(t)$, es decir con un

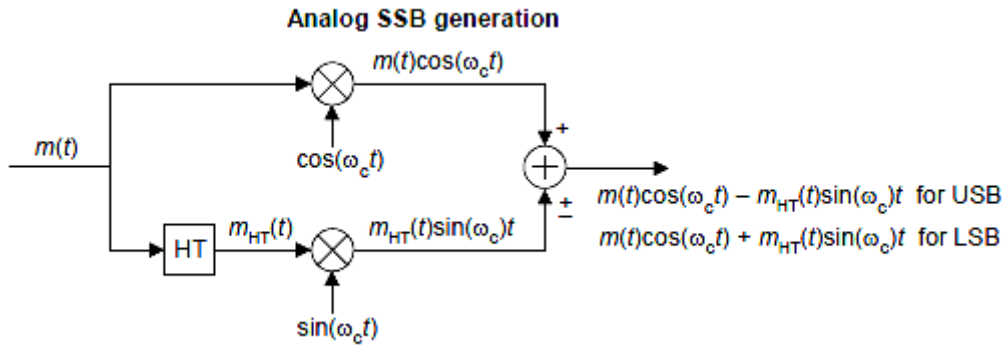


Figura 3.7: Esquema del modulador SSB mediante discriminación de fase. [11]

desfase de 90° respecto $m(t)$.

El transformador de Hilbert, sin entrar en detalle, es un desfaseador de banda ancha, que introduce en todo el ancho de banda de la señal $m(t)$ el mismo desfase.

Por separado se obtienen dos modulaciones DSB-SC, una con la componente en fase $m(t)$ usando la portadora en fase. Mientras que con la moduladora en cuadratura $m_{HT}(t)$ se realiza con la portadora en cuadratura.

El resultado de estas dos modulaciones resultan tener una banda en fase y la otra en contrafase. Por tanto, cuando se suman a la salida, una de las bandas se cancela y la otra se duplica.

Espectro de las modulaciones SSB. El espectro resultante de estas modulaciones es el mismo que en DSB-SC pero realizando un recorte debido al filtro y la cancelación de la banda lateral descartada, en la figura 3.8 se representa el espectro resultante, como hicimos en las modulaciones anteriores.

3.1.4. Desplazamiento de amplitud (ASK)

O *amplitude shift keying* en inglés. Esta modulación mostrada en la figura 3.9 es una versión digital de las modulaciones en amplitud, y consiste en multiplicar la portadora por un determinado número de niveles M , recibiendo la denominación M -ASK, siendo M el número de niveles o símbolos. Veamos los siguientes ejemplos de modulaciones ASK:

- **BASK:** Se trata de una modalidad *binaria*, donde se modula bit a bit, a un bit por símbolo, por lo tanto cuando se realiza la modulación, durante los bits a 1 se transmite la portadora con una amplitud, y durante los bits a 0 a otra. Lo habitual es hacerlo con un índice de modulación del 100 %, es decir, encender la portadora en los bits a 1 y apagarla en los bits a 0 o viceversa, recibiendo el nombre de *OOK*, *on-off keying*.

$$\begin{aligned} 0 &\rightarrow 0 \\ 1 &\rightarrow 1 \end{aligned}$$

- **4-ASK:** En este caso la modulación se realiza en cuatro niveles, dando lugar a cuatro símbolos de fuente y se corresponden con tomar un par de bits de la fuente y asignarles un nivel a cada uno:

$$\begin{aligned} 00 &\rightarrow 0 \\ 01 &\rightarrow 1 \end{aligned}$$

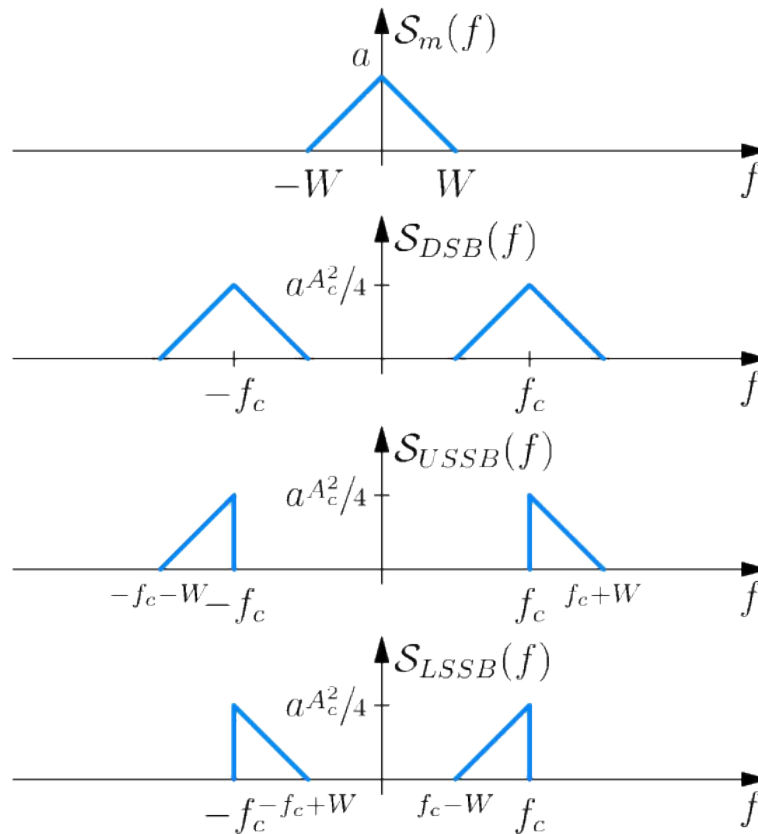


Figura 3.8: Distribución espectral de la potencia en SSB, respecto a DSB-SC. [1]

10 \rightarrow 2

11 \rightarrow 3

- 8-ASK: En este caso la modulación se realiza en 8 niveles, y se corresponden con tomar tres de bits de la fuente y asignarles un nivel a cada uno, habiendo ocho símbolos:

000 \rightarrow 0

001 \rightarrow 1

010 \rightarrow 2

011 \rightarrow 3

100 \rightarrow 4

101 \rightarrow 5

110 \rightarrow 6

111 \rightarrow 7

Con estos tres ejemplos es más que suficiente para comprender la mecánica. Cada grupo de bits que tomamos de la fuente se llama símbolo, por lo que modularemos símbolo a símbolo, en lugar de bit a bit, siempre que tengamos esquemas multinivel (no binarios). Cuantos más niveles tenemos, más aumenta la probabilidad de error, ya que los umbrales de decisión entre niveles quedan más próximos.

3.2. Modulaciones de fase

De las modulaciones de fase, que pertenecen a las denominadas modulaciones angulares, sólo vamos a estudiar una de las más utilizadas en las comunicaciones digitales:

- Desplazamiento de fase (PSK)

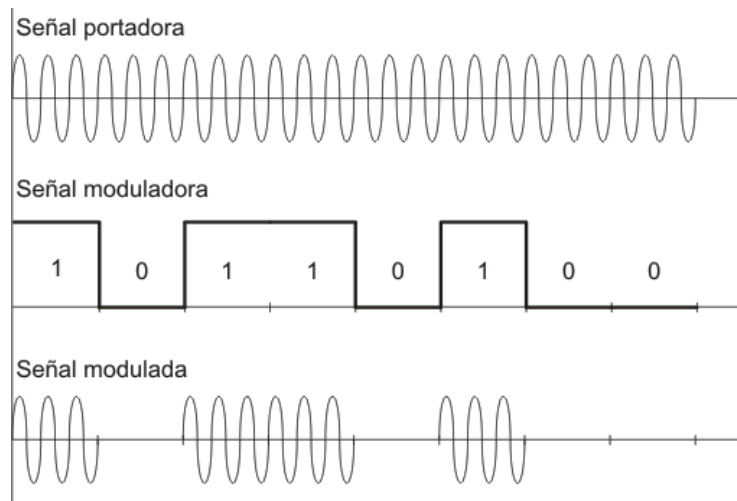


Figura 3.9: Modulación ASK (OOK) [7]

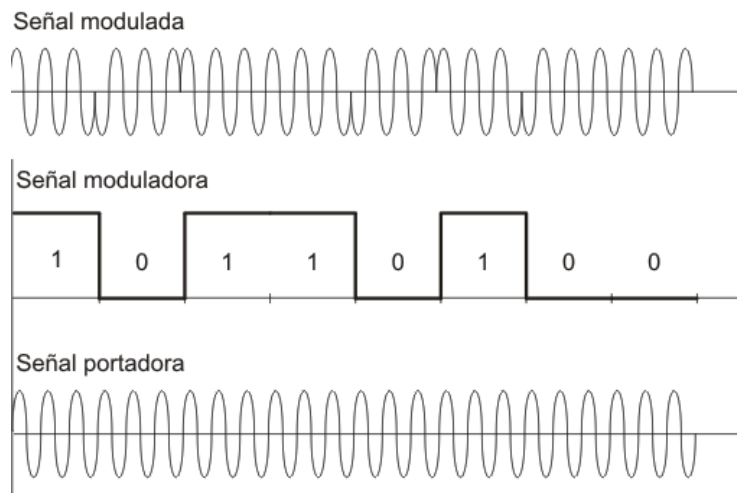


Figura 3.10: Modulación BPSK. [8]

3.2.1. Desplazamiento de fase (PSK)

O *phase shift keying* en inglés. Es una versión digital de fase modulada muy utilizada, que consiste en modificar la fase de una señal portadora dentro de un número finito de desfases, correspondientes a los diferentes símbolos de una fuente, como se muestra en la figura 3.10.

En esta modulación también podemos tener diferentes tamaños de símbolos, en estos ejemplos hemos tomado los desfases que hemos considerado pero no tienen por qué ser esos:

- BPSK: Se trata de una modalidad *binaria*, donde se modula bit a bit, a un bit por símbolo, por lo tanto cuando se realiza la modulación, durante los bits a 1 se transmite la portadora sin desfase, y durante los bits a 0 se realiza una inversión de la fase, esta es la modulación que se muestra como ejemplo en la figura 3.10 y la que vamos a implementar.

$$\begin{aligned} 0 &\rightarrow 0^\circ \\ 1 &\rightarrow 180^\circ \end{aligned}$$

- 4-PSK: En este caso se señala cada símbolo variando la fase de la portadora en cuatro posibles valores equidistantes, estos cuatro valores de fase se obtienen tomando los bits de la fuente binaria de dos en dos, dando lugar a una transmisión de cuatro símbolos.

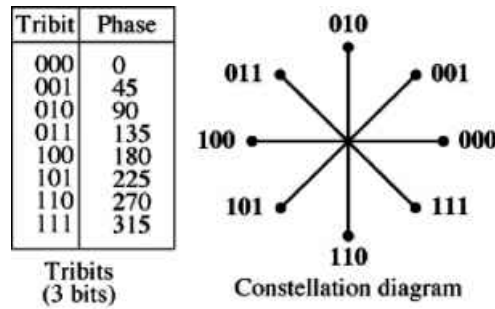


Figura 3.11: Diagrama de constelación de una modulación 8-PSK como la del ejemplo. [9]

$$\begin{aligned}
 00 &\rightarrow 0^\circ \\
 01 &\rightarrow 90^\circ \\
 10 &\rightarrow 180^\circ \\
 11 &\rightarrow 270^\circ
 \end{aligned}$$

- 8-PSK: En este caso la modulación, mostrada en la figura 3.11 se realiza usando ocho valores de fase diferentes, y se toman grupos de tres de bits de la fuente, es decir ocho símbolos.

$$\begin{aligned}
 000 &\rightarrow 0^\circ \\
 001 &\rightarrow 45^\circ \\
 010 &\rightarrow 90^\circ \\
 011 &\rightarrow 135^\circ \\
 100 &\rightarrow 180^\circ \\
 101 &\rightarrow 225^\circ \\
 110 &\rightarrow 270^\circ \\
 111 &\rightarrow 315^\circ
 \end{aligned}$$

3.3. Modulaciones de frecuencia

De las modulaciones de frecuencia, que también pertenecen a las denominadas modulaciones angulares, sólo vamos a estudiar la técnica más aplicada de todas:

- Frecuencia modulada (FM)

3.3.1. Frecuencia modulada (FM)

La frecuencia modulada, *frequency modulation* en inglés, se obtiene variando la frecuencia de la señal portadora, de frecuencia f_c en función de la amplitud de la señal moduladora $m(t)$. El rango de frecuencia en el que se mueve la portadora se llama máxima desviación, y la relación entre la amplitud de la moduladora $m(t)$ y la desviación de la portadora, se llama *sensibilidad del modulador FM* y se representa por f_d (Hz/v).

De forma que la frecuencia instantánea de la modulación FM, se puede escribir como:

$$f(t) = f_c + f_d \cdot m(t) \quad (3.16)$$

La modulación FM se obtiene utilizando un dispositivo llamado VCO *Voltage-controlled oscillator*[2], este dispositivo varia la frecuencia del tono que genera en función de un nivel de voltaje a su entrada, según la sensibilidad f_d de este, habiendo un máximo de desviación posible de frecuencia Δf :

$$\Delta f = f_d |m(t)|_{m\acute{a}x} < f_c \quad (3.17)$$

Es decir, la desviación de la portadora originada por el producto de la sensibilidad del modulador y la amplitud de $m(t)$, nunca debe ser superior a la frecuencia de reposo f_c , pues es obvio que de superarse, la frecuencia instantánea $f(t)$ se volvería negativa. Es decir el VCO trataría de oscilar a una frecuencia negativa. La expresión resultante de modular con una portadora coseno de amplitud A_c será:

$$y_{FM}(t) = A_c \cos \left[2\pi f_c t + 2\pi f_d \int_{-\infty}^t m(\tau) d\tau \right] \quad (3.18)$$

Cuando la moduladora es un tono senoidal:

En este caso la expresión de la señal modulada se puede expresar de la siguiente forma:

$$y(t) = A_c \cos [2\pi f_c t + D \cdot \sin(2\pi f_m)] \quad (3.19)$$

Donde D es el índice de modulación, que para una modulación FM es:

$$D = \frac{\Delta f}{f_m} = \frac{f_d A_m}{f_m} \quad (3.20)$$

Podemos comprobar, que cuando el mensaje de una moduladora es un tono senoidal f_m , la potencia se distribuye en la f_c y las frecuencias múltiplo de f_m contiguas, de la forma:

$$f_c \pm n f_m \quad (3.21)$$

Esto lo vamos a demostrar en el capítulo 5, utilizando el modulador de FM en GNU Radio.

El ancho de banda de una modulación angular, con una moduladora senoidal, se puede calcular con la regla de Carson, sabiendo que el número de frecuencias que aparecerán a cada lado de la f_c viene dado por $n_{m\acute{a}x} = D + 1$. Este ancho de banda incluye el 98% de la potencia total.

El ancho de banda de la modulación FM será:

$$B_T = 2(D + 1)f_m \quad (3.22)$$

La amplitud de las bandas en la representación de la densidad espectral de potencia la podemos calcular con las *funciones de Bessel*, en la figura 3.12 tenemos la tabla de la distribución de amplitud de las bandas según el índice de modulación, las bandas que aparecen en negativo están con la fase opuesta a la portadora.

Esta tabla la usaremos de nuevo en el capítulo 5 cuando realicemos pruebas con el modulador de FM, pues tomaremos medidas de las amplitudes de nuestras bandas y consultaremos para comprobar que se cumple.

x (m_f)	n OR ORDER															
	(CARRIER) J_0	J_1	Sidebands (J_2 J_3 J_4 J_5 J_6 J_7 J_8 J_9 J_{10} J_{11} J_{12} J_{13} J_{14} J_{15} J_{16})													
0.00	1.00	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
0.25	0.98	0.12	—	—	—	—	—	—	—	—	—	—	—	—	—	—
0.5	0.94	0.24	0.03	—	—	—	—	—	—	—	—	—	—	—	—	—
1.0	0.77	0.44	0.11	0.02	—	—	—	—	—	—	—	—	—	—	—	—
1.5	0.51	0.56	0.23	0.06	0.01	—	—	—	—	—	—	—	—	—	—	—
2.0	0.22	0.58	0.35	0.13	0.03	—	—	—	—	—	—	—	—	—	—	—
2.5	-0.05	0.50	0.45	0.22	0.07	0.02	—	—	—	—	—	—	—	—	—	—
3.0	-0.26	0.34	0.49	0.31	0.13	0.04	0.01	—	—	—	—	—	—	—	—	—
4.0	-0.40	-0.07	0.36	0.43	0.28	0.13	0.05	0.02	—	—	—	—	—	—	—	—
5.0	-0.18	-0.33	0.05	0.36	0.39	0.26	0.13	0.05	0.02	—	—	—	—	—	—	—
6.0	0.15	-0.28	-0.24	0.11	0.36	0.36	0.25	0.13	0.06	0.02	—	—	—	—	—	—
7.0	0.30	0.00	-0.30	-0.17	0.16	0.35	0.34	0.23	0.13	0.06	0.02	—	—	—	—	—
8.0	0.17	0.23	-0.11	-0.29	-0.10	0.19	0.34	0.32	0.22	0.13	0.06	0.03	—	—	—	—
9.0	-0.09	0.24	0.14	-0.18	-0.27	-0.06	0.20	0.33	0.30	0.21	0.12	0.06	0.03	0.01	—	—
10.0	-0.25	0.04	0.25	0.06	-0.22	-0.23	-0.01	0.22	0.31	0.29	0.20	0.12	0.06	0.03	0.01	—
12.0	0.05	-0.22	-0.08	0.20	0.18	-0.07	-0.24	-0.17	0.05	0.23	0.30	0.27	0.20	0.12	0.07	0.03
15.0	-0.01	0.21	0.04	-0.19	-0.12	0.13	0.21	0.03	-0.17	-0.22	-0.09	0.10	0.24	0.28	0.25	0.18

Source: E. Cambi, *Bessel Functions*, Dover Publications, Inc., New York, 1948. Courtesy of the publisher.

Figura 3.12: Amplitudes de las bandas según el índice de modulación, dadas por las funciones de Bessel. [12]

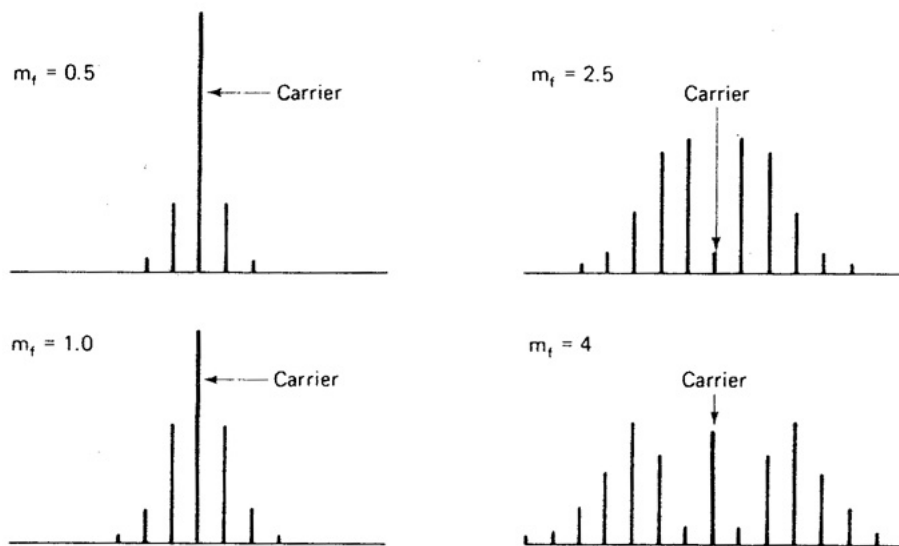


Figura 3.13: Espectros de las bandas para distintos índices de modulación. [12]

Capítulo 4

Radio definida por software (SDR)

4.1. Introducción a la Radio Definida por Software

SDR, *Software Defined Radio* en inglés o Radio Definida por Software, es un concepto que resulta de aplicar las técnicas de procesamiento digital de la señal (DSP) habituales, a la generación de señales de radiofrecuencia mediante síntesis por ordenador.

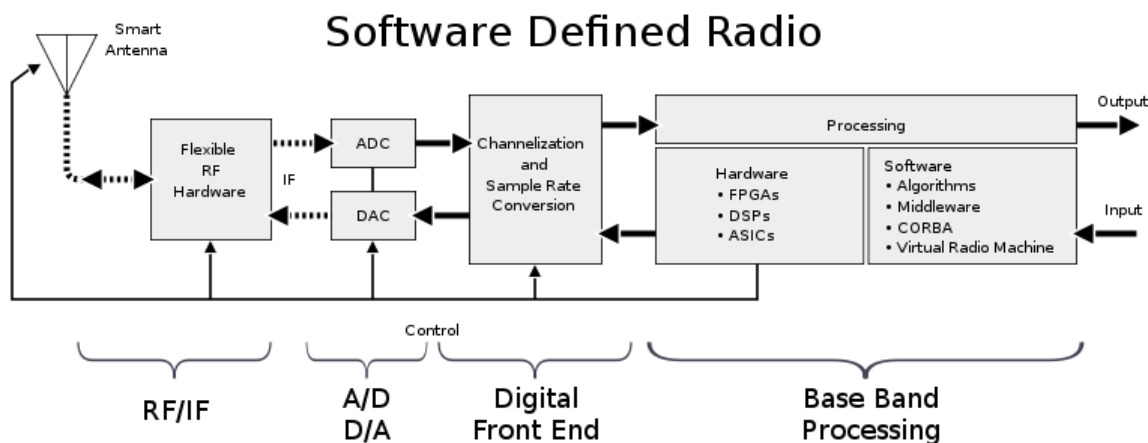


Figura 4.1: Esquema de un sistema de SDR típico. [13]

Podemos modelar directamente sistemas de radiofrecuencia empleando los modelos analíticos que conocemos del estudio de la electrónica, que están constituidos por componentes discretos, tales como condensadores, bobinas y resistencias. Figura 4.2.

En un sistema de SDR como el que vamos a utilizar distinguimos claramente dos partes fundamentales bien diferenciadas, el hardware de radio, y el ordenador con el software de procesamiento. En nuestro caso el hardware será un dispositivo experimental llamado *HackRF One*, y el software de SDR, *GNU Radio*.

El ordenador empleado puede ser un equipo de propósito general, aunque también se utilizan otros sistemas, como partes de un SoC, estos últimos se suelen emplear en los dispositivos móviles, de forma que el fabricante puede en cualquier momento modificar sus sistemas radio mediante actualizaciones de software.

Pero la ventaja no es solamente la posibilidad de actualizar el software, ya que de hecho esto no tiene por qué ocurrir durante la vida útil del sistema embebido, sino que de otra manera sería imposible obtener la integración que requieren muchas interfaces de comunicación inalámbricas, donde no

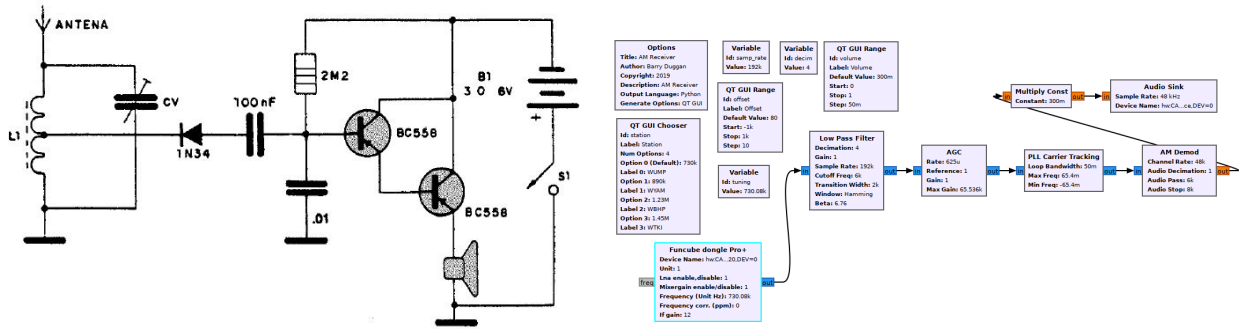


Figura 4.2: Comparación de un circuito demodulador AM, y un esquema de GNU Radio Companion para el mismo propósito, pero de características muy superiores. [14] [15]

se admite el uso de electrónica discreta para el tratamiento de señales.

Actualmente, casi todos los sistemas de RF lo son, o tienen etapas de SDR, sintonizadores de televisión, mandos, telemetría, conectividad inalámbrica, etc. muchos de estos sistemas son embebidos, y aunque no se distinguen sus partes, la arquitectura es la misma, su radio está definida a través de un software que está grabado en la memoria de este y utiliza una unidad de procesamiento. La figura 4.1 representa mediante bloques un modelo de sistema SDR típico.

El esquema básico de funcionamiento de un sistema de SDR con GNU Radio y HackRF One, se muestra en la figura 4.3 y sería el siguiente:

En recepción, tenemos una etapa analógica de entrada y filtrado de la señal de RF, que se introduce en un convertidor de frecuencia *down-converter* y la sitúa en banda base, la señal en banda base es digitalizada mediante un ADC que muestrea la señal, y la envía al software del ordenador, en algunos sistemas SDR el hardware de adquisición incluye sistemas de DSP para asumir tareas de procesamiento y filtrado digital previas a las que realiza el software.

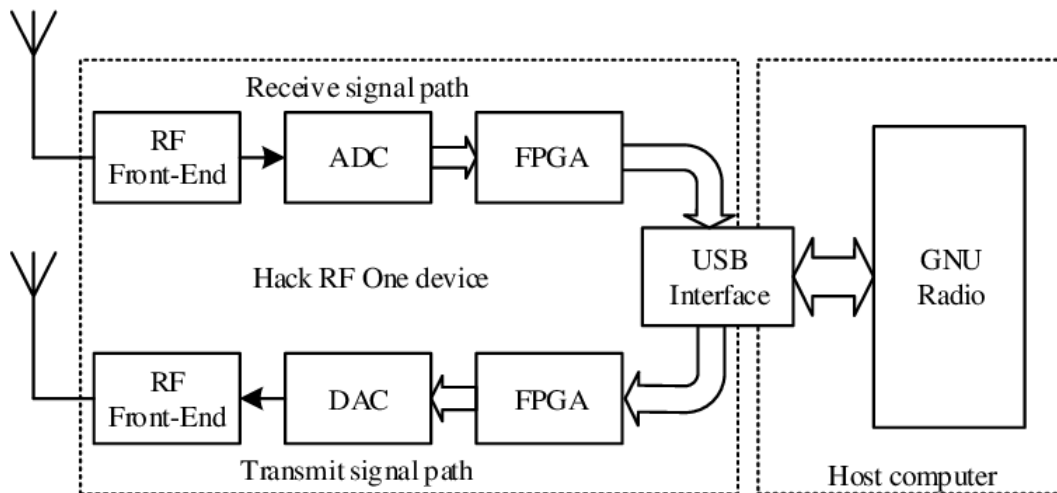


Figura 4.3: Esquema de transmisión y recepción del HackRF One con GNU Radio. [16]

En transmisión, el ordenador genera a partir del software de SDR una secuencia correspondiente a una señal en banda base, y la transfiere al DAC, la señal en banda base analógica resultante de dicha conversión se traslada a la frecuencia deseada de RF con un *up-converter* y se ajusta su potencia con

un amplificador.

Algunas ventajas significativas de la SDR son por tanto:

- Admite fabricación a gran escala.
- No es necesario ubicar componentes discretos.
- No degrada las señales.
- No requiere reajustes manuales.
- Es actualizable.

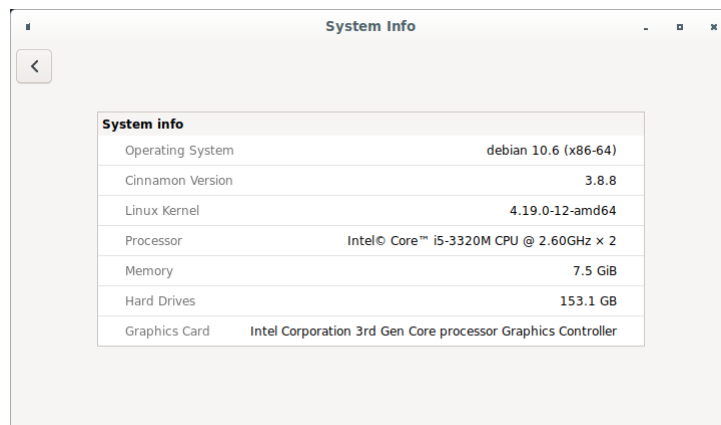
4.2. Modelado de SDR mediante GNU Radio (GR)



Figura 4.4: Logo del proyecto GNU Radio. [17]

GNU Radio (GR) es un conjunto de herramientas gratuitas, del proyecto GNU, que ofrece bloques de procesamiento y librerías de señal para implementar Radio Definida por Software.

El ámbito para el que está pensado es muy amplio, como su comunidad, ya que es utilizado en investigación, industria, sector académico, e incluso por los aficionados y experimentadores.



System info	
Operating System	debian 10.6 (x86-64)
Cinnamon Version	3.8.8
Linux Kernel	4.19.0-12-amd64
Processor	Intel® Core™ i5-3320M CPU @ 2.60GHz × 2
Memory	7.5 GiB
Hard Drives	153.1 GB
Graphics Card	Intel Corporation 3rd Gen Core processor Graphics Controller

Figura 4.5: Detalles del sistema operativo y las características del equipo que vamos a utilizar.

El sistema operativo que vamos a emplear es Debian GNU/Linux en su versión estable 10.6, con la versión de GNU Radio 3.7.13.4-4. No se trata de la más reciente, pero es la que está aprobada en el repositorio de *Debian Stable*, por lo que es muy conveniente para trabajar de forma estable y segura. Estos detalles se pueden comprobar en la figura 4.5 y 4.6.

