Manual de Instalación y uso de GNURadio

Realizado Por:
Ricardo M. Prieto

Temario:

A) Instalación de GNU/Radio y UHD
B) Entorno GNU/Radio
C) Tipos de Datos
D) Bloques iniciales
E) Modo inteligente de uso de variables
F) Visualización de Datos
G) Filter Design
H) Instalar Firmware en USRP
I) Configurar bloques del USRP
J) Conclusiones

A) Instalacion de Gnu-Radio y UHD

Primero debemos instalar Gnu-Radio ya que es el entorno de desarrollo y el que contiene las librerías para desarrollar nuestras aplicaciones en telecomunicaciones.

Por otra parte UHD es el driver necesario para comunicarnos con el dispositivo USRP, sin el cual no se pueden realizar las practicas en un ambiente real de comunicaciones.

Para esto utilizaremos un sistema operativo GNU/Linux basado en Debian, este manual debe ser por lo menos compatible con todos los sistemas ramificados de Debian. (Ubuntu, Linux Mint, Kali Linux, XBMC, gNewSense, entre otras)

1. Instalamos la librería pip para python
   sudo apt install python-pip
2. Actualizamos la librería pip
   sudo pip install -U pip
3. Instalamos PyBOMBS
   sudo pip install PyBOMBS
4. Iniciamos el prefijo por defecto
   sudo pybombs prefix init /usr/local -a usrlocal
5. Configuramos el prefijo
   sudo pybombs config default_prefix /usr/local
6. Agregamos los recipes desde github
   sudo pybombs recipes add gr-recipes git+https://github.com/gnuradio/gr-recipes.git
   sudo pybombs recipes add gr-etcetera git+https://github.com/gnuradio/gr-etcetera.git

7. Instalamos UHD y GNU-Radio finalmente
   sudo pybombs install uhd gnuradio
   sudo ldconfig

Una vez instalado Gnu-radio y UHD vamos a configurar UHD

1. Para asegurarnos que los drivers esten actualizados
   sudo "/usr/local/lib/uhd/utils/uhd_images_downloader.py"
2. Para activar la programacion en tiempo real debemos setear permisos
   nano /etc/security/limits.conf
3. En ese archivo ingresamos la siguiente linea
   @usrp - rtprio 99
4. Ademas en necesario crear un grupo usrp
   sudo groupadd usrp
   sudo adduser <usuario> usrp
5. <usuario> es el nombre del usuario que desea que use el USRP
6. Reiniciamos el sistema y tecleamos
   su - <usuario>

Ahora tenemos listo el uso para USRP para nuestro uso

B) Entorno de Gnu-Radio

Gnu radio nos permite trabajar mediante 2 maneras:

1) **Mediante lenguaje de programación**, este puede ser en Python (más común) o en C++, esta es una excelente opción cuando ya se conoce como utilizar las librerías ofrecidas por gnuradio, configurarlas y ademas se conoce, los principios de una programación orientada a objetos, permitiendo tener mayor control por parte del usuario, ademas que un mejor desarrollo de herramientas y objetos propios según requiera la investigación.

2) **Mediante un modo grafico** ofertado por la misma plataforma, es el modo mas básico de uso y es en el que se centrará principalmente el presente documento. El funcionamiento más simple es la conexión de

bloques, cada bloque representa un comportamiento distinto, y pueden ser colocados de manera paralela o serial.

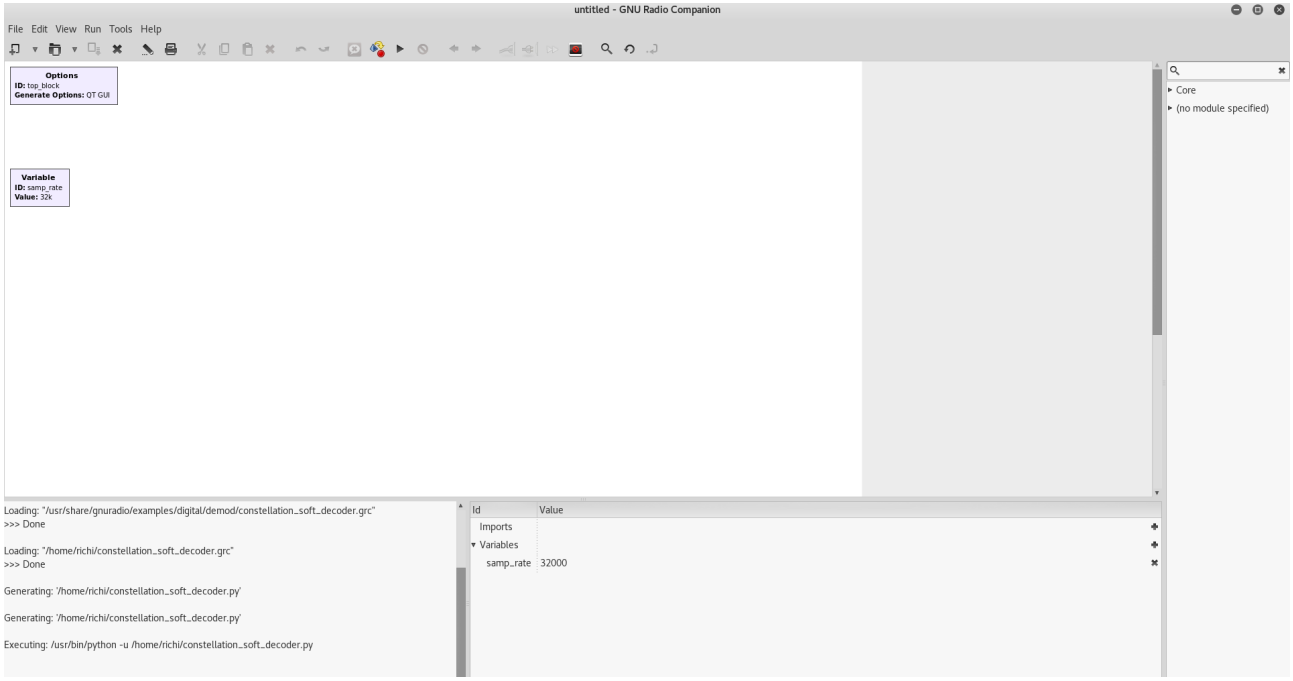

Fig1. Entorno Grafico Gnu-Radio

El entorno Grafico se compone de las siguientes partes principales

- **Canvas** aquí es donde se colocan los bloques de Gnu-radio Fig2

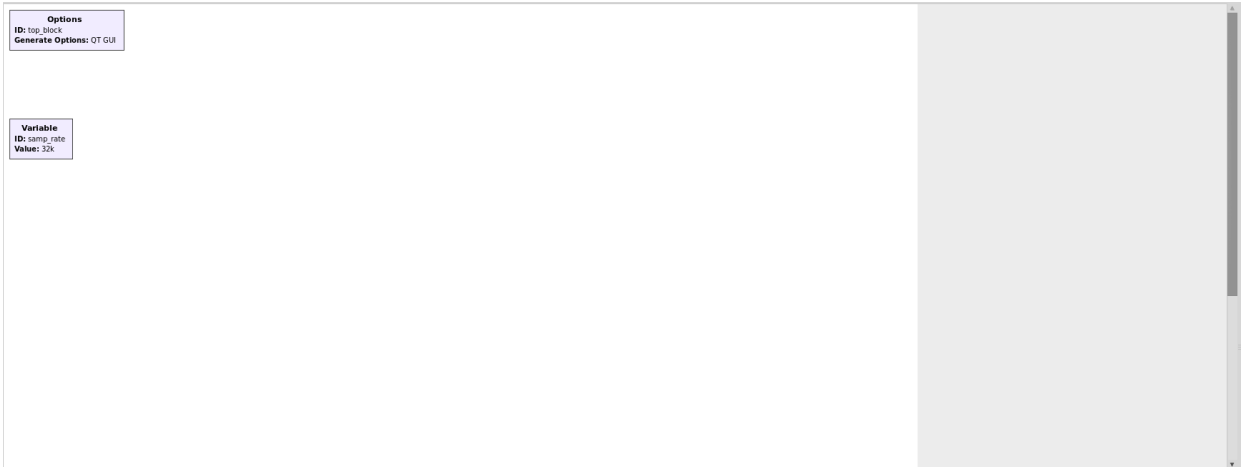

Fig2. Canvas

- **Consola** Es donde se muestran los avisos de ejecución del software que se esta desarrollando Fig3

Fig3. Consola

- **Entorno de Variables** donde se observa los valores que tienen o van adquiriendo las diferentes variables que utilizamos en nuestros desarrollos Fig4.

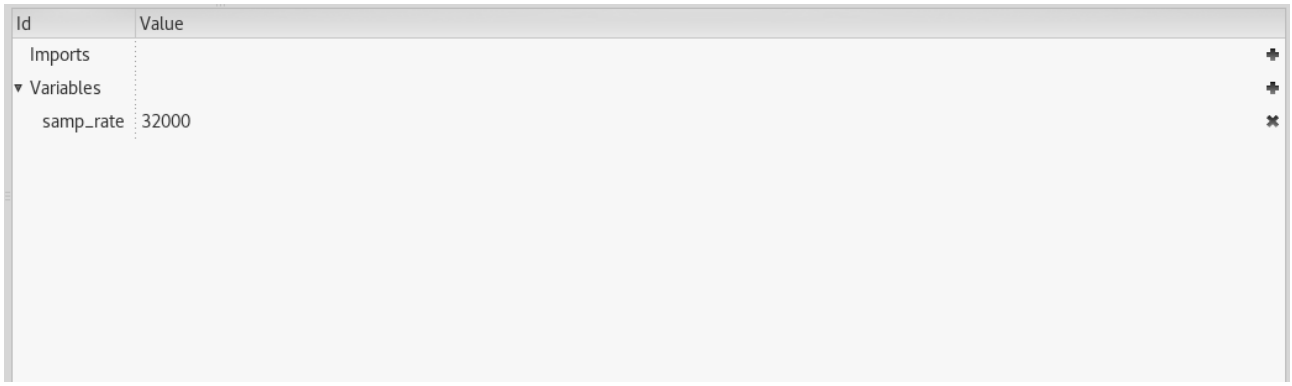

Fig4. Entorno de Variables

- **Entorno de bloques** es aquí donde encontramos todos los bloques que necesitamos para construir nuestro sistema de telecomunicaciones. Desde aquí arrastramos el bloque hasta el canvas Fig5.

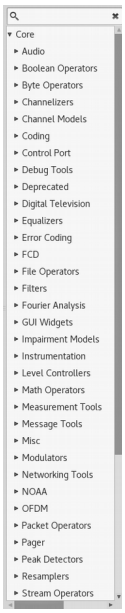

Fig5. Entorno de Bloques

- **Barra de herramientas** permite hacer ciertas modificaciones al programa y a los bloques Fig6a.

Fig6a. Barra de herramientas

De la barra de herramientas lo mas importante son los siguientes iconos.

- **Error flowgraph** Nos muestra notificaciones en donde se encuentra el error en nuestros bloques Fig6.

Fig6. Error flowgraph

- **Generate flowgraph** Nos permite generar un archivo, que sera el encargado de la ejecución se puede comparar a un compilar Fig7.

Fig7. Generate flowgraph

- **Run** Es el encargado de la ejecución del programa una vez este haya sido compilado sin errores Fig8.

Fig8. Run

- **Desconectar** muchas veces es necesario desconectar un bloque sin eliminarlo para probar el funcionamiento del programa Fig9. Si el bloque esta realmente desconectado se ve como la Fig10, cabe recalcar que para desconectar un bloque este debe estar en paralelo al flujo caso contrario nos dará un error por la no continuidad del flujo de datos.

Fig9. Desconectar

Fig10. Ejemplo de bloque desconectado.

- **Conectar** para conectar un bloque previamente desconectado se utiliza el siguiente botón Fig11. Al volver a conectarse el bloque debe de color normal, no de color gris Fig 12.

Fig11. Conectar

Fig12. Ejemplo de bloque conectado.

- **Bypass** muchas de las veces los bloques están en serie pero no deseamos eliminarlo, solo queremos probar nuestro diseño sin ese bloque, para esta opción podemos utilizar el botón bypass Fig13. Una vez utilicemos este bloque, el mismo debe de ponerse color amarillo Fig14.

Fig13. Bypass

Fig14. Bloque realizado bypass

C) Tipos de datos

Gnu Radio posee la capacidad de trabajar con diferentes tipos de datos según sea la necesidad de desarrollo:

- Byte - 1 byte of data (8 bits per element) Fig15.
- Short - 2 byte integer Fig16.
- Int - 4 byte integer Fig17.
- Float - 4 byte floating point Fig18.
- Complex - 8 bytes Fig19.

