

Maqueta de prácticas de comunicaciones digitales basada en el uso de USRP sobre Linux

Álvaro Pendás Recondo, Jesús A. López-Fernández

uo251853@uniovi.es, jelofer@uniovi.es

Departamento de Ingeniería Eléctrica, Electrónica, de Computadores y de Sistemas. Universidad de Oviedo.
Área de Teoría de la Señal y Comunicaciones. C\ Luis Ortiz Berrocal s/n Campus de Gijón 33203 Gijón (Asturias).

Abstract—Universal Software Radio Peripheral (USR) is a range of front-end devices designed by Ettus Research and used to implement Software Defined Radio (SDR) applications. The main advantages of both USRP and SDR are their remarkable flexibility and versatility when it comes to the system configuration and parameters like power, bandwidth or frequency band. The aim of this work is to make use of those advantages with educational purposes. In this regard, it is also interesting the possibility that SDR offers of simulating different types of channels. We have implemented several SDR systems to be used as a case study during one or multiple laboratory sessions. The main goal is that students can get a better general understanding of digital communications systems, familiarizing themselves with using USRP and SDR along the process. The USRP selected was the model B200 by Ettus Research, and the SDR environment, GNU Radio on Linux.

I. INTRODUCCIÓN

Habitualmente las prácticas de laboratorio sobre comunicaciones digitales suelen dividirse en dos modalidades: hardware y software. El uso principal de elementos de hardware ofrece un gran realismo al alumno. Por su parte, las ventajas principales del empleo de software son la versatilidad y, en general, el bajo coste. Justamente las ventajas de una modalidad coinciden con las desventajas de la otra. El equipamiento hardware suele llevar aparejado un coste elevado y no siempre es todo lo reconfigurable que sería deseable. A su vez, el empleo exclusivo de software aleja al alumno de los sistemas reales, dificultando así su comprensión de las limitaciones de los dispositivos físicos. El objetivo de este trabajo es ofrecer una alternativa que conjugue las ventajas de ambas modalidades.

La tecnología Software Defined Radio (SDR) [1], se basa en el diseño de sistemas de comunicaciones en los que los elementos de la capa física, tradicionalmente implementados mediante hardware electrónico, pasan a ser programados por software. Las principales ventajas que ofrece SDR son: abaratamiento de los costes, aumento de la versatilidad y robustez frente a fallos en los dispositivos físicos.

Por su parte, el nombre Universal Software Radio Peripheral (USR) hace referencia al conjunto de dispositivos hardware del fabricante Ettus Research [2] diseñados para ser programados mediante SDR. Dichos dispositivos implementan el front-end que toda aplicación SDR necesita, siendo su tarea principal la conversión digital/análogica (DAC del inglés Digital Analog Conversion), y analógica digital (ADC de Analog to Digital Conversion), así como las etapas de radiofrecuencia.

En este trabajo se ha desarrollado una maqueta de sistema de comunicaciones digitales sobre USRP con la finalidad

de ser usada como banco de prácticas. Los bloques de comunicaciones de SDR se han programado en un ordenador de sobremesa (PC) con sistema operativo Linux mediante GNU Radio [1]. El front-end de los sistemas generados es implementado mediante dos USRP modelo Ettus Research B200 [3] que ofrecen un buen compromiso entre prestaciones y precio. El canal físico de transmisión empleado es un cable coaxial que conecta entre sí los dos USRP y que apenas introduce ninguna perturbación. Asimismo, se ha introducido un emulador de canal que permite controlar las perturbaciones que sufre la señal transmitida y modificarlas en tiempo real. Esta conjunción entre canal físico y canal virtual ofrece versatilidad y realismo al sistema propuesto.

El presente artículo está organizado de la siguiente forma. En la Sección II se detallan las herramientas empleadas y se muestra el aspecto final de la maqueta propuesta. En la Sección III se describe la implementación de los sistemas SDR. La Sección IV incluye una serie de ejemplos de aplicación. Por último, la Sección V comprende las conclusiones extraídas del presente trabajo.

II. HERRAMIENTAS EMPLEADAS Y MAQUETA PROPUESTA

En este apartado se detallan las herramientas, tanto software como hardware, usadas en el sistema implementado y se muestra el aspecto final que tiene la maqueta propuesta.