Para instalar GNU Radio solamente hemos tenido que acudir a la consola y realizar su instalación mediante apt-get, el encargado de mantener el sistema actualizado y satisfacer las dependencias de software en Debian:

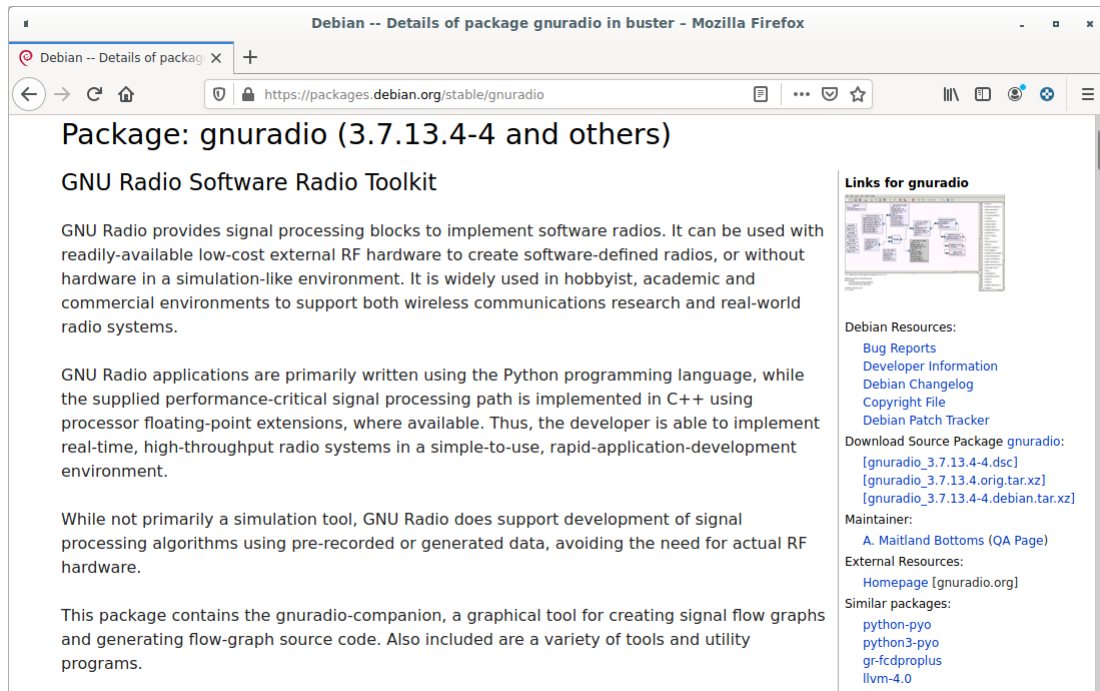


Figura 4.6: Descripción del paquete GNU Radio 3.7.13.4-4 en el repositorio oficial del sistema operativo Debian.

```
usuario@debian:~$ sudo apt-get update
usuario@debian:~$ sudo apt-get install gnuradio
```

Por tanto, no es necesario instalar nada más, ya que apt-get satisficará las dependencias de software, es decir, que instalará todo lo necesario para ejecutar GNU Radio. Si se necesita información sobre dichas dependencias o paquetes opcionales es posible consultar en la página de la figura 4.6:

<https://packages.debian.org/stable/gnuradio>

4.2.1. Introducción a GNU Radio Companion (GRC)

La forma habitual de trabajar con GNU Radio es a través del programa *GNU Radio Companion (GRC)*, que es un editor de diagramas, al estilo *Simulink* de MATLAB o *LabVIEW*, este programa genera scripts en Python, a partir de las librerías de GNU Radio, por lo que cuando acabemos de diseñar nuestros programas de SDR, simplemente necesitaremos una distribución de GNU Radio instalada en el sistema para ejecutarla, sin necesidad de cargar el diagrama de bloques.

El software correctamente liberado al público, que necesita de GNU Radio, lo indica en su lista de dependencias, por lo que si intentamos instalar cualquier software en el sistema operativo que requiera GNU Radio, el gestor de dependencias de software lo instalará automáticamente, pues es una característica que tienen todos los sistemas Linux habituales, para evitar que el usuario tenga que instalar librerías o programas necesarios.

La versión de GNU Radio Companion que viene en la instalación de GNU Radio es 3.7.13.4.

Vamos a detallar el funcionamiento básico de GRC a modo de resumen, para conocer las funciones principales, en la documentación del wiki oficial existe una explicación extensa y actualizada de todas las funciones y procedimientos realizables, tanto de GNU Radio a nivel de programación e implementación de bloques, como del software GNU Radio Companion con los bloques oficiales, por lo que es recomendable haberla al menos ojeado un poco y tenerla siempre a nuestra disposición:

<https://wiki.gnuradio.org>



Figura 4.7: Información sobre la versión de GRC instalada por el gestor de paquetes.

4.2.2. Entorno de trabajo

El entorno de trabajo de GNU radio cumple con la estructura habitual del software de edición de diagramas, y en general de cualquier software informático típico, puede verse en la captura de la figura 4.8.

En la parte superior de la ventana tenemos la barra de menús, donde tenemos acceso a todas las funcionalidades del programa.

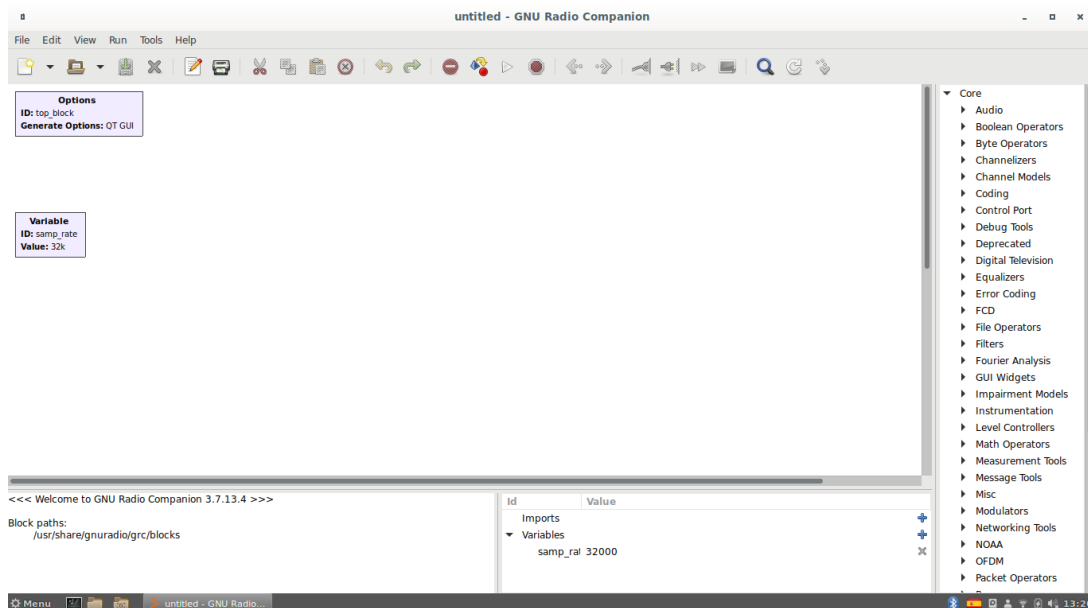


Figura 4.8: Espacio de trabajo de un proyecto nuevo en GNU Radio Companion, sobre el escritorio de Debian.

En el menú *File* tenemos todas las opciones para *crear*, *cargar* y *guardar* los proyectos en formato *.grc*, así como una opción que nos permite hacer una *captura de pantalla*, que consiste en una imagen en formato PNG con el diagrama, del tamaño que hayamos especificado que es nuestro espacio de trabajo. También disponemos de la opción de cerrar el proyecto y terminar el programa. Las funciones más importantes de estos menús aparecen en la fila de iconos que hay debajo de la barra. Si ponemos el cursor sobre ellos nos dirá para que sirven.

A la hora de arrastrar los bloques que se encuentran organizados en la lista de la derecha, los encontraremos de tres tipos fundamentales, de *entrada o source*, de *procesado*, y de *salida o sink*.

4.2.3. Bloques de entrada o source

Los bloques tipo source, introducen señales en el diagrama, desde señales generadas por el propio bloque como puede ser un bloque de generación de funciones, la señal del micrófono, paquetes de Internet, o la lectura de un archivo almacenado en el disco.

4.2.4. Bloques de procesado

Estos bloques realizan operaciones a partir de una señal de entrada, y devuelven una salida.

Distinguimos tres casos posibles de bloques de procesado según el ratio de muestras de entrada y salida:

- **Igual a igual:** Estos bloques permiten realizar procesados, donde el resultado de la operación realizada a la señal mantiene la misma relación de muestras de entrada y salida.
- **Diezmadores:** Estos bloques generan una relación a la salida inferior a 1, el número de muestras que sale es menor que el que entra, es decir, son compresores y bajan la tasa de muestreo.
- **Interpoladores:** Estos bloques generan una relación a la salida superior a 1, el número de muestras que sale es mayor que el que entra, es decir, son expansores, por lo que elevan la tasa de muestreo.

No podemos pasar por alto el carácter diezmador o interpolador de los bloques, si la tasa de muestreo no es correcta, las frecuencias digitales dejan de corresponderse con las equivalentes analógicas y los parámetros que dependen de ajustes de frecuencia analógicos, como un generador de señal por ejemplo, no funcionarían correctamente. Lo cual sería aún más grave si la señal tiene que visualizarse en un gráfico de densidad espectral de potencia o enviarse a un hardware como una tarjeta de SDR o la propia tarjeta de sonido del equipo.

4.2.5. Bloques de salida o sink

Los bloques tipo sink, recogen muestras del diagrama y funcionan como una interfaz con el resto del sistema, es decir, son capaces de enviar muestras a los dispositivos y funciones del sistema operativo. Por ejemplo hay bloques sink para enviar la señal digital a un hardware de RF, hay bloques para enviar sonido a la tarjeta de sonido, hay bloques para escribir las muestras en un fichero, incluso hay bloques sink para enviar los datos a través de Internet usando el protocolo UDP.

4.3. Dispositivo para SDR, HackRF One

Para finalizar este capítulo vamos a presentar el hardware que vamos a utilizar en este trabajo. Se trata del HackRF One, es un dispositivo de uso experimental, diseñado por Great Scott Gadgets. El dispositivo es Open Hardware, por lo que es posible encontrar en su página el diagrama electrónico.

4.3.1. Características técnicas del HackRF One

Según Great Scott Gadgets, el HackRF One, posee las siguientes características: [20]

- frecuencia de operación, desde 1 MHz hasta 6 GHz
- transceptor half-dúplex
- hasta 20 millones de muestras por segundo
- muestras de 8 bits en cuadratura (8 bits I y 8 bits Q)

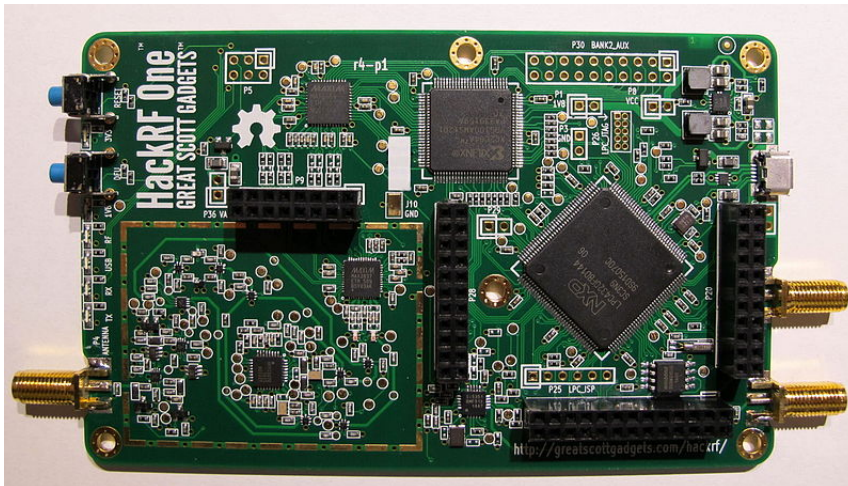


Figura 4.9: Circuito impreso del HackRF One al descubierto.



Figura 4.10: Antena ANT500 que vamos a usar para las pruebas. [20]

- compatible con GNU Radio, SDR#, y más
- ganancia configurable por software en RX y TX y filtro en banda base
- puerto de antena con potencia controlada por software (50 mA a 3.3 V)
- conector SMA hembra de antena
- conector SMA hembra para entrada de reloj y salida para sincronización
- botones para el modo programación
- pines internos de expansión
- USB 2-0 de alta velocidad
- Alimentado por USB
- hardware open source

La antena que vamos a usar se llama ANT500 y está diseñada para usar en el HackRF One, aunque se vende por separado. Se trata de una antena telescópica de 50 Ohms, que funciona entre 75 MHz y 1 GHz.

Aunque para el caso del modulador de AM comercial no cubra, la potencia radiada es suficiente para realizar la prueba.

Para obtener más información sobre el dispositivo se puede consultar la página web:

<https://greatscottgadgets.com/hackrf/one/>



Figura 4.11: Aspecto exterior del HackRF One sin la antena conectada. [20]

4.3.2. Instalación del software de HackRF en Debian

En nuestro caso vamos a instalar el software de un proyecto llamado *gr-osmosdr*. Este proyecto de software libre consiste en dar soporte a una conjunto de dispositivos de SDR de uso extendido.

La lista de hardware soportado por el bloque osmocom que aparece en la página del proyecto es la siguiente: [18]

- FunCube Dongle a través de libgnuradio-fcd
- FUNcube Dongle Pro+ a través de gr-fcdproplus
- sysmocom OsmoSDR Devices a través de libosmosdr
- Nuand LLC bladeRF a través de libbladeRF library
- **Great Scott Gadgets HackRF a través de libhackrf**
- Ettus USRP Devices a través de Ettus UHD library
- Fairwaves UmTRX a través de Fairwaves' fork of Ettus' UHD library
- RFSPACE SDR-IQ, SDR-IP, NetSDR (con opción X2 incluida)
- Dongles RTL2832U para DVB-T a través de librtlsdr
- Servidor de espectro RTL-TCP (consultar el proyecto librtlsdr)
- Dongles MSi2500 para DVB-T a través de libmirisdr
- SDRplay RSP a través de SDRplay API library
- Dongles AirSpy R820t a través de libairspy
- Entradas .cfile de gnuradio a través de libgnuradio-blocks

En la página se explica como compilar el proyecto para instalarlo, pero como existe un paquete precompilado en el repositorio de Debian, el sistema que estamos usando, es posible instalar *gr-osmosdr* de la forma siguiente:

```
usuario@debian:~$ sudo apt-get update
usuario@debian:~$ sudo apt-get install gr-osmosdr
```

Si fuera necesario consultar la información del proyecto, está disponible en la página oficial:

<https://osmocom.org/projects/gr-osmosdr/wiki>

Capítulo 5

Simulación de moduladores en GNU Radio

En este capítulo vamos a implementar diferentes moduladores, utilizando GNU Radio Companion, el propósito es el de mostrar el proceso de implementación y el funcionamiento del programa con determinados ejemplos de modulaciones.

Los moduladores que vamos a implementar en esta sección son una selección de siete de los más comunes y serán: *AM*, *DSB-SC*, *USB*, *LSB*, *ASK*, *PSK* y *FM*.

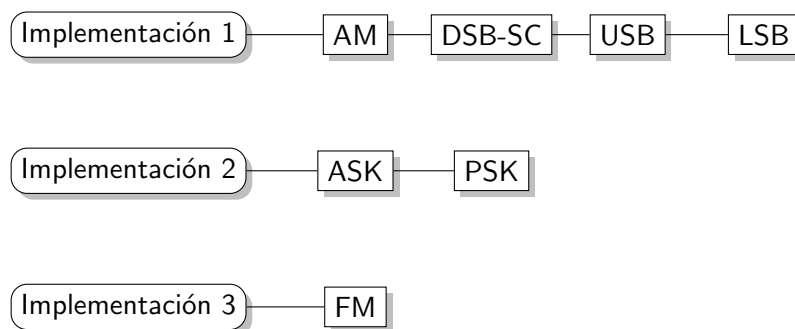


Figura 5.1: Orden de desarrollo de los moduladores derivados.

Vamos a trabajar sobre tres diseños generales, como se indica en la figura 5.1:

- Partiremos de un modulador AM, con el que veremos detalladamente cual es el proceso de implementación de un sistema. Una vez comprobado su funcionamiento, realizaremos las modificaciones necesarias para transformarlo en un modulador DSB-SC, posteriormente lo derivaremos en un modulador USB mediante una técnica de discriminación de banda, y finalmente con una última modificación, en un modulador LSB.
- Después implementaremos un nuevo diseño, que consistirá en un modulador ASK, el cual mediante una modificación aritmética se convertirá en modulador PSK.
- Finalmente realizaremos un sencillo modulador de FM.

5.1. Amplitud modulada (AM)

GNU Radio Companion estima que por defecto vamos a trabajar en un tipo de datos al que llama Complex, los tipos de datos Complex, se componen de dos señales, que se corresponden a una componente en fase (I) almacenada en la parte real, y la otra en cuadratura (Q) almacenada en la

imaginaria. GRC utiliza el color azul para señalar las entradas y salidas configuradas en Complex.

Para el caso que nos ocupa, como es un modulador AM, no es necesario emplear bloques de tipo Complex, por lo que vamos a realizar el montaje usando bloques Float. Para elegir el tipo de datos de un bloque simplemente tenemos que acceder a su configuración usando el botón derecho o pinchando dos veces sobre él. Cuando hayamos configurado los bloques para Float, sus entradas y salidas serán de color naranja.

5.1.1. Desarrollo del esquema del modulador

Creemos un nuevo proyecto, lo primero que vamos a encontrar es dos bloques ya predefinidos, uno de ellos es la configuración general del proyecto, y el otro es una variable llamada *samp_rate*, esta variable no es imprescindible, pero la probabilidad de usarla es tan alta, que GRC nos la ofrece cuando creamos un nuevo proyecto. En la figura 5.2 se observan estos bloques.

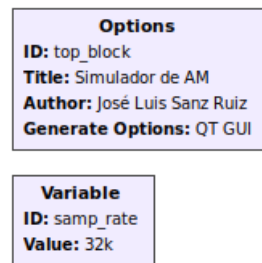


Figura 5.2: Configuración de proyecto *Simulador de AM* y la variable *samp_rate* incluida por defecto.

En primer lugar añadimos desde la librería de bloques situada en la parte derecha, al esquema, todos los bloques necesarios para realizar la modulación a nivel de señal (ver apartado 3.1.1):

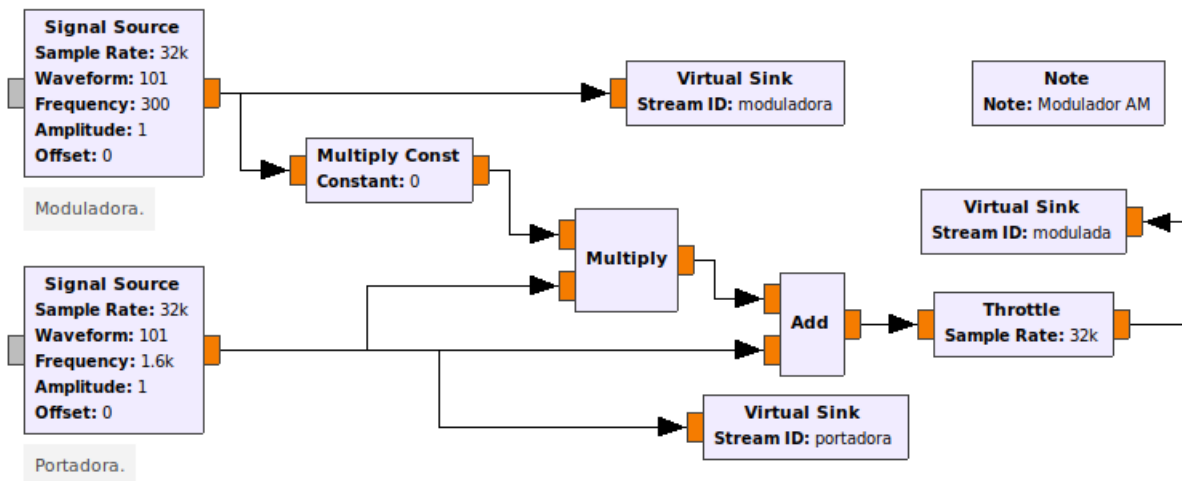


Figura 5.3: Esquema del modulador AM.

- **1 Mezclador:** Core → Maths Operators → Multiply.
- **1 Sumador:** Core → Maths Operators → Add.
- **1 Control de ejecución:** Core → Misc → Throttle.
- **2 Fuentes de señal:** Core → Waveform Generators → Signal Source.

- **1 Constante multiplicadora:** Core → Maths Operators → Multiply Const.
- **3 Salidas virtuales:** Core → Misc → Virtual Sink.

Podemos emplear el buscador de bloques pinchando sobre el icono de la lupa, así no será necesario buscar en la lista cuando ya sepamos el nombre.

El *mezclador* se encargará de realizar la modulación con la moduladora que vendrá de una *fente de señal*, y la portadora, que vendrá de la otra *fente de señal*.

El *multiplicador*, que se encargará de multiplicar por k la moduladora, y actuará como sensibilidad del modulador para ajustar el índice de modulación.

El *sumador* de reinsertar la portadora.

El *control de ejecución* hará que el diagrama se ejecute a una determinada tasa de muestreo sincronizada con el reloj del ordenador. Emplearemos siempre este bloque cuando no exista ningún bloque que sea capaz retener la ejecución, si no utilizamos este bloque, GNU Radio intentará ejecutar el diagrama tan rápido como pueda.

En primer lugar vamos a configurar la moduladora, pinchando dos veces sobre el bloque destinado a serlo, una vez abierta la configuración observamos como es posible establecer los valores de funcionamiento. En nuestro caso no vamos a realizar un circuito simple, ya que queremos exprimir al máximo las posibilidades, por lo que en lugar de valores vamos a emplear variables, que luego podremos modificar empleando interfaces gráficas, modificando así los valores en tiempo de ejecución.

De la misma forma añadimos la configuración de la portadora.

Por tanto rellenamos la configuración ambos bloques *Signal Source* con la configuración, como en el cuadro 5.1:

ID	moduladora	portadora
Output Type	Float	Float
Waveform	forma_onda_m	forma_onda_c
Frequency	F_m	F_c
Amplitude	A_m	A_c
Offset	offset_m	0

Cuadro 5.1: Configuración de los bloques generadores de moduladora y portadora.

En la figura 5.4 se observa una captura de la interfaz de configuración en GRC, correspondiente al bloque *Signal Source*, donde van introducidos los datos expresados en el cuadro 5.1. A partir de ahora usaremos cuadros para describir la configuración de los bloques que sea necesario configurar, en lugar de usar capturas de pantalla.

Como se observa, hemos usado el subfijo m, para nombrar a las variables correspondientes a la moduladora, y c, para nombrar a las variables correspondientes a la portadora, siguiendo la misma nomenclatura, es importante hacer esto a la hora de mantener el orden.

Al principio de poner estos valores, GRC nos marcará en rojo los bloques, pues detectará que las variables no están definidas. No deberíamos preocuparnos de esto hasta que no esté todo definido y los bloques conectados.

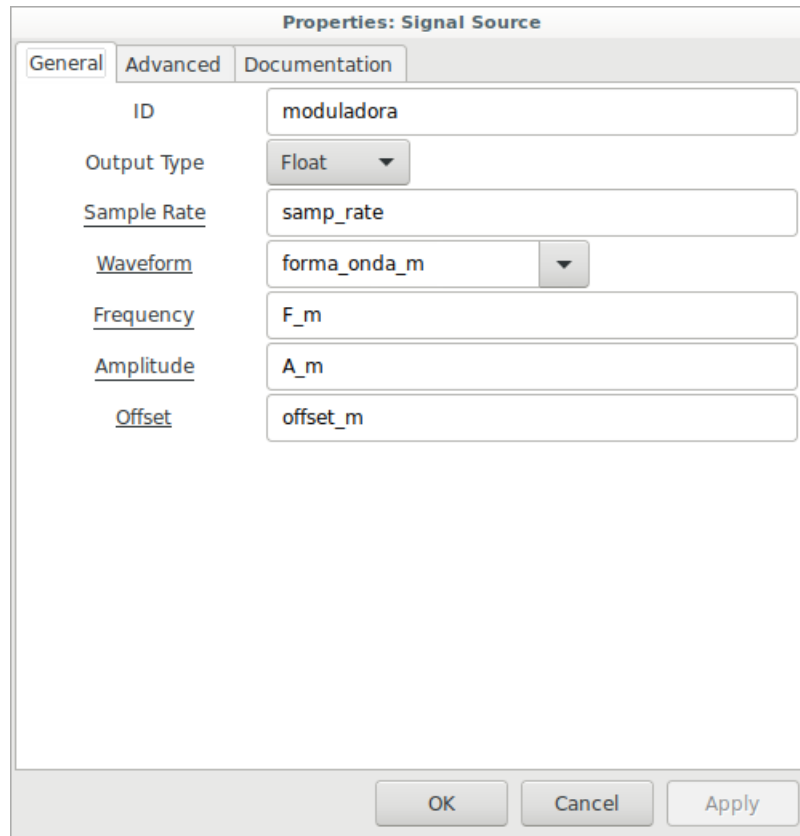


Figura 5.4: Interfaz de configuración del bloque *Signal Source* destinado a la señal moduladora.

El bloque de control de ejecución *Throttle* va configurado con el valor de la variable *samp_rate*, predefinida a 32.000, por lo que no es necesario intervenir.

Lo que sí es necesario, es configurar las salidas virtuales, *Virtual Sink* con una Stream ID cada una. Nos permitirán recuperar la señal que entra en ellas, usando un bloque de fuente virtual *Virtual Source* en otra parte del diagrama, sin necesidad de cruzar conexiones. Este Stream ID será una cadena de texto con caracteres de cualquier tipo.

El bloque *Multiply Const* tendrá valor k , ya que luego le daremos el valor que queramos que tome.

5.1.2. Interfaz de control del modulador

Esta es quizás la parte más complicada del simulador, si se quiere realizar correctamente. Pues normalmente, cuando se empieza a programar en GRC, lo normal es no prestar atención al mecanismo de ordenación de widgets que utiliza el gestor QT a la hora de organizar tanto las visualizaciones, como los controles que muestra la interfaz. Esto se debe a que cuando no especificamos nada en la configuración, la distribución se realiza de forma automática.

QT es un framework multiplataforma, de código abierto, que es ampliamente usado para desarrollar programas con interfaces gráficas, y en el cual se apoya GNU Radio a la hora de implementar interfaces gráficas.

En la configuración de los bloques que emplean QT, existe siempre una opción llamada *GUI hint*, esta opción admite una sintaxis que nos va a permitir organizar nuestras interfaces gráficas de manera cómoda y sencilla.

Si consideramos la ventana una cuadrícula, la cual está dividida en filas y columnas, siempre tendremos que especificar cuatro valores: fila, columna, relleno de fila y relleno de columna, tal como se muestra en la figura 5.5:

- **Fila:** es el número de fila, empezando por 0, donde queremos ubicar el widget.
- **Columna:** es el número de columna empezando por 0, donde queremos ubicar el widget.
- **Relleno de fila:** es el número de filas que va a ocupar el widget.
- **Relleno de columna:** es el número de columnas que va a ocupar el widget.

(0,0)	(0,1)	(0,2)	(0,3)
(1,0)	(1,1)	(1,2)	(1,3)
(2,0)	(2,1)	(2,2)	(2,3)

Waveform Selector (0,0,2,1)	Offset Slider (0,1,1,1)
	Frequency Slider (1,1,1,1)
Time Display (2,0,1,1)	Frequency Display (2,1,1,1)

Figura 5.5: Dibujo de la documentación sobre GUI hint en GNU Radio. [21]

No vamos a reparar en especificar la estrategia de organización, pero sí mostraremos los argumentos utilizados en los GUI hint de cada bloque de interfaz gráfica.

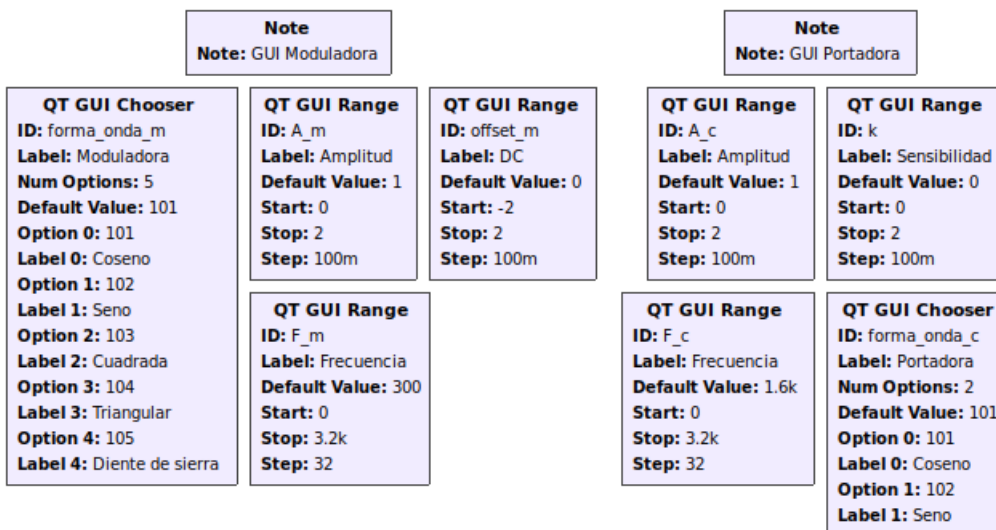


Figura 5.6: Bloques de control del simulador AM.

Los bloques utilizados para el control del simulador AM son los que se muestran en la figura 5.6:

- **2 Selectores:** Core → GUI Widgets → QT → QT GUI Chooser.
- **6 Deslizantes:** Core → GUI Widgets → QT → QT GUI Range.

En realidad el bloque *QT GUI Range* permite diferentes tipos de controles, pero nosotros vamos a utilizar unos que son deslizantes, y al mismo tiempo nos permitirán introducir valores por teclado,

o pulsando flechas.

En primer lugar vamos a configurar los dos selectores, estos dos selectores nos van a permitir realizar la selección de forma de onda. En la moduladora estará disponible: *coseno*, *seno*, *onda cuadrada*, *onda triangular* y *diente de sierra*. Para la portadora sólo pondremos *coseno* y *seno*.

Empezando por el selector de la moduladora, la ID de este bloque generará una variable, que tomará un valor según la opción que se escoja en el selector, esta variable, es la que anteriormente habíamos asignado a la forma de onda de la moduladora, y que finalmente a partir de este paso quedará declarada. Configurarlos es sencillo, seleccionamos el número de opciones que queremos y las cumplimentamos.

Para hacer referencia a las formas de onda hemos usado 101, 102, 103, 104 y 105, pues en un principio

ID	forma_onda_m	forma_onda_p
Label	Moduladora	Portadora
Type	Integer	Integer
Num Options	5	2
Default Value	101	101
Option 0	101	101
Label 0	Coseno	Coseno
Option 1	102	102
Label 1	Seno	Seno
Option 2	103	103
Label 2	Cuadrada	–
Option 3	104	–
Label 3	Triangular	–
Option 4	105	–
Label 4	Diente de sierra	–
Widget	Combo Box	Combo Box
GUI Hint	0,0,1,1	1,0,1,1

Cuadro 5.2: Configuración de los bloques *QT GUI Chooser*, para controlar la forma de onda en la moduladora y la portadora.

intentamos referenciarlas por el nombre que tienen en el desplegable de configuración del bloque de generador de señal *Waveform*, tras investigar llegamos a la conclusión de que las diferentes opciones se podían elegir tecleando un entero sobre el desplegable, en el caso de este bloque, 100 significa una forma de onda constante, 101 un coseno, y así sucesivamente, como hemos configurado en el cuadro 5.2.

