

# Sistemas de Comunicaciones basados en Radio Definida por Software (SDR)

Dr. Ing. Alejandro José Uriz

Uso de GNU Radio Companion



GNU Radio es un software libre y abierto que consta de herramientas para el procesamiento digital de señales que permiten implementar sistemas de comunicaciones.

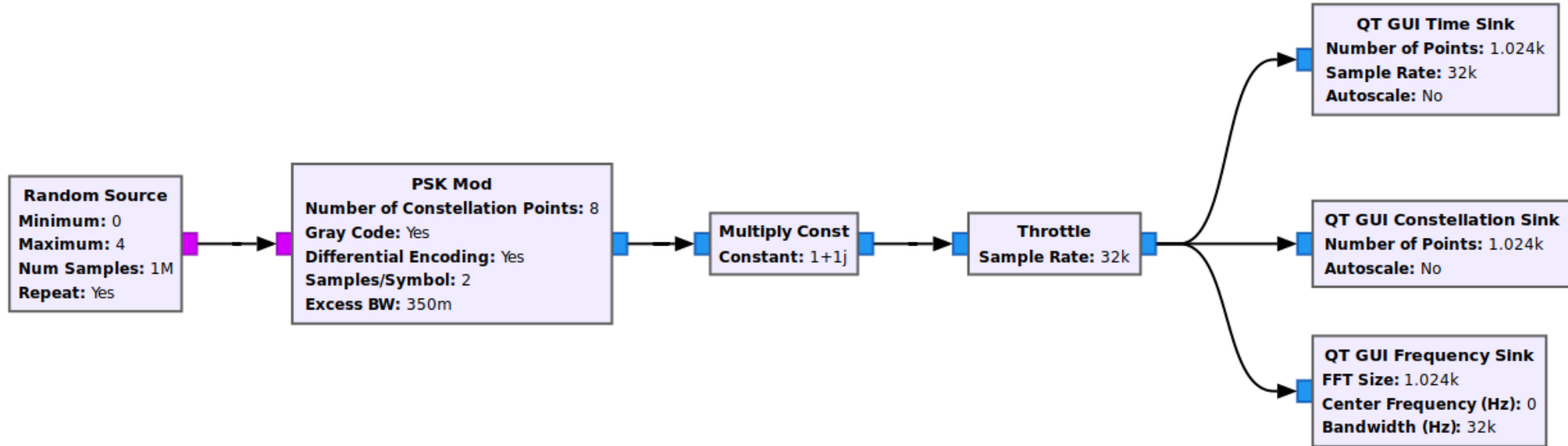
El sitio de Internet del proyecto es:  
<https://www.gnuradio.org/>

Esta plataforma es ampliamente utilizado para simulación de sistemas de comunicaciones. Pero también dispone de la capacidad de integrar radios definidas por software y por lo tanto implementar sistemas de comunicaciones.

Las ventajas previamente descriptas hacen que sea ampliamente usado para investigación, docencia, industria y también por radioaficionados.

**Options**  
**Title:** Test  
**Author:** lac068  
**Output Language:** Python  
**Generate Options:** QT GUI

**Variable**  
**Id:** samp\_rate  
**Value:** 32k



# Waveform Generators

- Constant Source
- Noise Source
- Signal Source (e.g. Sine, Square, Saw Tooth)

# Modulators

- AM Demod
- Continuous Phase Modulation
- WBFM Receive / NBFM Receive
- PSK Mod / Demod
- GFSK Mod / Demod
- GMSK Mod / Demod
- QAM Mod / Demod

# Instrumentation (i.e., GUIs)

- Constellation Sink
- Frequency Sink
- Histogram Sink
- Number Sink
- Time Sink
- Waterfall Sink

# Channel Models

- Channel Model
- Fading Model
- Dynamic Channel Model
- Frequency Selective Fading Model

# Filters

- Band Pass / Reject Filter / Low / High Pass Filter
- IIR Filter
- Generic Filterbank
- Hilbert
- Decimating FIR Filter
- Root Raised Cosine Filter
- FFT Filter

# Fourier Analysis

- FFT / Log Power FFT
- Goertzel (Resamplers) / Fractional Resampler / Polyphase Arbitrary Resampler / Rational Resampler (Synchronizers)
- Clock Recovery MM / Correlate and Sync
- Costas Loop
- FLL Band-Edge / PLL Freq Det / PN Correlator / Polyphase Clock Sync



La modularidad de los bloques de GNU Radio permiten implementar sistemas de comunicaciones de una forma muy ágil.