Cada tipo de datos vienen definidos por diferentes colores en los extremos de los bloques:

Fig15. Byte

Fig16. Short

Fig17. Int

Fig18. Float

Fig19. Complex

D) Bloques iniciales

Al iniciar un nuevo proyecto, siempre inician dos bloques, el primero es de configuración y el segundo de una variable que se denomina samp_rate (sample rate) Fig20.

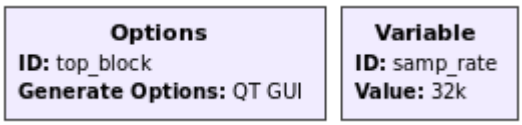

Fig20. Bloques iniciales

- **Options** el bloque options tiene opciones de configuración inicial Fig21. En el mismo se pueden configurar los siguientes parámetros
  - ID: Identificador del bloque es un nombre único asignado a cada bloque
  - Title: Titulo del proyecto
  - Author: Nombre del autor del proyecto
  - Description: Descripción del proyecto
  - Canvas Size: Tamaño de nuestra área de trabajo
  - Generate Options: Formato de salida del modo grafico cuando generemos nuestro proyecto
  - Run: Manera en la que arrancar nuestro proyecto
  - Max Number of Output: máximo número de salidas
  - Realtime Scheduling: Activar o desactivar la programación en tiempo real. (se debe tener configurado para activar esta opción)
  - QSS Theme: El tema grafico de nuestro entorno

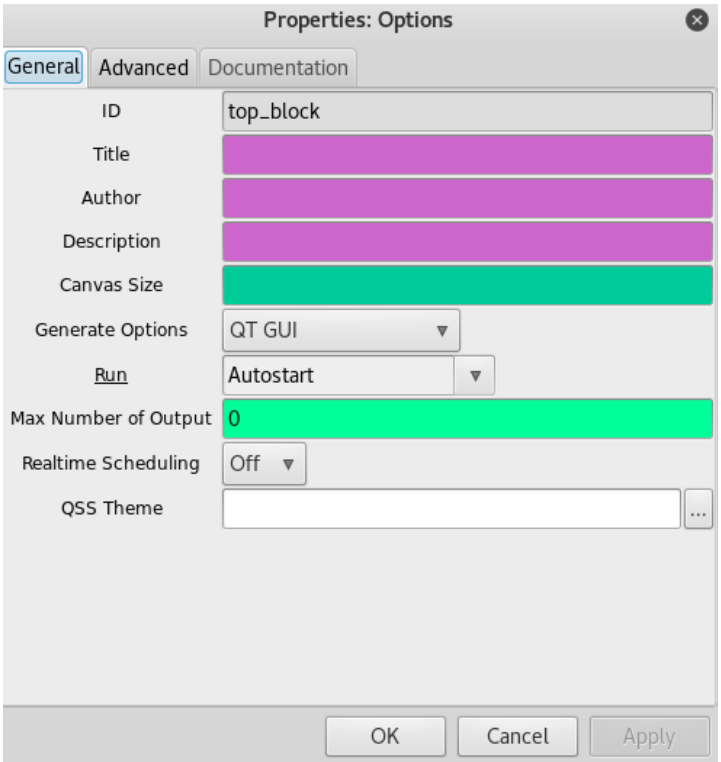

Fig21. Entorno de options

- **Variable** en este bloque vamos a configurar el parámetro inicial de nuestro proyecto, el mismo que es la taza de muestreo, por default viene a 32k, pero puede aumentarse o disminuirse al gusto Fig22. Tiene 2 valores editables

    - ID: identificador único para el bloque
    - value: un valor entero que contiene la taza de muestreo

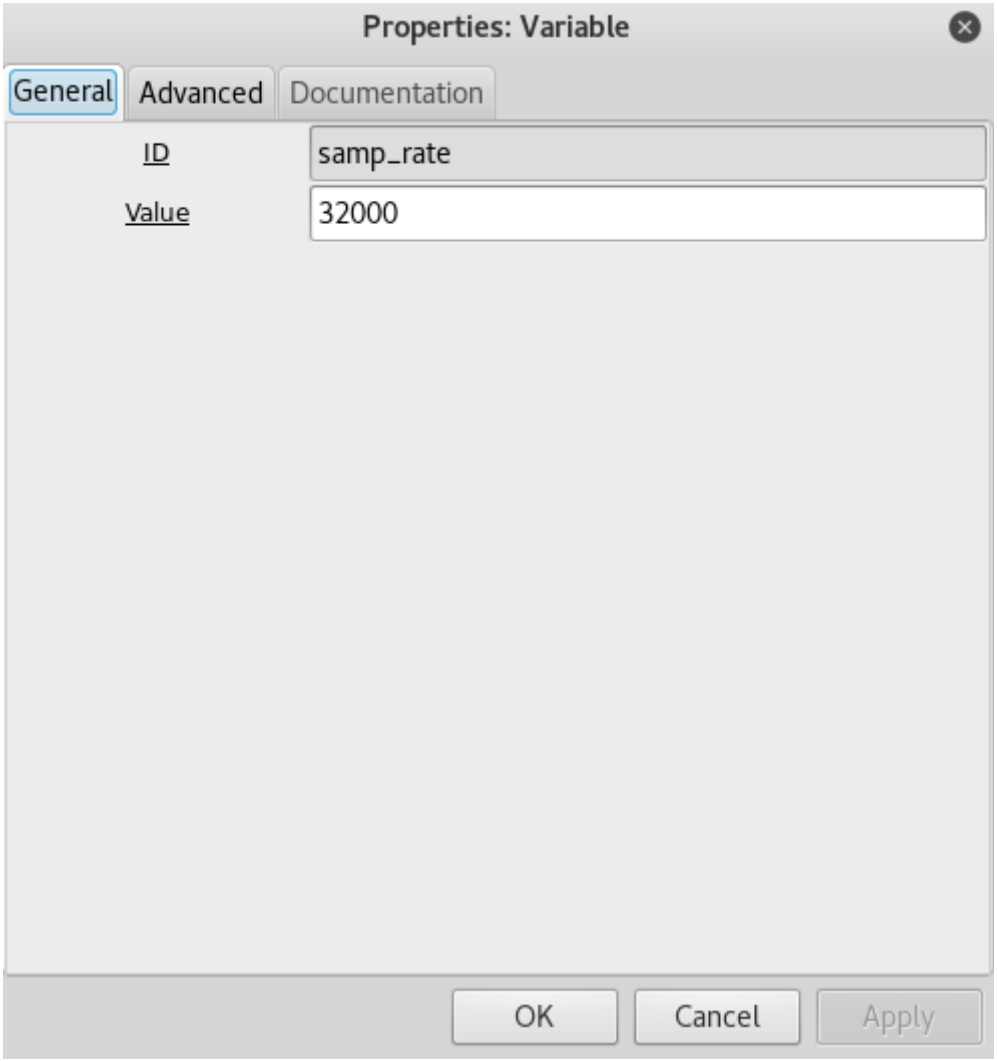

Fig22. Variable samp_rate

E) Modo inteligente de uso de variables

Cuando se esta trabajando en proyectos es muy común tener que cambiar en medio de las pruebas los valores de algunas variables, el problema surge cuando se utiliza una valor para múltiples bloques, cambiar el numero a mano es una opción pero no es la mas óptima, para esto utilizamos los bloques variables, de esta manera al cambiar el valor de ese bloque todos los valores asignados a ese ID único serian reemplazados por el nuevo valor. Veamos un ejemplo:

Primero vamos a ir a la dirección en donde se encuentra el bloque Variable Fig23.

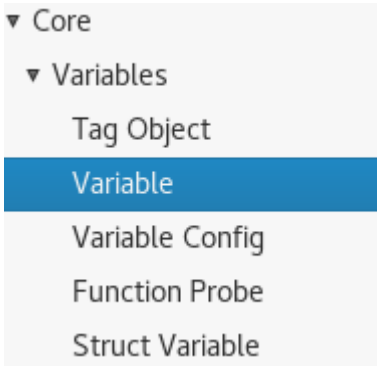

Fig23. Dirección bloque Variable

Hacemos click sobre variable y la llevamos a nuestro canvas

Una vez realizado esto, nos saldrá un bloque similar al de la Fig24.

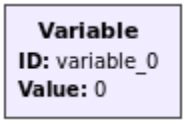

Fig24. Bloque variable

Ahora hacemos doble click en el bloque y editamos el mismo, en el ID le cambiamos por un identificador que recordemos, en este caso lo llamaremos muestreo. Y en el cuadro Value vamos a colocar el valor que deseemos. Fig25.

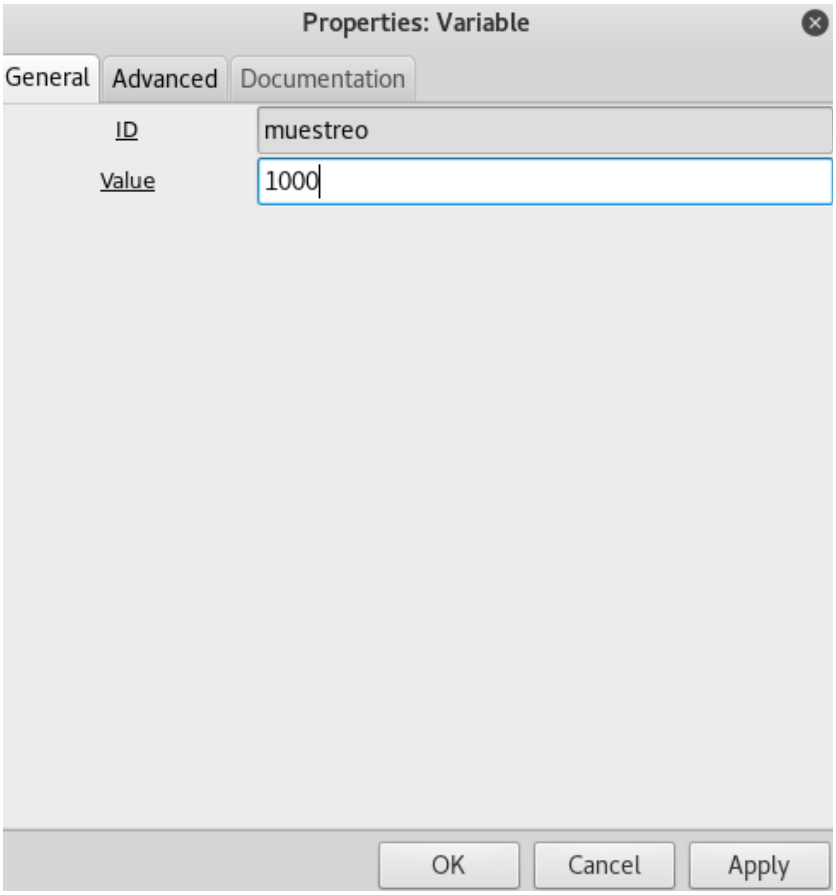

Fig25. Variable muestreo con valor de 1000

Para ejemplificar se mostrara varios bloques escogidos al azar y se les colocara el ID “muestreo” en su configuración, cambiando su propiedad Sample Rate. Fig26.