ID	A_m	F_m	offset_m
Label	Amplitud	Frecuencia	DC
Type	Float	Float	Float
Default Value	1	300	0
Start	0	0	-2
Stop	2	3.2e3	2
Step	0.1	32	0.1
Widget	Counter + Slider	Counter + Slider	Counter + Slider
Minimum Length	200	200	200
GUI Hint	0,1,1,1	0,3,1,1	0,2,1,1

Cuadro 5.3: Control de Amplitud, Frecuencia y Offset, de la moduladora.

ID	A_c	F_c	k
Label	Amplitud	Frecuencia	Sensibilidad
Type	Float	Float	Float
Default Value	1	1600	0
Start	0	0	0
Stop	2	3.2e3	2
Step	0.1	32	0.1
Widget	Counter + Slider	Counter + Slider	Counter + Slider
Minimum Length	200	200	200
GUI Hint	1,1,1,1	1,3,1,1	1,2,1,1

Cuadro 5.4: Control de Amplitud y Frecuencia de la portadora. Control de sensibilidad del modulador.

Para controlar la *frecuencia*, *amplitud* y el *offset*, tanto de la moduladora como de la portadora, (el offset sólo en la moduladora) usaremos tres y tres deslizantes, en cuyo caso la configuración que tomarán será del cuadro 5.3 para la moduladora y el cuadro 5.4 para la portadora.

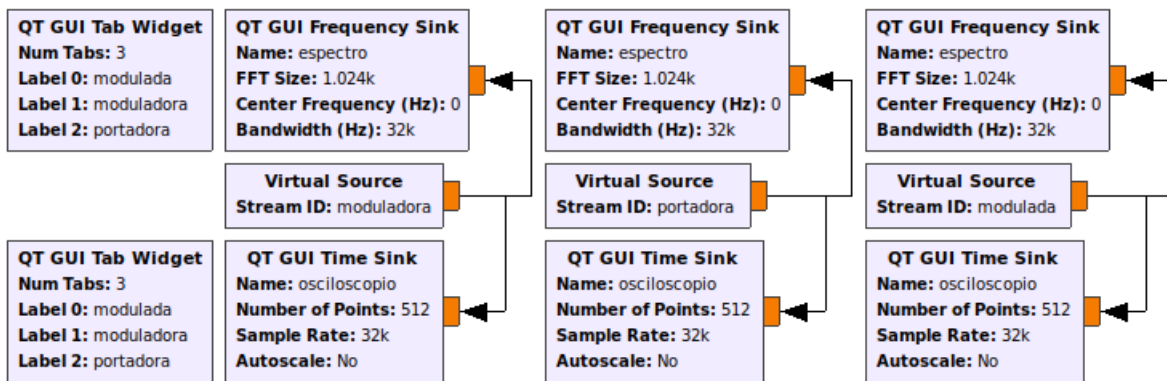


Figura 5.7: Bloques de visualización del simulador AM.

Con esto quedan definidos los controles del simulador AM.

5.1.3. Interfaz de visualización del modulador

Para visualizar las señales que genera nuestro simulador de AM, vamos a emplear tres gráficos para señal en dominio temporal y otros tres gráficos para la señal en el dominio de la frecuencia, los cuales serán para visualizar la señal moduladora, la portadora y la modulada. Para no ocupar toda la pantalla, haremos uso de las pestañas (tabs). Tres fuentes virtuales *Virtual Source* nos permitirán reinsertar las señales producidas anteriormente en el modulador, a las cuales ya habíamos asignado un Stream ID, como "moduladora", "portadora" y "modulada". El esquema empleado es el de la figura 5.7.

Elementos a importar de la librería de bloques:

- **3 Osciloscopios:** Core → Instrumentation → QT → QT GUI Time Sink.
- **3 Analizadores de espectro:** Core → Instrumentation → QT → QT GUI Frequency Sink.
- **2 Pestañas:** Core → GUI Widgets → QT → QT GUI Tab Widget.
- **3 Fuentes virtuales:** Core → Misc → Virtual Source.

Si antes explicábamos la necesidad de una sintaxis en los campos de configuración GUI Hint, en este caso, será necesario ir más allá, y extender la sintaxis para el uso de pestañas.

Partimos de un bloque QT GUI Tab Widget, donde seleccionamos el número de pestañas que deseamos y cumplimentamos los campos Label 0, Label 1, Label 2, etc. con los títulos de las pestañas, en el campo GUI Hint de la configuración del bloque tecleamos el lugar donde queremos situarlo.

Para introducir widgets dentro de una pestaña, en el campo GUI Hint del bloque que haya que destinar usaremos la sintaxis "ID@PESTAÑA", donde ID será la identidad del bloque QT GUI Tab que ofrece las pestañas, y PESTAÑA el número de pestaña partiendo de 0.

Si queremos además posicionar el widget dentro de la pestaña, usaremos dos puntos, y a continuación los cuatro números que posicionan el widget. Por ejemplo para posicionar un widget de 1x1 en la primera fila y columna de la segunda pestaña de un QT GUI Tab Widget cuya ID es "tablas", emplearemos: "tablas@1:0,0,1,1".

ID	selector_tiempo	selector_frecuencia
Num Tabs	3	3
Label 0	modulada	modulada
Label 1	moduladora	moduladora
Label 2	portadora	portadora
GUI Hint	2,0,1,2	2,2,1,2

Cuadro 5.5: Configuración de los bloques *QT GUI Tab Widget*, para conmutar entre las tres señales.

Una vez comprendido esto, ya podemos utilizar dos *QT GUI Tab Widget*, uno para integrar la visualización en función del tiempo de las tres señales, y otro para el de la frecuencia. La configuración de los QT GUI Tab Widgets es la que se encuentra definida en el cuadro 5.5.

ID	moduladora_scope	portadora_scope	modulada_scope
Type	Float	Float	Float
Name	osciloscopio	osciloscopio	osciloscopio
Y Axis Label	Amplitud	Amplitud	Amplitud
Y Axis Unit	""	""	""
Number of Points	2**9	2**9	2**9
Sample Rate	samp_rate	samp_rate	samp_rate
Grid	Yes	Yes	Yes
Autoscale	No	No	No
Y min	-2	-2	-2
Y max	2	2	2
Number of Inputs	1	1	1
Update Period	0.10	0.10	0.10
Disp. Tags	No	No	No
GUI Hint	selector_tiempo@1	selector_tiempo@2	selector_tiempo@0

Cuadro 5.6: Configuración de los bloques *QT GUI Time Sink* para mostrar las señales en el dominio del tiempo.

El bloque QT GUI Time Sink permite realizar visualizaciones temporales de las señales, pues representa en una gráfica los diferentes puntos de la señal. Para lograr una buena presentación de los datos lo más recomendable es realizar ajustes hasta obtener la visualización esperada, en mi caso la configuración empleada fue la que se explica en el cuadro 5.6.

ID	moduladora_spect	portadora_spect	modulada_spect
Type	Float	Float	Float
Name	espectro	espectro	espectro
FFT Size	1024	1024	1024
Spectrum Width	Half	Half	Half
Window Type	Blackman-harris	Blackman-harris	Blackman-harris
Center Frequency (Hz)	0	0	0
Bandwidth (Hz)	samp_rate	samp_rate	samp_rate
Grid	Yes	Yes	Yes
Autoscale	No	No	No
Average	None	None	None
Y min	-140	-140	-140
Y max	10	10	10
Y label	Ganancia Relativa	Ganancia Relativa	Ganancia Relativa
Y units	dB	dB	dB
Number of Inputs	1	1	1
Update Period	0.10	0.10	0.10
GUI Hint	selector_frecuencia@1	selector_frecuencia@2	selector_frecuencia@0
Show Msg Ports	No	No	No

Cuadro 5.7: Configuración de los bloques *QT GUI Frequency Sink* para mostrar las señales en el dominio de la frecuencia.

El bloque *QT GUI Frequency Sink* permite visualizar el espectro de la señal, para ello realiza la FFT de esta y la muestra en una gráfica. La configuración que hemos dado a los bloques *QT GUI Frequency Sink* es la del cuadro 5.7.

Los valores que se han utilizado son orientativos, con lo cual en futuras implementaciones se deja a criterio del usuario el optar por modificarlos.

5.1.4. Prueba del simulador de AM

Una vez terminado, pulsaremos en el botón de *play*, que hay en la barra de iconos. También podemos pulsar F6, o acudir al menú *Run*, y elegir *Execute*.

Utilizaremos el botón central del ratón, para configurar la visualización, de forma que podremos establecer los triggers o los tamaños de FFT, si necesitamos aumentar la precisión de la gráfica, también podremos recuadrar con el botón izquierdo del ratón, para realizar zoom en las gráficas.

Para probar el modulador hemos realizado la siguiente modulación:

- Forma onda: coseno
- Amplitud de moduladora: 0.3
- Frecuencia de moduladora 192 Hz
- Amplitud de portadora: 1.5
- Frecuencia de portadora: 1600 Hz
- Sensibilidad: 1
- Ventana de FFT: Hamming

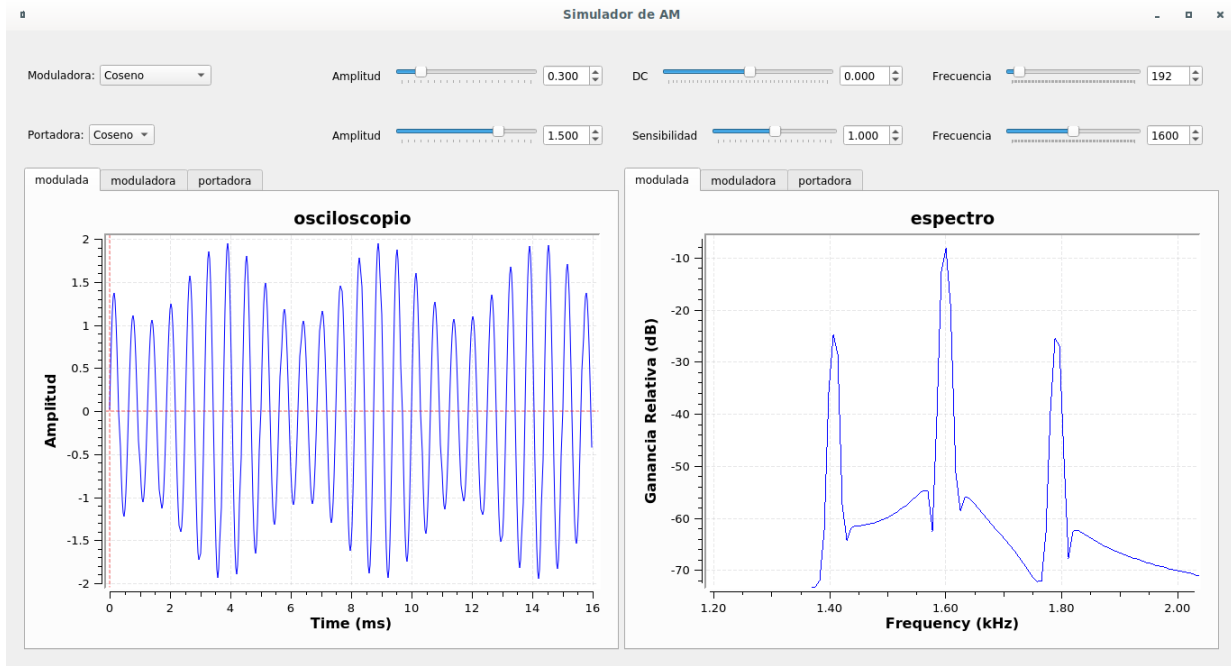


Figura 5.8: Primer ejemplo del modulador AM.

El índice de modulación para estos valores de amplitud es (apartado 3.1.1):

$$m = \frac{A_m \cdot k}{A_p} = \frac{0,3 \cdot 1}{1,5} = 0,2$$

Los resultados obtenidos son los de la figura 5.8, observamos en el espectro la componente de la portadora situada en 1.6 kHz, la banda inferior debida a la moduladora a 1.408 kHz y la superior 1.792 kHz. Es decir a una distancia de 192 Hz. Midiendo con el cursor obtenemos que la portadora está a -8.5 dB, y las bandas a -25 dB. Podemos entender la distribución de potencia que se realiza en el espectro de la modulación AM, entre la portadora y las dos bandas laterales. No obstante al realizar el siguiente modulador, vamos a comparar los resultados.

Para experimentar un poco más con este modulador, ahora los vamos a configurar para que produzca sobremodulación:

- Forma onda: coseno
- Amplitud de moduladora: 0.8
- Frecuencia de moduladora 300 Hz
- Amplitud de portadora: 1
- Frecuencia de portadora: 2000 Hz
- Sensibilidad: 1.4
- Ventana de FFT: Hamming

Para este caso el índice de modulación será:

$$m = \frac{A_m \cdot k}{A_p} = \frac{0,8 \cdot 1,4}{1} = 1,12$$

La portadora queda a -11.3 dB, centrada en 2 kHz, y las bandas laterales en 1.7 y 2.3 kHz, a -17.5 dB.

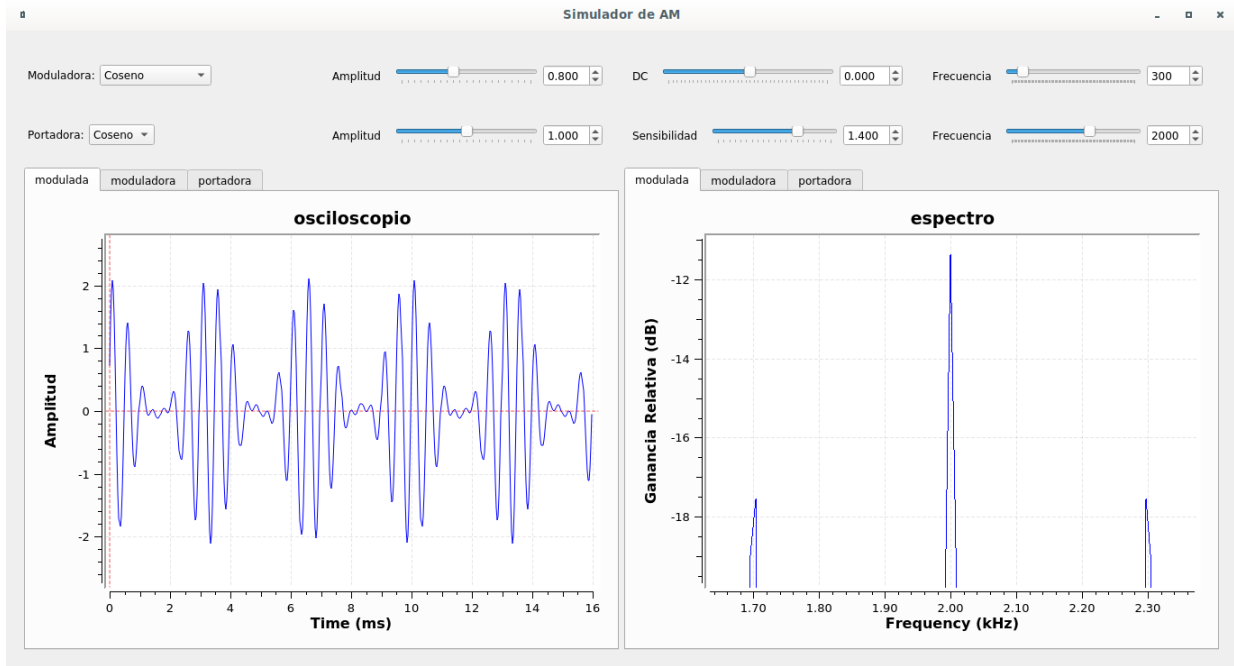


Figura 5.9: Segundo ejemplo del modulador AM.

5.2. Doble banda lateral con portadora suprimida (DSB-SC)

Para realizar este modulador no vamos a iniciar un nuevo proyecto, ya que sería muy tedioso volver a repetir todo, sino que partimos del simulador AM del apartado anterior.

Antes de hacer nada, para evitar alterar el ya desarrollado simulador AM, vamos a ir al menú *file* y vamos a usar la opción de *Save As*, para guardar una copia nueva del proyecto anterior, con un nuevo nombre.

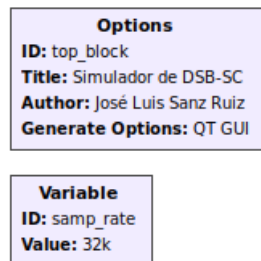


Figura 5.10: Bloques de configuración globales del modulador DSB-SC.

Después vamos a actualizar la configuración como se muestra en la figura 5.10, y ya podemos empezar a trabajar sin perder el diseño previo.

5.2.1. Transformación del modulador AM a DSB-SC

Como sabemos de teoría, en un modulador AM la portadora se suma a la señal que sale del mezclador. La portadora como vimos en teoría ya no es necesaria (apartado 3.1.2), y por lo tanto, vamos a eliminar el sumador que realiza la reinserción de la portadora, y la constante de sensibilidad de AM quedando el diagrama, como se muestra en la figura 5.11.

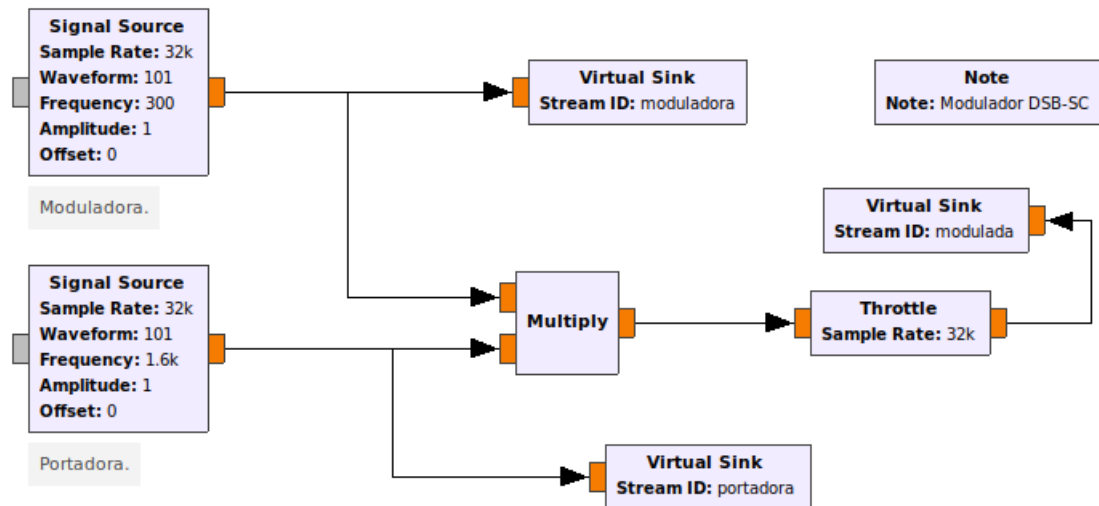


Figura 5.11: Esquema del modulador DSB-SC.

Sustituimos en el QT GUI Range, el control del valor de k , por el valor de offset de la portadora, para no dejar el hueco en la interfaz.

El resto del modulador no tiene ningún cambio, por lo que no vamos a mostrar los demás esquemas.

5.2.2. Prueba del simulador de DSB-SC

Como en el anterior, pulsaremos en el botón de *play*, o F6, o menú *Run* y después *Execute*.

Para probar el modulador hemos realizado la misma modulación que con el de AM, ya que queremos comparar las medidas, pues el propósito de trabajar con estos sistemas es el de observar si se cumple la teoría:

- Forma onda: coseno
- Amplitud de moduladora: 0.3
- Frecuencia de moduladora 192 Hz
- Amplitud de portadora: 1.5
- Frecuencia de portadora: 1600 Hz
- Sensibilidad: 1
- Ventana de FFT: Hamming

El resultado de la modulación se encuentra en la figura 5.12, en este caso ya no se observa la señal portadora, sólo el espectro de la moduladora en las dos bandas laterales. Estas bandas laterales están a -22.5 dB. Si comparamos con la modulación AM, teníamos -25 dB en las bandas laterales.

Vamos a replicar ahora el segundo ejemplo que hemos realizado en AM:

- Forma onda: coseno
- Amplitud de moduladora: 1.12
- Frecuencia de moduladora 300 Hz

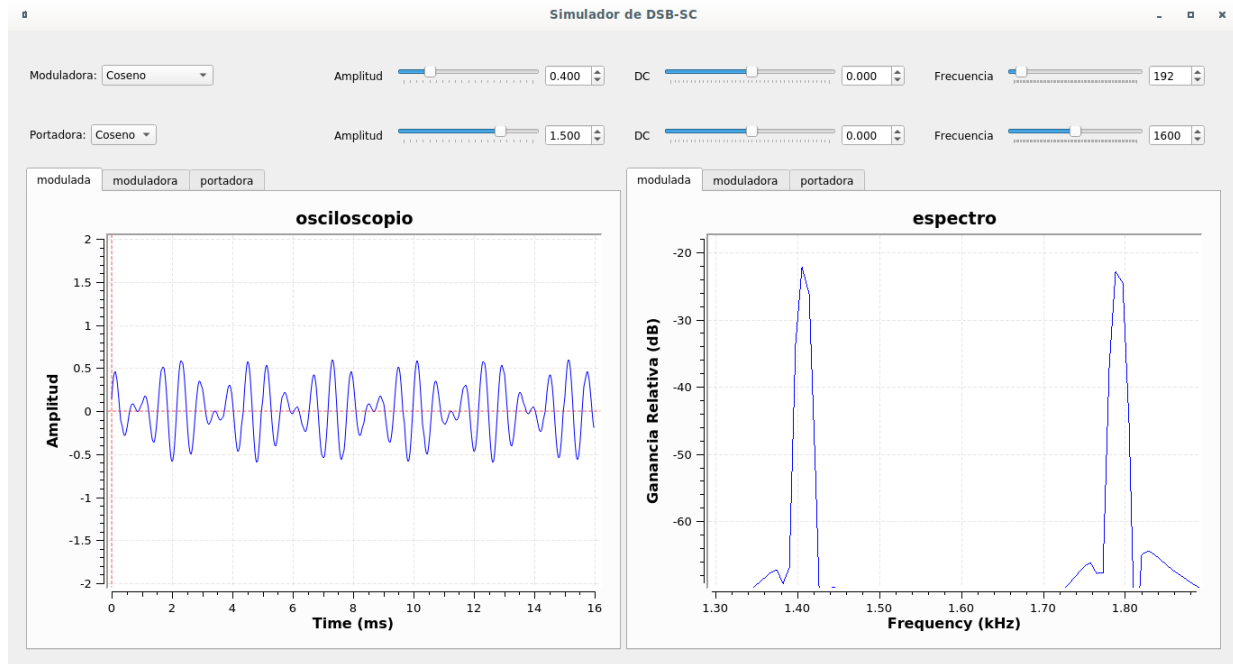


Figura 5.12: Primer ejemplo del modulador DSB-SC.

- Amplitud de portadora: 1
- Frecuencia de portadora: 2000 Hz
- Ventana de FFT: Hamming

Como no tenemos control de sensibilidad, vamos a introducir la moduladora como $0,8 \cdot 1,4 = 1,12$.

En este ejemplo, como muestra la figura 5.13 tenemos las dos bandas laterales a $-16,5$ dB, mientras que en el ejemplo de AM estaban a $-17,5$ dB.

La ventaja vuelve a ser la misma, no se desperdicia potencia en la portadora, y que además en este caso no hay sobremodulación, mientras que en AM sí la había.

5.3. Banda lateral superior (USB)

Ahora que hemos realizado el modulador DSB-SC a partir del modulador AM, vamos a ir un paso más allá, para transformar el modulador DSB-SC en un modulador USB.

Como en el caso anterior, realizamos una nueva copia del simulador de DSB-SC, en la opción *Save As* del menú *file* para modificarlo sin alterar el original.

Sin olvidar que es necesario actualizar el bloque de configuración con el título del nuevo simulador. Figura 5.14.

5.3.1. Transformación del modulador DSB-SC a USB

La transformación para realizar el modulador USB a partir del DSB-SC es sencilla, de hecho son prácticamente iguales, si no fuera porque en el modulador USB vamos a emplear la técnica de discri-

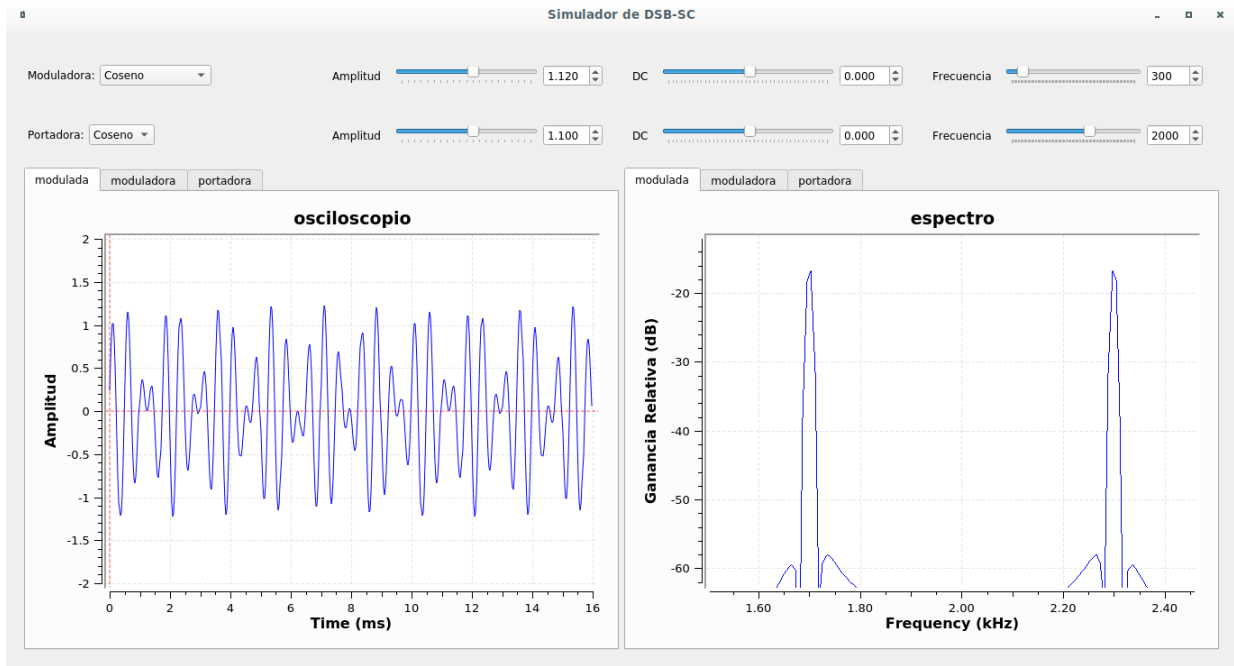


Figura 5.13: Segundo ejemplo del modulador DSB-SC.

Options

ID: top_block
Title: Simulador de USB
Author: José Luis Sanz Ruiz
Generate Options: QT GUI

Variable

ID: samp_rate
Value: 32k

Figura 5.14: Bloques de configuración globales del modulador USB.

minación de banda lateral basada en la fase de la señal (apartado 3.1.3).

Para poder desarrollar el método de discriminación de fase, tenemos que trabajar en fase y cuadratura, es decir, vamos a transformar casi todos los bloques del diagrama de la parte del modulador DSB-SC de Float a Complex. De esta forma automáticamente GNU Radio comienza a trabajar en fase y cuadratura.

En la figura 5.15 y en la figura 5.16 se observan los bloques que han pasado a ser Complex. Hemos mantenido la moduladora, y la modulada, así como los osciloscopios y analizadores de espectro correspondientes a ellas en Float.

ID	hilbert_fc_0
Num Taps	1200
Window	Blackman-harris
Beta	6.76

Cuadro 5.8: Configuración del transformador de hilbert.

Hemos añadido un transformador de hilbert, cuya configuración está en el cuadro 5.8, que es un desfaseador de banda ancha, el cual realiza un desfase de 90° durante todo el ancho de banda de la

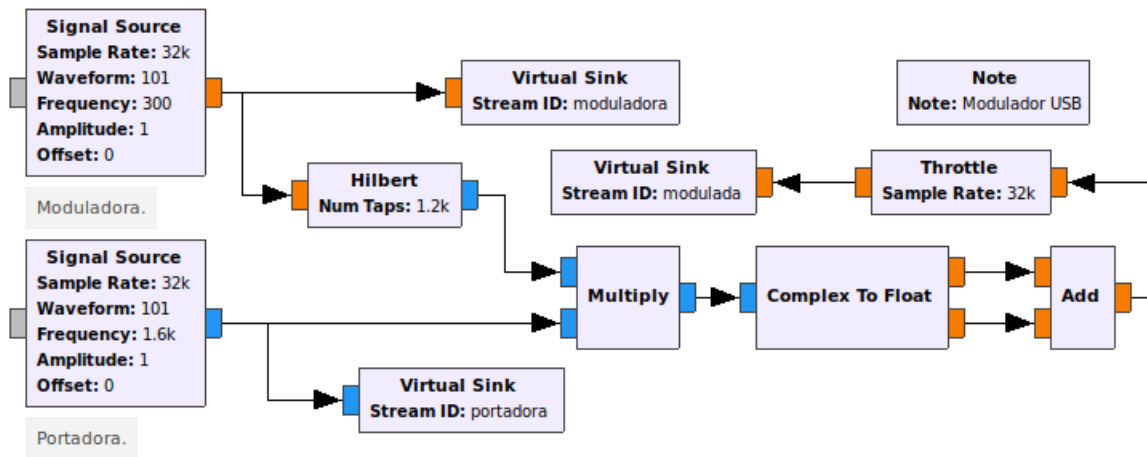


Figura 5.15: Diagrama del modulador USB.

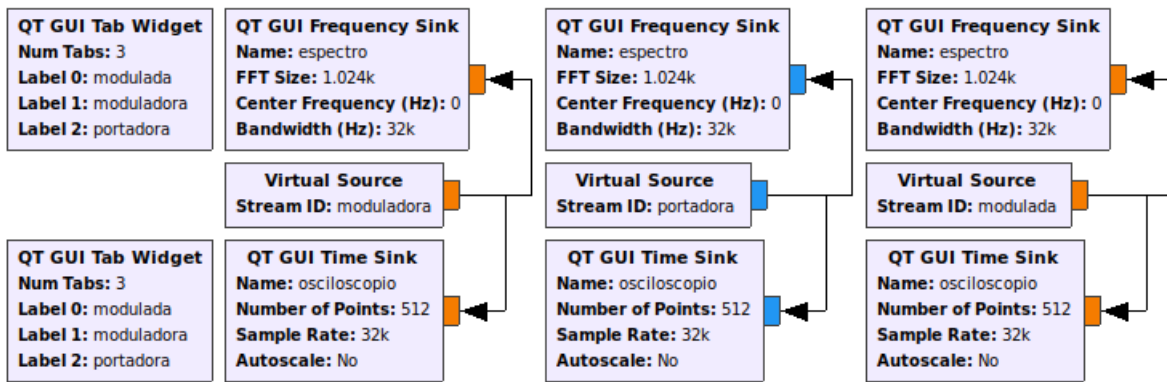


Figura 5.16: Bloques de visualización del modulador USB.

moduladora.