Asimismo por ser de código abierto existe gran variedad de códigos y librerías generadas por la comunidad disponibles.

Los bloques y librerías pueden ser creados y editados en C++ y/o Python.

File Edit Build Help

Options  
ID: top\_block

Variable  
ID: samp\_rate  
Value: 32k

Signal Source  
Sample Rate: 32k  
Waveform: Square  
Frequency: 1k  
Amplitude: 1  
Offset: 0

Signal Source  
Sample Rate: 32k  
Waveform: Sine  
Frequency: 10k  
Amplitude: 1  
Offset: 0

Noise Source  
Noise Type: Gaussian  
Amplitude: 1  
Seed: 42

Add

Add

Scope Sink  
Title: Scope Plot  
Sample Rate: 32k  
V Scale: 0  
T Scale: 0

Área de Trabajo

Blocks

- [ Sources ]
- [ Sinks ]
- [ Graphical Sinks ]
  - Number Sink
  - Scope Sink
  - FFT Sink
  - Constellation Sink
  - Waterfall Sink
  - Histo Sink
- [ Operators ]
- [ Type Conversions ]
- [ Stream Conversions ]
- [ Misc Conversions ]
- [ Synchronizers ]
- [ Level Controls ]
- [ Filters ]
- [ Modulators ]
- [ Error Correction ]
- [ Line Coding ]
- [ Probes ]
- [ USRP ]
- [ Variables ]
- [ Misc ]

>>> warning: this flow graph may not have flow control: no audio or usrp blocks found. Add a Misc->inrottle block to your flow graph to avoid CPU congestion.

Executing: "/home/weissalissa/Desktop/new proj/reqd/top\_block.py"

>>> Done

Información de ejecución y variables

Barra de librerías y módulos

De esta forma hay desde bloques aptos para implementar sistemas de comunicaciones clásicos como AM, FM, PSK, FSK y QAM, hasta más complejos como OFDM.

También existen librerías para aplicaciones y protocolos específicos. Algunos son:

- Televisión digital.
- Comunicaciones satelitales.
- Telefonía celular.
- ADS-B.
- AIS.

# Tipos de variables

Los bloques de GNU Radio son aptos para distintos tipos de variables, dependiendo de su naturaleza. Los tipos de variables que dispone de el entorno son:

- Complex (8 bytes).
- Float (4 byte floating point).
- Int (4 byte integer).
- Short (2 byte integer).
- Byte (1 byte of data).

**Signal Source**  
 Sample Rate: 32k  
 Waveform: Cosine  
 Frequency: 1k  
 Amplitude: 1  
 Offset: 0  
 Initial Phase (Radians): 0

Terminal Azul: Tipo Complex

**Signal Source**  
 Sample Rate: 32k  
 Waveform: Cosine  
 Frequency: 1k  
 Amplitude: 1  
 Offset: 0  
 Initial Phase (Radians): 0

Terminal Naranja: Tipo Float

**Signal Source**  
 Sample Rate: 32k  
 Waveform: Cosine  
 Frequency: 1k  
 Amplitude: 1  
 Offset: 0  
 Initial Phase (Radians): 0

