



GNU Radio (GRC)

Profesor: Mgt. Ing. Rodolfo Salvatore

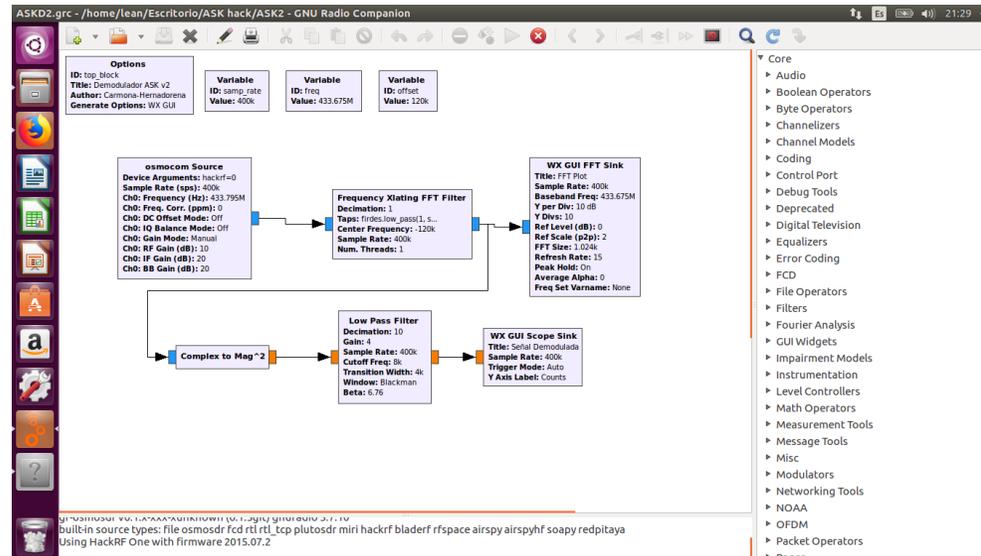
Ayudantes: Ing. Leandro Carmona

Ing. Ignacio Hernandorena

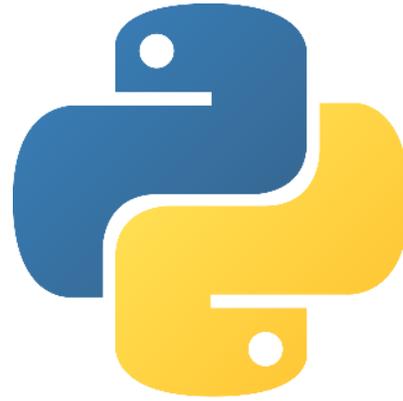
2021

GNU Radio.

- GNU Radio es un software de desarrollo que proporciona bloques de procesamiento de señales para la implementación de SDR.
- El entorno de trabajo del GNU Radio permite diseñar, simular y desarrollar sistemas reales de radio, tales como comunicaciones móviles, satelitales, sistemas de radar, GSM, LTE.



GNU Radio

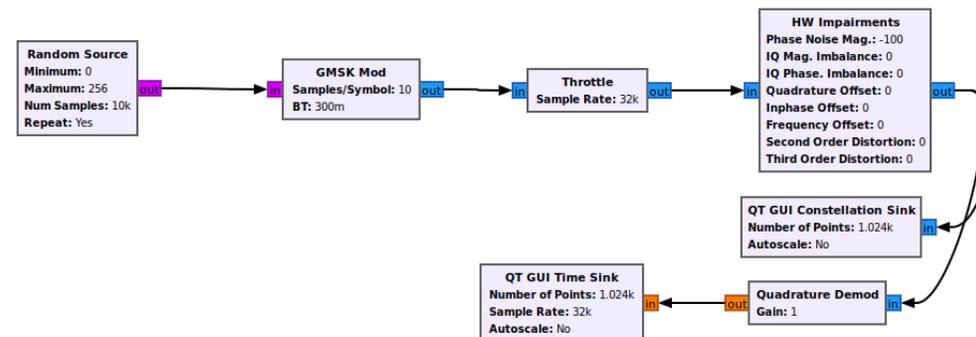
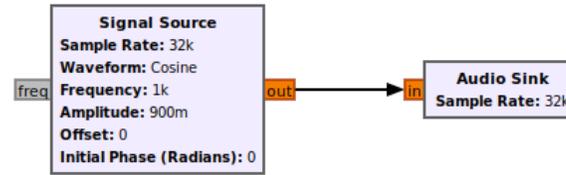


- C++ y Python son las dos capas principales que se utilizan en GNU Radio. Los códigos de alto nivel se escriben en Python y se usan como un lenguaje interpretable, principalmente realizan el trabajo de conectar los diversos bloques de señal para crear gráficos de flujo. Python no requiere ningún tiempo de compilación adicional, por lo que se utiliza principalmente para el funcionamiento rápido del gráfico de flujo.