A. Herramientas empleadas

En primer lugar, como herramienta de SDR se optó por el empleo de GNU Radio ejecutado en Linux. GNU Radio es una herramienta libre y de código abierto de desarrollo de software que proporciona bloques de procesado de señal digital [1]. GNU Radio define también la interacción entre dichos bloques y permite la programación de bloques propios en los lenguajes C++ y/o Python. Por otro lado, la interfaz gráfica GNU Radio Companion (GRC), permite el manejo y conexión de bloques de manera amigable para el usuario. Además, algunos de los bloques implementados son específicos para el manejo y configuración de dispositivos USRP, si bien GNU Radio es compatible con una amplia lista de otros dispositivos hardware de distinta gama y fabricante [4]. Las principales ventajas que ofrece GNU Radio son: alta configurabilidad y potencia; gratuidad y disponibilidad de consulta de su código fuente [5]; y amplia extensión en el ámbito académico, empresarial e investigador.

En cuanto a los elementos hardware, se ha optado por utilizar dos USRP Ettus Research B200 [3]. Este modelo presenta una funcionalidad y versatilidad notables en relación

a su coste. No obstante, la práctica podría realizarse con otros modelos USRP sin más que realizar unos ligeros ajustes. El receptor se basa en un amplificador de bajo ruido como primer elemento, seguido de un amplificador ajustable. Tras esto, incorpora un demodulador homodino Fase/Quadratura (I/Q) seguido de la etapa ADC. El esquema en transmisión es completamente análogo, empleándose, claro está, una etapa de DAC. Las características más destacables son un ancho de banda de 56 MHz y un rango en frecuencia de 70 MHz a 6 GHz [3]. Una guía completa para la instalación e integración de todo el software necesario se puede encontrar en [6].

La elección de Linux como sistema operativo se debe a que así es recomendado encarecidamente por los desarrolladores de GNU Radio, siendo el único sistema operativo para el que se ofrece soporte. En este caso se ha optado por la distribución Ubuntu, que también es compatible con el hardware seleccionado. Otras distribuciones como Debian o Fedora son también válidas [6].

B. Maqueta propuesta

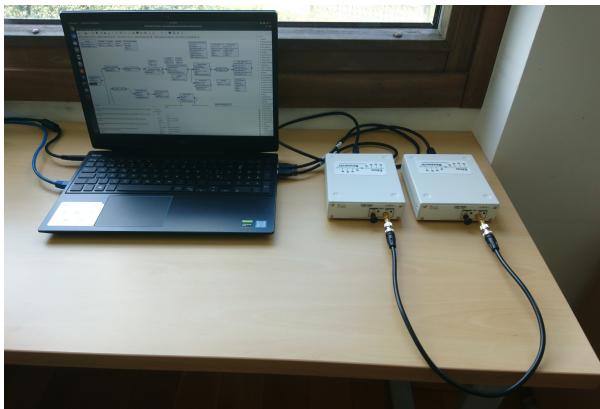


Fig. 1. Ilustración de la maqueta de prácticas propuesta.

El montaje físico de la maqueta propuesta se representa en la fotografía de la Figura 1. En dicha fotografía se observa que hay dos unidades B200, transmisor y receptor, conectadas entre sí mediante un cable coaxial. Cada una de las unidades B200 se conecta al PC mediante un cable USB (preferiblemente USB 3.0 porque ofrece un mayor ancho de banda en la comunicación [7]).

Una vez preparado el montaje, la ejecución de un único archivo de GNU Radio desde el PC permite el control simultáneo de ambos USRP. Por un lado, se generarán las muestras digitales a mandar al transmisor. Por otro, se recibirán y procesarán digitalmente las muestras recibidas desde el receptor. En realidad, para la realización de la práctica sería posible el empleo de un único USRP B200, ya que se trata de un dispositivo full-duplex, capaz de transmitir y recibir simultáneamente. Sin embargo, esto tendría dos consecuencias no deseadas. Por un lado, se reduciría el ancho de banda máximo a la mitad. Por otro, se trataría de un entorno menos realista que podría solucionar problemas de sincronización entre transmisor y recepción.

III. IMPLEMENTACIÓN DE LOS SISTEMAS

En este apartado se explicará de manera general la implementación en GNU Radio de los sistemas desarrollados.