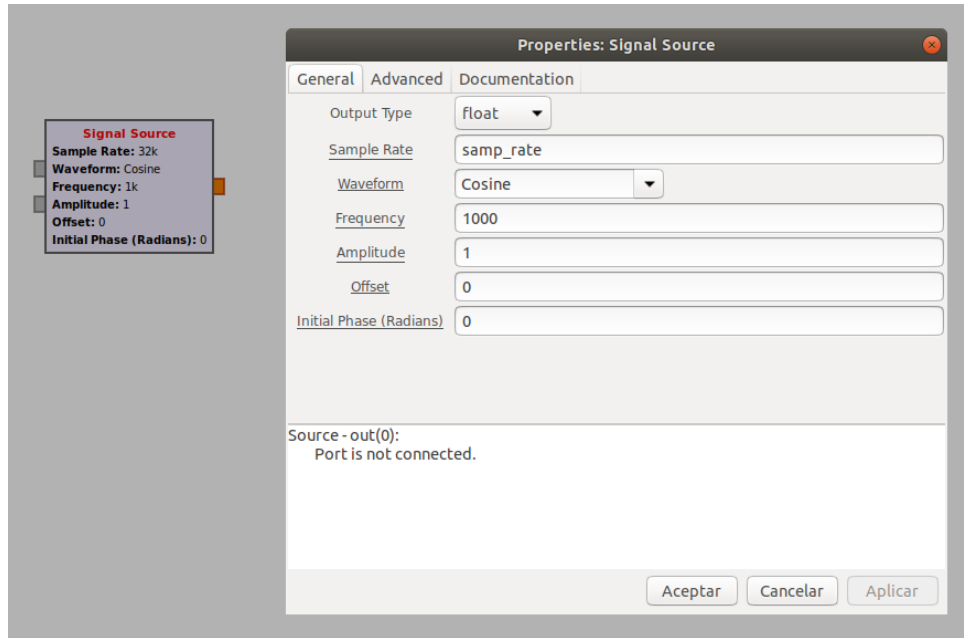
Terminal Verde: Tipo Int

**Signal Source**  
 Sample Rate: 32k  
 Waveform: Cosine  
 Frequency: 1k  
 Amplitude: 1  
 Offset: 0  
 Initial Phase (Radians): 0

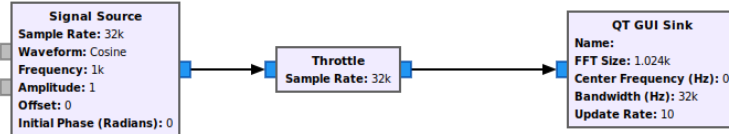
Terminal Amarillo: Tipo Short

**Signal Source**  
 Sample Rate: 32k  
 Waveform: Cosine  
 Frequency: 1k  
 Amplitude: 1  
 Offset: 0  
 Initial Phase (Radians): 0

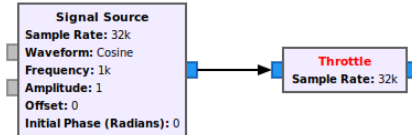
Terminal Magenta: Tipo Byte



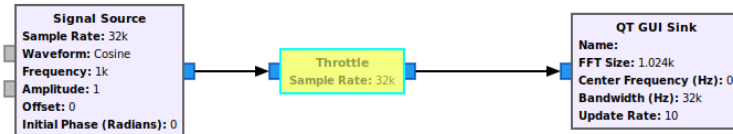
# Uso de bloques



Fuente negra en el título del bloque: Correcto



Fuente roja en el título del bloque: Hay un error



Bloque sombreado en amarillo: Bloque con By-pass.

# Creación de un proyecto

Cuando se crea un nuevo proyecto, se deben configurar dos bloques: el “Options” y el de una variable que se denomina “samp\_rate” (sample rate).