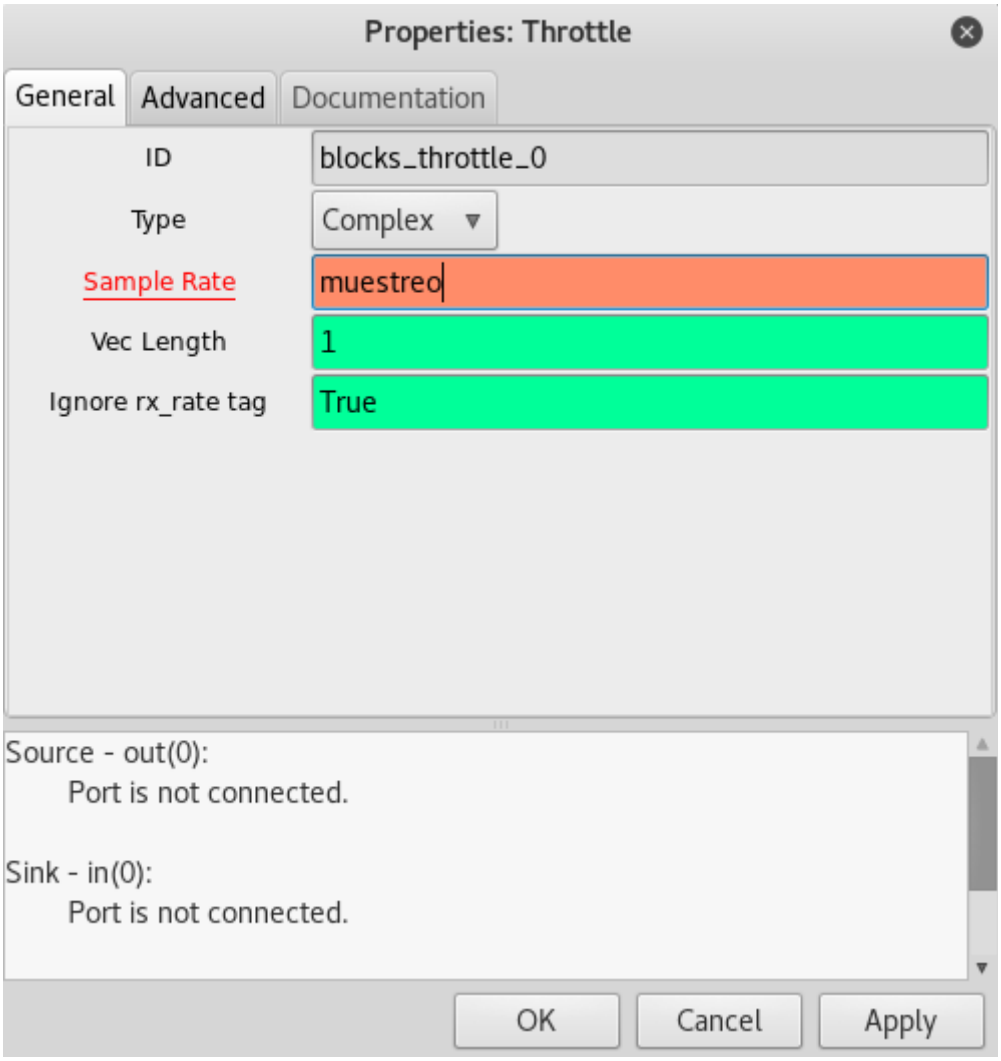

Fig26. Cambiando la propiedad sample rate por el ID de nuestra variable “muestreo”

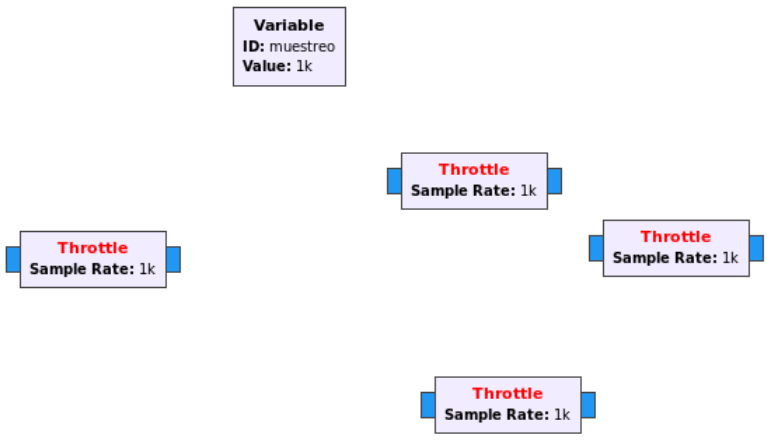

Fig27. Bloques Throttle asignados en Sample Rate la variable Muestreo

Ahora bien procedemos a cambiar el valor de la variable con ID “muestreo” la misma que cambiara a todos los bloques su valor, por e nuevo asignado Fig28.

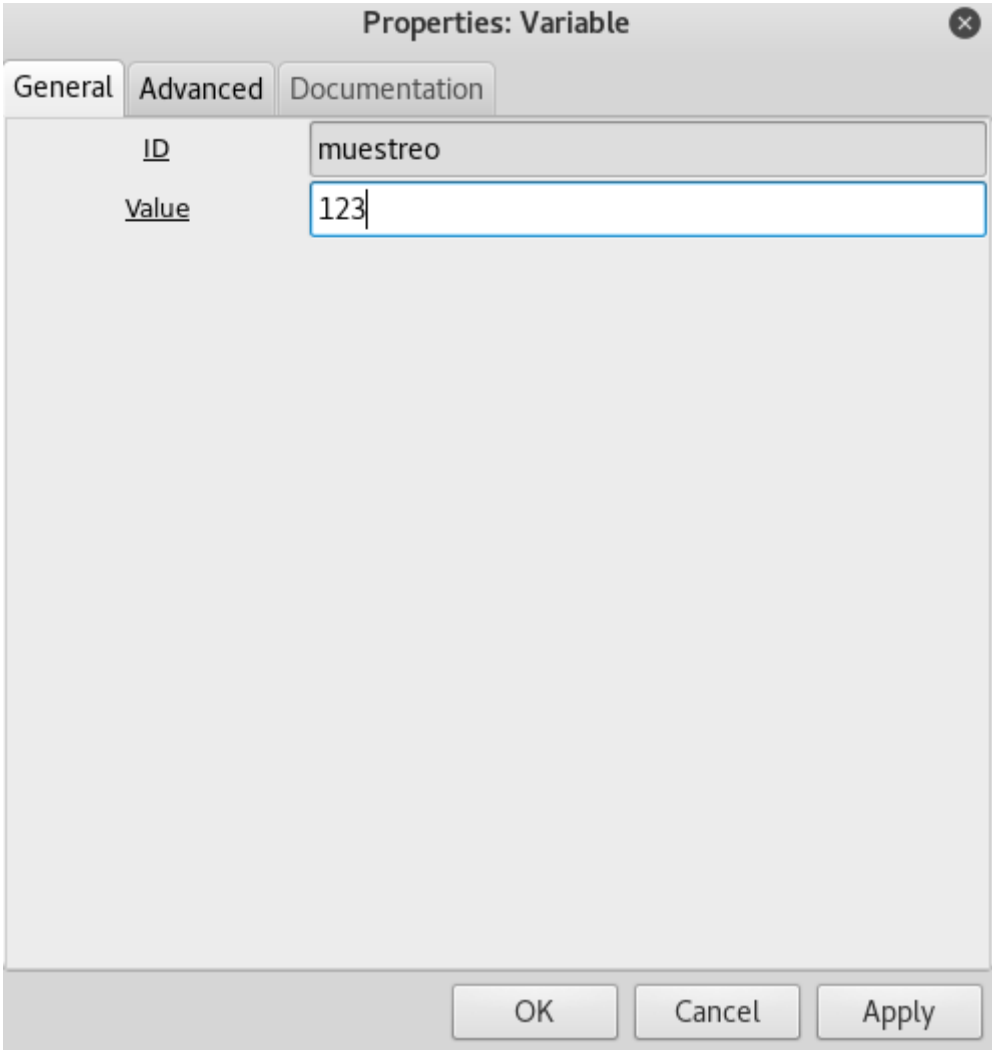

Fig28. Cambio de valor de la variable “muestreo”

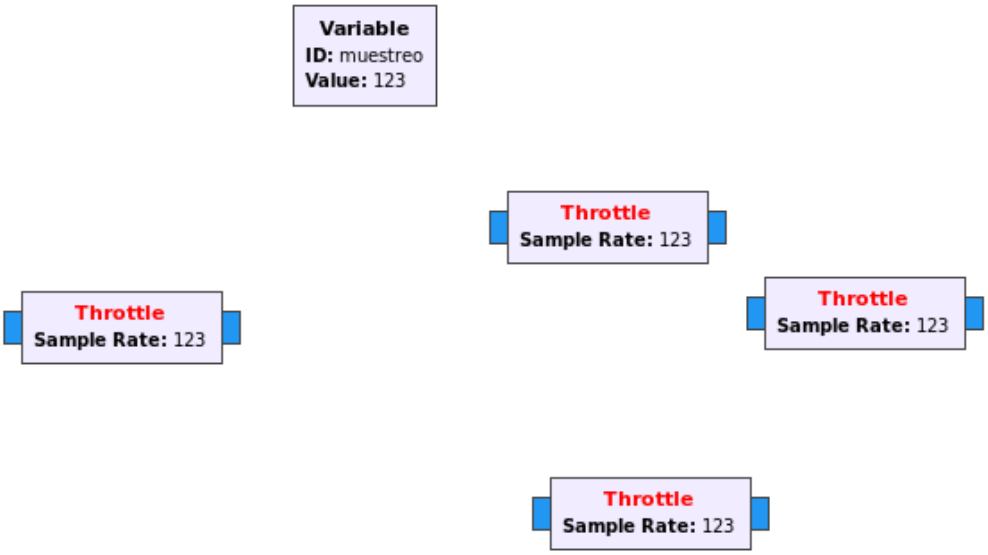

Fig29. Todos los bloques cambiaron instantáneamente por el nuevo valor.

F) Visualización de datos

La mayoría de las veces nosotros deseamos ver las graficas de las señales que estamos adquiriendo y tratando, Gnu-Radio nos trae una gran cantidad de herramientas distintas y fáciles de utilizar para visualizar o tener un control visual de las variables que estamos utilizando un proyecto.

Veamos un ejemplo.

Es un ejemplo simple, utilizamos una fuente cosenoidal conectada a una salida de audio, esto nos genera una frecuencia que ha sido configurada a 250Hz con una amplitud de 0.5 Fig30. Al iniciar el script, vamos a observar una ventana en blanco, y escucharemos el sonido que generamos.

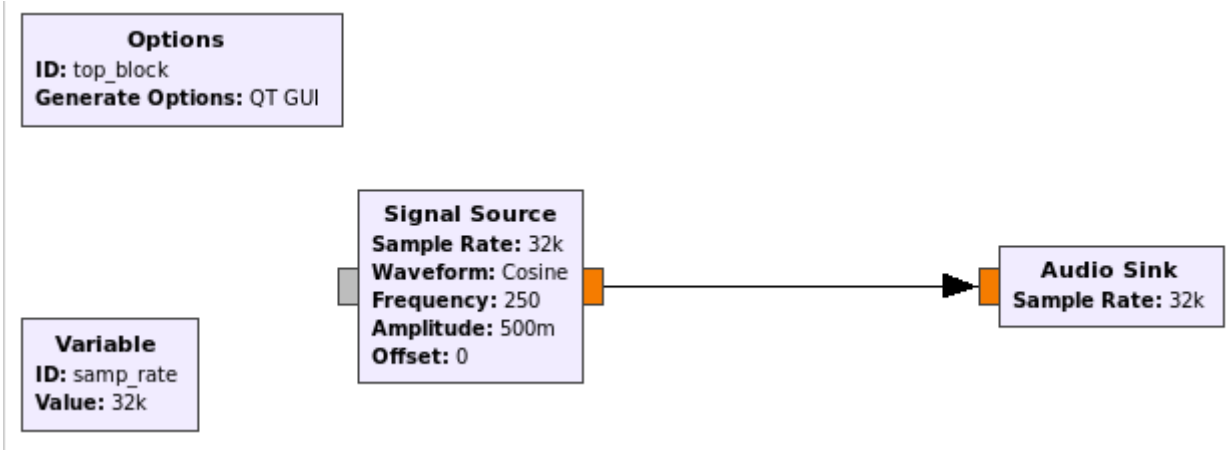

Fig30. Ejemplo de generación y escucha de una señal Cosenoidal

Hay que tomar en cuenta en el bloque Options, el modo grafico que esta aceptado, vemos en Generate Options y dice QT GUI, por lo tanto todo nuestros visualizadores deben ser del tipo QT, ya que existen otros como por ejemplo WX, esto lo cargamos dese nuestro entorno de bloques Fig31.

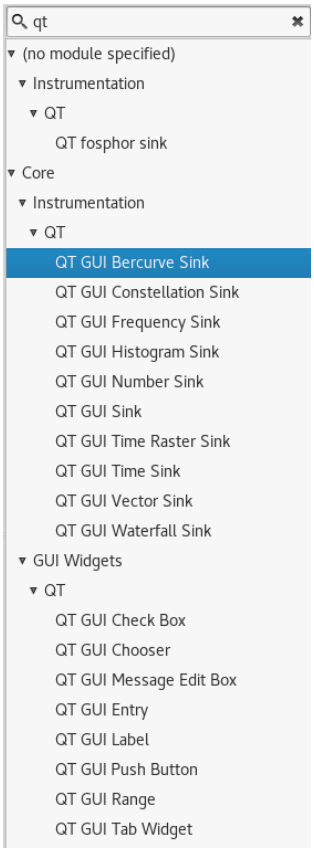

Fig31. Entorno de Bloques con las opciones QT

Ahora vamos a proceder a insertar algunos bloques.