El esquema genérico para un sistema digital tomado como referencia se recoge en la Figura 2, tal y como se establece en [8]. A partir de dicho esquema se implementaron las modulaciones de fase QPSK y 8-PSK, así como las de amplitud en cuadratura 16-QAM y 64-QAM [8].

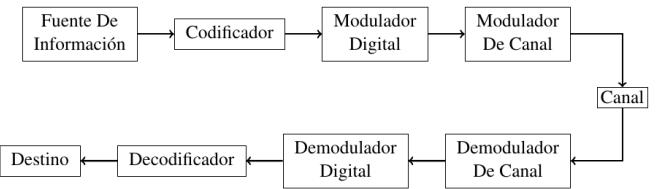


Fig. 2. Esquema de sistema de comunicaciones digitales empleado como referencia en este trabajo.

El bloque *Random Source* de GNU Radio implementa una fuente de bytes aleatoria que actúa como Fuente de Información. Para reagrupar dichos bytes en grupos de bits según la modulación se utiliza el bloque *Packed to Unpacked*.

El Modular Digital se compone de tres bloques: el *Differential Encoder*, que se encarga de la codificación diferencial [8]; el *Chunks to Symbols*, que asigna las agrupaciones de bits a valores complejos según las constelación empleada; y el *Polyphase Arbitrary Resampler*, que modula los símbolos a transmitir empleando como función base la raíz de coseno alzado [8]. Por otro lado, la tarea del Modulador de Canal es realizada por el propio USRP, que se maneja mediante el bloque *UHD:USRP Sink*. A su vez, el Canal se corresponde con el canal físico, en este caso un medio cableado. Sin embargo, al final del Modulador Digital se añade el bloque *Channel Model* que permite añadir a la señal transmitida perturbaciones adicionales.

El Demodulador de Canal es implementado por el USRP en recepción, controlado mediante el bloque *UHD: USRP Source*. Las muestras generadas por la etapa ADC son manejadas por el resto de bloques de software, esto es, el Demodulador Digital. Sin embargo, antes de dicha etapa, es necesario un escalado digital para ajustar los valores enviados por el USRP dentro de los rangos apropiados para GNU Radio. Con dicho fin se programó un bloque específico que se denotó como *Power Scale*.

A continuación, el bloque *Polyphase Clock Sync* implementa la sincronización temporal y el filtro adaptado. Esto supone la primera fase del Demodulador Digital. Nótese que el USRP en recepción no tiene información a priori acerca de los instantes temporales en los que comienza y termina cada símbolo enviado por el transmisor. Por ello, el USRP realiza un sobremuestreo, generando múltiples muestras por período de símbolo. En una etapa posterior, el *Polyphase Clock Sync* selecciona la muestra óptima para cada símbolo. Esto es esencial para evitar interferencia entre símbolos (ISI – Intersymbol Interference) [8].

Tras el sincronismo temporal, se implementa un igualador ciego adaptativo para compensar los efectos del canal. Para 64-QAM el igualador empleado fue el *LMS Equalizer*. Para el resto de modulaciones se optó por el *CMA Equalizer*. Tras el igualador, para las modulaciones QPSK y 8-PSK, se implementó el bloque *Costas Loop*, dedicado a corregir desviaciones en fase y/o frecuencia. Dicho bloque es sustituido

por el *Constellation Receiver* para los casos de 16-QAM y 64-QAM. Por último, el bloque *Constellation Decoder* realiza las tareas de decisor, asignando los valores sobre la constelación a grupos de bits. Estos son decodificados diferencialmente por el bloque *Differential Decoder*.

El código fuente de los bloques empleados, a excepción del *Power Scale*, se encuentra recogido en [5]. Para una información explicativa más en detalle de cada uno de ellos se recomienda consultar [9].

En general, los parámetros más relevantes se mantienen fijos para todos los sistemas. Se empleó así siempre una velocidad de símbolo $v_s = 750 \text{ ksimb s}^{-1}$ y un filtro en raíz de coseno alzado con un factor de *roll-off* $\alpha = 0.35$. Respecto a la configuración de los USRP, se empleó una frecuencia de transmisión y recepción de 1 GHz y se configuró la ganancia por defecto en transmisión en 30 dB para QPSK y 8-PSK, 40 dB para 16-QAM, y 45 dB para 64-QAM. En recepción, la ganancia se fijó en 30 dB para todos los casos. También en recepción, la tasa de muestreo del USRP se fijó en 3 MS s^{-1} , lo que da lugar a 4 muestras por símbolo.