Al utilizar el filtro transformador de hilbert para conseguir la señal moduladora en cuadratura, y multiplicar por la portadora en fase y la portadora en cuadratura respectivamente, obtendremos dos señales moduladas, donde las bandas laterales inferiores se encuentran en contrafase. Al sumarse se producirá la cancelación de la banda inferior.

Para sumar ambas partes hemos separado la fase y la cuadratura usando el bloque *Complex To Float*, de forma que mediante un sumador *Float* podamos sumar ambas.

5.3.2. Prueba del simulador de USB

Como en el anterior, pulsaremos en el botón de *play*, o F6, o menú *Run* y después *Execute*.

Para probar el modulador hemos realizado la misma modulación que con el de AM y DSB-SC, ya que queremos comparar las medidas con los anteriores, pues el propósito de trabajar con estos sistemas es el de observar si se cumple la teoría:

- Forma onda: coseno
- Amplitud de moduladora: 0.3
- Frecuencia de moduladora 192 Hz

- Amplitud de portadora: 1.5
- Frecuencia de portadora: 1600 Hz
- Ventana de FFT: Hamming

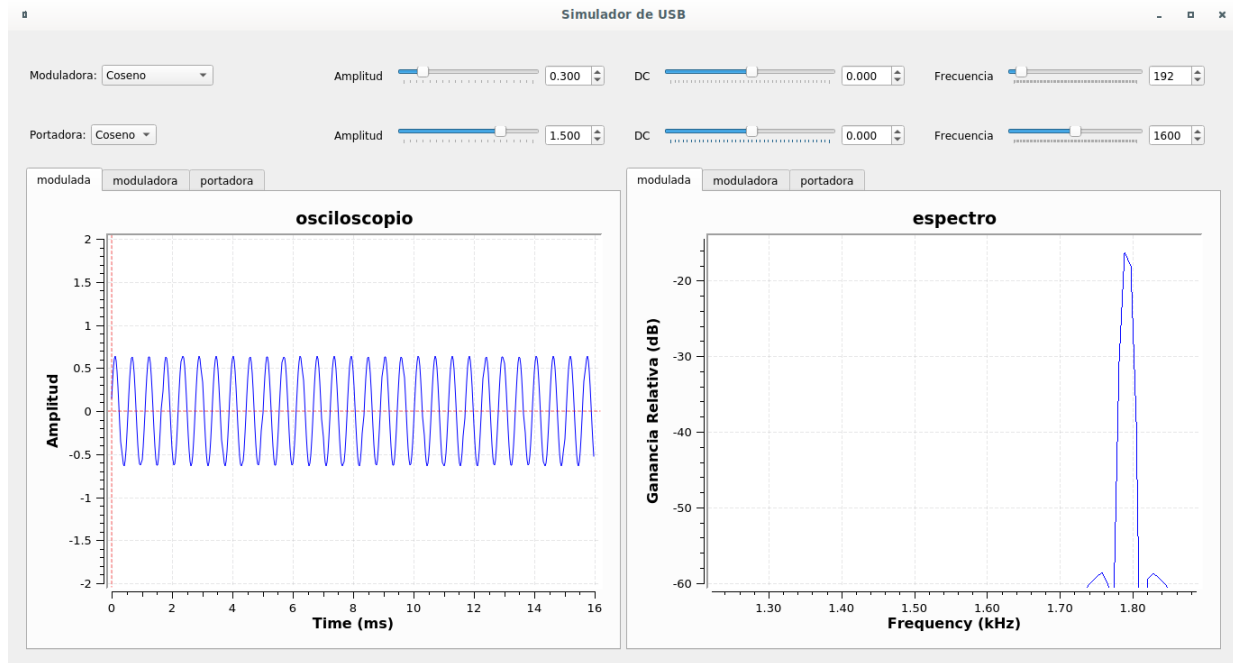


Figura 5.17: Primer ejemplo en el modulador USB.

El resultado de la modulación se encuentra en la figura 5.17, en este caso ya no se observa la señal portadora ni la banda inferior, sólo el espectro de la moduladora en la banda superior que está a 1.792 kHz. Las bandas laterales en DSB-SC estaban a -22.5 dB. En AM, teníamos -25 dB en las bandas laterales. Ahora tenemos -16 dB en la única banda lateral.

La conclusión es que se aprecia el efecto constructivo del método de discriminación empleado sobre la banda lateral superior.

El segundo ejemplo lo vamos a realizar con el siguiente modulador.

5.4. Banda lateral inferior (LSB)

El simulador de LSB se consigue mediante una modificación directa del USB.

Guardamos con el nuevo nombre y editamos título en el bloque de configuración, como hemos hecho anteriormente.

5.4.1. Transformación del modulador USB a LSB

Como explicábamos antes, el método que hemos usado supone la cancelación de una de las bandas mediante una suma destructiva. Por tanto para conseguir LSB a partir de USB, simplemente tenemos que invertir la fase de la señal moduladora en cuadratura, o lo que es lo mismo cambiarla de signo, de forma que la banda eliminada sea la superior. Para ello hemos introducido el bloque *Complex Conjugate*, que cambia de signo a la parte compleja, haciendo que la señal modulada en cuadratura cambie

de signo.

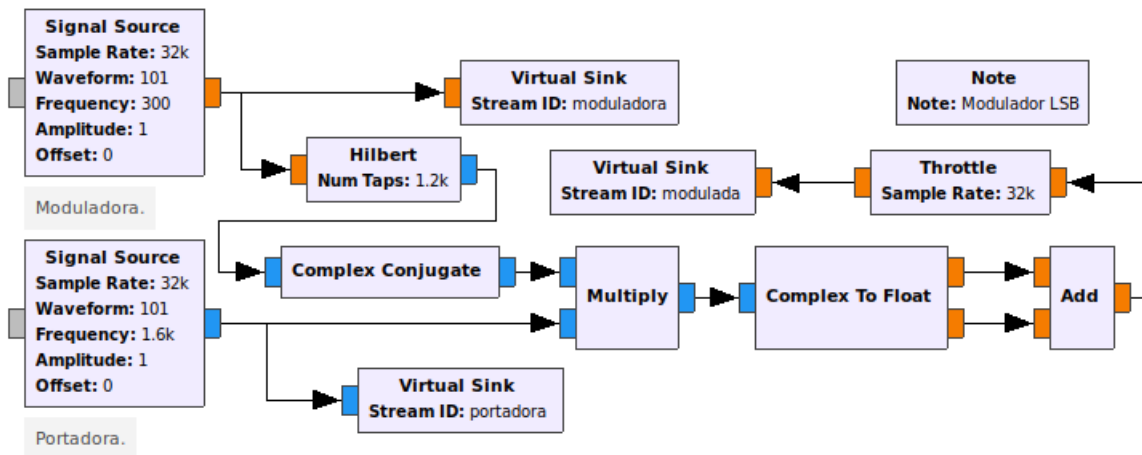


Figura 5.18: Esquema del modulador LSB.

En la figura 5.18 se observa la modificación realizada en el modulador USB, para conseguir el modulador LSB (apartado 5.3).

5.4.2. Prueba del simulador de LSB

Vamos a replicar ahora el segundo ejemplo que hemos realizado en AM:

- Forma onda: coseno
- Amplitud de moduladora: 1.12
- Frecuencia de moduladora 300 Hz
- Amplitud de portadora: 1
- Frecuencia de portadora: 2000 Hz
- Ventana de FFT: Hamming

Como la modulación es LSB observamos en el espectro de la figura 5.19, que en 2 kHz no tenemos nada, pero encontramos la componente espectral a 1.7 kHz, por tanto estamos detectando el tono de 300 Hz de la moduladora 300 Hz por debajo de la frecuencia portadora que era de 2000 Hz, luego efectivamente se trata de una modulación LSB. En este ejemplo, para las otras modulaciones teníamos las dos bandas laterales a -16.5 dB en DSB-SC, mientras que en el ejemplo de AM estaban a -17.5 dB. Ahora medimos -8.5 dB, de nuevo se puede considerar que existe una ganancia debido al método de cancelación.

5.5. Desplazamiento de amplitud (ASK)

Para implementar la modulación ASK podríamos haber seguido diversos caminos, pero tras reiteradas pruebas e intentos, hemos determinado una estrategia de diseño que vamos a detallar.

Primeramente iniciamos un proyecto nuevo, y rellenamos el bloque de configuración como en la figura 5.20.

El primer paso para realizar este simulador es generar una señal binaria. Un bloque para lograr esto de manera directa podría ser *Vector Source* (figura 5.21), este bloque permite insertar en el diagrama

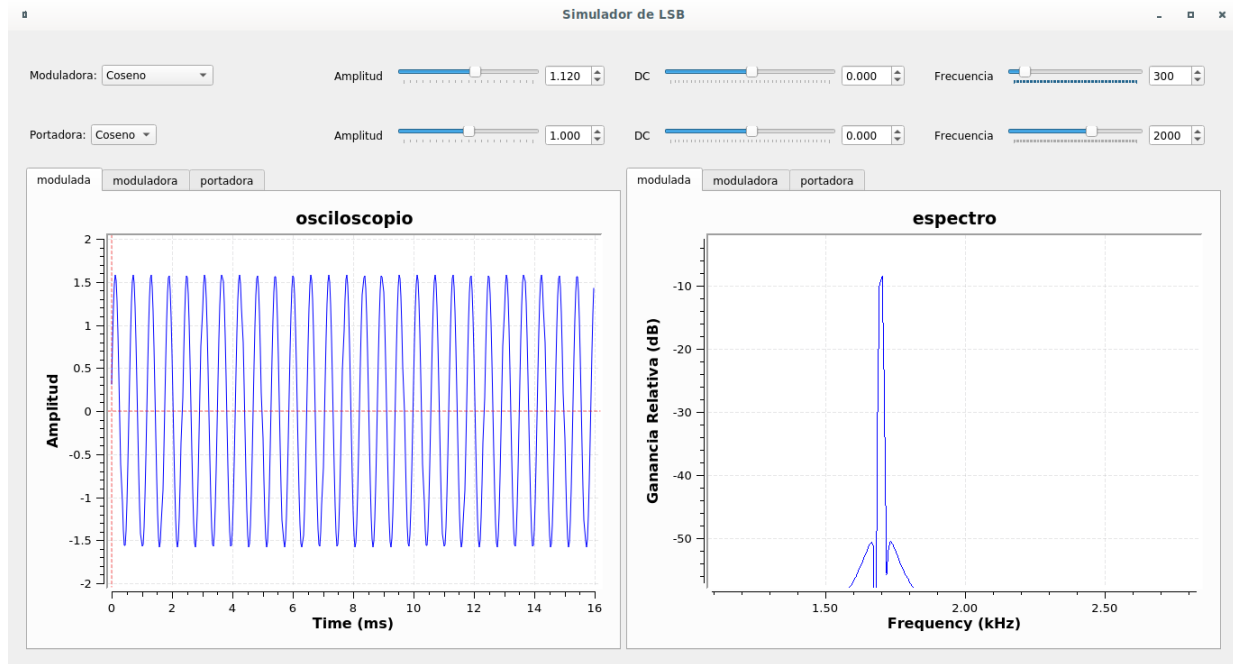


Figura 5.19: Segundo ejemplo en el modulador LSB.

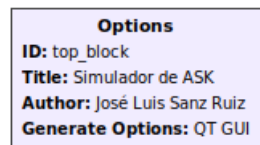


Figura 5.20: Configuración del proyecto simulador de ASK.

un vector periódico a modo de secuencia. Mediante variables podríamos insertar los valores binarios de la secuencia que queremos en cada momento y se repetiría infinitas veces. Por tanto es el bloque que recomendaría usar en la mayoría de casos, sin embargo no es el bloque que vamos a utilizar, ya que en nuestro caso preferimos utilizar como entrada un entero y convertirlo a binario.

5.5.1. Generación de secuencias binarias a partir de un entero

En la figura 5.22 se encuentra la parte del diagrama correspondiente al generador de secuencias binarias.

Hemos introducido un entero a través de la QT GUI Range, y usando una fuente *Signal Source*. En modo constante genera el entero de 16 bits deseado por el usuario, en tipo de datos Short (amarillo). Para la aplicación que le queremos dar, su buffer de salida requiere ser lo más pequeño posible. Si no hacemos el buffer lo más pequeño posible, cuando cambiemos el valor de entrada, el sistema habrá acumulado tantas muestras del valor anterior que no tendrá una respuesta en tiempo real. Previamente hemos intentado de alguna manera evitar esto, pero finalmente lo único que hemos podido hacer, es establecer el buffer de salida al mínimo valor permitido que es 2048 muestras. Para que el tiempo de respuesta sea fluido, y al mismo tiempo no haya sobrecarga en la CPU hemos situado el control de flujo *Throttle*, justo después a 2048 muestras por segundo (variable `read_rate`), esto nos garantiza que cada segundo que pasa se vacía el buffer, por tanto el tiempo de respuesta entre que el usuario cambia el valor de entrada y se visualiza en la gráfica final, será de un segundo aproximado. Para establecer el Max Output Buffer a 2048, vamos a la configuración del bloque *Signal Source* y a la pestaña *Advanced*.

Una vez tenemos el valor de 16 bits generado, este se entrega al bloque *Packet to Unpacked*, lo que

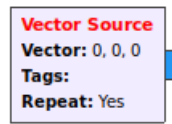


Figura 5.21: Bloque *Vector Source*.

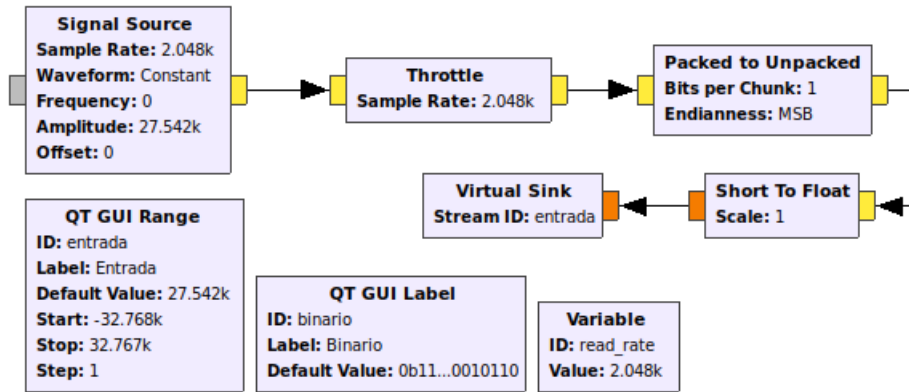


Figura 5.22: Generador de secuencias del modulador ASK.

hace *Packet to Unpacket* es trocear el dato de 16 bits (o del tamaño que fuera), en trozos de longitud de el número de bits que le pongamos en el campo Bits per Chunk, por lo que cada dato de 16 bits dará lugar a 16 muestras, de unos y ceros, aunque estas muestras serán cada una también de 16 bits ya que el tipo de datos sigue siendo Short. La velocidad de muestreo a la que trabaja GNU Radio se habrá multiplicado por 16, es decir, pasamos a 32768 muestras por segundo.

Detrás del bloque *Packet to Unpacked*, hemos colocado un bloque *Short To Float*, este bloque sirve para pasar las muestras tipo Short a Float, que es el tipo de datos con el que vamos a trabajar en la etapa de modulación. Hemos cerrado la etapa de generación de secuencias con una salida virtual *Virtual Sink*, que la hemos llamado entrada, ya que va a ser la entrada del modulador ASK.

5.5.2. Implementación de la modulación ASK

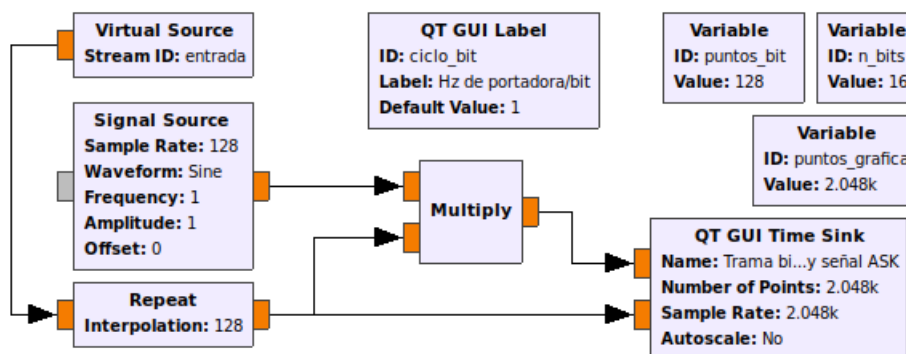


Figura 5.23: Diseño del modulador ASK.

La parte de modulación se corresponde con la parte superior del diagrama de la figura 5.23.

La entrada que hemos adaptado al modulador ASK hemos dicho que se trata de una secuencia de muestras en Float con unos y ceros, pues bien, la duración del bit es unitaria, es decir cada muestra que

entra por la *Virtual Source* entrada tiene la información de un bit. Nosotros vamos a modular multiplicando el valor de los bits, por una señal portadora, por tanto tenemos que expandir la señal de entrada, al tamaño necesario para que se pueda multiplicar muestra a muestra en el mezclador con la portadora.

Para adaptar el tiempo que dura cada bit al de un número determinado de ciclos de la señal portadora, vamos a usar el bloque *Repeat*, este bloque cuando recibe una muestra, realiza una interpolación copiando la muestra las veces que indiquemos en el campo *Interpolation*.

Los valores que hemos dado a estos campos han sido calculados de forma estratégica, si bien estos parámetros no se pueden alterar durante la ejecución, porque se pierde la sincronización entre la portadora y la secuencia, antes de iniciar el simulador ASK es posible determinar:

- El número de bits que queremos estudiar, mediante la variable `n_bits`, que la hemos puesto a 16.
- El número de puntos que queremos usar para representar cada bit, mediante la variable `puntos_bit`, puesto a 128, la tasa de muestreo aquí se multiplica por 128, quedando en 4194304.
- El número de Hz de portadora por bit, puesto a 1 `ciclo_bit`.

Para generar una portadora que esté sincronizada a un determinado número de ciclos por cada bit de entrada, usamos un bloque *Signal Source* en forma de onda senoidal, y vamos a establecer el Sample Rate del generador al valor de la variable `puntos_bit` (128), de esta manera le estamos indicando al generador que la tasa de muestreo para él es exactamente lo que dura un bit. Por lo tanto cuando indiquemos 1 Hz en el generador, estaremos generando una onda de exactamente un ciclo en lo que dura un bit, si ponemos 2 Hz, generaremos dos ciclos, y así sucesivamente, el número de Hz por bit lo hemos indicado en la etiqueta *QT GUI Label* con `ID ciclo_bit` y como hemos dicho antes, se puede configurar antes de iniciar el simulador. La amplitud del generador la hemos puesto en 1, para tener una salida normalizada entre 1 y -1.

El principio con el que generamos ASK es que cuando multiplicamos la señal cuadrada que vale uno o cero modificamos la amplitud de la portadora, en este caso entre 0 y 1.

Para visualizar la salida usamos *QT GUI Time Sink*, con el número de puntos al valor de la variable `puntos_grafica`. La variable `puntos_grafica` es la operación `n_bits × puntos_bit`.

5.5.3. Prueba del simulador ASK

Para probar el simulador, simplemente ejecutamos el diagrama, y si todo está correcto ya podemos probar a meter valores y ver su modulación, como en la figura 5.24, tal como se vio en el apartado de teoría 3.1.4.

5.6. Desplazamiento de fase (PSK)

El modulador PSK que vamos a hacer es de tipo BPSK, es decir PSK binario, variaremos la fase de la portadora entre dos fases que se encuentren equidistantes, es decir 0° y 180° . Este modulador resulta altamente sencillo de implementar, pues como la moduladora es un seno o coseno, se puede lograr un salto de fase cambiandola de signo.

5.6.1. Transformación del modulador ASK a PSK

Para lograr que nuestro anterior modulador ASK trabaje como un modulador PSK, sólo tenemos que modificar la entrada al mezclador, para que en lugar de multiplicar a la portadora por 1 y por 0 lo haga por 1 y -1. Una sencilla forma de lograr esto es desplazar la señal -0.5, para centrarla entre 0.5 y -0.5, después sólo tenemos que elevar la amplitud de la portadora a 2.

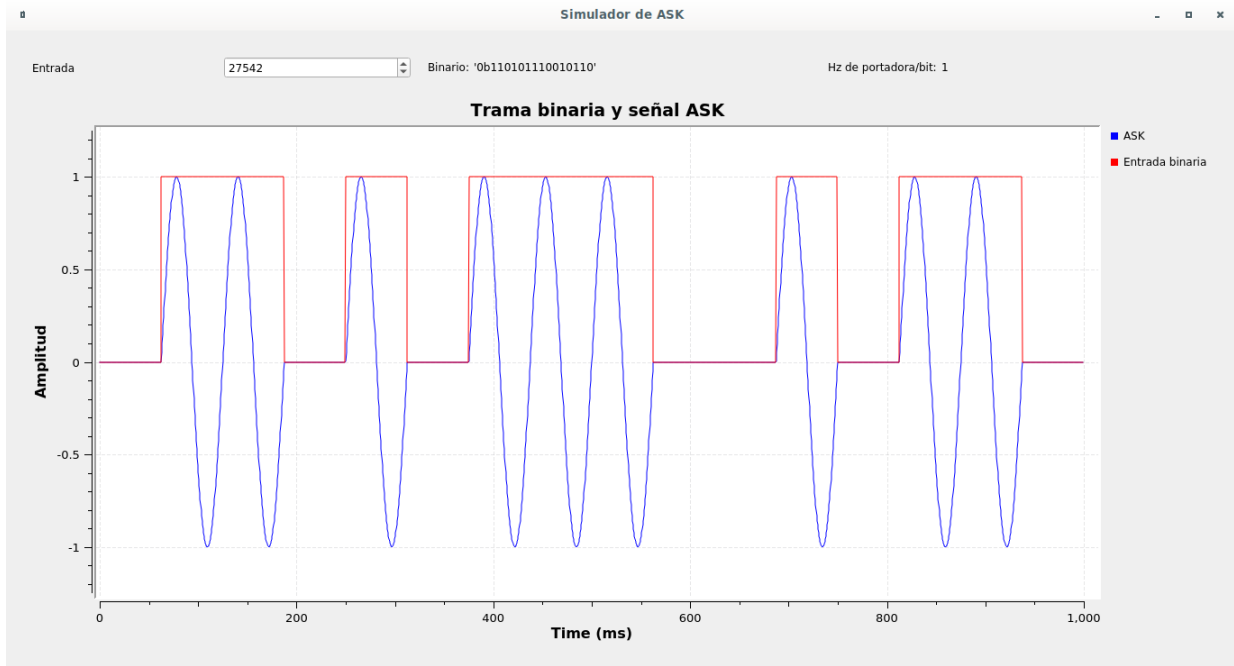


Figura 5.24: Ejemplo del modulador ASK.

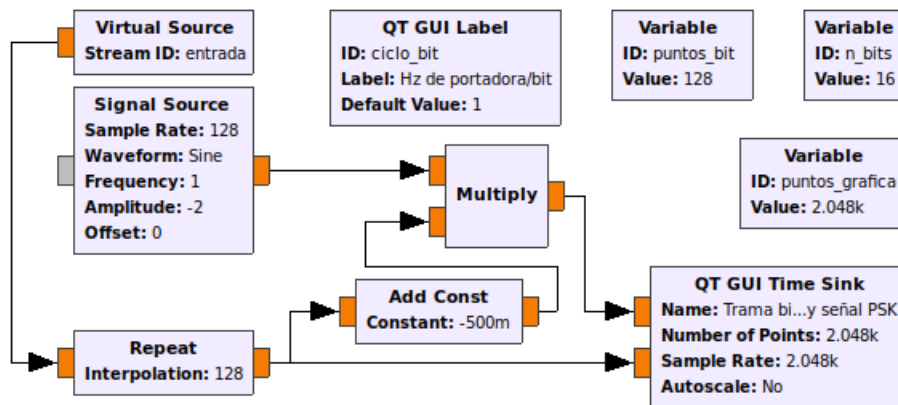


Figura 5.25: Modificación del modulador ASK para convertirlo en PSK.

De esta manera se realiza la inversión de fase de 180° que estamos buscando para modular en PSK binario.

5.6.2. Prueba del simulador PSK

El simulador PSK lo probamos con la misma secuencia y comprobamos como realiza la inversión de la fase, como muestra la figura 5.26. Este modulador lo hemos explicado teóricamente en el apartado 3.2.1.

5.7. Frecuencia modulada (FM)

5.7.1. Desarrollo del esquema del modulador

Para este simulador vamos a reutilizar la interfaz de los moduladores en amplitud, por ejemplo podemos usar el DSB-SC, ya que tiene todos los bloques configurados como tipo de datos Float.

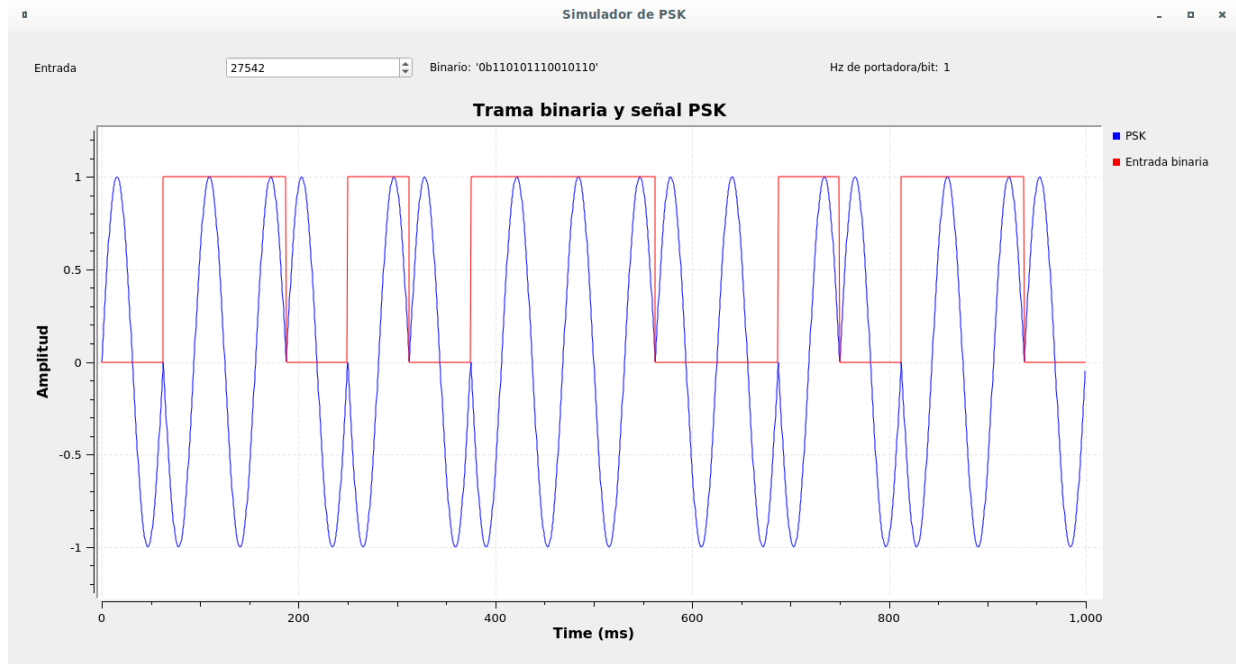


Figura 5.26: Ejemplo del modulador PSK.

El primer paso ya aprendido, es generar un proyecto a partir de la opción *save as* del menú *file* y escribir el título del proyecto, figura 5.27.

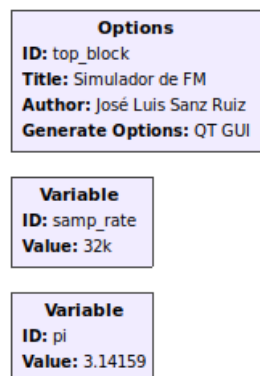


Figura 5.27: Bloques de configuración globales del modulador FM.

En este caso hemos declarado una variable llamada *pi*, para disponer del valor de π en precisión suficiente, 3.14159, sin tener que teclearlo a mano en las expresiones.

Para realizar el modulador de frecuencia, emplearemos un bloque *VCO*. Este bloque genera un tono con una frecuencia instantanea dada por el producto de la sensibilidad que tiene en su configuración en *radians/sec/volt*, y la muestra a la entrada. De forma que lo vamos a establecer a $2 \cdot \pi$, para que podamos introducir el valor en Hz por la línea de entrada del VCO.

Para establecer la frecuencia de reposo del modulador, vamos a añadir una constante justo a la entrada del VCO, que vamos a llamar F_c y que no es más que la frecuencia portadora, ya que en ausencia de moduladora, el VCO quedará oscilando a esa frecuencia. Además, como el bloque *VCO* no permite variar la sensibilidad en tiempo de ejecución, vamos a multiplicar la moduladora por una constante f_d , antes de sumarle la constante de reposo del VCO. El motivo de insertar la constante después de la sensibilidad, es para que la sensibilidad de la modulación no altere el punto de reposo al multiplicarse.

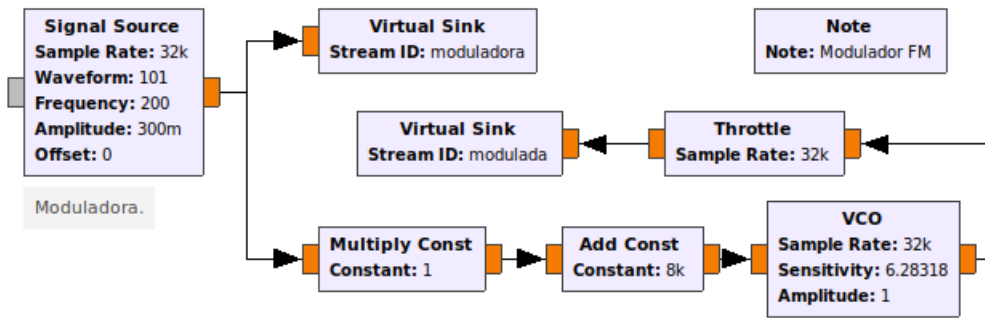


Figura 5.28: Esquema del modulador FM.