## Options

**Title:** Not titled yet

**Author:** lac068

**Output Language:** Python

**Generate Options:** QT GUI

## Variable

**Id:** samp\_rate

**Value:** 32k

# Bloque Options

## Options

**Title:** Not titled yet

**Author:** lac068

**Output Language:** Python

**Generate Options:** QT GUI



Properties: Options

General Advanced Documentation

Id Prueba\_1

Title Prueba

Author Alejandro

Copyright

Description

Output Language Python ▾

Generate Options QT GUI ▾

Run Autostart ▾

Max Number of Output 0

Realtime Scheduling Off ▾

QSS Theme

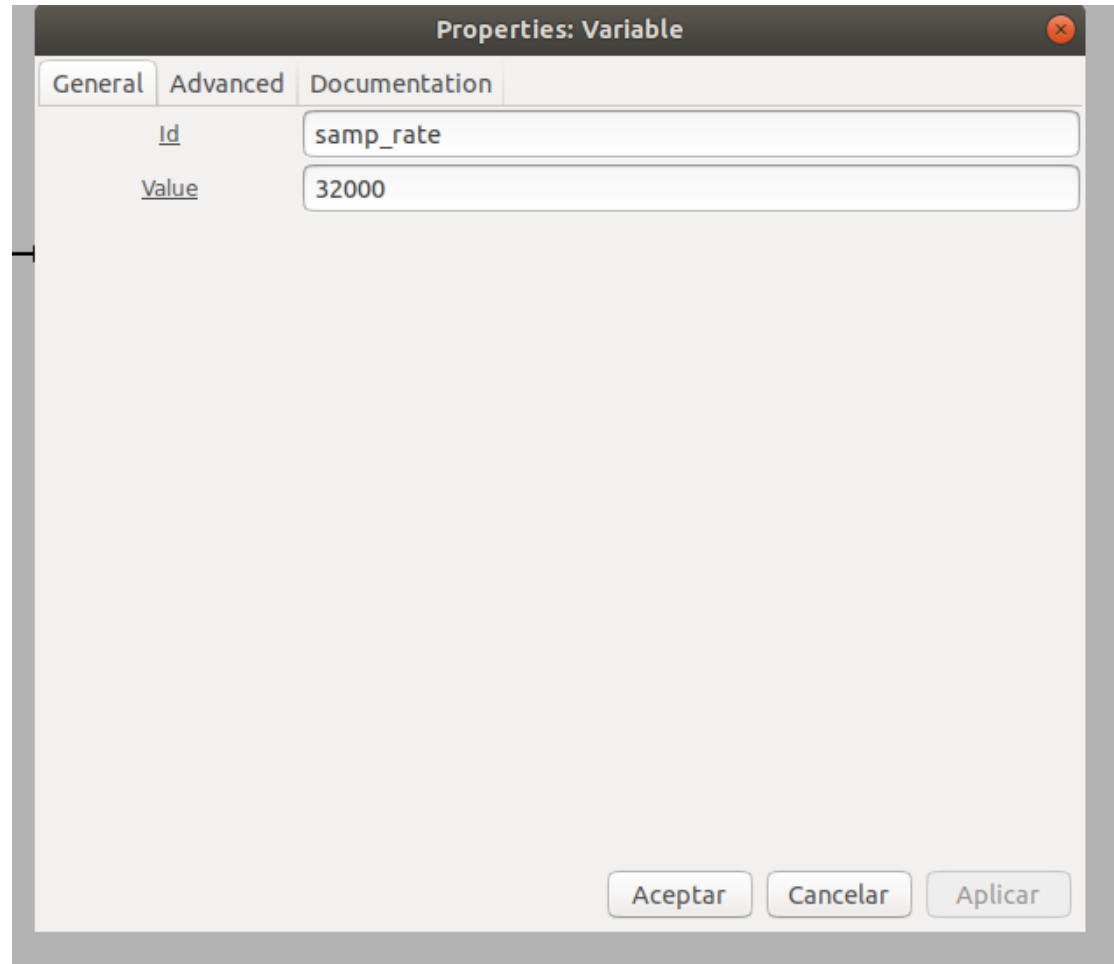
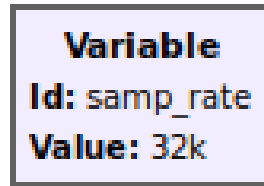
Aceptar Cancelar Aplicar



El bloque Options contiene opciones de configuración inicial.  
Se pueden configurar los siguientes parámetros

- **ID:** Nombre único asignado a cada bloque
- **Title:** Título del proyecto
- **Author:** Autor del proyecto
- **Description:** Descripción del proyecto
- **Canvas Size:** Tamaño del área de trabajo
- **Generate Options:** Formato de salida del modo grafico.
- **Run:** Forma de inicio de la ejecución del proyecto
- **Max Number of Output:** Máximo número de salidas
- **Realtime Scheduling:** Activar/desactivar programación en

## Bloque variable

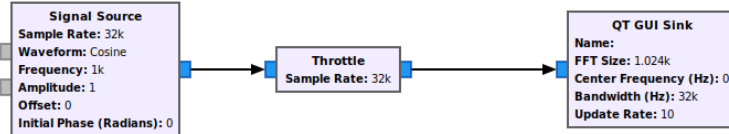


A screenshot of a software dialog box titled "Properties: Variable". It has three tabs: "General", "Advanced", and "Documentation". The "General" tab is selected. It contains two input fields:

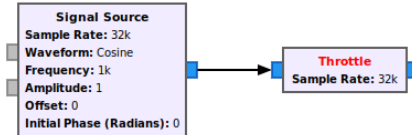
- Id:** samp\_rate
- Value:** 32000

At the bottom right, there are three buttons: "Aceptar", "Cancelar", and "Aplicar".

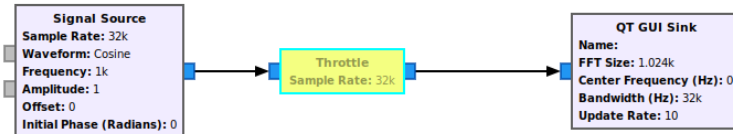
# Uso de bloques



Fuente negra en el título del bloque: Correcto



Fuente roja en el título del bloque: Hay un error



Bloque sombreado en amarillo: Bloque con By-pass.