- **QT GUI Frequency Sink** este bloque nos permite visualizar la transformada de fourier de nuestra señal, solo agregamos el bloque, configuramos el tipo de dato de entrada, en este caso float y le damos guardar Fig32.

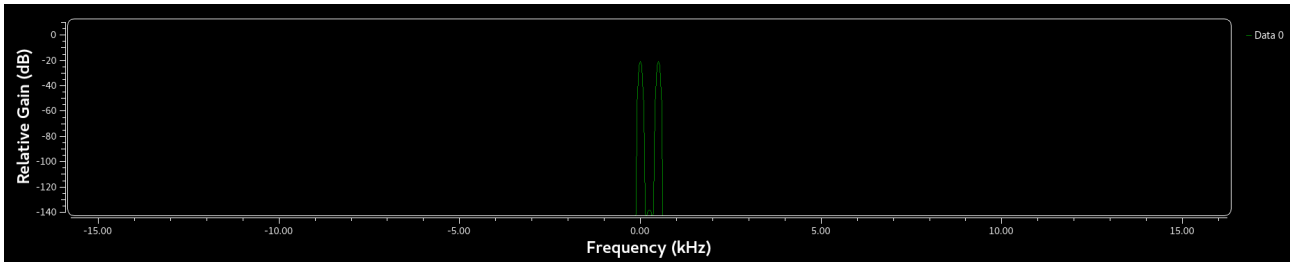

Fig32. QT GUI Frecuency Sink

- **QT GUI Time Sink** este bloque nos permite ver el comportamiento en el tiempo de nuestra señal, el procedimiento de configuración el el mismo en todos los casos Fig33.

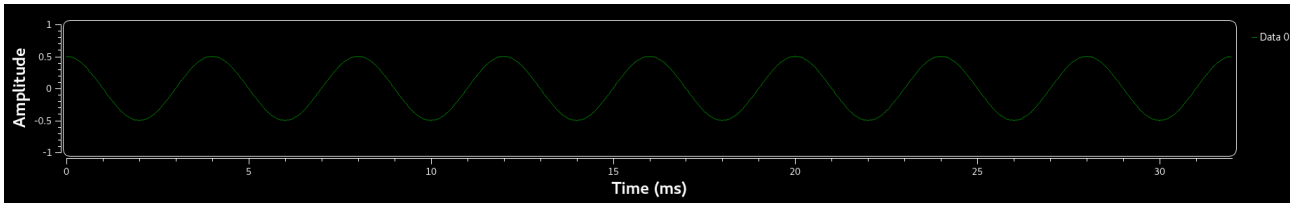

Fig33. QT GUI Time Sink

- **QT GUI Waterfall Sink** este bloque nos permite una visualización del tipo waterfall de nuestra señal. Fig34.

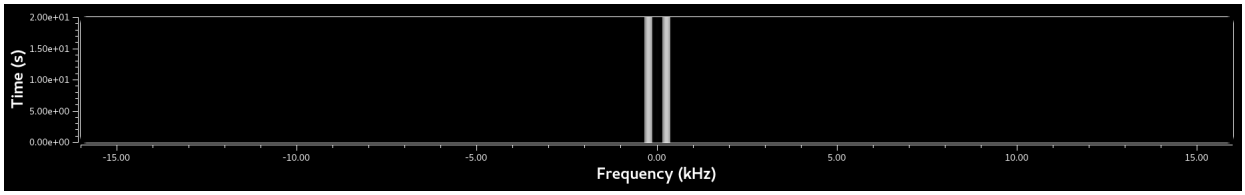

Fig34. QT GUI Waterfall Sink

Existen múltiples otras opciones pero estas son las más importantes, el funcionamiento de las demas, muy similar. El diagrama de bloques quedaría de una manera similar a esta Fig35.

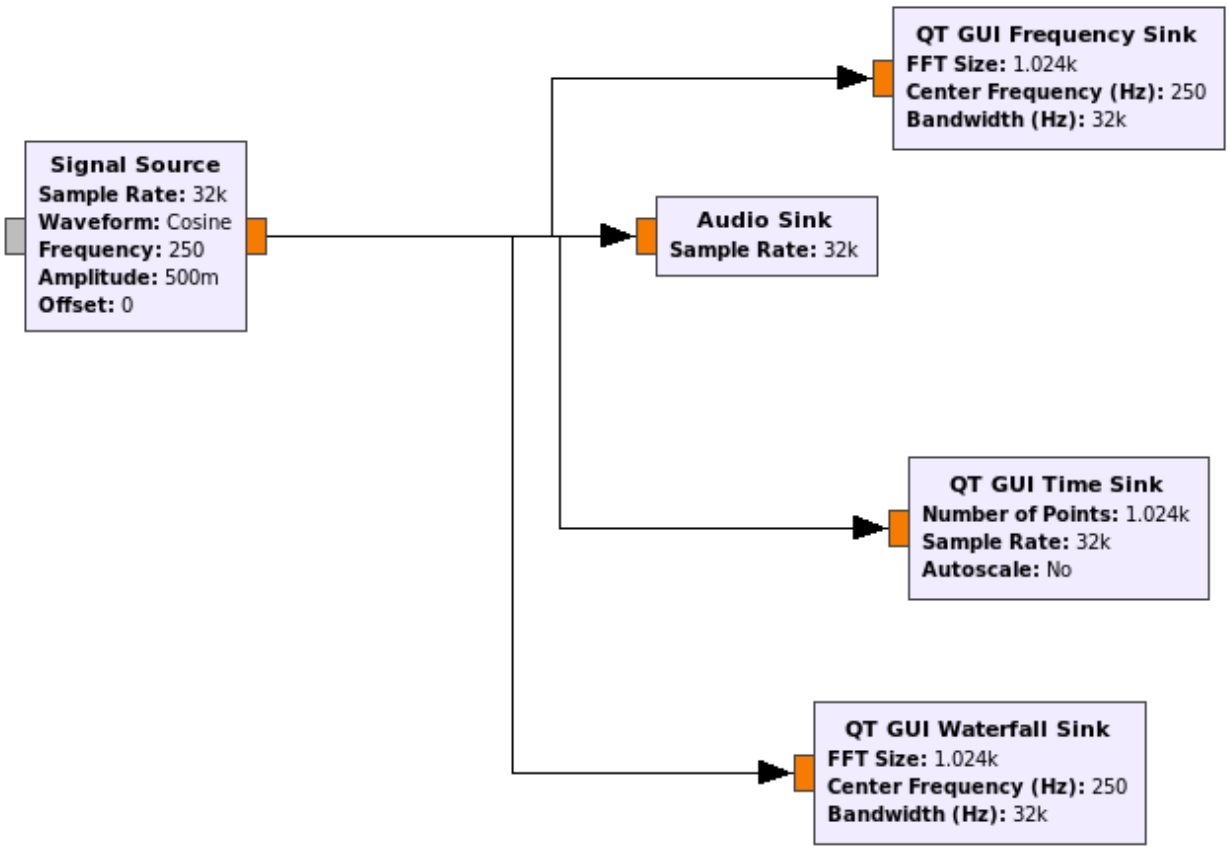

Fig35. Diagrama de Bloques

En cuanto a la posición de los elementos, el diagrama de bloques se acomoda automáticamente, esta es la posición de nuestros elementos conjuntos Fig36.

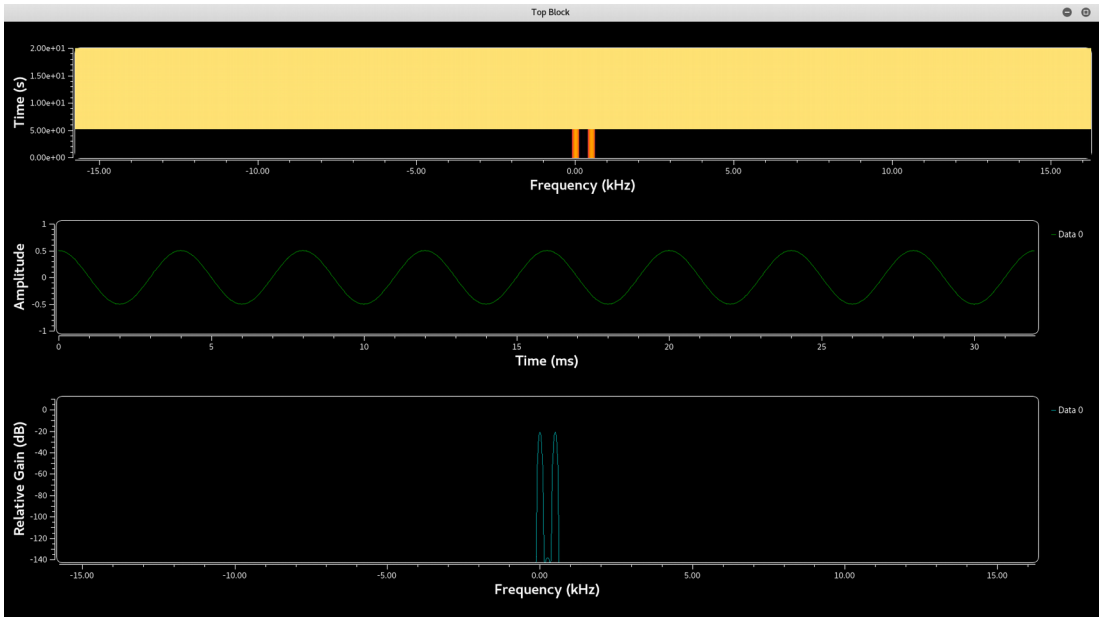

Fig36. Posición de los elementos acomodados automáticamente

Para utilizar por ejemplo WX se debe cambiar en el bloque Options la opcion QT Gui por la opcion WX GUI y todos los bloques QT reemplazarlos WX, en la Fig37. se observa como cambia el modo grafico dentro del proyecto que estamos realizando pero los valores, naturalmente serán los mismos.

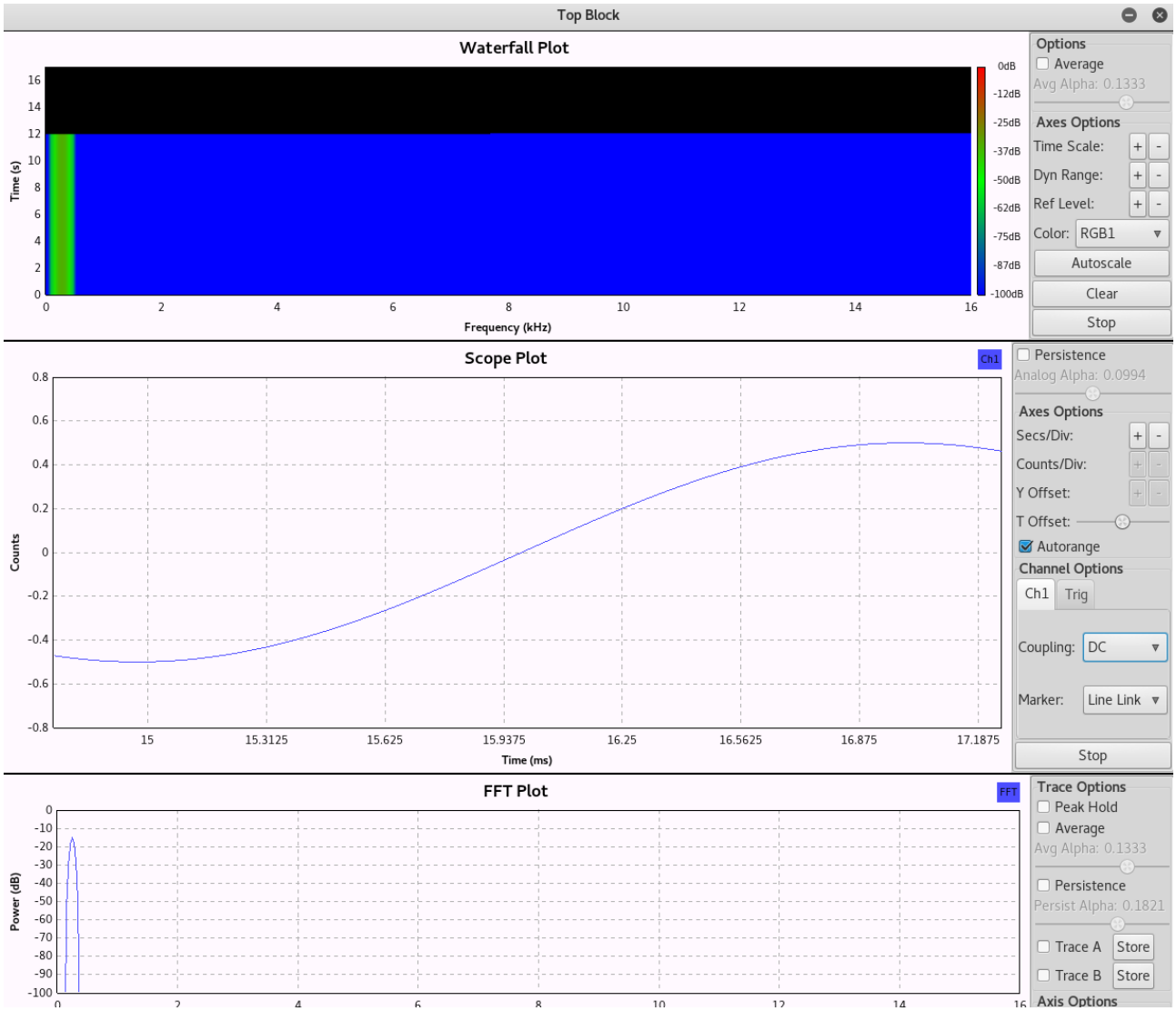

Fig37. Bloques en formato WX

Otra de las opciones interesantes dentro de Gnu Radio, es que permite cambiar variables, en tiempo de ejecución, esto quiere decir que mientras estamos probando nuestro proyecto, podemos ir variando sus valores mientras lo estamos ejecutando, sin la necesidad de detener nuestra programa, cambiar el valor de la variable y volver a inicia. Para esto utilizaremos el Bloque QT GUI Range Fig38.