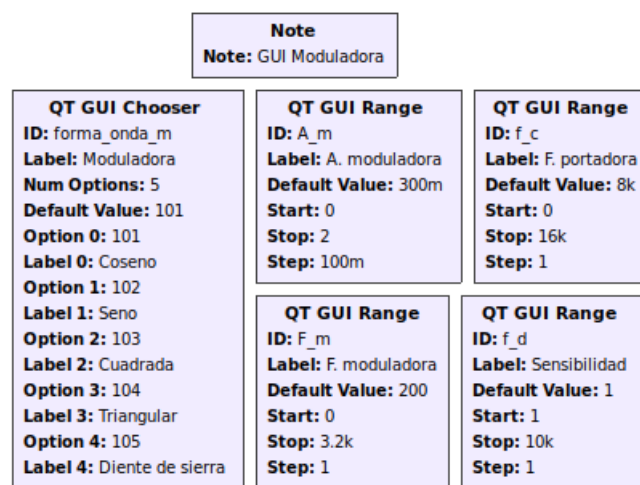


Figura 5.29: Bloques de control del modulador FM.

El resto de controles los hemos heredado de los moduladores de amplitud, además hemos suprimido el visor de la portadora, ya que al usar el VCO no existe realmente una portadora original sin modular.

5.7.2. Prueba del simulador FM

Para probar el simulador de FM vamos a usar los siguientes ejemplos: En primer lugar vamos a modificar la frecuencia de muestreo a 320k para obtener una representación más suave y el tamaño de la FFT del espectro a 4096.

Para comprobar la teoría vista en el apartado 3.3.1, realizaremos diferentes pruebas y comprobaremos con diferentes índices de modulación:

$$D = \frac{f_d \cdot A_m}{f_m}$$

D = 1

- Amplitud de moduladora: 0.5
- Frecuencia de moduladora: 1000 Hz
- Frecuencia de portadora: 10000 Hz
- Sensibilidad: 2000

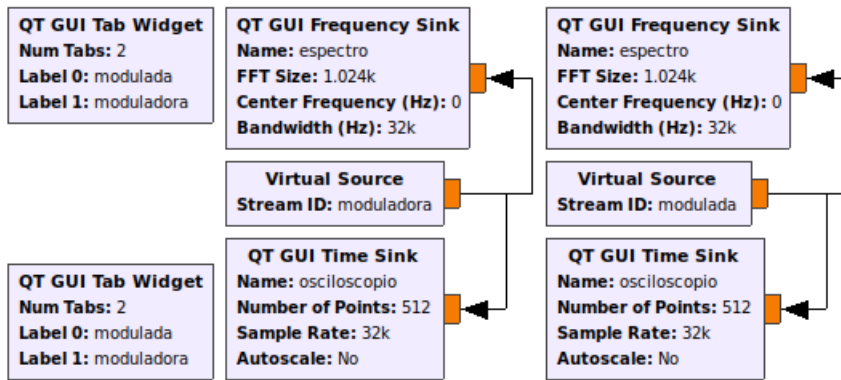


Figura 5.30: Bloques de visualización del modulador FM.

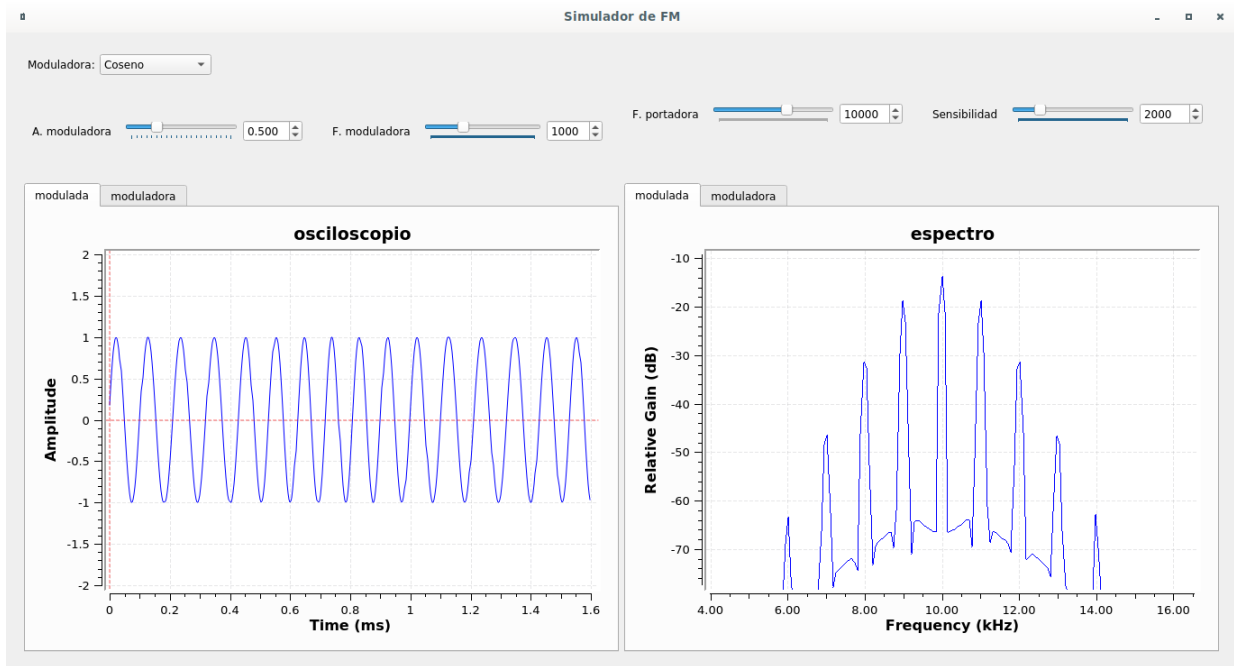


Figura 5.31: Ejemplo en el modulador FM con $D=1$.

Para este teóricamente:

$$B_T = 2(D + 1)f_m = 4kHz$$

$$n_{m\acute{a}x} = D + 1 = 2$$

Y hemos medido en la gráfica:

	Frecuencia	Ganancia
n_c	10 kHz	-13 dB
n_1	11 kHz	-19 dB
n_2	12 kHz	-31 dB

Lo que hemos visualizado coincide con los cálculos, ya que tenemos la potencia en la banda central y en la primera y segunda. Luego la potencia ya cae mucho, además la segunda banda coincide con el ancho de banda obtenido de 4 kHz, ya que va desde 8 a 12 kHz.

$D = 2$:

- Amplitud de moduladora: 1

- Frecuencia de moduladora: 1000 Hz
- Frecuencia de portadora: 10000 Hz
- Sensibilidad: 2000

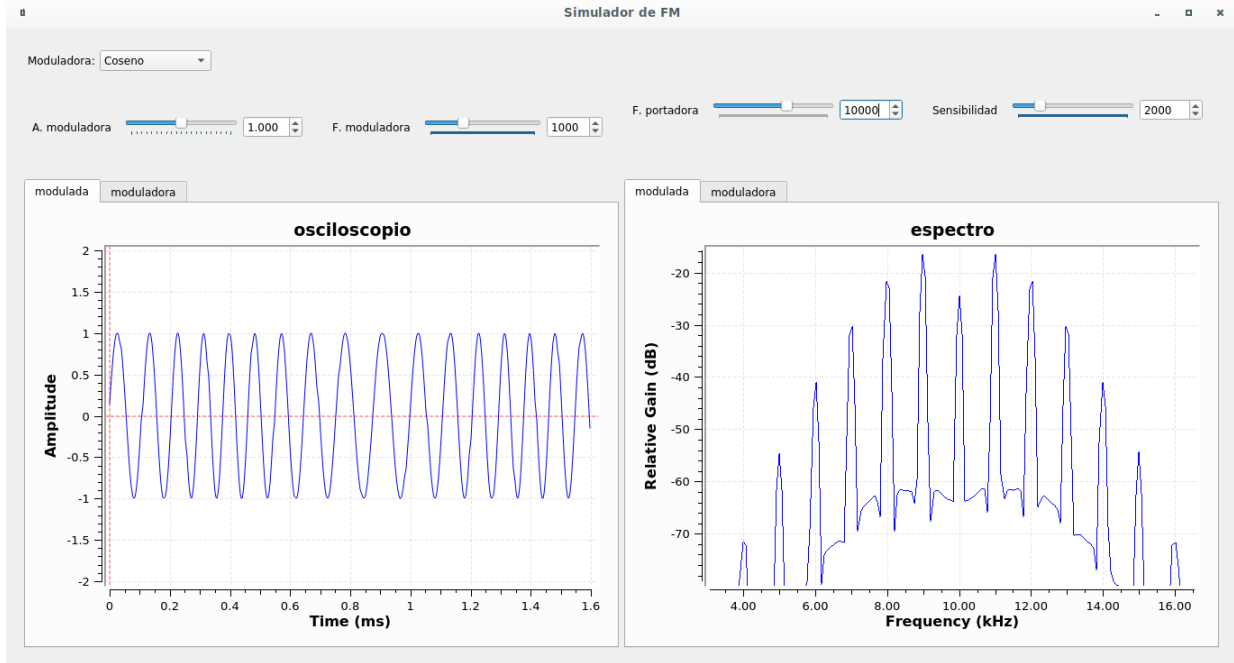


Figura 5.32: Ejemplo en el modulador FM con $D=2$.

Para este teóricamente:

$$B_T = 2(D + 1)f_m = 6\text{kHz}$$

$$n_{m\acute{a}x} = D + 1 = 3$$

Y hemos medido en la gráfica:

	Frecuencia	Ganancia
n_c	10 kHz	-24 dB
n_1	11 kHz	-17 dB
n_2	12 kHz	-21 dB
n_3	13 kHz	-30 dB

Lo que hemos visualizado coincide con los cálculos, ya que tenemos la potencia en la banda central y en la primera, segunda y tercera. Luego la potencia ya cae mucho, además la segunda banda coincide con el ancho de banda obtenido de 6 kHz, ya que va desde 7 a 13 kHz.

$D = 5$:

- Amplitud de moduladora: 1.25
- Frecuencia de moduladora: 500 Hz
- Frecuencia de portadora: 10000 Hz
- Sensibilidad: 2000

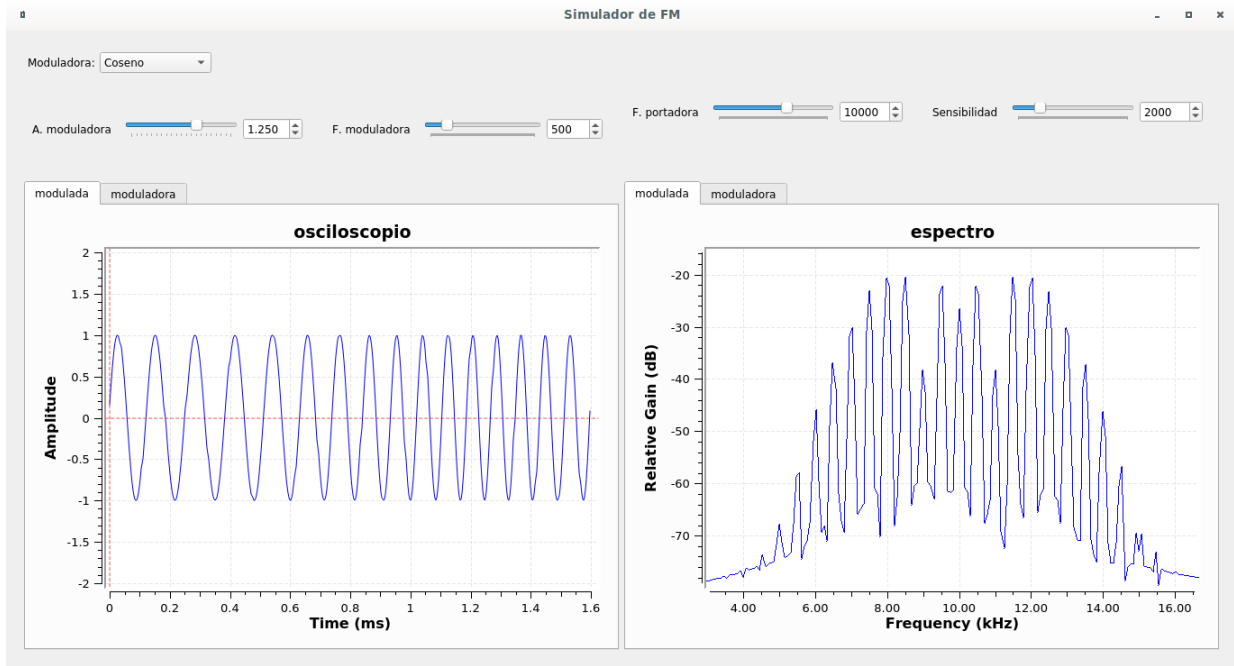


Figura 5.33: Ejemplo en el modulador FM con $D=5$.

Para este teóricamente:

$$B_T = 2(D + 1)f_m = 6\text{kHz}$$

$$n_{m\acute{a}x} = D + 1 = 6$$

Y hemos medido en la gráfica:

	Frecuencia	Ganancia
n_c	10 kHz	-27 dB
n_1	10.5 kHz	-22 dB
n_2	11 kHz	-39 dB
n_3	11.5 kHz	-21 dB
n_4	12 kHz	-21 dB
n_5	12.5 kHz	-23 dB
n_6	13 kHz	-30 dB

Lo que hemos visualizado coincide con los cálculos, ya que tenemos la potencia en la banda central y en las seis siguientes bandas. Luego la potencia ya cae mucho, además la sexta coincide con el ancho de banda obtenido de 6 kHz, ya que va desde 7 a 13 kHz.

Capítulo 6

Sistemas de RF con HackRF One

6.1. Introducción

Para demostrar la capacidad que tiene GNU Radio a la hora de implementar sistemas de RF, hemos decidido diseñar un modulador de radio AM comercial en onda media, y otro en FM comercial, con capacidad estéreo. En este capítulo veremos su desarrollo, y realizaremos algunas pruebas prácticas que nos permitan mostrar su funcionamiento.

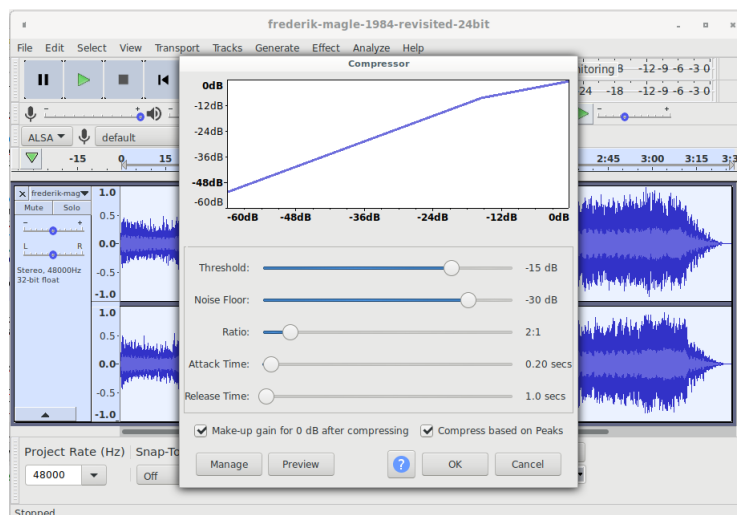


Figura 6.1: Aplicación de un compresor a la pista de sonido que usaremos para modular. Software Audacity.

6.1.1. Obtención y adecuación del fichero de sonido para modular

El primer paso es obtener un fichero de sonido fuente, hemos elegido una pista titulada *1984 Revisted* de *Frederik Magle*. El interés principal en dicha pista de sonido es que está muestreada a 48 KHz, sin compresión, y el autor permite su uso libre no comercial, con atribución. Se puede obtener en:

<https://www.magle.dk/free-soundtracks.html#electronic>

En Internet hay numerosos sitios web donde encontrar este tipo de obras, si se desea se puede usar cualquier sonido propio que esté en formato *WAV (Microsoft) signed 16-bit PCM*.

Una vez descargada la pista la abriremos en el software Audacity. Este software gratuito de edición de audio nos permite realizar un sin fin de tareas relacionadas con el procesamiento digital de sonido.



Figura 6.2: Arriba: Philips MC148/22, abajo a la izquierda: Sony ICD-PX333, abajo a la derecha: Sony ICF-S10MK2.

Audacity nos va a permitir asegurarnos que el fichero está muestreado a la frecuencia de muestreo que necesitamos, en caso de necesitar cambiar la frecuencia de muestreo, podemos usar la opción de remuestrear y elegir otra frecuencia diferente.

Nosotros hemos aplicado un compresor a la pista, ya que en radiodifusión es habitual buscar un elevado nivel sonoro relativo, evitando el silencio que suele dar paso al ruido de fondo.

Exportamos como WAV, usando la codificación *WAV (Microsoft) signed 16-bit PCM*, esto último es imprescindible ya que la versión de GNU Radio que estamos usando espera este formato. Le hemos llamado *input.wav*.

6.1.2. Receptores de radio comercial para pruebas

Para probar el modulador comercial de AM vamos a utilizar un receptor de radio portátil tipo transistor, con dial analógico, marca Sony, modelo ICF-S10MK2.

Para probar el modulador comercial de FM vamos a utilizar un sistema de sonido doméstico marca Philips, modelo MC148/22.

Para grabar el resultado de la modulación conectaremos la salida de auriculares de cada receptor a la entrada de línea de una grabadora de sonido digital marca Sony, modelo ICD-PX333.

Los tres dispositivos aparecen fotografiados en la figura 6.2.

Mostraremos en el software Audacity la forma de onda obtenida en cada caso comparada con el fichero original y el espectro de dicha señal sonora.

6.2. Radiodifusión AM comercial en onda media

Para realizar el modulador de radiodifusión en AM, vamos a comenzar un proyecto nuevo, y lo vamos a titular Modulador de AM comercial, como en la figura 6.3.

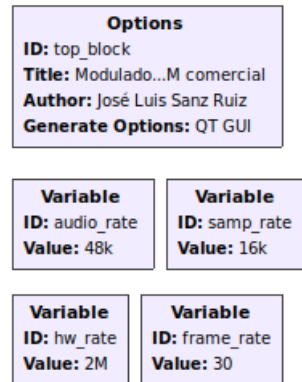


Figura 6.3: Configuración y variables.

Además vamos a crear cuatro variables para tener el control de la tasa de muestreo de cada una de las partes:

- audio_rate: 48000, que será la tasa de muestreo del fichero de audio que vamos a emplear en la transmisión.
- samp_rate: 16000, que será la tasa de muestreo del modulador AM.
- hw_rate: 2000000, que será la tasa de muestreo del DAC del HackRF One.
- frame_rate: 30, que será la tasa de refresco de los gráficos (opcional, pero mejora la visualización si se ajusta bien).

6.2.1. Adecuación de la señal de audio al modulador

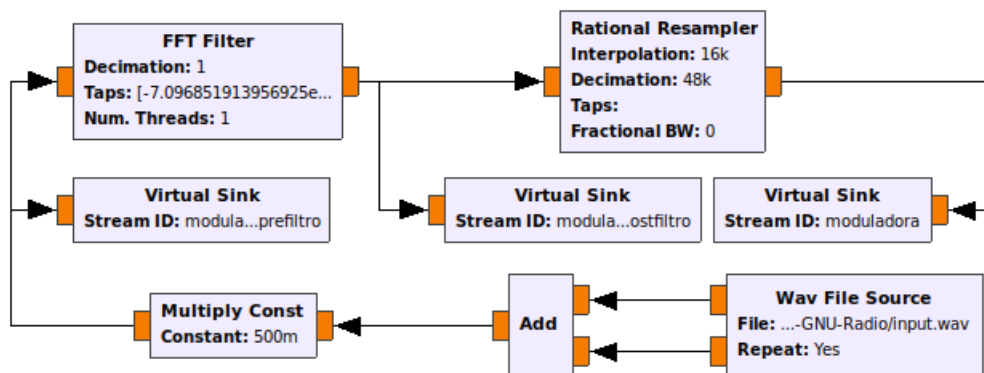


Figura 6.4: Etapa de entrada del modulador AM comercial.

En este paso, usando el diagrama de bloques de la figura 6.4, vamos a insertar la señal de audio. Siguiendo este, observamos como mediante un bloque *Wav File Source* se lee el fichero WAV que hemos preparado y que nos dará muestras de entre -1 y 1 en cada canal. Se realiza la suma de canales, para obtener una conversión de señal estereofónica a monofónica y multiplicamos por un control de ganancia, que a su vez se multiplica por 0.5 para compensar la amplitud resultante de la suma de los

dos canales. Con el control de ganancia vamos a evitar siempre que la amplitud exceda de entre -1 y 1, evitando la sobremodulación. Esta ganancia es equivalente al valor k , la sensibilidad del modulador AM vista en la teoría.

Obtenemos una copia de la señal prefiltro para inyectarla luego en un módulo de visualización, y así tener conocimiento de como es la señal de entrada original. Al mismo tiempo vamos a pasar la señal por un filtro FFT, que vamos a diseñar como pasa-bajo, mediante la herramienta de diseño que incorpora GRC, y que se abre en el menú *Tools*. Después de filtrar la señal, volvemos a generar una copia postfiltro y la original la sometemos mediante un cambiador de frecuencia de muestreo a un diezmado de 48k a 16k. El bloque *Rational Resampler* se encarga de generar la cadena necesaria para realizar el cambio de frecuencia de muestreo de forma automática, sólo hay que especificar los valores de interpolación y diezmado en su configuración.

Diseño del filtro

Vamos a diseñar el filtro paso-bajo a 4.5 kHz, para ello usaremos la siguiente configuración:

- FIR
- Low Pass
- Hamming Window
- Sample Rate (sps): 48000
- Filter Gain: 1
- End of Pass Band (Hz): 4500
- Start of Stop Band (Hz): 5000
- Stop Band Attenuation (dB): 30

Abrimos la herramienta de diseño de filtros y una vez tenemos los datos introducidos en el programa, pulsamos Design. Una vez calculado el filtro podemos visualizar su diagrama de polos y ceros, así como su respuesta en frecuencia, fase, respuesta al impulso etc. como muestra la figura 6.5.

En la pestaña *Filter Coeficients* tenemos una lista para copiar los coeficientes y pegarlos en la configuración *taps* del bloque *FFT Filter*. Si el filtro tiene un número excesivo de coeficientes es posible que hayamos elegido bandas de transición estrechas y abruptas, en cuyo caso tendremos problemas de rendimiento si intentamos utilizarlo.

6.2.2. Etapa de modulación

Como hemos visto, el HackRF One, es un dispositivo que realiza una conversión de frecuencia, por tanto nosotros construimos señales en banda base y él las desplaza en frecuencia a la frecuencia RF deseada. Lo cual hace que el proceso de modulación quede totalmente simplificado para AM como se ve en la figura 6.6. Para obtener la portadora una vez actúe el conversor de frecuencia, lo único que tenemos que hacer es añadirla como componente continua en banda base, por tanto, lo único que vamos a hacer previamente es poner un sumador, y sumar un valor constante a la moduladora. Mediante el bloque *QT GUI Range* hemos permitido ajustar ese valor de componente continua, para facilitar la experimentación.

Al realizar la modulación con una moduladora contenida entre 1 y -1, la amplitud de la señal después de sumar la componente continua, va a variar entre 0 y 2. Si entregamos al HackRF One una señal que no esté normalizada a 1 se produce saturación. Por lo que es necesario corregir esto, y que la señal no se salga del rango de amplitud permitido. Para ello hemos añadido el multiplicador por la

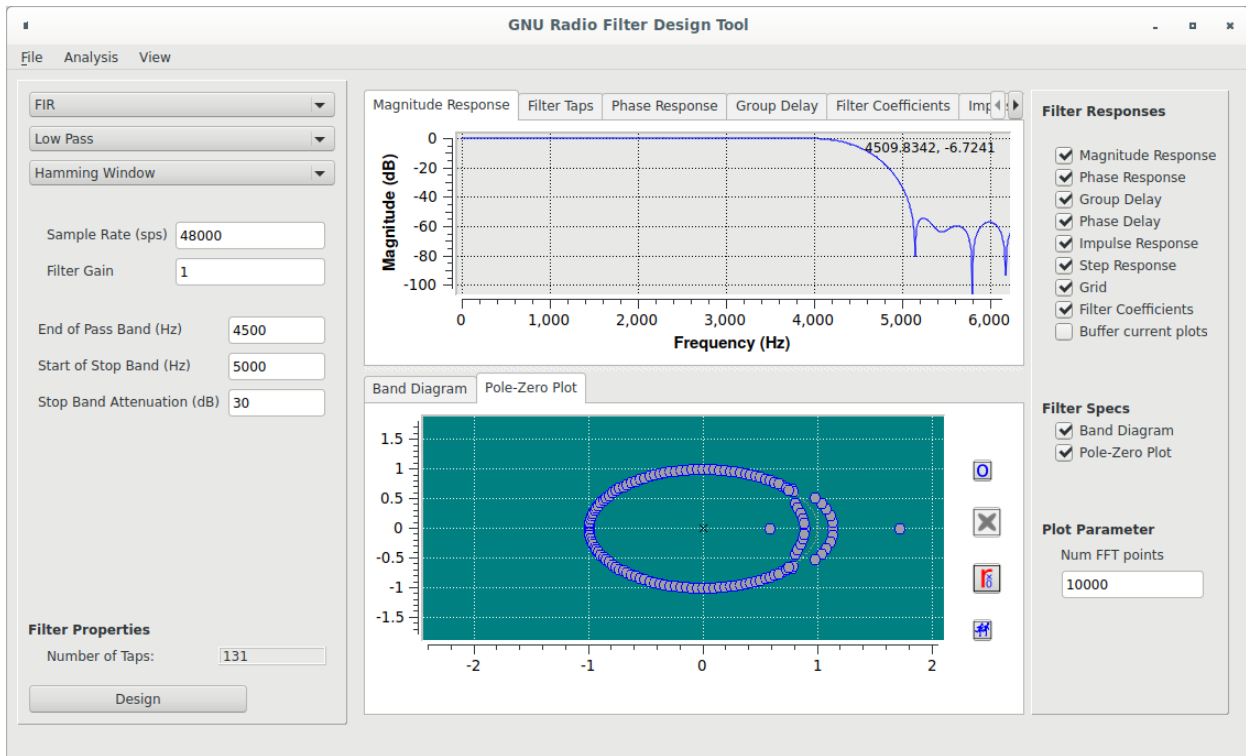


Figura 6.5: Utilidad de GRC para el diseño de filtros de muchos tipos.

constante 0.5.

Una vez que tenemos la señal que vamos a enviar al DAC, tenemos que adecuarla a su frecuencia de muestreo, el HackRF funciona solamente en un rango de frecuencias de muestreo que va desde las 2 Msps hasta las 20 Msps y sólo configurable a ciertos múltiplos de frecuencia, este valor ya lo hemos definido en la variable `hw_rate`. Y vamos a enviar dos millones de muestras cada segundo al DAC del HackRF One.

Como nuestro modulador lo hemos hecho trabajar a 16 ksps, no podemos enviar la señal directamente al hardware, por tanto hemos pasado la señal modulada por un *Rational Resampler* que realiza la interpolación necesaria hasta 2 Msps (factor $L = 125$), para poder satisfacer las necesidades del DAC. Si no generamos muestras suficientes en tiempo de ejecución podríamos generar una situación llamada *underrun* [10], que se produce cuando un bloque se queda sin datos en su buffer de entrada y a su vez otra parte del sistema le está exigiendo una salida inmediata, como podría ser el reloj del DAC del HackRF One exigiendo muestras.

Para no usar el modo en fase y cuadratura, cuando empleamos el HackRF One, de momento la única solución que existe es la de tomar un bloque que convierte Float a Complex, y utilizar solamente el conversor en fase, a través de la parte real, dejando el conversor en cuadratura sin entrada, o sea la parte imaginaria del bloque.

Para que el HackRF One funcione vamos a configurar el bloque *osmocom Sink*, tenemos que especificar la tasa de muestreo, en nuestro caso va con la variable `hw_rate` como hemos mencionado más arriba que supone lo mismo que escribir 2000000 o $2e6$. Después tenemos que establecer la frecuencia en hercios a la que queremos obtener la señal, en el campo Frequency (Hz), para ello hemos agregado un *QT GUI Range* con *ID* sintonía, que nos permite establecer una frecuencia de transmisión dentro de los límites de la onda media, y a su vez, dentro de los límites del oscilador local del HackRF One:

- Onda media: 530 kHz a 1710 kHz.

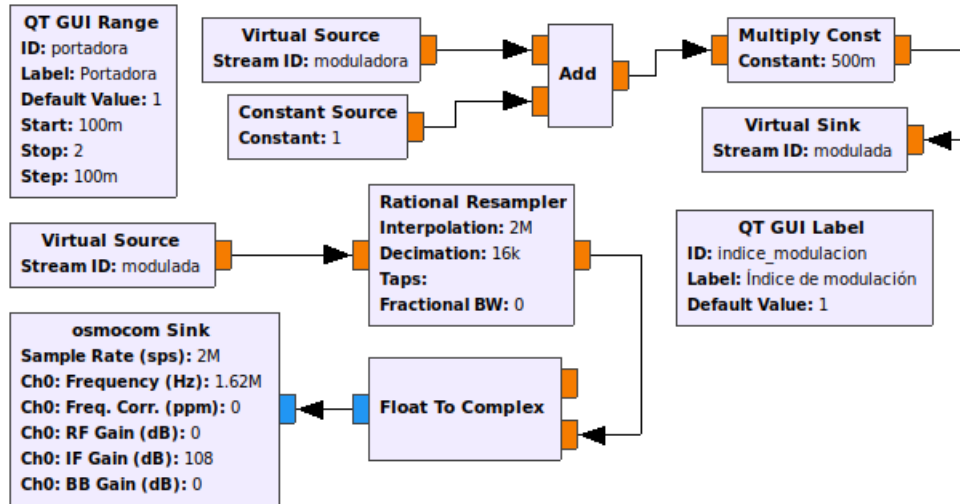


Figura 6.6: Etapa de modulación y transmisión de la señal.

- Oscilador local del HackRF One: 1 MHz a 6 GHz.
- Rango posible: 1000 kHz a 1710 kHz.
- Rango excluido: 530 kHz a 1000 kHz.

Por lo que sólo vamos a poder trabajar media banda de radio AM en onda media. Otra solución es utilizar onda corta, que está totalmente contemplada dentro del rango de HackRF One, pero la hemos descartado porque no es común disponer de receptores de onda corta, mientras que sí lo es de onda media. También podemos desplazar la señal por software mediante LSB en banda base.