# Ejemplo básico

## Options

**Title:** Ejemplo-1

**Author:** Alejandro

**Output Language:** Python

**Generate Options:** QT GUI

## Variable

**Id:** samp\_rate

**Value:** 32k

## Signal Source

**Sample Rate:** 32k

**Waveform:** Cosine

**Frequency:** 1k

**Amplitude:** 1

**Offset:** 0

**Initial Phase (Radians):** 0

## Throttle

**Sample Rate:** 32k

## QT GUI Sink

**Name:**

**FFT Size:** 1.024k

**Center Frequency (Hz):** 0

**Bandwidth (Hz):** 32k

**Update Rate:** 10

# Ejemplo básico

## Options

**Title:** Ejemplo-1

**Author:** Alejandro

**Output Language:** Python

**Generate Options:** QT GUI

## Variable

**Id:** samp\_rate

**Value:** 32k

Datos tipo FLOAT

## Signal Source

**Sample Rate:** 32k

**Waveform:** Cosine

**Frequency:** 1k

**Amplitude:** 1

**Offset:** 0

**Initial Phase (Radians):** 0

## Throttle

**Sample Rate:** 32k

## QT GUI Sink

**Name:**

**FFT Size:** 1.024k

**Center Frequency (Hz):** 0

**Bandwidth (Hz):** 32k

**Update Rate:** 10

**Options****Title:** Ejemplo-1**Author:** Alejandro**Output Language:** Python**Generate Options:** QT GUI**Variable****Id:** samp\_rate**Value:** 32k

General Advanced Documentation

Output Type float

Sample Rate samp\_rate

Waveform Cosine

Frequency 1000

Amplitude 1

Offset 0

Initial Phase (Radians) 0

Aceptar

Cancelar

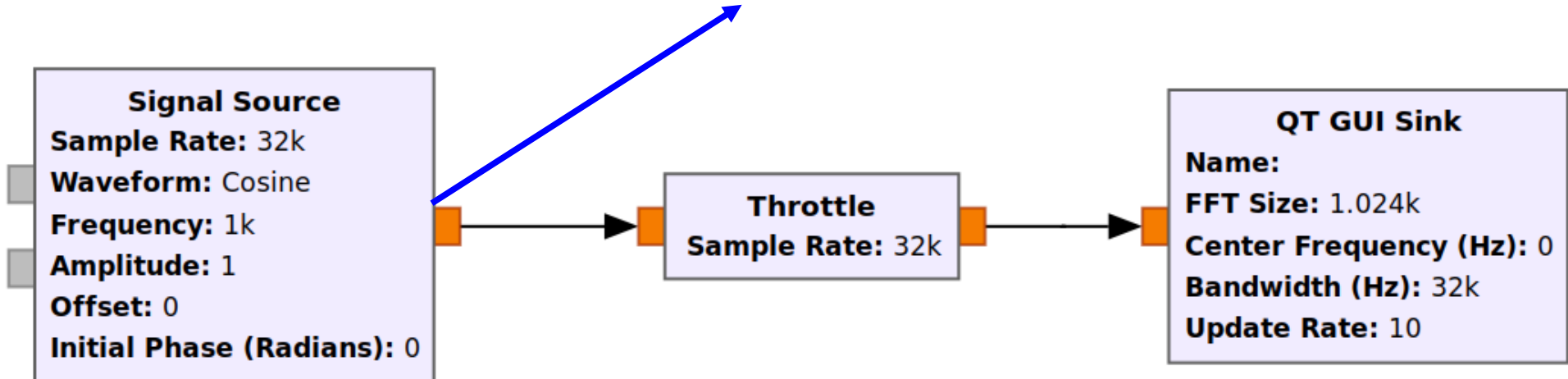
Aplicar

**Generador de Señales****Signal Source****Sample Rate:** 32k**Waveform:** Cosine**Frequency:** 1k**Amplitude:** 1**Offset:** 0**Initial Phase (Radians):** 0**Throttle**  
**Sample Rate:** 32k**QT GUI Sink****Name:****FFT Size:** 1.024k**Center Frequency (Hz):** 0**Bandwidth (Hz):** 32k**Update Rate:** 10

## Bloque THROTTLE

Este bloque se suele usar a la salida de generadores de señales que no estén implementadas en hardware.

Se utiliza para limitar la tasa de generación de muestras. De lo contrario, el uso del CPU se eleva y el GNU podría fallar.



**Options**  
**Title:** Ejemplo-1  
**Author:** Alejandro  
**Output Language:** Python  
**Generate Options:** QT GUI

**Variable**  
**Id:** samp\_rate  
**Value:** 32k

Graficador de señales