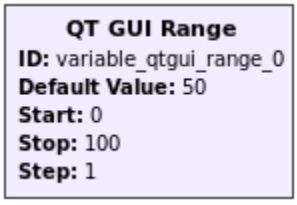

Fig38. QT GUI Range

En este bloque nos permite configurar una serie de opciones, a continuación veremos las mas importantes Fig39.

- ID: Como ya hemos visto anteriormente la opción más importante, el nombre de identificación unico dentro del proyecto.
- Default Value: Valor de arranque de la variable
- Start: Valor mínimo que tomará la variable
- Stop: Valor máximo que tomará la variable
- Step: El tamaño de los pasos en la variable, ejemplo: 1 en 1, 2 en 2, 5 en 5.
- Widget: El estilo grafico que queremos visualizar

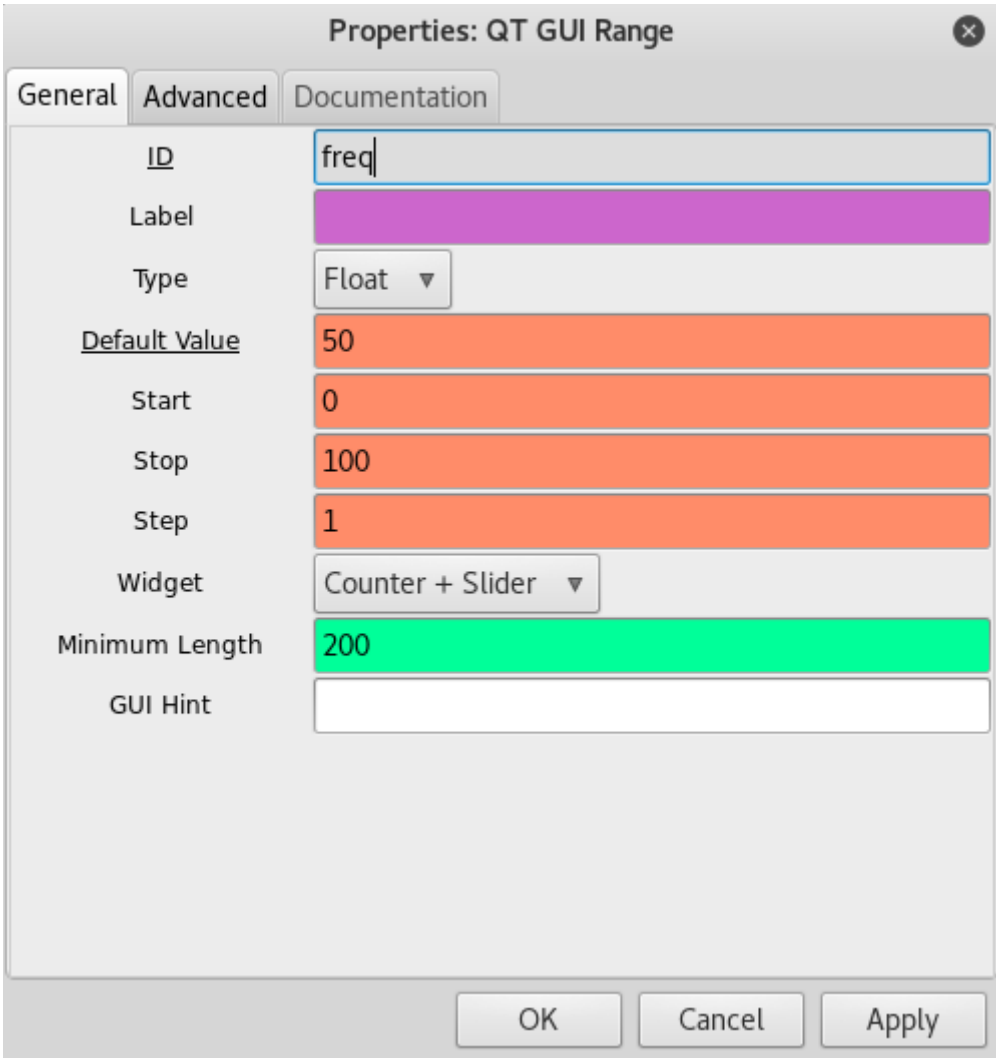

Fig39. Configuración del bloque QT GUI Range

Una vez configurados nuestros parametros corremos el script anterior y comenzamos a variar la frecuencia en tiempo de ejecución Fig40., Fig41

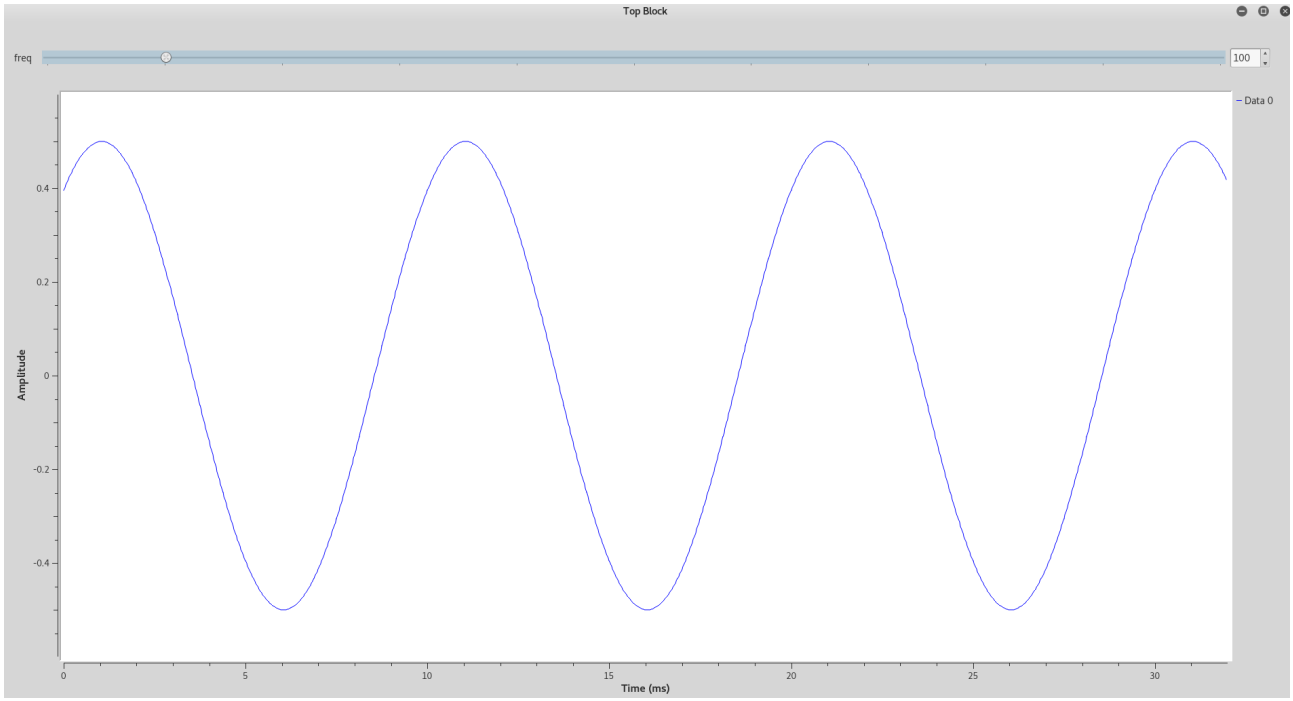

Fig40. Inicio de frecuencia en tiempo de ejecución

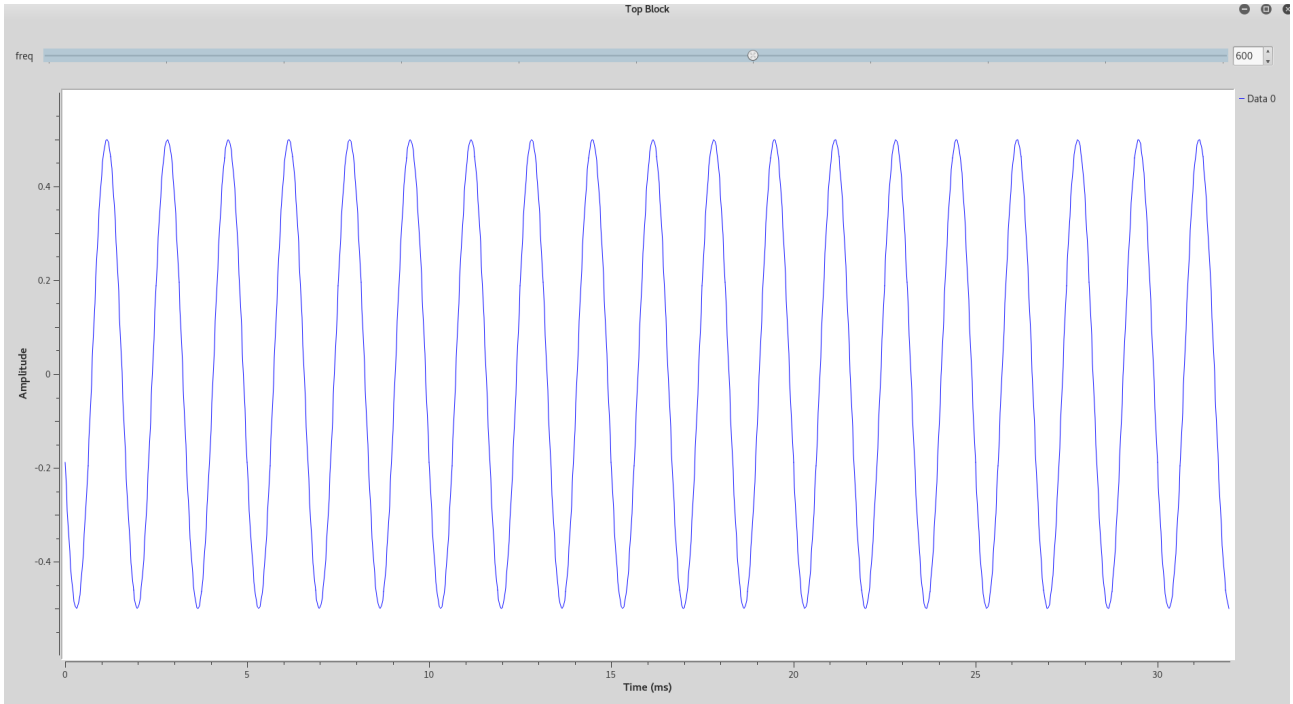

Fig41. Cambio de frecuencia en tiempo de ejecución

G) Filter Design

Los filtros digitales son de suma importancia, por que son utilizados con frecuenca en los sistemas de telecomunicaciones, estos tienen como función, manipular y modificar el espectro de frecuencia de la señal de entrada para obtener en la salida la función que se requiera aplicar al sistema.
Gnu-Radio tiene la opcion de diseñar filtros desde la misma plataforma, para utilizarlo basta con ir a la opción Tools –> Filter Design Tool Fig42.