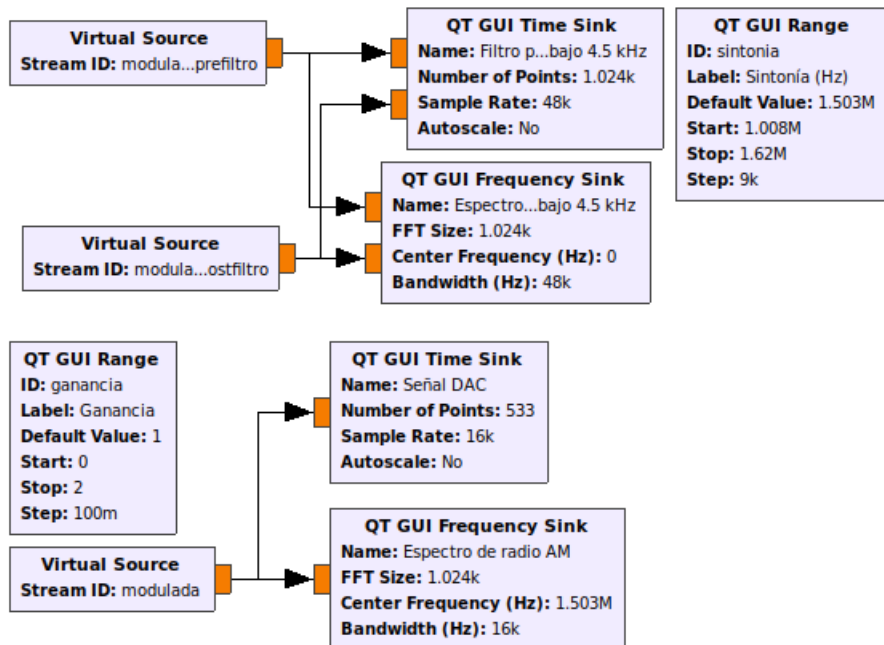


Figura 6.7: Visualización.

En la figura 6.7, se detallan los bloques que hemos usado para la visualización, se encuentra el bloque de sintonía que hemos mencionado antes, y el de la ganancia de entrada para ajustar el índice de modulación final. Para mostrar el cálculo del índice de modulación hemos empleado un *QT GUI*

Label, mostrando la división de la ganancia de la moduladora dividido entre la constante que hemos usado como portadora en banda base, este bloque se encuentra en la parte del modulador, es decir, en la figura 6.6.

Hemos configurado los visores a sus correspondientes frecuencias de muestreo, tamaños de FFT para los de espectro y puntos para los de tiempo. Para controlar el refresco de las gráficas hemos escrito en el periodo de actualización $1/\text{frame_rate}$, de forma que se actualizan 30 veces cada segundo.



Figura 6.8: Situación de los dispositivos durante el experimento de transmisión y recepción de radio en AM.

6.2.3. Prueba del modulador AM comercial

Para probar el modulador hemos montado un experimento que consiste en situar el HackRF One con la antena sobre la mesa, y aproximar el receptor de AM alineando de forma paralela a la antena la barra de ferrita que utiliza este para la recepción.

No hemos podido separar mucho el receptor de la antena transmisora, porque había demasiado ruido de interferencias, que enmascaraban la transmisión. También hemos procurado alimentar el equipo informático con la batería integrada, para no introducir ruidos de la fuente de alimentación. Hemos alejado el HackRF One lo máximo posible del ordenador, y también hemos evitando acercarlo demasiado al HackRF One por la parte de la carcasa, ya que se produce interferencia de los osciladores y el fenómeno es claramente audible en el receptor.

Hemos tomado la grabadora y la hemos configurado para grabar desde la entrada de línea, la calidad que hemos elegido es mp3 a 192 kbps. También nos hemos asegurado que no esté activado ningún perfil de reducción de ruido. Después hemos conectado un cable de tres hilos y dos conectores tipo

plug, uno al jack de la grabadora y el otro al de la salida de auriculares del receptor AM. Finalmente hemos conectado unos auriculares a la grabadora para monitorizar el sonido. Todo lo que se describe aquí se puede ver en la figura 6.8.

Para terminar hemos conectado el HackRF One al equipo mediante el cable USB, hemos ejecutado el modulador y ha aparecido en pantalla la GUI que hemos creado para visualizar y ajustar los parámetros. La figura 6.9 es una captura de pantalla de la GUI.

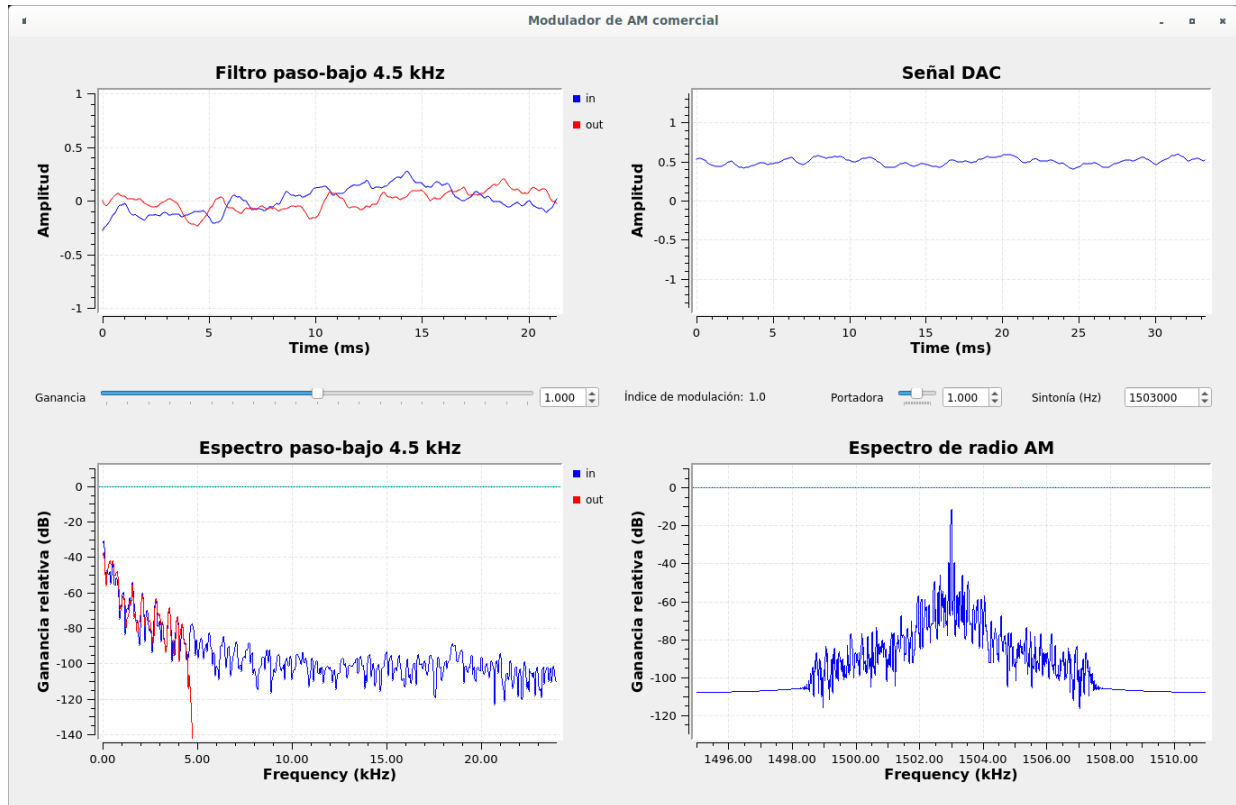


Figura 6.9: Prueba del modulador AM transmitiendo en el 1620 kHz de la onda media.

A la izquierda de la pantalla hemos colocado un visor de la señal que entra y sale del filtro, superpuestas, tanto en tiempo como en frecuencia. De forma que podemos ver la diferencia visual entre ambas señales, pues efectivamente se observa como el filtro recorta la señal a 4.5 kHz. Con esto estamos demostrando que la utilidad de GRC para diseño de filtros es buena.

A la derecha hemos colocado un visor de la señal que se envía al DAC y en él se puede observar tanto el nivel de continua como la envolvente que se forma. En la parte inferior se muestra el espectro que se envía al DAC, aunque hemos configurado el bloque para que muestre como frecuencia central el valor de la variable sintonía, realmente lo que se está mostrando es el gráfico de espectro de la señal en banda base, con la frecuencia del oscilador local del HackRF One etiquetada como frecuencia central, en lugar de 0 Hz.

Para evaluar la calidad de la transmisión a primera vista, mostraremos en Audacity la forma de onda de la señal original en suma monofónica, comparado con lo que hemos registrado en la grabadora digital, figura 6.10. También hemos representado el espectro en la figura 6.11, con la herramienta de análisis de espectro de Audacity. Se puede apreciar el recorte.

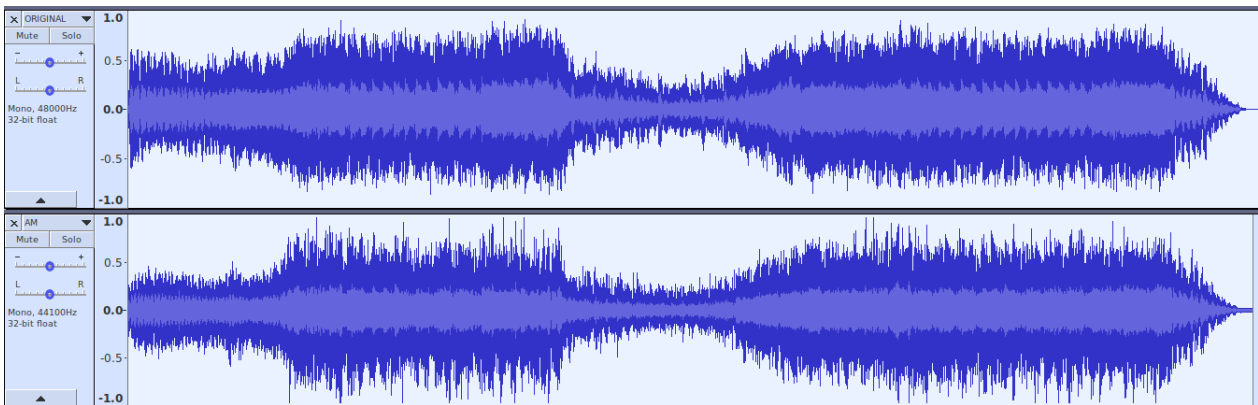


Figura 6.10: Comparación de la forma de onda original y la registrada con la grabadora digital tras la modulación AM.

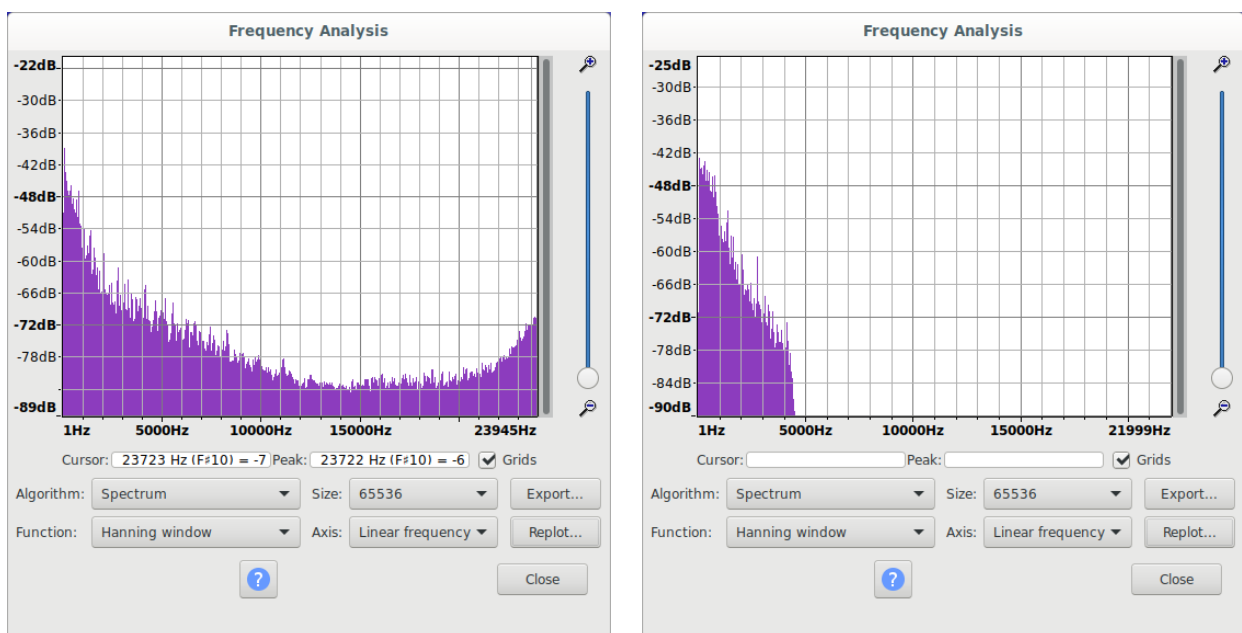


Figura 6.11: Espectro de la pista original y espectro de la pista en el receptor de AM.

6.3. Radiodifusión FM estéreo comercial

Este modulador es el más complicado de realizar, porque requiere del uso de muchos subsistemas, y hay que conseguir que todos estén correctamente implementados y ajustados para que el resultado sea satisfactorio.

Se trata de un modulador capaz de realizar modulaciones con sonido estereofónico, empleando el formato estándar mundial que se utiliza para la radiodifusión en FM. A modo de resumen, el modulador toma una entrada de sonido estereofónica, la filtra pasabaja, le aplica preénfasis, y obtiene la señal suma y diferencia de los canales. Después produce la señal múltiplex de FM con la señal suma en banda base, el piloto de 19 kHz y la señal diferencia modulada DSB-SC a 38 kHz. El múltiplex se utiliza como moduladora en un modulador FM basado en un VCO, y la señal se transmite al aire usando el HackRF One. Veamos detenidamente como está realizado.

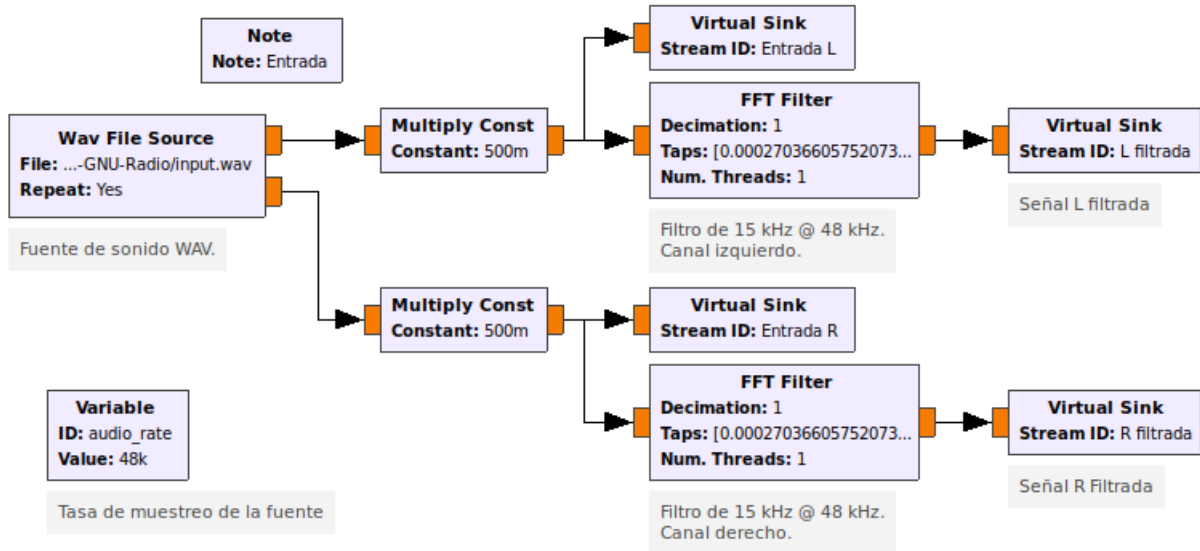


Figura 6.12: Ajuste de ganancia y filtro entrada del modulador FM estéreo.

6.3.1. Tratamiento previo de la señal estereofónica

En la figura 6.12 observamos como hemos declarado una primera variable `audio_rate`, que representa la frecuencia de muestreo del fichero de sonido que vamos a transmitir. El fichero que vamos a transmitir es estéreo, por lo que tiene dos canales, cada canal lo multiplicaremos por una constante, que está controlada por un *QT GUI Range*, permitiendo ajustar la ganancia de entrada general. Después de ajustar la ganancia, pasamos la señal por un filtro paso-bajo y la recortamos a 15 kHz, usando el mismo método que para el modulador AM comercial. Solamente tenemos que establecer la frecuencia de corte en 15 kHz y se procede de forma análoga. De esta etapa sacamos tanto la señal de entrada en bruto como la señal filtrada, para más tarde poder ver el efecto del filtro en la representación gráfica.

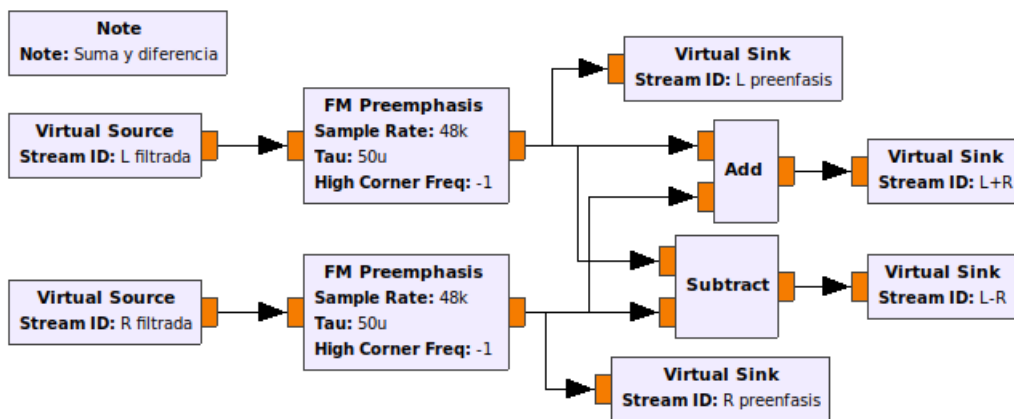


Figura 6.13: Preénfasis, señal suma y diferencia.

En la figura 6.13 se realiza el preénfasis, que consiste en realzar las frecuencias más altas de la señal, para preservarlas del ruido, de modo que en el receptor se realiza la operación inversa. El valor que hemos ajustado es tau 50 microsegundos, que es el valor que se usa en Europa. Si fuéramos a transmitir en América usaríamos 75 microsegundos.

Después del preénfasis, con un bloque de suma y otro de resta se generan la señal suma, que es la señal monofónica, y la señal diferencia, que resulta ser la diferencia entre los dos canales. Esto se realiza

así, porque el receptor mediante suma y resta podrá regenerar ambas señales y ofrecer el sonido estéreo.

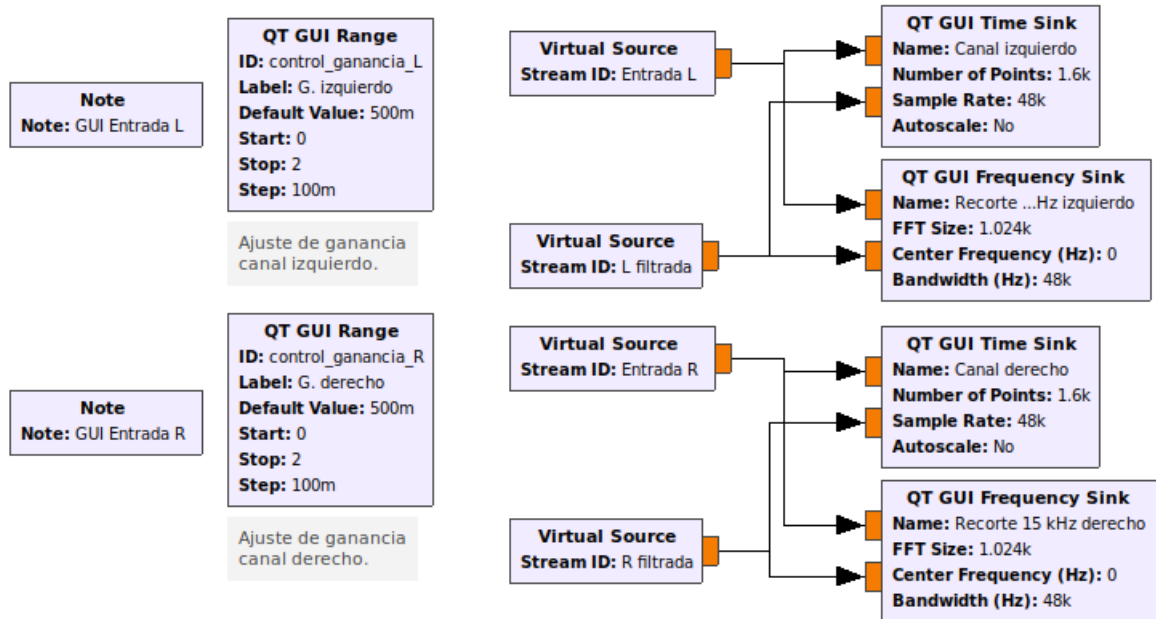


Figura 6.14: Bloques de visualización y ajustes de la etapa de entrada al modulador.

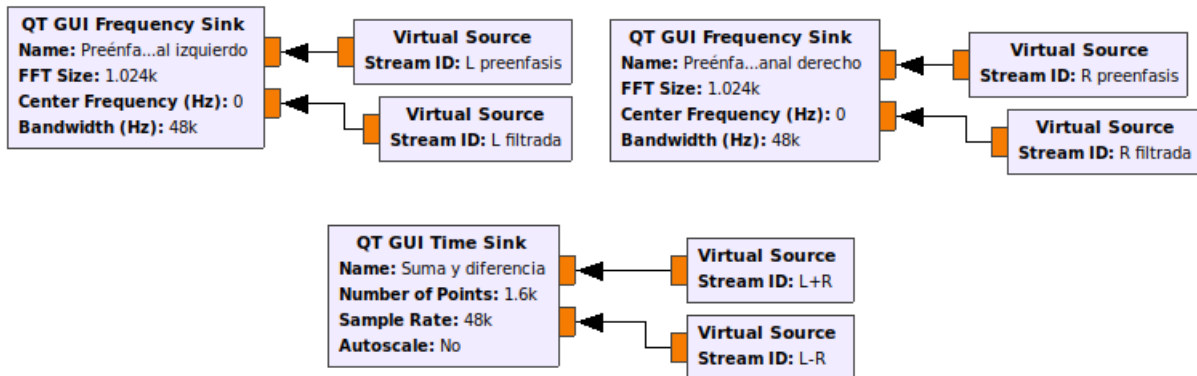


Figura 6.15: Bloques de visualización del preénfasis y señales suma y diferencia.

En la figura 6.14 y 6.15 hemos realizado los controles y la visualización mediante gráficas del proceso que acabamos de describir, como venimos haciendo desde el principio del trabajo, y que cuando ejecutemos el modulador podremos ver en funcionamiento para explicarlo más detenidamente. Hemos agregado en esta parte también los *QT GUI Range* encargados de ajustar la ganancia de entrada aplicada en la figura 6.12.

6.3.2. Generación del múltiplex de FM estéreo

Como el múltiplex de FM puede tener hasta 100 kHz de ancho de banda cuando se le añaden ciertos servicios, aunque nosotros no lo vamos a usar, vamos a situar la frecuencia de muestreo en esta etapa a 200 ksp/s, por lo que tendremos que realizar una subida de la tasa de muestreo a las señales que intervengan. Estas son, la señal suma y la señal diferencia, que en la figura 6.16 se muestra como elevamos la tasa de muestreo con el *Rational Resampler*.

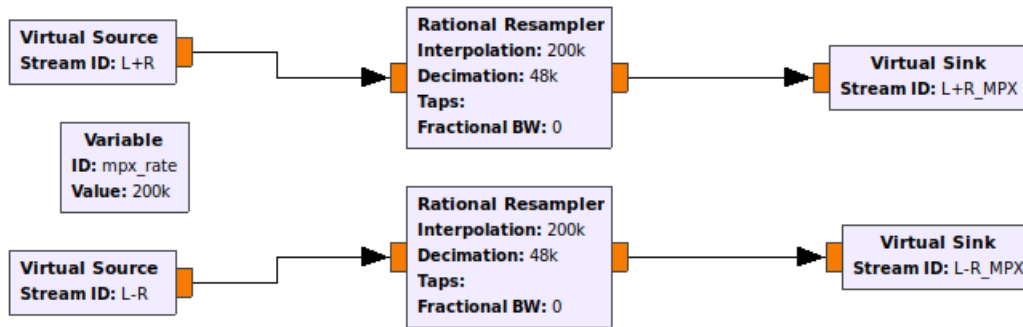


Figura 6.16: Reajuste de la tasa de muestreo de la señal suma y diferencia para entrar al múltiplex.

Una vez que las señales suma y diferencia están a la tasa de muestreo de 200 kps ya es posible realizar el múltiplex con ellas. En la figura 6.17 se encuentra detallado el proceso de realización de la señal múltiplex: se genera un tono de 19 kHz, la señal diferencia se modula en DSB-SC usando una subportadora de 38 kHz y junto con la señal suma se suman las tres, pero antes repartimos la potencia controlando la amplitud de las tres señales con un QT GUI Range, con un bloque de multiplicación por una constante. Para que funcione bien, las señales suma y diferencia tienen que tener el 45 % de la potencia cada una, y el piloto el 10 %, por lo que usaremos 0.45, 0.45 y 0.1 como valor por defecto en los QT GUI Range que controlan dichos multiplicadores.

Cabe destacar que hemos añadido un QT GUI Check Box que cuando se desmarca su variable `disable_stereo` toma valor `false`, y nos sirve para poder quitar el piloto y la señal diferencia en un momento determinado, ya que se encuentra como valor de amplitud de la subportadora de 38 kHz y el piloto de 19 kHz.

Mediante el bloque de representación generamos una representación del espectro del múltiplex.

6.3.3. Modulación FM de la señal múltiplex de FM

Finalmente, en la figura 6.18, la señal del múltiplex se usa como moduladora en un VCO de tipo Complex, subiendo primero la tasa de muestreo de la señal a 1 Msps, este VCO tipo Complex nos dará una señal en fase y cuadratura, que nos servirá para aprovechar la arquitectura en fase y cuadratura del HackRF One. El VCO tiene la sensibilidad ajustada a $2 \cdot \pi \cdot 75000$ para lograr una desviación máxima de 75 kHz. En esta etapa del modulador hemos añadido algunos controles para poder variar la configuración del HackRF One desde la GUI del modulador y hacer así un proyecto en GRC lo más completo posible.

Estos controles son útiles para probar aspectos del HackRF One como los amplificadores de RF internos que incorpora. Y una enorme visualización del espectro de la modulación, así como visualización de la tasa de muestreo de todas las etapas. La frecuencia de muestreo usada en el DAC, como en el proyecto anterior de AM, es la mínima de hackRF One, 2 Msps, que también la hemos declarado como una variable.

6.3.4. Alternancia de las pestañas de la interfaz gráfica

Para poder alternar entre las diferentes pestañas que nos van a permitir visualizar todo el modulador de FM comercial, hemos usado un bloque QT GUI Tab Widget. Para ubicar los diferentes widgets, recordemos como vimos en capítulos anteriores, que había que mencionar en el GUI Hint la ID del bloque de pestañas, una @, y el número de pestaña. Si queremos agregar los valores para posicionar el widget dentro, usamos dos puntos (:) y los cuatro valores de posición y relleno separados por comas.

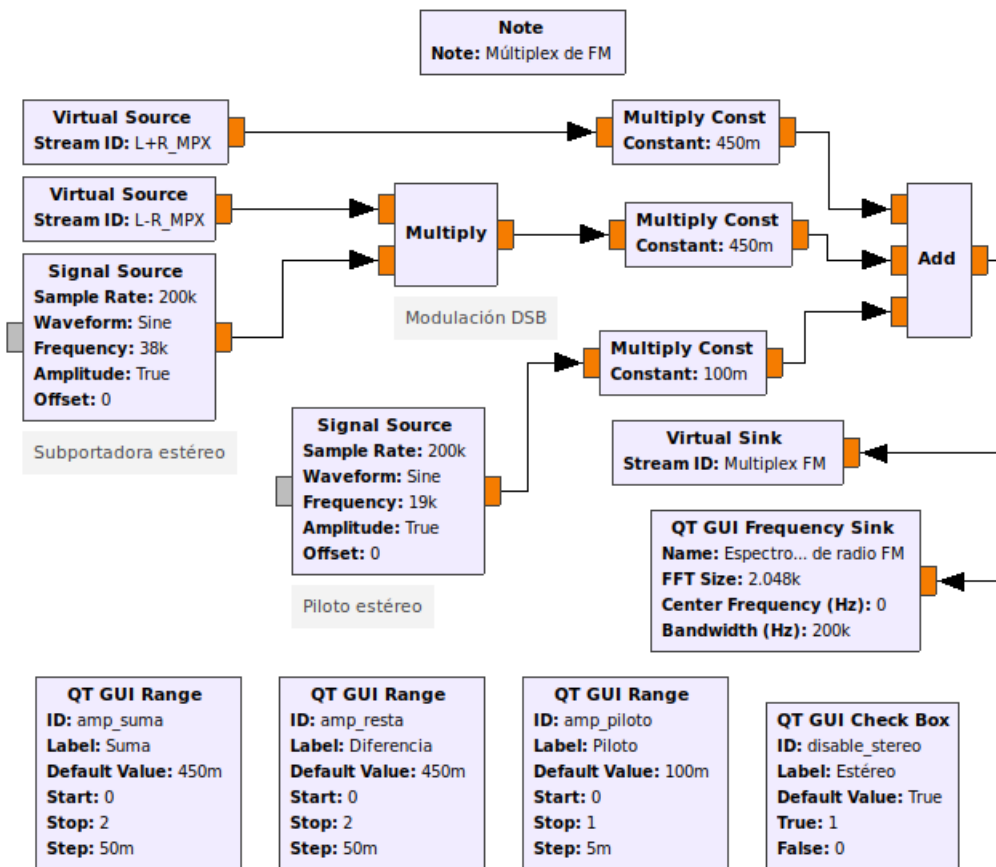


Figura 6.17: Generación del múltiplex de FM.