Un enfoque modular

- El software proporciona un conjunto de bloques de procesamiento de señal que se pueden agregar para construir gráficos de flujo. Estos bloques están escritos en C++ y se ejecutan en Python, lo que brinda varias ventajas, como una fácil instanciación y conexión de bloques existentes y fácil creación de GUI (Interfaz gráfica de usuario). Alrededor de 250 bloques en GNU Radio incluyen varias aplicaciones como bloques simples de operaciones matemáticas, bloques moduladores/demoduladores, bloques de codificación de canales, códec de voz y otros. Los bloques de entrada / salida son una clase especial de bloques.



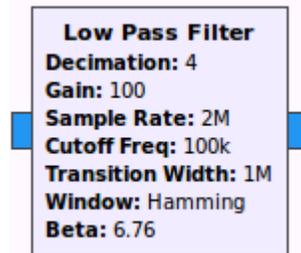
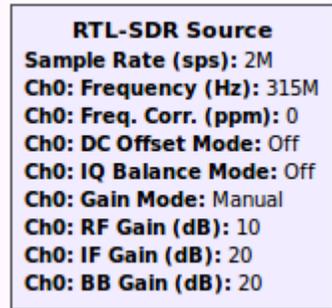
Bloques

- **Signal Source:** Es la fuente que nos entrega la señal que provee el SDR o un archivo. Por ejemplo, la fuente Osmocom, la fuente USRP y la fuente del archivo son tipos comunes de bloques signal source.
- **Signal Sink:** El sink se usa al final del diagrama de flujo. Por ejemplo, el USRP Sink, el sink de archivos son tipos comunes de bloques sink.

RTL-SDR Source
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 315M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

File Sink
File: /usr/codigo
Unbuffered: On
Append file: Overwrite

Bloques



- **RTL-SDR Source**

Representa un dispositivo SDR genérico que provee las muestras de entrada. Puede ajustarse la tasa de muestreo, la frecuencia del mezclador, las ganancias de RF/IF, etc.

- **Filtro FIR**

Puede ser pasa bajos, pasa altos, pasa banda, etc. Se configura el tipo de ventana (aproximación al filtro ideal), la frecuencia de corte, la ganancia de la banda pasante, y un factor de decimación.

- **Throttle**

Limita el flujo de datos que llegan a los diferentes sinks.

Bloques

- **Conversor Complejo/Real**

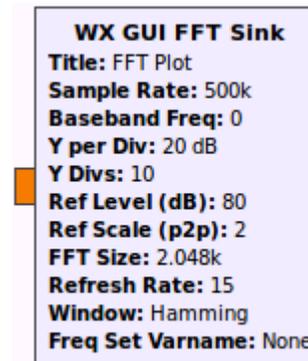
GNU Radio trabaja con varios tipos de datos; el SDR provee datos en forma de canales IQ, que mediante este bloque se transforman en valores de punto flotante.

- **Valor absoluto**

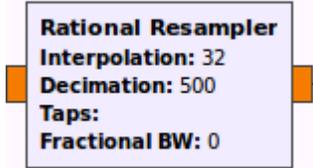
Recupera el valor absoluto o módulo de la señal (análogo a un puente rectificador ideal)

- **Analizador de espectro**

Muestra en una interfaz gráfica el espectro resultante de aplicar la Transformada Rápida de Fourier (FFT) a la señal de entrada. Se puede configurar el span, el nivel de referencia, el tamaño de la FFT, etc. tanto desde el bloque como desde la interfaz gráfica



Bloques



- **Resampler**

Un resampler es un bloque propio del procesamiento digital cuya función es cambiar la tasa de muestras por segundo que circulan. Para un Resampler Racional como el de la figura, se especifican dos valores:

Decimación: El resampler submuestra la señal según este valor. Por ejemplo, para un factor de decimación de 500, se toma 1 muestra cada 500 recibidas

Interpolación: Luego de la decimación, el resampler efectúa una interpolación lineal para reconstruir parcialmente la señal. Por ejemplo, para un factor de interpolación de 32, el resampler regenera 32 puntos entre dos submuestras consecutivas.



- **Comparador con Histéresis**

Funciona de forma análoga a un Schmitt Trigger: si la señal supera el umbral High, genera una salida en 1. Sin embargo, la salida no vuelve a 0 hasta que la señal no caiga por debajo del umbral Low.

Bloques

- **Osciloscopio**

Muestra la señal en función del tiempo a través de una interfaz gráfica, de modo similar al Analizador de Espectro. Las escalas pueden ajustarse tanto desde el bloque como desde la interfaz.

- **Agregado de constante**

Suma un valor constante a la señal.

- **Convertor real/char**

Transforma un valor entero a un número de 8 bits que representa un ASCII.

- **Salida a archivo**

Las muestras pueden guardarse en un archivo para su posterior representación y análisis.