IV. EJEMPLOS DE APLICACIÓN

En esta sección se plantean algunos ejemplos de aplicación de la maqueta de comunicaciones digitales propuesta. Se consideran dos tipos de canal: canal gaussiano y canal lineal, pudiendo ser este último invariante o variante en el tiempo [8]. Para cada ejemplo de aplicación se muestra el diagrama de dispersión de la constelación recibida que sirve como medida cualitativa de la calidad de la comunicación.

A. Canal Gaussiano

En primer lugar, se considera únicamente el canal físico que supone el cable coaxial mostrado en la Figura 1, el cual se puede aproximar por un canal gaussiano. Los distintos sistemas implementados se distinguen según su modulación, a saber: QPSK, 8-PSK, 16-QAM y 64-QAM. En estos primeros sistemas no se añade la etapa de igualación de canal, eliminando el bloque destinado a tal efecto. Mediante GNU Radio es posible representar en tiempo real los resultados de las distintas etapas, tanto en transmisión como en recepción.

La Figura 3 muestra el diagrama de dispersión de la constelación recibida para una 8-PSK. La dispersión de los símbolos de la constelación se debe, principalmente, al ruido (aproximadamente blanco y gaussiano) producido en la cadena de recepción y, en menor medida, al *jitter* y a pequeños errores en el enganche de fase. A modo de experimento, se puede añadir ruido gaussiano simulado digitalmente mediante el bloque *Channel Model*. La Figura 4 muestra el resultado para 64-QAM tras añadir ruido de forma que $E_b/N_0 = 20 \text{ dB}$, en el transmisor, sea igual a 20 dB. Es posible variar en tiempo real el nivel de ruido introducido por *Channel Model* y comprobar que las constelaciones menos densas son las que, a igualdad de potencia media transmitida, soportan un mayor nivel de ruido. Para mantener la potencia media transmitida, basta con asignar en todos los casos idéntica ganancia en el transmisor.

B. Canal lineal

Mediante el bloque *Channel Model* es posible también añadir el efecto correspondiente a una propagación multicamino. Con este fin se ha empleado un canal lineal e invariante en el

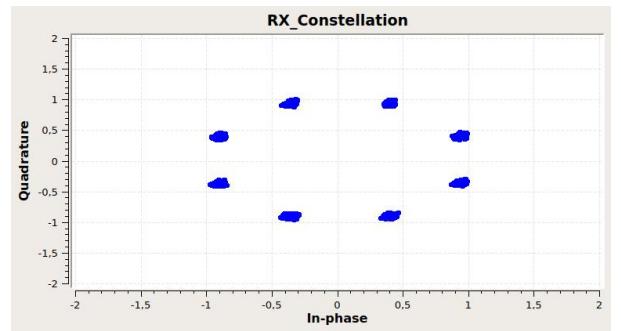


Fig. 3. Constelación recibida para una 8-PSK.

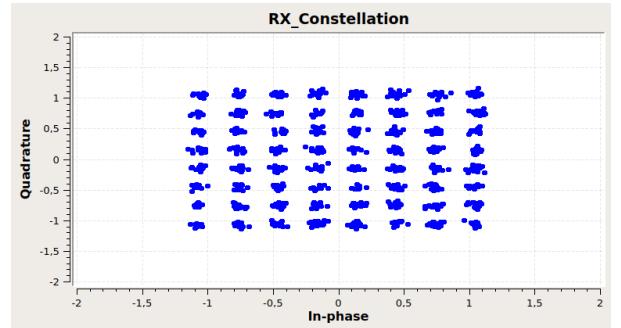


Fig. 4. Constelación recibida para una 64-QAM añadiendo ruido tal que en el transmisor $E_b/N_0 = 20 \text{ dB}$.

tiempo cuya respuesta discreta equivalente viene dada en (1). En la Figura 5 se muestra la constelación recibida para una QPSK bajo este canal y sin emplear igualación. Se observa que la calidad de la recepción se degrada notablemente por causa de la ISI.