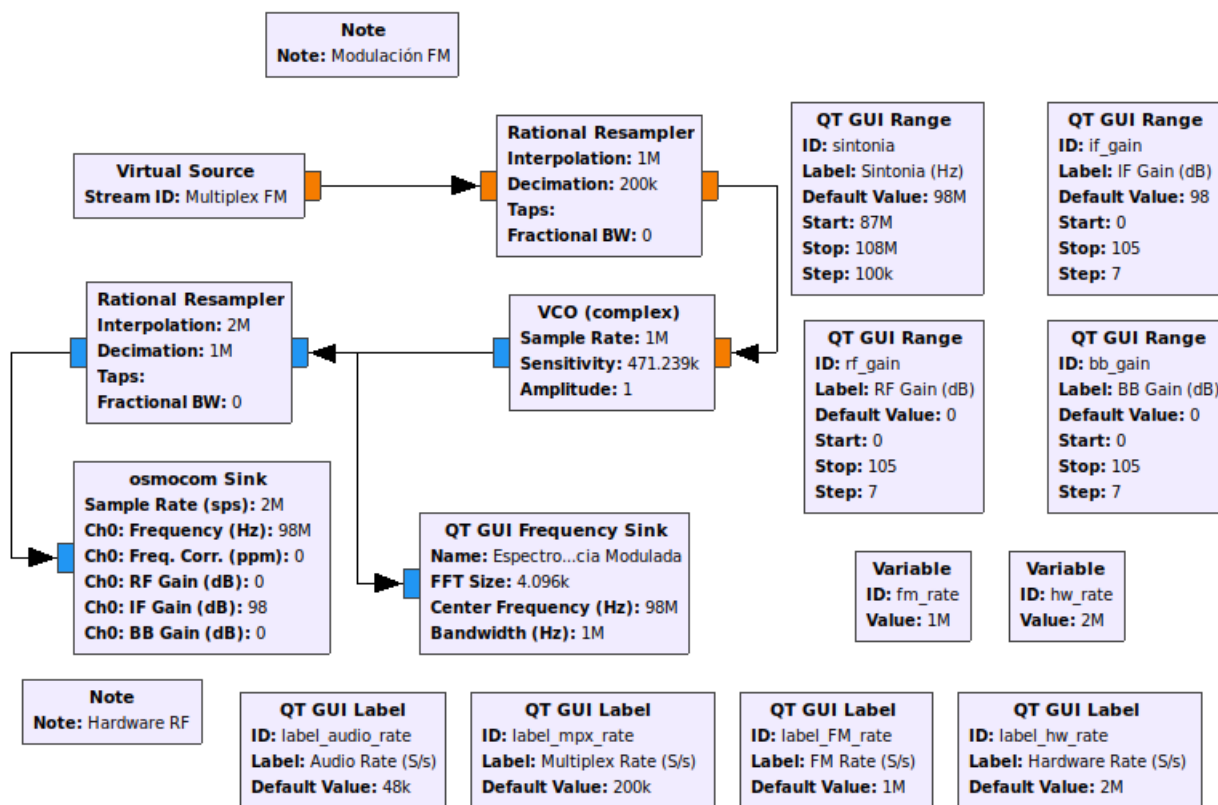


Figura 6.18: Etapa de modulación y transmisión de la señal FM.

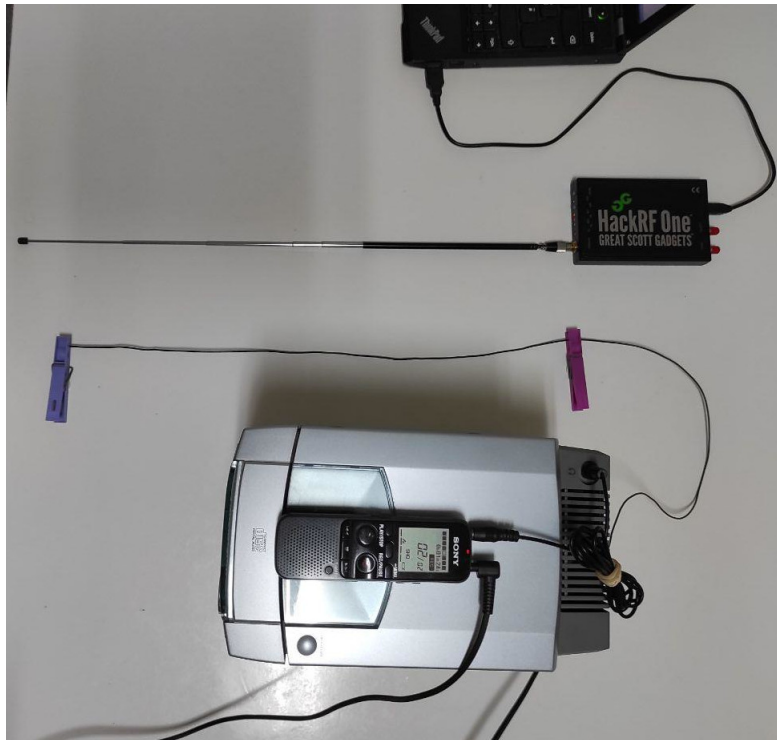


Figura 6.19: Montaje del experimento con el modulador FM comercial, el receptor de radio FM y la grabadora digital.

6.3.5. Prueba del modulador FM estéreo comercial

Para probar este modulador, continuamos con el plan iniciado para el modulador de radio AM comercial. Tomamos el HackRF One y posicionamos la antena, extendida un metro aproximadamente, y tendemos junto a esta la antena de hilo que tiene conectado el receptor de FM. Conectamos la grabadora digital a la toma de auriculares y el HackRF One al puerto usb del equipo. Figura 6.19.

Ya podemos iniciar el modulador de FM estéreo. Acudimos a la última pestaña de nuestra GUI (figura 6.20) y establecemos una frecuencia de transmisión donde hemos comprobado que no existe ninguna estación transmitiendo. En ese momento ya se debería estar escuchando la transmisión en el receptor.

Antes de continuar con la prueba de sonido vamos a examinar la GUI que hemos creado para visualizar y controlar el modulador FM comercial.

Empezamos por la pestaña donde hemos configurado la frecuencia de transmisión, *Transmisión FM y ajustes de RF*, en esa pestaña, además de eso, podemos visualizar la gráfica espectral de la modulación FM y configurar los parámetros de ganancia del HackRF One. Por algún motivo, si subimos la ganancia de RF, a partir de 7 dB el HackRF One deja de transmitir correctamente la señal. El valor que nos ayuda a ajustar la potencia de forma eficaz es el de la ganancia en frecuencia intermedia (IF Gain). La ganancia en banda base no surte ningún efecto al modificarla. En dicha pestaña podemos observar también, como el ancho de banda aproximado de la señal modulada ocupa aproximadamente 150 kHz entre los puntos de corte.

Si cambiamos a la primera pestaña, que hemos titulado *Ganancia de entrada y filtros* (figura 6.21) observamos cuatro gráficas que representan el funcionamiento del filtro de entrada, este filtro recorta la señal paso-bajo a 15 kHz, entonces en las gráficas superiores podemos ver en función del tiempo la señal de entrada y como sale filtrada del filtro pasa-bajo. Aunque se produce un efecto de desfase

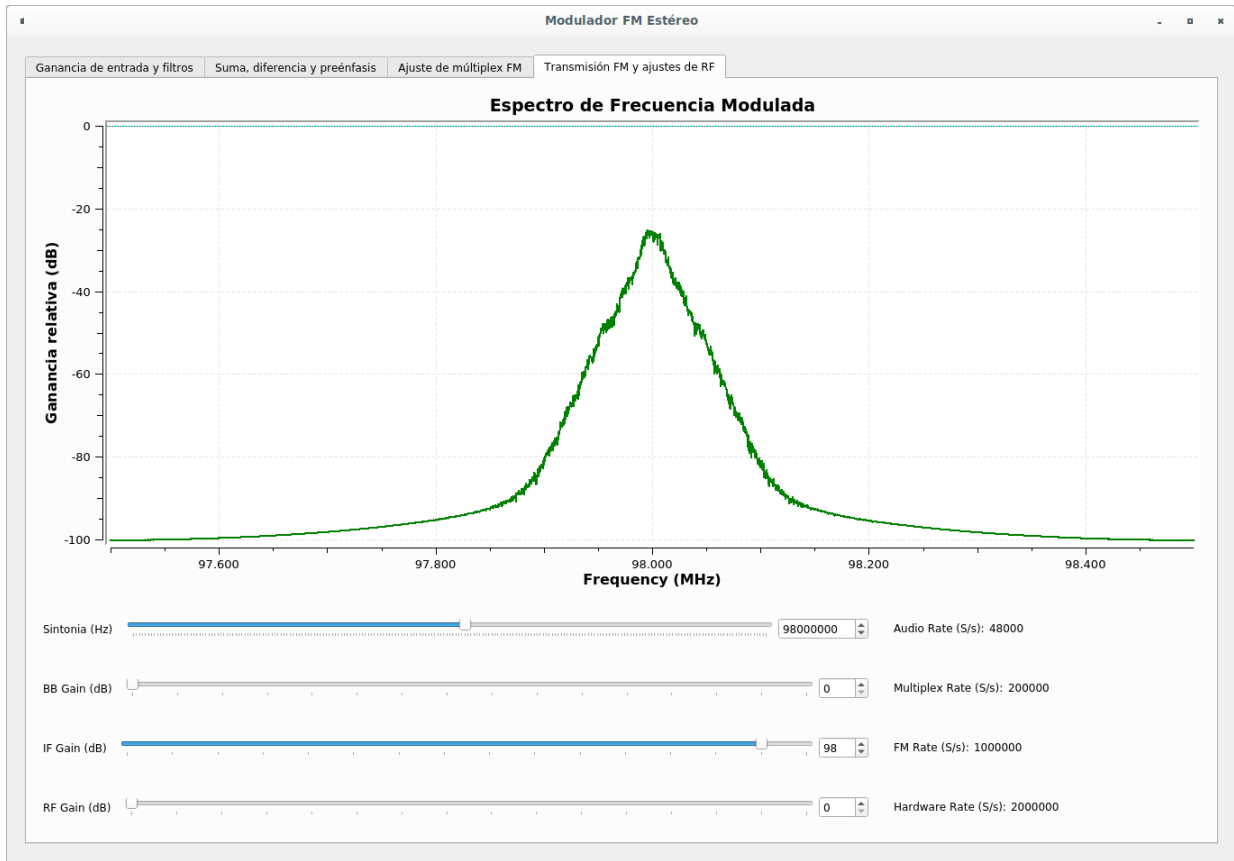


Figura 6.20: Transmisión FM y ajustes de RF.

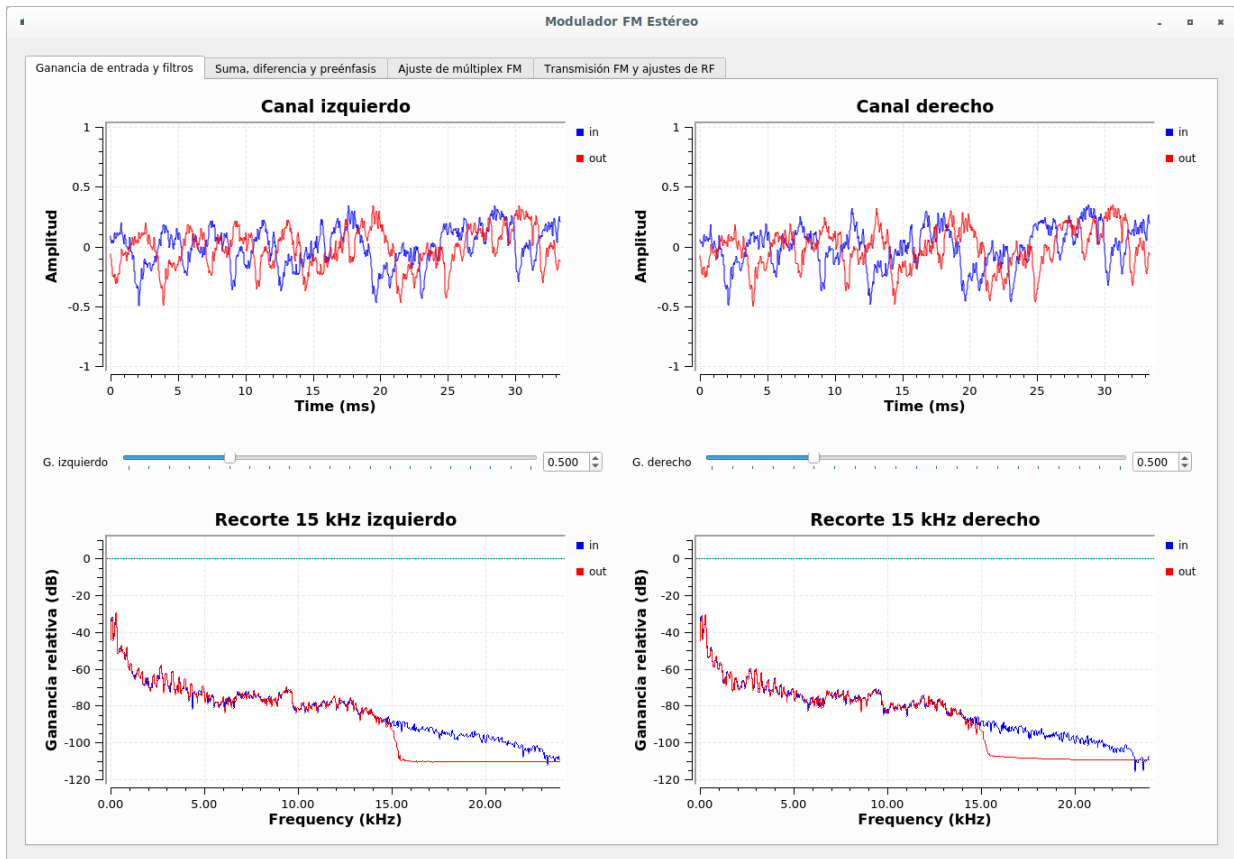


Figura 6.21: Ganancia de entrada y filtros.

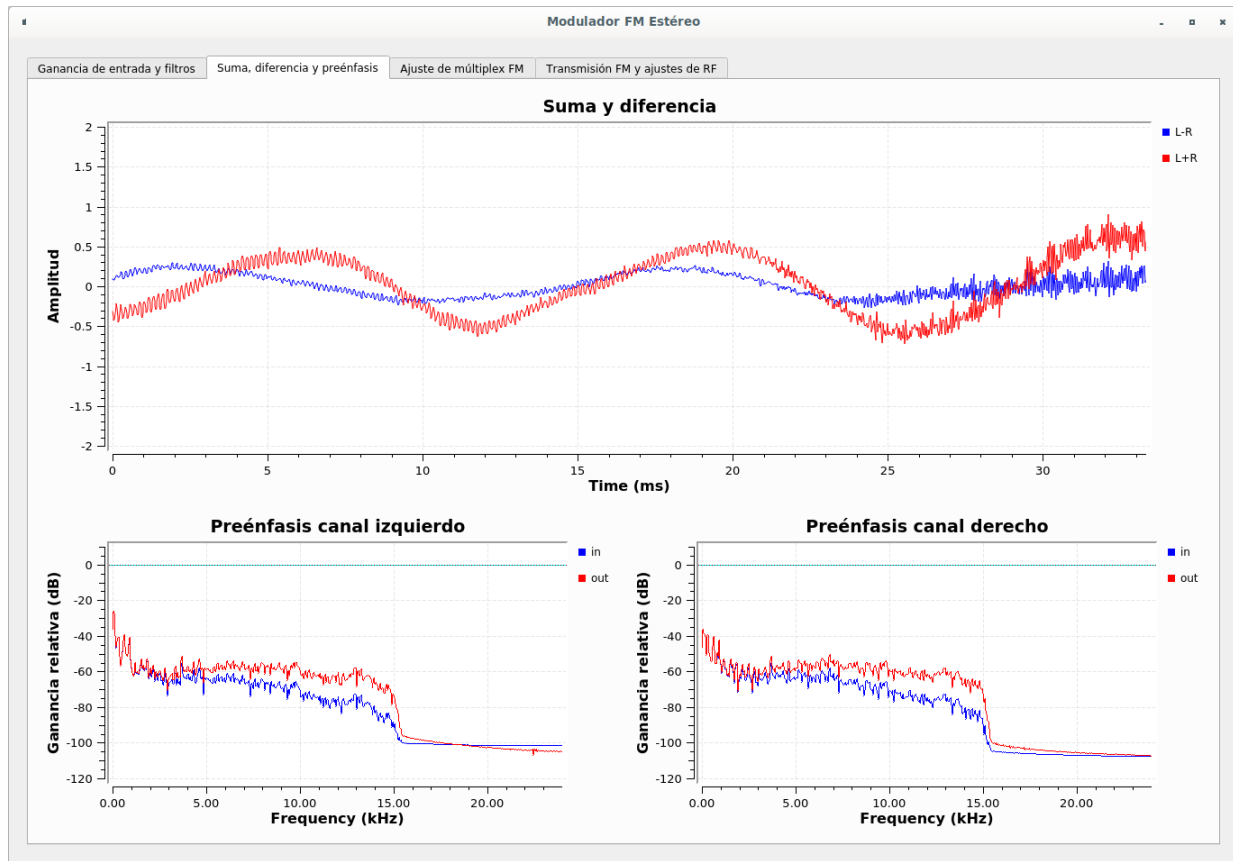


Figura 6.22: Suma, diferencia y preénfasis.

apreciable debido al procesamiento que retiene muestras. En la parte inferior se visualiza el espectro de la señal de entrada al filtro y la de salida, observándose claramente como está el corte a 15 kHz.

En la siguiente pestaña (figura 6.22), titulada *Suma, diferencia y preénfasis*, es posible visualizar en la gráfica superior tanto la señal suma, como la señal diferencia, y en la parte inferior la entrada y la salida del filtro preénfasis de cada canal. Es claramente visible el repunte de las frecuencias altas de la señal, lo cual demuestra que el preénfasis funciona correctamente.

Cuando las señales que se escuchan por los dos canales son muy diferentes entre sí, porque el efecto estéreo es muy intenso, se puede apreciar como la señal diferencia crece bastante. Cuando por ejemplo se está modulando una voz humana, la señal diferencia es cero y la suma toma sus valores más altos.

En la pestaña *Ajuste de múltiplex FM* (figura 6.23) podemos ver el espectro multiplexado y si queremos podemos ajustar la potencia de las tres señales por separado, y así comprobar que efectos tiene en el receptor.

Con las pruebas realizadas, hemos llegado a la conclusión de que el margen de potencias tolerado a la hora de modificar la ganancia de las señales del múltiplex es muy estrecho, pues durante otras pruebas previas de toma de contacto, estuvimos probando como afecta a diferentes modelos de receptores el hecho de subir el tono piloto o la señal diferencia modulada.

Una de las pruebas consistía en transmitir una pista con todo el sonido en un solo canal, pues bien, si la potencia de la modulación de la señal diferencia variaba lo mínimo, la cancelación del canal silenciado se perdía casi por completo.

También comprobamos el nivel mínimo de detección del piloto. En algunos receptores tiene un efecto de enganche muy acusado, quiere decir que cuando detectan el tono, podemos bajarlo progresivamente

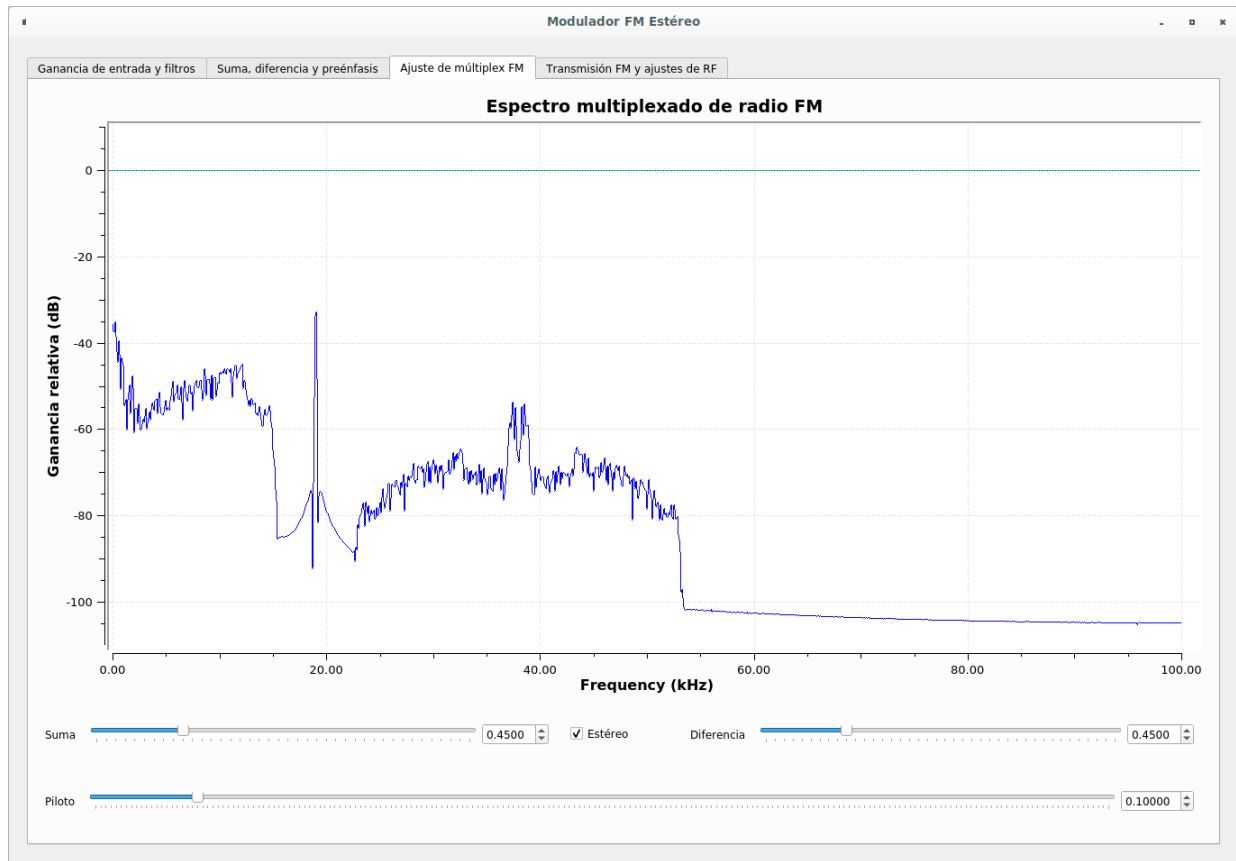


Figura 6.23: Ajuste de múltiplex FM.

y el nivel al que dejan de detectarlo es mucho menor que al que lo empiezan a detectar.

Es importante no saturar la potencia del tono piloto, si hacemos eso en una estación real de FM, los receptores lejanos que no tengan un buen nivel de recepción activaran de todos modos el circuito de estéreo, ya que lo activa el piloto, introduciendo ruido al tratar de demodular la señal diferencia.

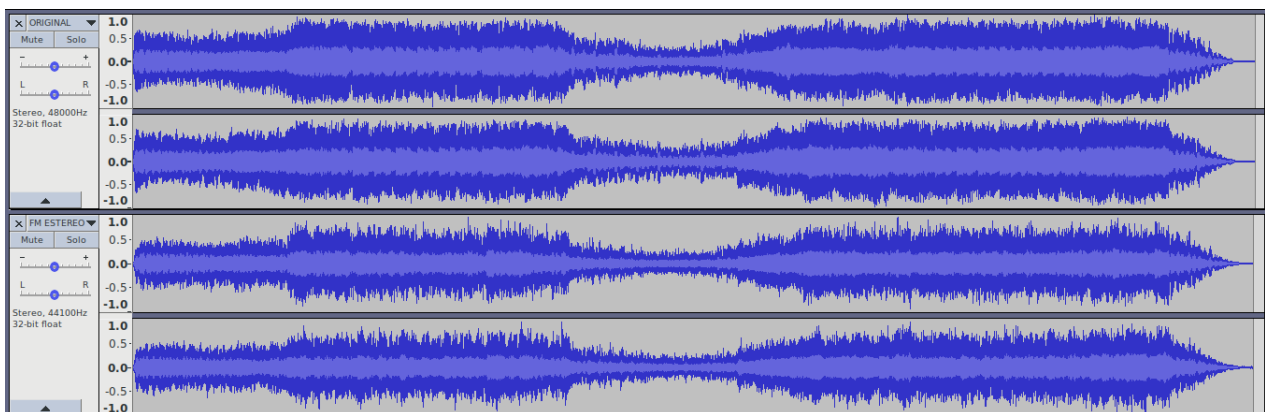


Figura 6.24: Comparación de la forma de onda original y la registrada con la grabadora digital tras la modulación FM.

Para evaluar la calidad de la transmisión, de nuevo, tal y como hicimos en AM, mostraremos en Audacity la forma de onda de la señal original, esta vez con los dos canales de audio, comparado con lo que hemos registrado en la grabadora digital, figura 6.24. También hemos representado el espectro en la figura 6.25. Esta vez el recorte queda a 15 KHz.

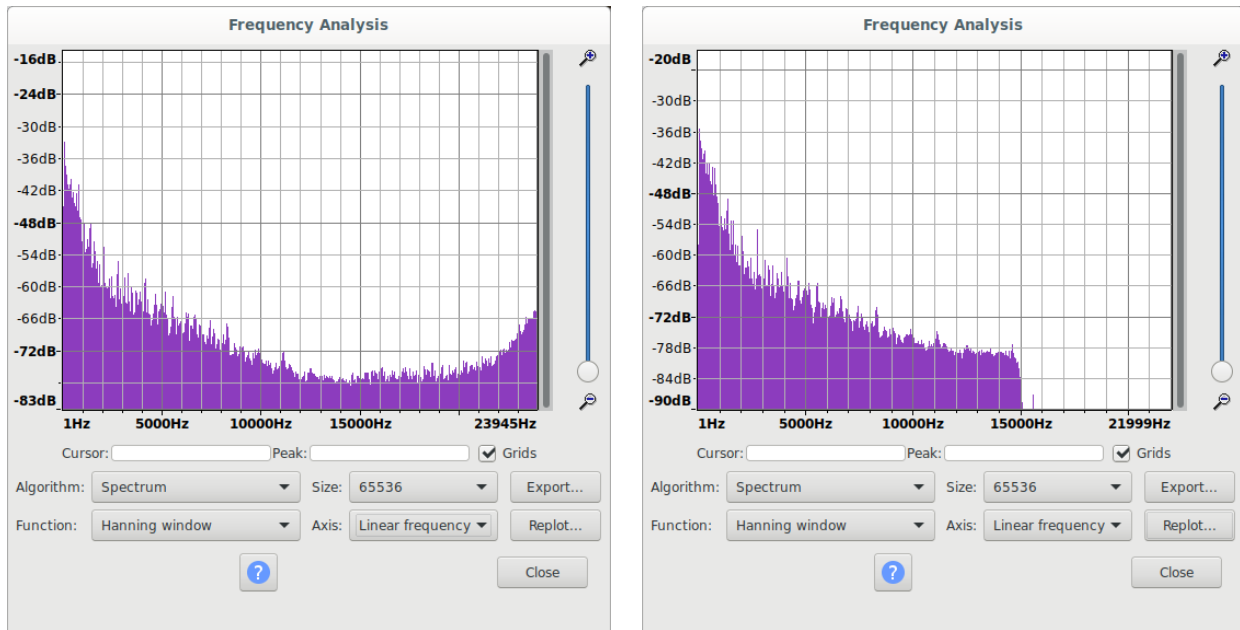


Figura 6.25: Espectro de la pista original y espectro de la pista en el receptor de FM.

El sonido en estéreo parece un poco ruidoso en cuanto empeora la recepción, debido al circuito de estéreo, pero en FM monofónico la calidad de sonido es excepcional, y no se llega a percibir ruido en ningún momento.

Con los resultados obtenidos queda de manifiesto que la modulación FM es superior a la AM en el uso de radiodifusión analógica, las razones son muy evidentes. Pese a ello la modulación AM ha utilizado 9 kHz, para transmitir una señal de 4.5 kHz, mientras que la FM ha utilizado un canal de RF de casi 200 kHz para transmitir 30 kHz. Lo que marca la diferencia no es por tanto el ancho de banda sino la SNR que se obtiene con la FM. Pues generamos señales de mayor ancho de banda, pero la técnica de modulación es muy robusta frente al ruido y las señales adyacentes, pues en nuestro experimento hemos modulado muy próximos a las frecuencias de otras dos estaciones, lo cual no ha significado ningún problema, una vez que el receptor queda enganchado a nuestra portadora de FM.

Capítulo 7

Conclusiones y futuras líneas

7.1. Conclusiones

Concluimos este trabajo habiendo demostrado que el software GNU Radio, es muy útil para las implementaciones que se realizan en el ámbito académico de teoría de la señal.

Se han implementado diferentes modulaciones analógicas y digitales a modo de simulación, mediante GNU Radio Companion, y se ha podido comprobar su funcionamiento de forma teórica.

Se ha implementado un modulador de radiodifusión AM comercial, utilizando GNU Radio Companion y además el hardware HackRF One. Se ha comprobado su correcto funcionamiento con receptores de AM comerciales en campo cercano.

También se ha implementado un modulador de radiodifusión FM estéreo, en GNU Radio Companion y el hardware HackRF One, comprobando su funcionamiento completo con receptores comerciales, así como los efectos en el ajuste de los diferentes parámetros de la modulación.

7.2. Futuras líneas

Al incorporar GNU Radio a la realización de ensayos y prácticas se abre un gran abanico de futuras líneas en la modalidad sólo software:

- Realizar un curso básico donde se trabaje con bloques de alto nivel, de forma que se realicen pruebas típicas, como por ejemplo generar una onda con diferentes formas y escucharlas a través de la tarjeta de sonido, muestrear el micrófono y guardar la señal en un fichero .wav, procesar una señal de audio con un filtro, usar módulos que directamente realicen la modulación y demodulación para tener una perspectiva panorámica de los conceptos sin forzar a la implementación minuciosa, etc.
- Desarrollar demoduladores de forma minuciosa como en este trabajo, para ampliar cada uno de los ejemplos, e idear una forma de conectar los diagramas usando bloques de comunicación.
- Estudio de modelos de canal, y otros efectos típicos de los sistemas de telecomunicación usando los bloques que modifican la señal, para comprobar cuál es el resultado de pasar la señal modulada por uno de ellos.
- Aprender a programar dentro de los bloques que soportan scripts Python.