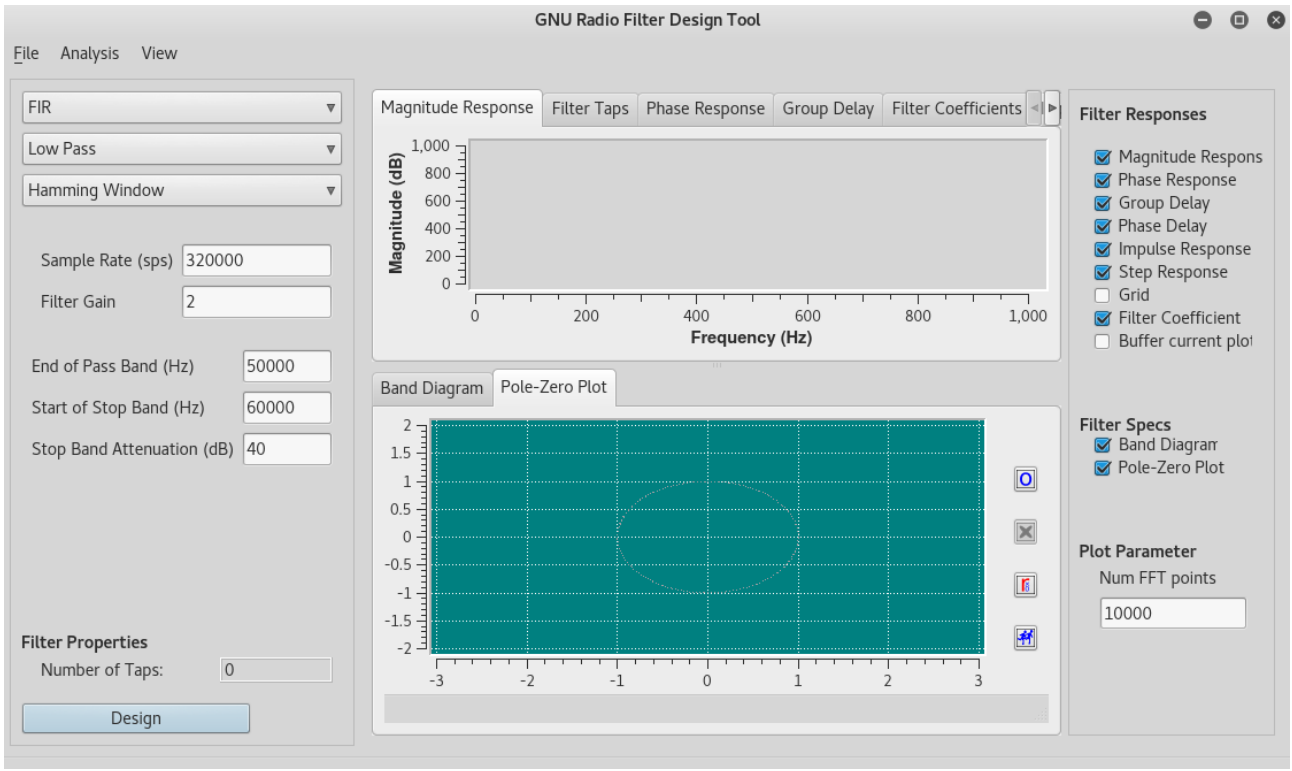

Fig42. Imagen de la herramienta de diseño del filtro

A continuación vamos a ver las opciones más interesantes que nos trae esta herramienta.
Primero nos permite elegir entre dos tipos de Filtros:

- **Filtros FIR:** Filtro de respuesta finita al impulso
- **Filtros IIR:** Filtro de respuesta infinita al impulso

Ademas de eso podemos escoger entre distintos tipos de filtro

- **Pasa Bajo:** se define como el termino de la banda de paso, el inicio de la banda de corte, el ripple en la banda de paso y la atenuación den la banda de corte
- **Pasa Banda:** define el primer banda de corte, el inicio de la banda de paso, el fin de la banda de paso, el inicio de la segunda banda de corte, el ripple en la banda de paso, la atenuación en la banda de corte.
- **Elimina Banda:** define el fin de la primera banda de paso, el inicio de la banda de corte el fin de la banda de corte, el inicio de la segunda banda de paso, el ripple en la banda de paso y el corte en la banda de atenuación.
- **Pasa Alto:** define el fin de la banda de corte, el inicio de la banda de paso, el ripple en la banda de paso y el corte en la banda de atenuación.

También tenemos otras opciones:

- **Sample Rate (sps):** Valor de la taza de muestreo

- **Filter Gain:** Ganancia del filtro
- **Start of Pass Band:** Inicio de la banda de paso en Hz
- **End of pass band:** Fin de la banda de paso en Hz
- **Transition Width:** Tamaño de la transición en Hz
- **Atenuación en la banda de corte:** La atenuación que deseamos en dB

Una vez hemos configurado los parámetros nos diseña el filtro Fig43.

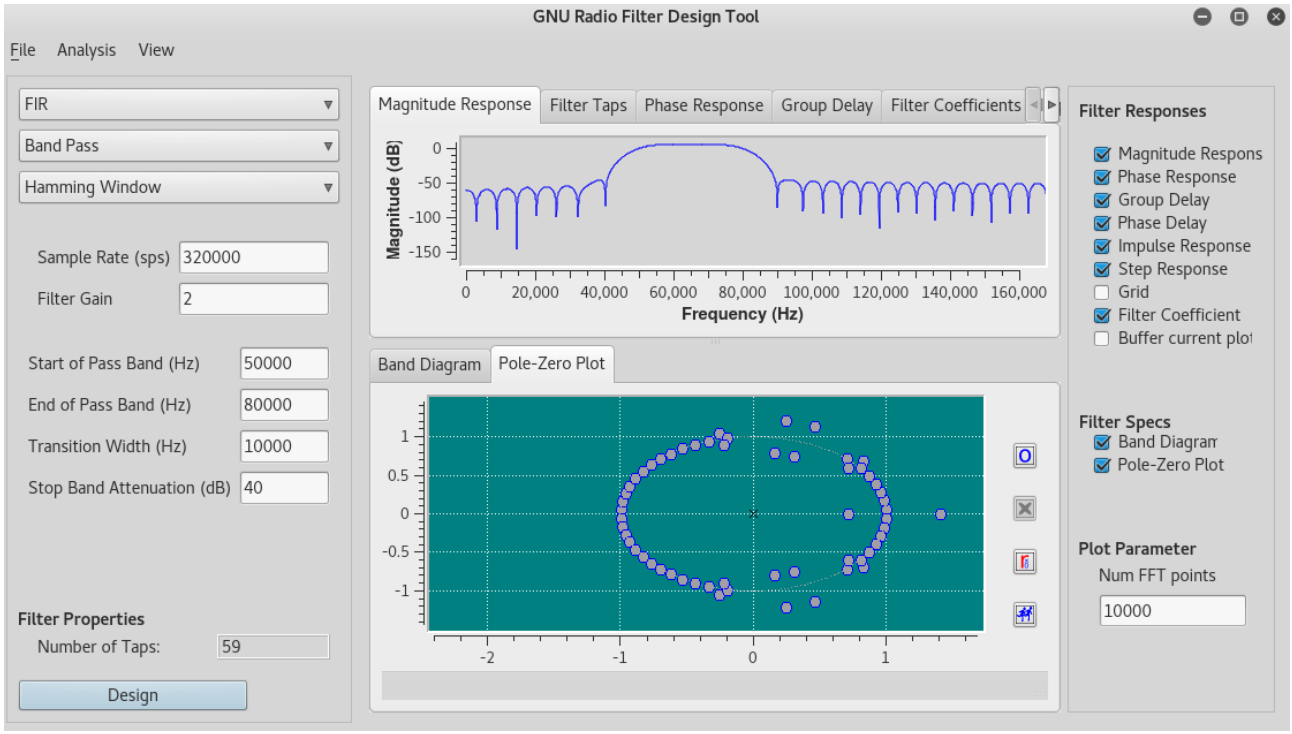

Fig43 Filtro diseñado

- **Respuesta en Magnitud** Brinda una grafica de la respuesta en magnitud Fig44.

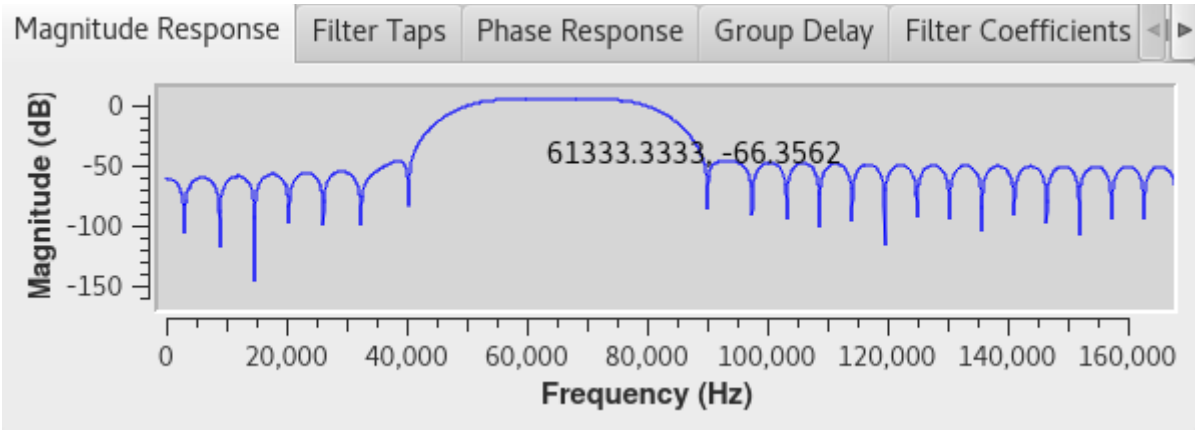

Fig44. Magnitude Response

- **Respuesta en fase:** Gráfica de la respuesta en fase Fig45.

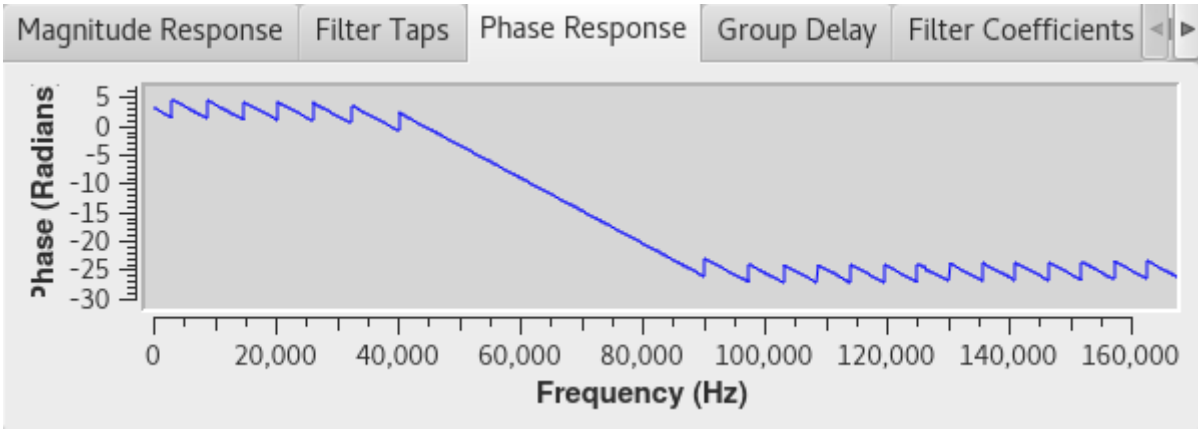

Fig45. Respuesta en Fase

- **Retraso de Grupo:** Gráfica del retraso de grupo en el filtro Fig46.