$$h[n] = 0.85\delta[n] + 0.526\delta[n - 3]. \quad (1)$$

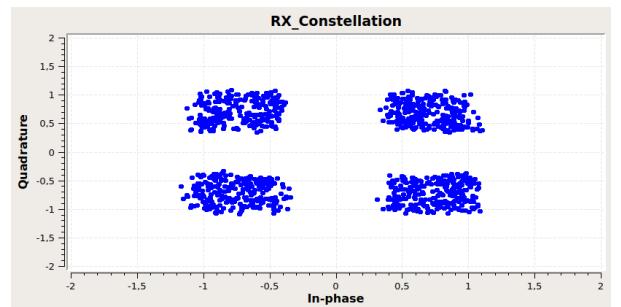


Fig. 5. Constelación recibida para una QPSK sobre $h[n]$ de (1) sin etapa de igualación.

Para mitigar los efectos de la ISI basta con incluir el igualador de canal correspondiente. En la Figura 6 se muestra el resultado, esta vez para una 16-QAM.

Otro caso de propagación interesante es el que tiene lugar cuando se producen desvanecimientos de potencia y/o desplazamiento Doppler, asociados a canales lineales variantes en el tiempo [8]. Modificando, en tiempo real y desde GNU Radio, la ganancia del amplificador del transmisor se generan desvanecimientos de potencia en el tiempo. El sistema es bastante robusto frente a desvanecimientos gracias al escalado

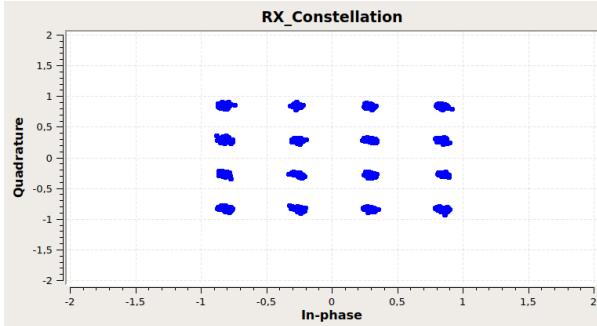


Fig. 6. Constelación recibida para una 16-QAM sobre $h[n]$ de (1).

adaptativo que realiza el bloque *Power Scale*. Así, desva- necimientos del orden de 10 dB únicamente afectan a la relación E_b/N_0 , pero no modifican la posición de los símbolos respecto de las fronteras de decisión.

Respecto al efecto Doppler, es posible generar un desplazamiento en frecuencia variable también mediante *Channel Model*. Dicho desplazamiento puede ser positivo o negativo, y ajustado en tiempo real desde GNU Radio. Como ejemplo final, se comprueba el efecto conjunto de todas las pertur- baciones consideradas: multicamino dado por (1), variación del ruido y de la ganancia del transmisor en tiempo real, así como desplazamiento en frecuencia. La Figura 7 muestra el resultado para QPSK sobre el canal (1), $E_b/N_0 = 15$ dB en el transmisor y un desplazamiento en frecuencia igual $0.02 v_s$. Nótese que para una velocidad de símbolo $v_s = 750 \text{ ksymb s}^{-1}$, dicho desplazamiento en frecuencia equivale a 15 kHz.

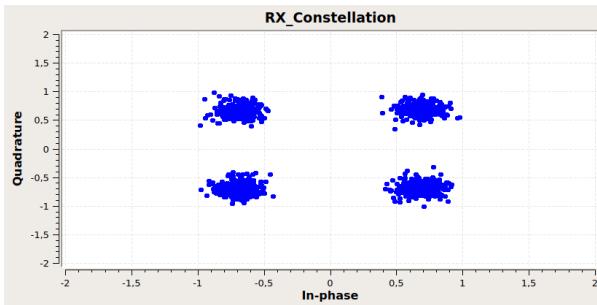


Fig. 7. Constelación recibida para una QPSK sobre $h[n]$ de (1), $E_b/N_0 = 15$ dB en el transmisor y desplazamiento en frecuencia $0.02 v_s$.