Y teniendo en cuenta el uso de hardware, algunas futuras líneas podrían ser:

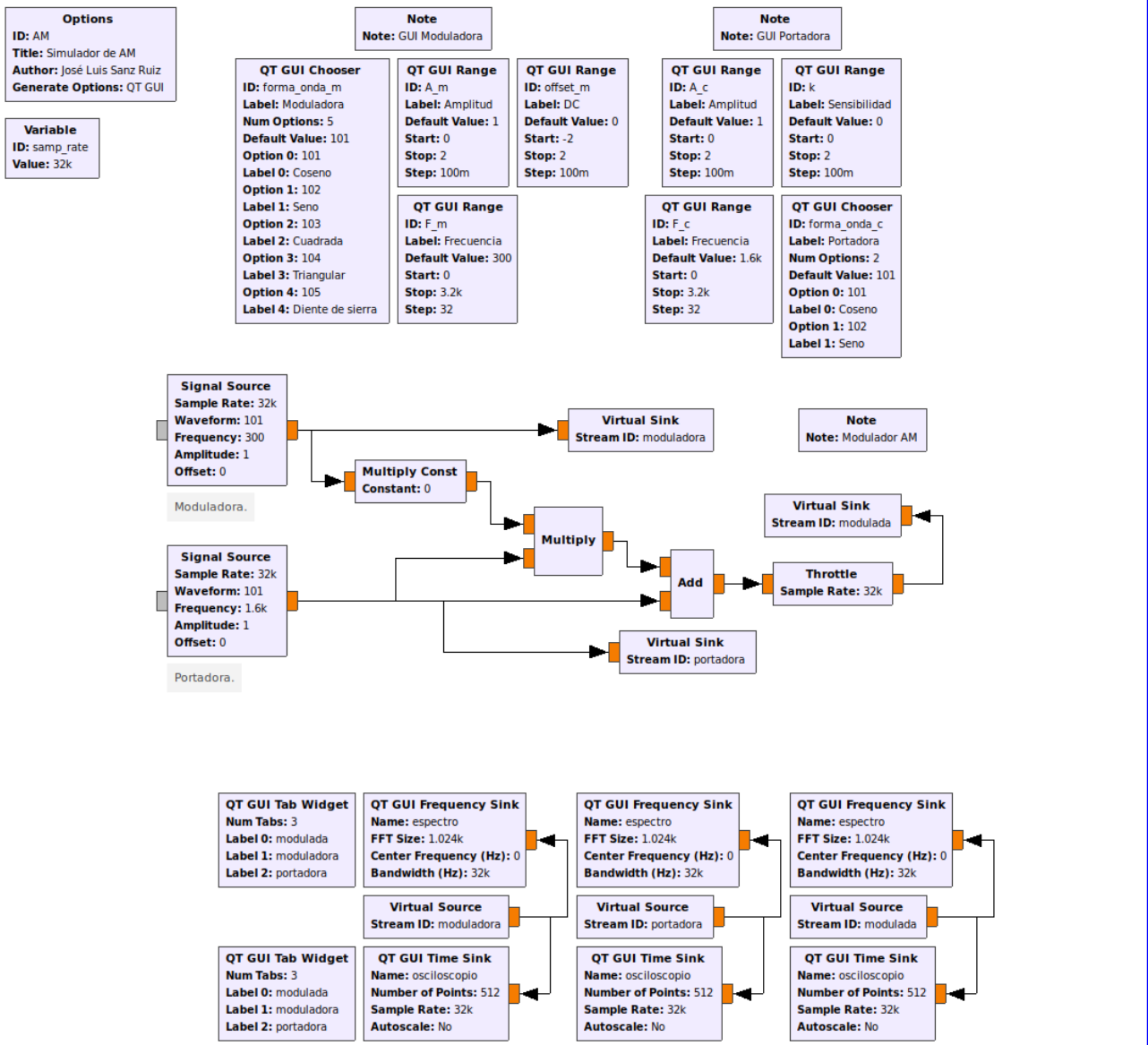
- Implementaciones de receptores de AM comercial y FM estéreo, para cumplimentar los moduladores vistos en este trabajo.
- Explotar las propiedades de recepción del hardware para realizar el estudio de señales de radio, sin necesidad de usar analizadores de espectro, y modelar receptores de diversos tipos.
- Promover el uso de alguna herramienta de las que hay publicadas en los repositorios de aplicaciones y que están basadas en GNU Radio pensadas para CW, TTY y comunicaciones espaciales, por ejemplo.
- Mejorar el hardware o diseñar un hardware para poder trabajar con GNU Radio.

Apéndice A

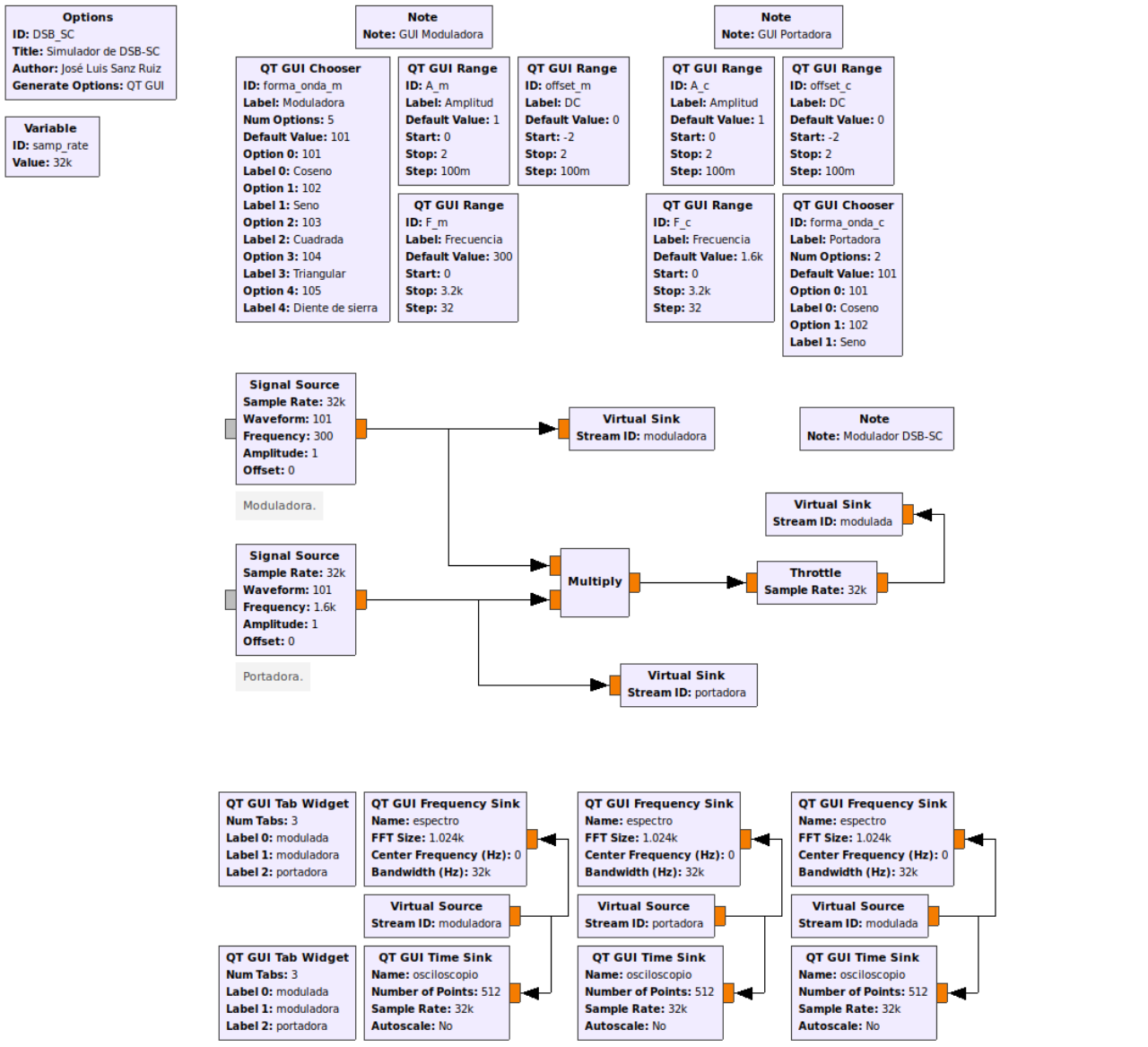
Imágenes de los esquemas realizados

En este apéndice se encuentran todas las imágenes de los esquemas sin recortar, tal como se exportan de GNU Radio Companion.

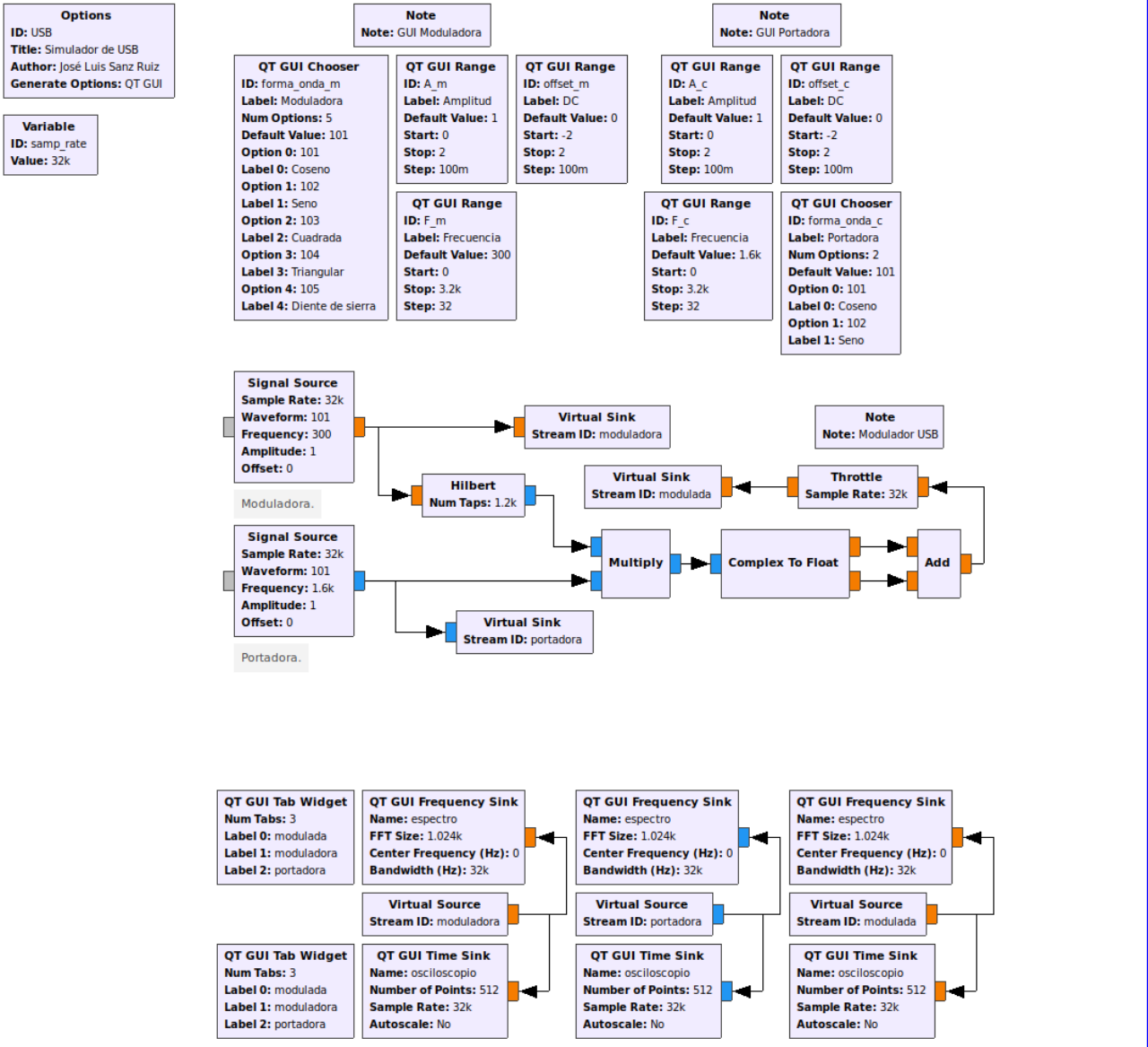
AM.png



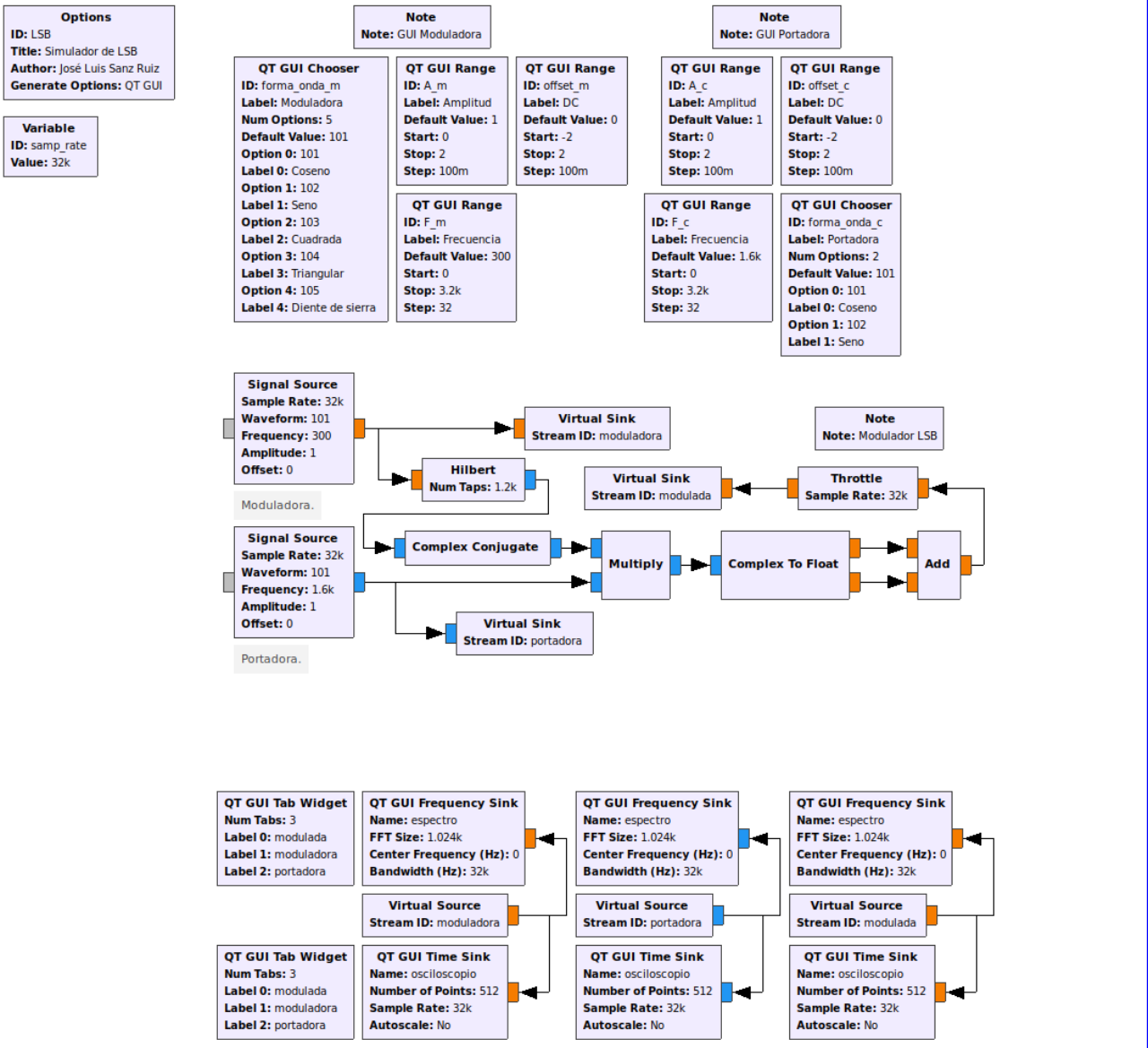
DSB-SC.png



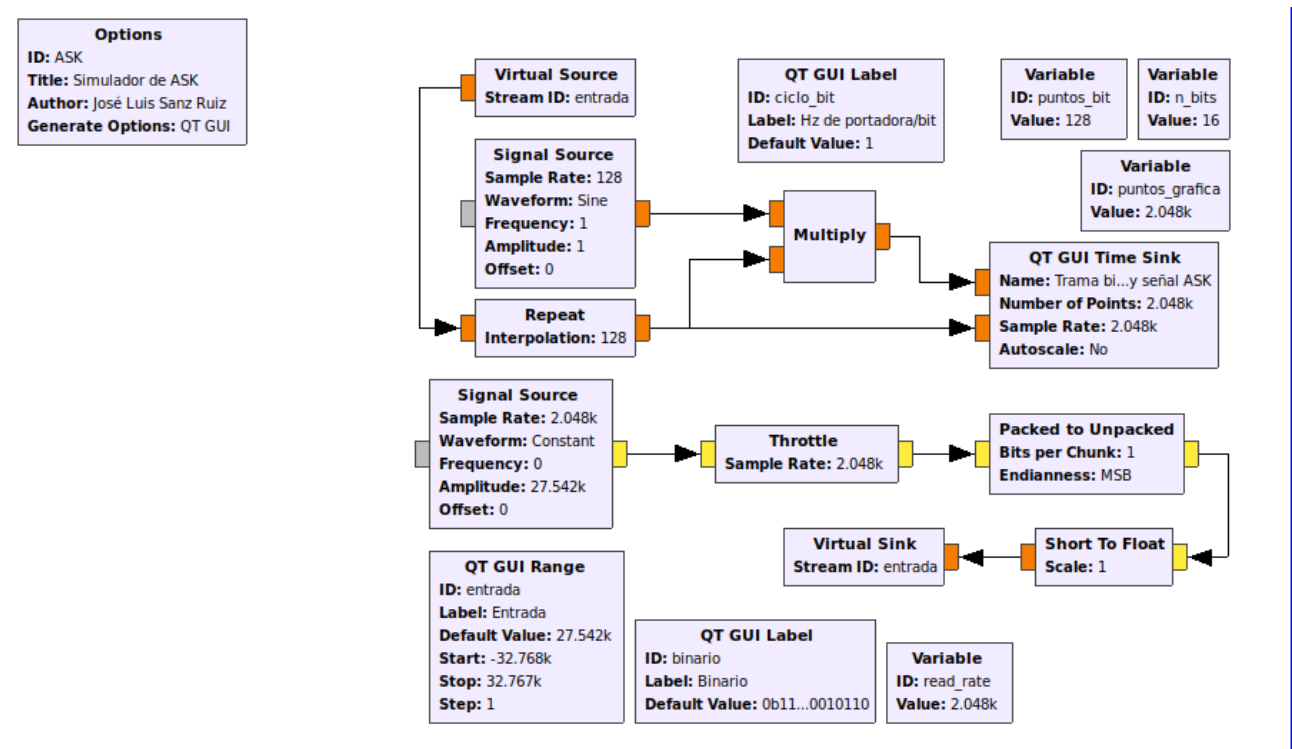
USB.png



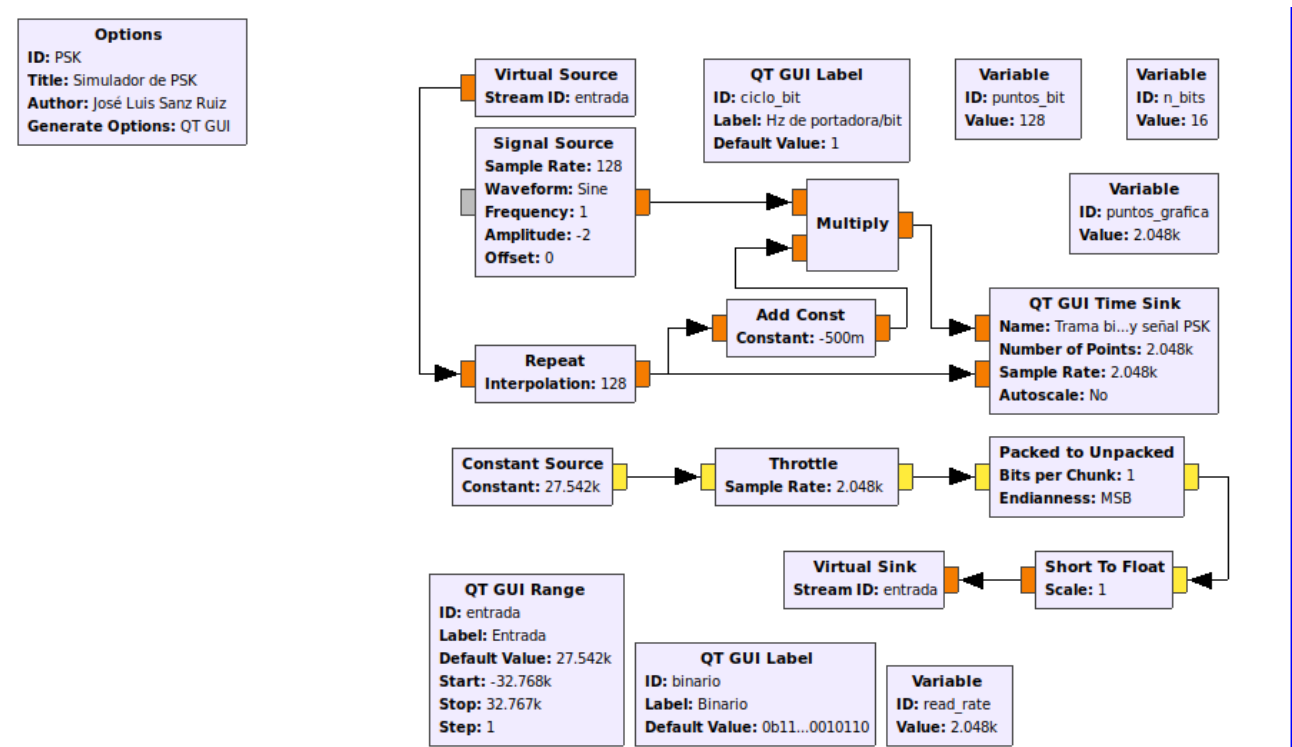
LSB.png



ASK.png



PSK.png



FM.png

Options
 ID: FM
 Title: Simulador de FM
 Author: José Luis Sanz Ruiz
 Generate Options: QT GUI

Variable
 ID: samp_rate
 Value: 320k

Variable
 ID: pi
 Value: 3.14159

Note
 Note: GUI Moduladora

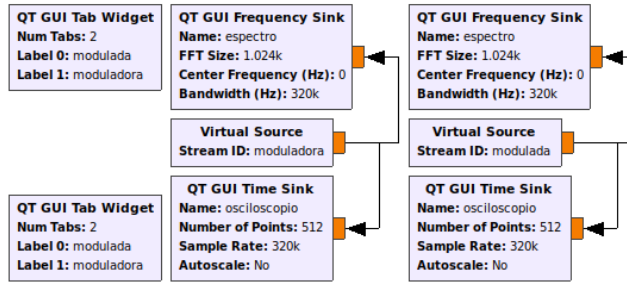
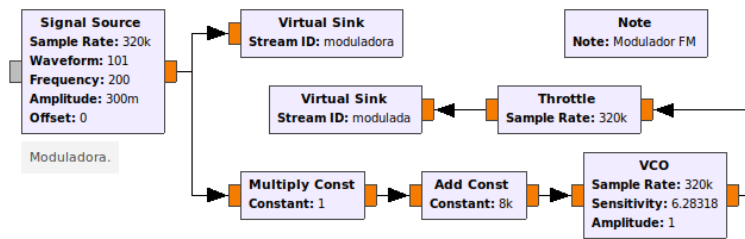
QT GUI Chooser
 ID: forma_onda_m
 Label: Moduladora
 Num Options: 5
 Default Value: 101
 Option 0: 101
 Label 0: Coseno
 Option 1: 102
 Label 1: Seno
 Option 2: 103
 Label 2: Cuadrada
 Option 3: 104
 Label 3: Triangular
 Option 4: 105
 Label 4: Diente de sierra

QT GUI Range
 ID: A_m
 Label: A. moduladora
 Default Value: 300m
 Start: 0
 Stop: 2
 Step: 100m

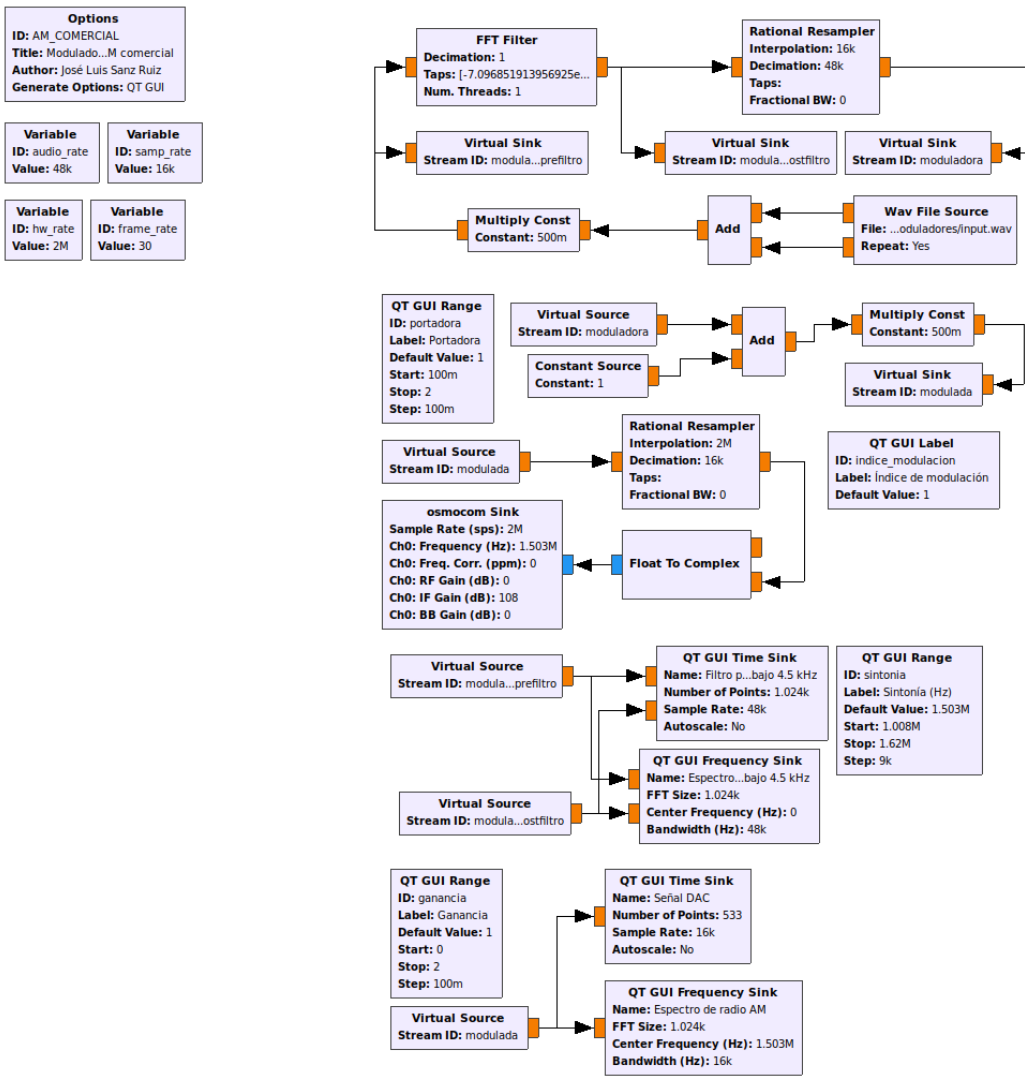
QT GUI Range
 ID: f_c
 Label: F. portadora
 Default Value: 8k
 Start: 0
 Stop: 16k
 Step: 1

QT GUI Range
 ID: F_m
 Label: F. moduladora
 Default Value: 200
 Start: 0
 Stop: 3.2k
 Step: 1

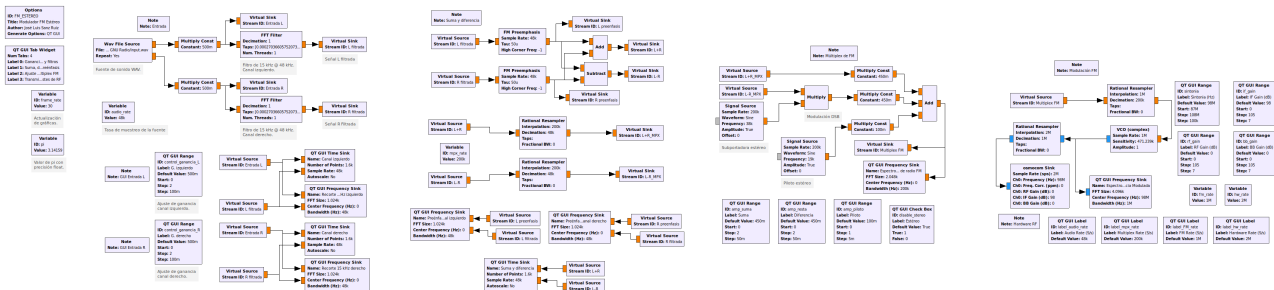
QT GUI Range
 ID: f_d
 Label: Sensibilidad
 Default Value: 1
 Start: 1
 Stop: 10k
 Step: 1



AM COMERCIAL.png



FM ESTEREO.png



Bibliografía

- [1] Leandro Juan Llácer, Juan Morales Sánchez. BII. Analog Communications: Continuous Wave Modulation (Teoría de la Comunicación), *Universidad Politécnica de Cartagena*, 2020
- [2] Ramón Ruiz Merino, Vicente Garcerán Hernández, Juan Hinojosa Jiménez. Apuntes de Electrónica para Telecomunicaciones, *Universidad Politécnica de Cartagena*, 2017
- [3] Wikipedia: Ruido de color, https://es.wikipedia.org/wiki/Ruido_de_color
- [4] José María Drake Moyano, Ruidos e Interferencias: Técnicas de reducción. Univesidad de Cantabria. https://www.ctr.unican.es/asignaturas/instrumentacion_5_IT/IEC_4.pdf
- [5] Wikipedia: Ancho de banda, https://es.wikipedia.org/wiki/Ancho_de_banda
- [6] Wikipedia: Espectro electromagnético, https://es.wikipedia.org/wiki/Espectro_electromagn%C3%A9tico
- [7] EcuRed: Modulación ASK, https://www.ecured.cu/Modulaci%C3%B3n_ASK
- [8] Wikipedia: Modulación por desplazamiento de fase, https://es.wikipedia.org/wiki/Modulaci%C3%B3n_por_desplazamiento_de_fase
- [9] Fisher Telecommunication: Constellation diagram, <https://www.fishercom.xyz/division-multiplexing/constellation-diagram.html>
- [10] GNU Radio FAQ, <https://wiki.gnuradio.org/index.php/FAQ>
- [11] DSP RELATED.com: Understanding the 'Phasing Method' of Single Sideband Demodulation, <https://www.dsprelated.com/showarticle/176.php>
- [12] Experiment 6: Frequency Modulation (FM), Generation and Detection, <https://wwwusers.ts.infn.it/~milotti/Didattica/Segnali/Exp93f.pdf>
- [13] Wikipedia: Radio definida por Software, https://es.wikipedia.org/wiki/Radio_definida_por_software
- [14] Receptor de AM con transistores PNP con BC548, <http://www.pesadillo.com/pesadillo/?p=9709>
- [15] GNU Radio: AM Demod, https://wiki.gnuradio.org/index.php/AM_Demod
- [16] https://www.researchgate.net/figure/Block-diagram-of-GNU-Radio-software-and-HackRF-One-hardware-system_fig1_336578019
- [17] GNU Radio, <https://www.gnuradio.org/>
- [18] Proyecto osmosdr, <https://osmocom.org/projects/gr-osmosdr/wiki>
- [19] Imagen HackRF One, https://commons.wikimedia.org/wiki/File:SDR_HackRF_one_PCB.jpg
- [20] Great Scott Gadgets: HackRF One, <https://greatscottgadgets.com/hackrf/one/>
- [21] GNU Radio: GUI Hint, https://wiki.gnuradio.org/index.php/GUI_Hint