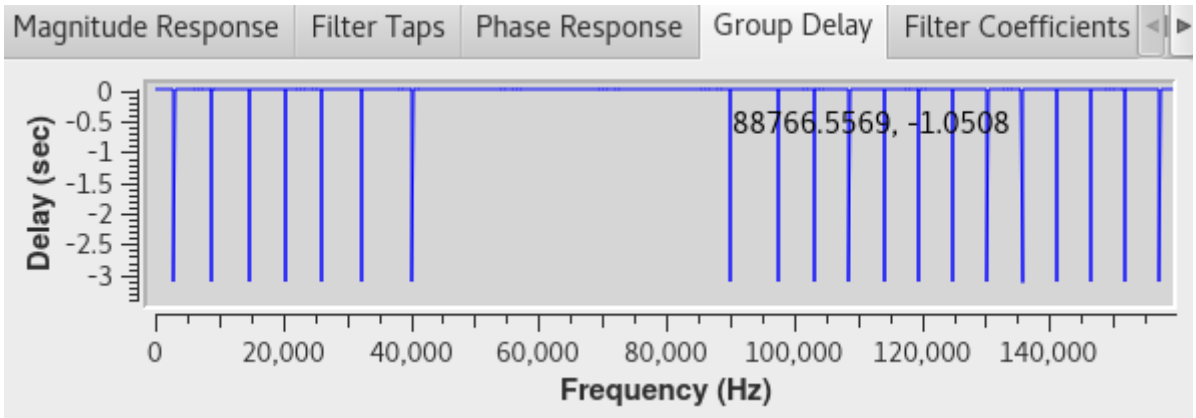

Fig46. Retraso de Grupo

- **Coeficientes del Filtro:** Una de las mejores opciones brindadas por la herramienta, nos da la oportunidad de copiar todos los coeficientes del filtro Fig47.

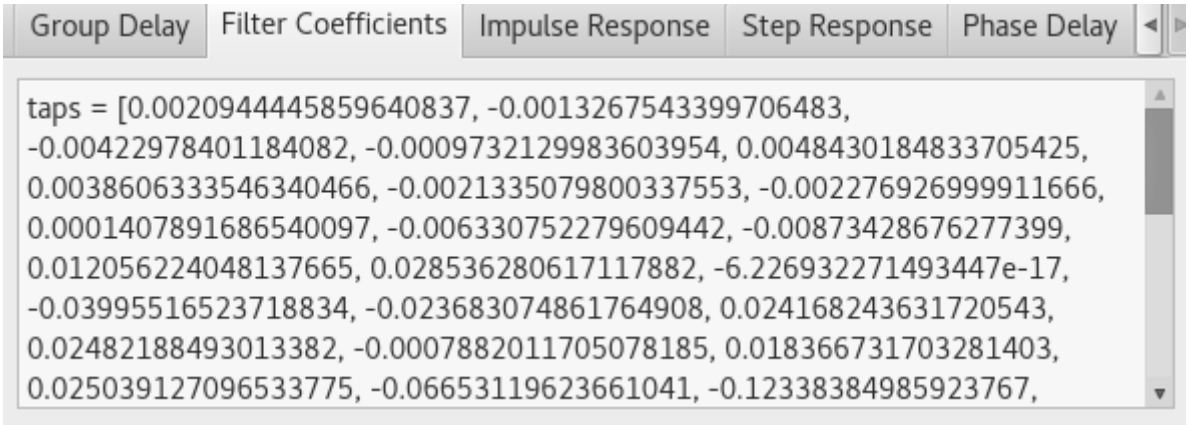

Fig47. Coeficientes del filtro

- **Respuesta al impulso:** Gráfica de la respuesta al impulso Fig48.

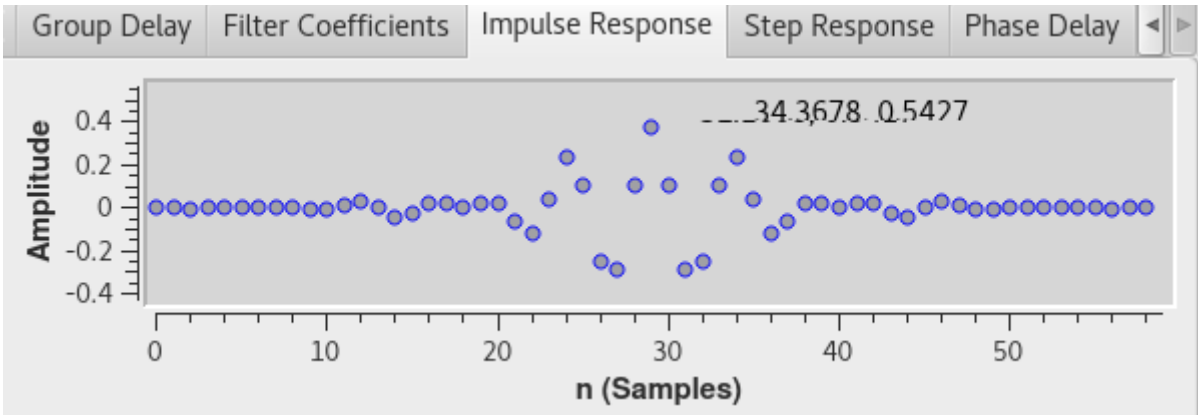

Fig48. Respuesta al impulso

- **Respuesta al paso:** Gráfica la respuesta al paso Fig49.

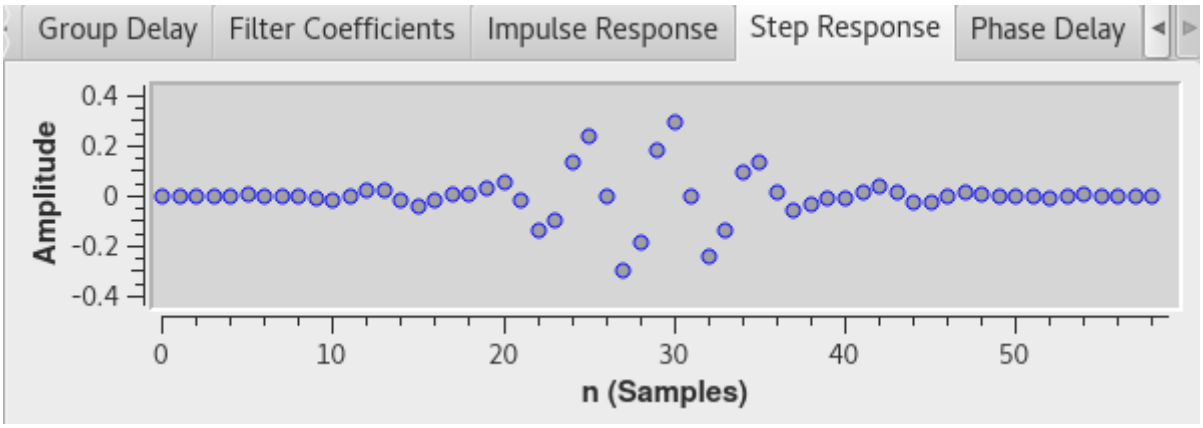

Fig49. Respuesta al paso

- **Retraso de fase:** Gráfica el retraso de fase en el filtro fig50.

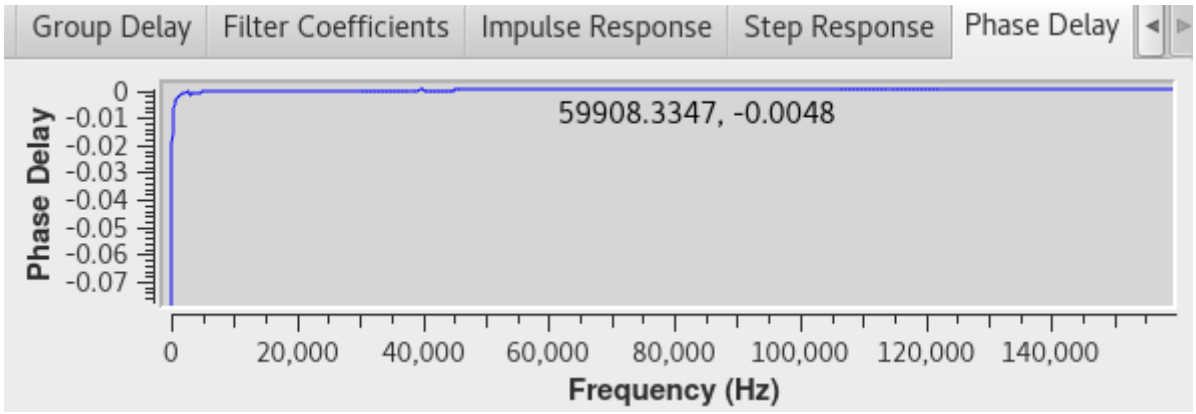

Fig50. Retraso de Fase

H) Instalar Firmware en USRP 2920 y 2921

Para utilizar el USRP primero se debe instalar el firmware, para eso vamos a seguir los siguientes pasos:

- Conectamos fisicamente el USRP
- Configuramos la red para que este dentro del mismo grupo que el USRP para eso podemos escribir en consola
  sudo ifconfig 192.168.10.10
- Una vez realizado esto vamos a verificar que se detecta el software
  uhd_find_devices
- Si todo ha salido bien va a detectar el USRP y nos va a dar la dirección IP y el ID del dispositivo
- A continuación nos vamos a mover hacia la dirección en donde se encuentran las imágenes:
  cd /usr/share/uhd/images/
- Procedemos a cargar la imagen del firmware y del fpga al USRP
  uhd_image_loader –args="type=usrp2,addr=<ip-USRP>" --fw-path usrp_n210_fw.bin --fpga-path usrp_n210_r4_fpga.bin
- Recordar que donde dice **<ip-USRP>** colocamos la ip del USRP que ya conocemos con anterioridad.
- Desconectamos el cable de PODER del USRP y lo volvemos a conectar, esto es muy importante para que se reinicie el USRP caso contrario no va a funcionar.
- Finalmente podemos proceder a hacer nuestras instalaciones.

I) Configurar Bloques USRP

Para realizar la configuración del USRP en nuestro proyecto nos vamos a centrar en los bloque necesarios para transmisión y recepción de las señales.

- **Bloque TX UHD: USRP Sink**
  Este bloque es el que nos permitira transmitir nuestra información mediante USRP Fig51.

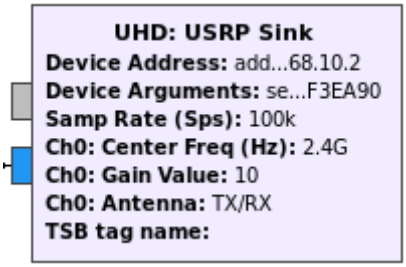

Fig51. Bloque UHD: USRP Sink

En este caso se ha creado una variable que contiene la direccion IP del USRP Fig52.

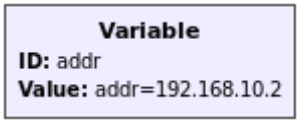

Fig52. Variable nombre “addr”

Una vez agregamos nuestro bloque USRP Sink a nuestro proyecto, lo vamos a configurar haciendo doble click sobre el mismo.

Tenemos multiples opciones que trataremos a continuación Fig53., Fig54.

- **ID:** Identificador unico de nuestro bloque
- **Input Type:** Tipo de datos en la entrada (Complex float32)
- **Device Address:** Dirección ip del dispositivo USRP
- **Device Arguments:** Multiples argumentos para el USRP como ejemplo esta el numero serie.
- **Sync:** Tipo de Sincronización
- **Num Channels:** Numero de Canales
- **Samp Rate (Sps):** Taza de muestreo