Como comparación final entre las distintas modulaciones, se pueden realizar experimentos basados en fijar los paráme- tros del canal a excepción de uno y comprobar los límites de este último. Por ejemplo, manteniendo las condiciones descritas en la Figura 7, pero variando la frecuencia de desplazamiento. Se observa que QPSK y 8-PSK soportan un desplazamiento de hasta $0.04 v_s$. Por su parte 16-QAM y 64- QAM soportan hasta $0.02 v_s$. En el caso de 64-QAM dicho límite se corresponde para un escenario sin multicamino, ya que para el canal $h[n]$ de (1) el desplazamiento en frecuencia soportado es despreciable. En general, se comprueba que, para unos niveles determinados en las perturbaciones, el efecto de degradación es mayor para las constelaciones más densas, lo que da lugar a una mayor probabilidad de error.

V. CONCLUSIONES

El empleo de dispositivos USRP manejados mediante SDR permite implementar sistemas reales muy flexibles en rela- ción a parámetros como son: la modulación empleada, la potencia transmitida, el ancho de banda, o la frecuencia de transmisión. Esto sumado a la posibilidad de añadir un canal simulado digitalmente permite la configuración de múltiples escenarios cuyo análisis resulta de interés desde un punto de vista docente, así como investigador. Además, el empleo de GNU Radio sobre Linux forma un tandem para el desarrollo de aplicaciones de SDR muy competitivo por ser de libre distribución y de código abierto.

La maqueta y los ejemplos de aplicación aquí propuestos pueden servir como base para el desarrollo de prácticas de comunicaciones digitales con dos objetivos principales. En primer lugar, facilitar al alumno el estudio y análisis de las características de las modulaciones digitales I/Q sobre un sistema de comunicaciones físico bajo distintos escenarios de propagación fácilmente parametrizables. En segundo lugar, familiarizar al alumno con el entorno GNU Radio, así como con el manejo básico de un USRP. La línea futura inmediata del presente trabajo sería por tanto la implementación de la maqueta propuesta en prácticas de laboratorio reales, con el fin de comprobar dichos resultados sobre el aprendizaje. Para ello se asumirían unos costes aproximados de 800 euros por cada dispositivo B200 [3], si bien se recuerda que otras opciones más baratas podrían ser adecuadas, siempre y cuando sean compatibles con GNU Radio [4].

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia, Innovación y Universidades bajo el proyecto TEC2017-86619 (ARTEINE) y por el Gobierno del Principado de Asturias/FEDER mediante el proyecto GRUPIN-IDI/2018/000191.

REFERENCIAS

- [1] GNU Radio, *About Software Defined and GNU Radio*. Fecha de consulta: 4 de Mayo de 2020 [Online]. Disponible en: <https://www.gnuradio.org/>
- [2] Ettus Research, *Ettus Research official page*. Fecha de consulta: 4 de Mayo de 2020 [Online]. Disponible en: <https://www.ettus.com/>
- [3] Ettus Research, *USRP B200 de Ettus Research*. Fecha de consulta: 4 de Mayo de 2020 [Online]. Disponible en: <https://www.ettus.com/all-products/ub200-kit/>
- [4] GNU Radio, *A Quick Guide to Hardware and GNU Radio*. Fecha de consulta: 4 de Mayo de 2020 [Online]. Disponible en: <https://wiki.gnuradio.org/index.php/Hardware>
- [5] GNU Radio, *GNU Radio Github repository*. Fecha de consulta: 4 de Mayo de 2020 [Online]. Disponible en: <https://github.com/gnuradio/gnuradio>
- [6] Ettus Research, *Building and Installing the USRP Open-Source Toolchain (UHD and GNU Radio) on Linux*. Fecha de consulta: 4 de Mayo de 2020 [Online]. Disponible en: [https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_\(UHD_and_GNU_Radio\)_on_Linux](https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux)
- [7] Ettus Research, *About USRP Bandwidths and Sampling Rates*. Fecha de consulta: 4 de Mayo de 2020 [Online]. Disponible en: https://kb.ettus.com/About_USRP_Bandwidths_and_Sampling_Rates
- [8] John G. Proakis, Masoud Salehi, *Digital Communications*, Mc Graw Hill, Fifth Edition, International Edition, 2008.
- [9] GNU Radio, *GNU Radio Wiki*. Fecha de consulta: 4 de Mayo de 2020 [Online]. Disponible en: https://wiki.gnuradio.org/index.php/Main_Page