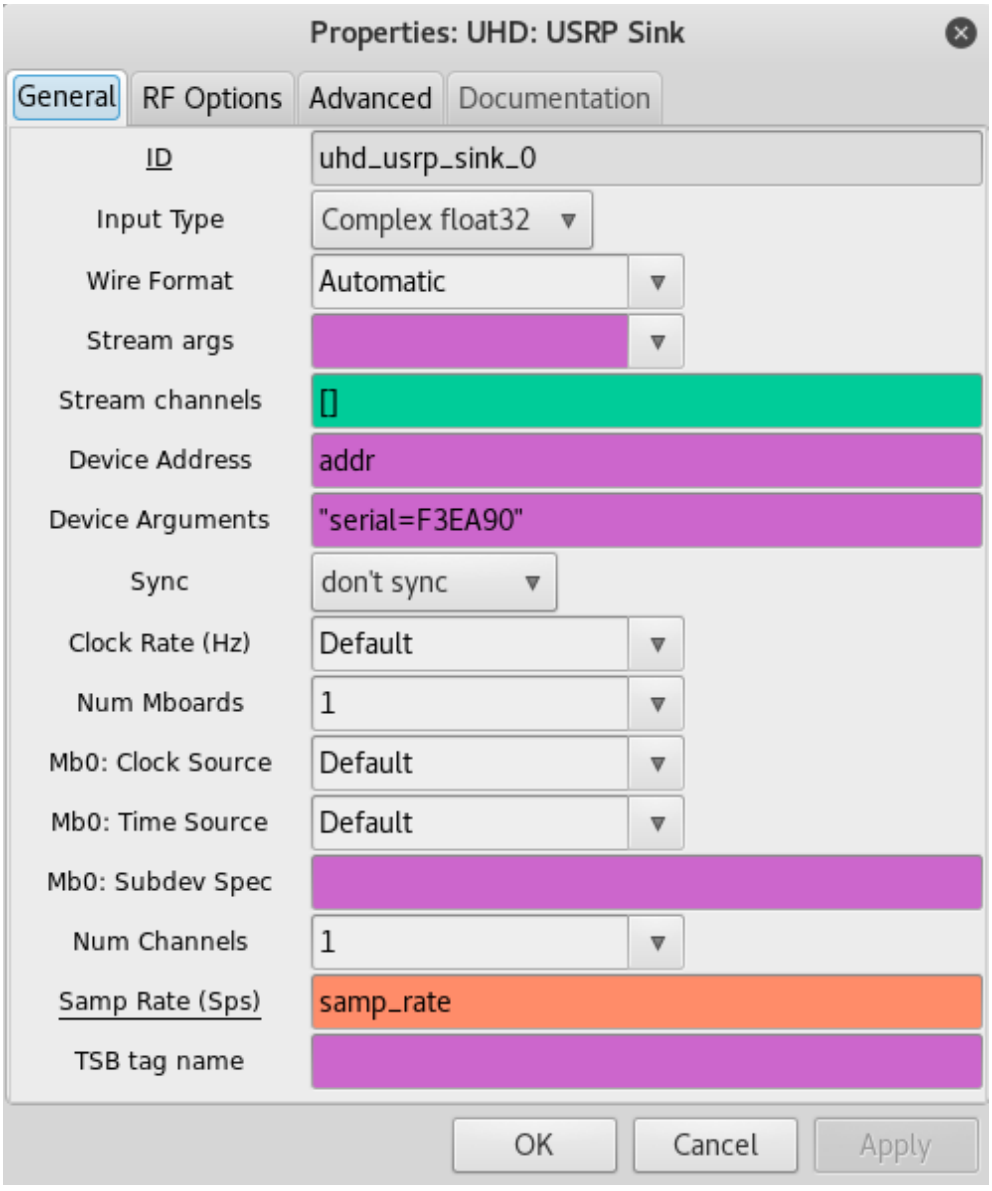

Fig53. Propiedades Generales USRP

- **Ch0: Center Freq(Hz):** Frecuencia central del canal 0
- **Ch0: Gain Value:** Valor de Ganancia
- **Ch0: Gain Type:** Tipo de Ganancia
- **Ch0: Antenna:** Permite escoger la antena

- **Ch0: Bandwidth(Hz):** Ancho de banda en Hz

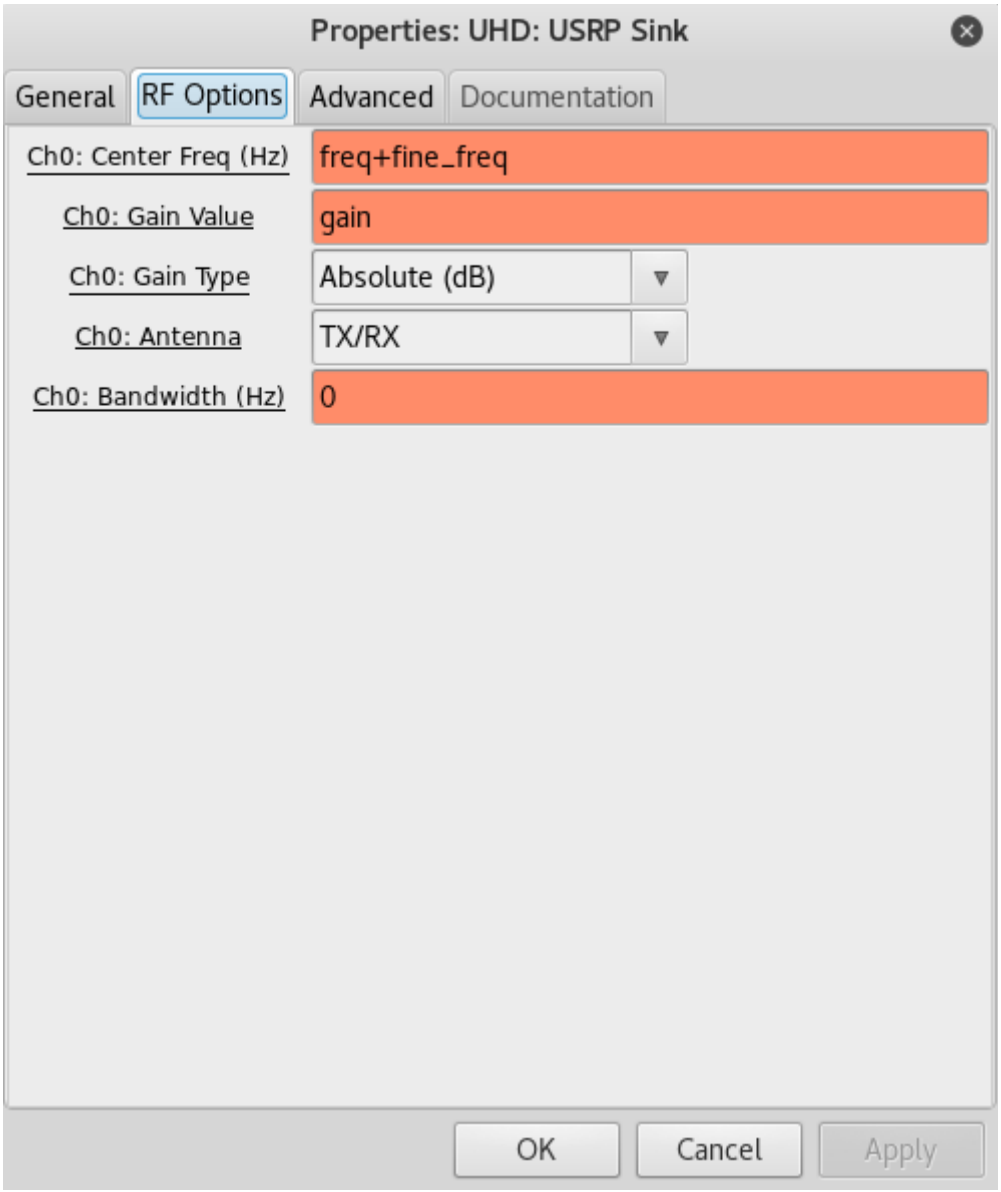

Fig54. Propiedades RF Options USRP

- **Bloque RX UHD: USRP Source**

  Este bloque es el que nos permitira transmitir nuestra información mediante USRP Fig51.

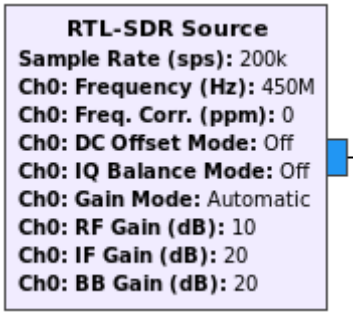

Fig51. Bloque USRP Source

Vamos a proceder a describir brevemente como se configura este bloque para recepción de datos en el USRP Fig52.

- **ID:** Identificador unico de nuestro bloque
- **Input Type:** Tipo de datos en la entrada (Complex float32)
- **Device Arguments:** Multiples argumentos para el USRP como ejemplo esta el numero serie.
- **Sync:** Tipo de Sincronización
- **Num Channels:** Numero de Canales
- **Samp Rate (Sps):** Taza de muestreo
- **Ch0: Center Freq(Hz):** Frecuencia central del canal 0
- **Ch0: RF Gain (dB):** Ganancia de Radio frecuencia
- **Ch0: Antenna:** Se puede colocar la antena con la cual se recibe o dejar por default
- **Ch0: Bandwidth(Hz):** Ancho de banda en Hz

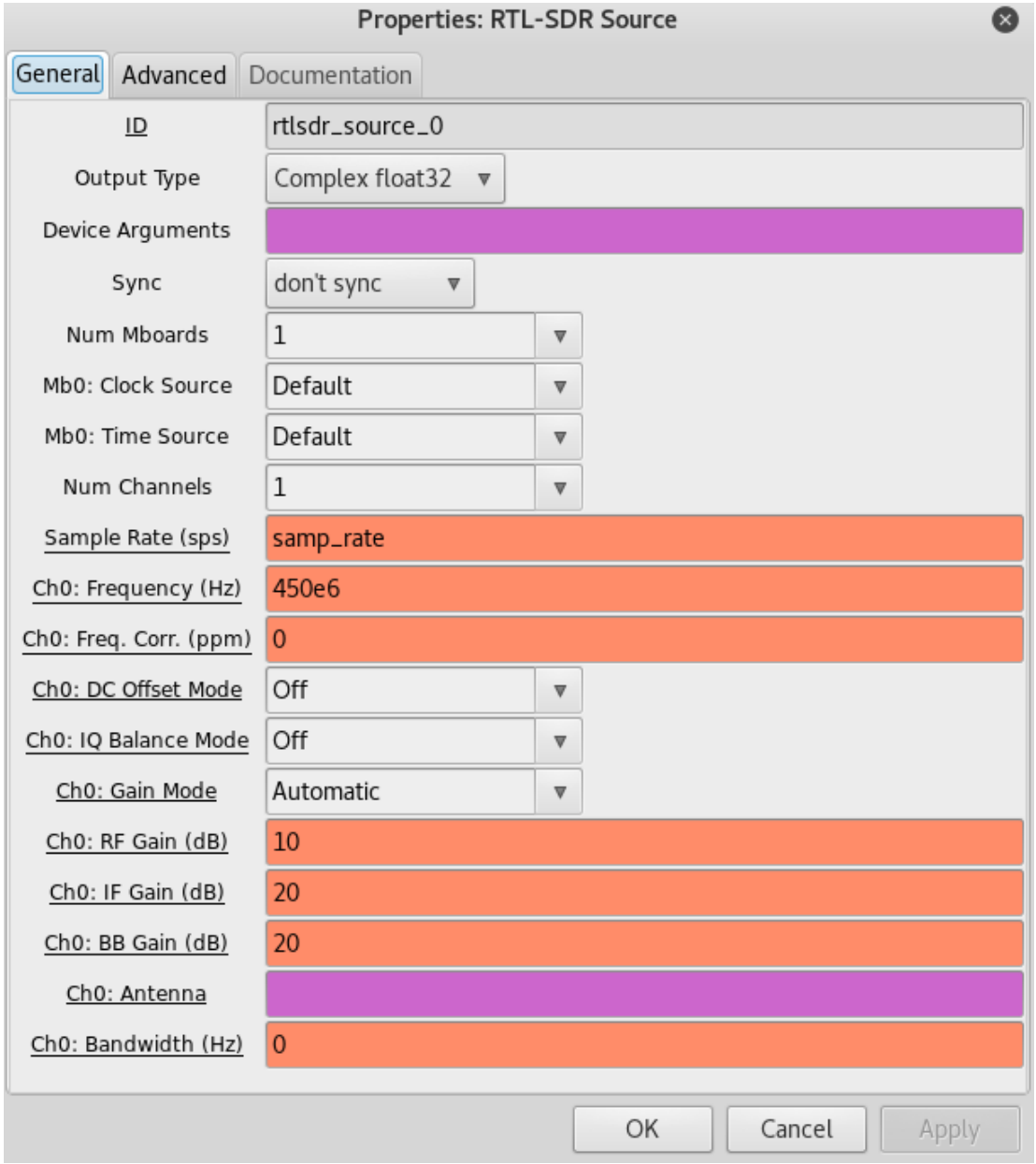

Fig52. Opciones de Configuración del Receptor

J) Conclusiones

Como se puede observar GNU-Radio es un framework de software libre para desarrollo de aplicaciones en tratamiento de señales y telecomunicaciones muy versatil y flexible.

Posee una gran variedad de librerías con las que se puede desarrollando proyectos muy interesantes, tiene la capacidad de ser programado mediante código y así mismo de programarse mediante bloques, de manera grafica.

Lo mas recomendable para iniciar el estudio y el uso de este software es usar la versión de desarrollo en bloques, puesto que brinda mas sencillez al explorar las diferentes librerías que se tienen incluidas en el Core del programa.

Permite crear nuestras propias librerías e incluirlas en el software para su posterior uso, ademas existen amplia variedad de librerías disponibles en Github, desarrollado por gente de todo el mundo.

El soporte que dan hacia los dongles como el RTL-SDR o fun cube son bastante buenas, así mismo tiene soporte para SDR de mas alta gama como los USRP creados por etthus o National Instuments, permitiendo de esta manera que el desarrollo no solo se realice en simulación, si no también en ambientes reales de telecomunicaciones.