UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

DEPARTAMENTO DE
ELECTRONICA

---

ELO241: Laboratorio de Comunicaciones

# Certamen 2

**Alumno:** Brian Montero 201830015-2
**Profesor:** Patrick Guicharrousse
**Fecha:** 1 de julio de 2024

---

# Índice

# 1. Problema 1: Modulación en amplitud

**Se tiene disponible el archivo *prueba_audio_1*, el cual ha sido generado en GNU Radio mediante el bloque Wav File Sink. Este archivo contiene cuatro señales transmitidas en AM de forma simultánea, cada una de ellas modulada con diferentes parámetros.**

## 1.1. Dibuje un diagrama de bloques en el que se explique qué estrategia utilizará para recuperar las señales de audio disponibles en el archivo *prueba_audio_1*. Explique su estrategia y especifique qué valores debe utilizar para recuperar la señal de audio original. Recuerde que los audios fueron modulados con diferentes parámetros de frecuencia e índice de modulación.

Se tiene el siguiente diagrama de bloques en el cual tenemos como entrada el archivo de audio *prueba_audio_1*, el cual sera procesado mediante filtros y un demodulador para poder separar las señales en diferentes rangos de frecuencia. El proceso se divide en cuatro etapas (ya que son cuatro señales), cada una de las etapas esta diseñada para analizar un rango especifico de frecuencias. Para entender con claridad se presenta a continuación el diagrama:

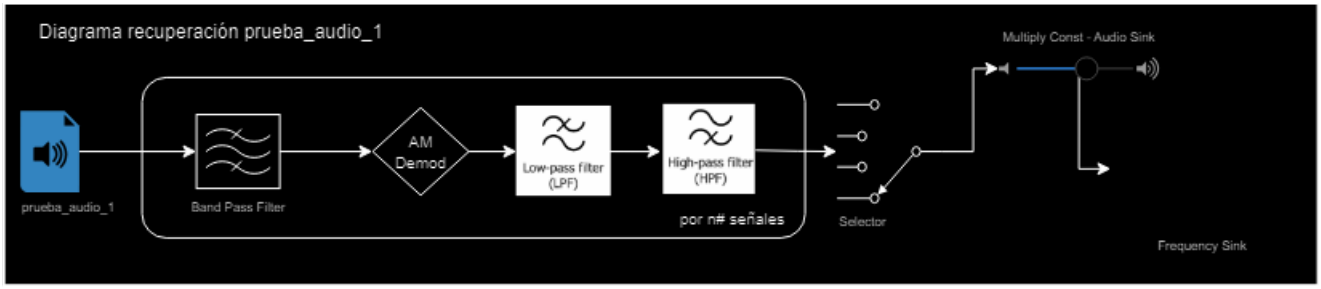



Figura 1: Diagrama de Bloques

Podemos observar que el diagrama de bloques se compone de los siguientes componentes:
**Band Pass Filter + AM Demod + Low Pass Filter + High Pass Filter**

- *Band Pass Filter:* Este filtro selecciona un rango específico de frecuencias del archivo de audio. Se ajustan las frecuencias bajas y altas de corte, filtrando así las frecuencias fuera de este rango. En nuestro caso, se utilizan cuatro filtros de paso de banda para extraer los cuatro rangos de frecuencia distintos:
  - 3-7 kHz
  - 7-11 kHz
  - 13-17 kHz
  - 18-22 kHz
- *AM Demod:* Este bloque en Python realiza la demodulación de amplitud de la señal filtrada. Incluyo una variable modulation_index, la cual puedo ir variando con un Range hasta que se escuche mejor.

C: > Users > prodb > AppData > Local > Temp > epy_block_0_0_gvdc052r.py > ...

```python
import numpy as np
from gnuradio import gr

class am_demod(gr.sync_block):
    def __init__(self, mod_index=1.0):
        gr.sync_block.__init__(
            self,
            name="AM Demod",
            in_sig=[np.float32],
            out_sig=[np.float32]
        )
        self.mod_index = mod_index

    def work(self, input_items, output_items):
        in0 = input_items[0]
        out = output_items[0]

        # AM demodulation: absolute value of the signal divided by the modulation index
        out[:] = np.abs(in0) / self.mod_index

        return len(output_items[0])
```

Figura 2: Bloque AM Demod en Python para GNU Radio

- *Low Pass Filter:* Este filtro elimina las componentes de alta frecuencia que resultan de la demodulación de amplitud, permitiendo que solo las frecuencias bajas (la señal de interés) sean pasadas a la siguiente etapa.
- *High Pass Filter:* Este filtro asegura que las frecuencias muy bajas, que podrían estar presentes como ruido o interferencia, sean eliminadas, dejando solo la señal de frecuencia útil después de la demodulación y filtrado.

Finalmente, las cuatro señales procesadas se conectan a un selector, el cual permite reproducir el audio y visualizar el espectro de frecuencia de la señal seleccionada. El propósito de utilizar estos bloques es separar el archivo permitiendo un análisis detallado de las diferentes partes de *prueba_audio_1*.

## 1.2. Genere un archivo en GNU Radio que actúe como demodulador de las señales. Al ejecutar el archivo, se debe escuchar el mensaje transmitido. El archivo debe tener una forma de seleccionar el audio que se está escuchando mientras se ejecuta.

En una primera instancia se observa el espectro de la frecuencia para saber mas o menos en que rango están las señales que se encuentran en *prueba_audio_1*, estos valores serán utilizados en los diferentes Band Pass Filter como vimos en el inciso anterior, este análisis se lleva a la práctica en el siguiente esquema de GNU Radio el cual se encuentra disponible en Referencias.

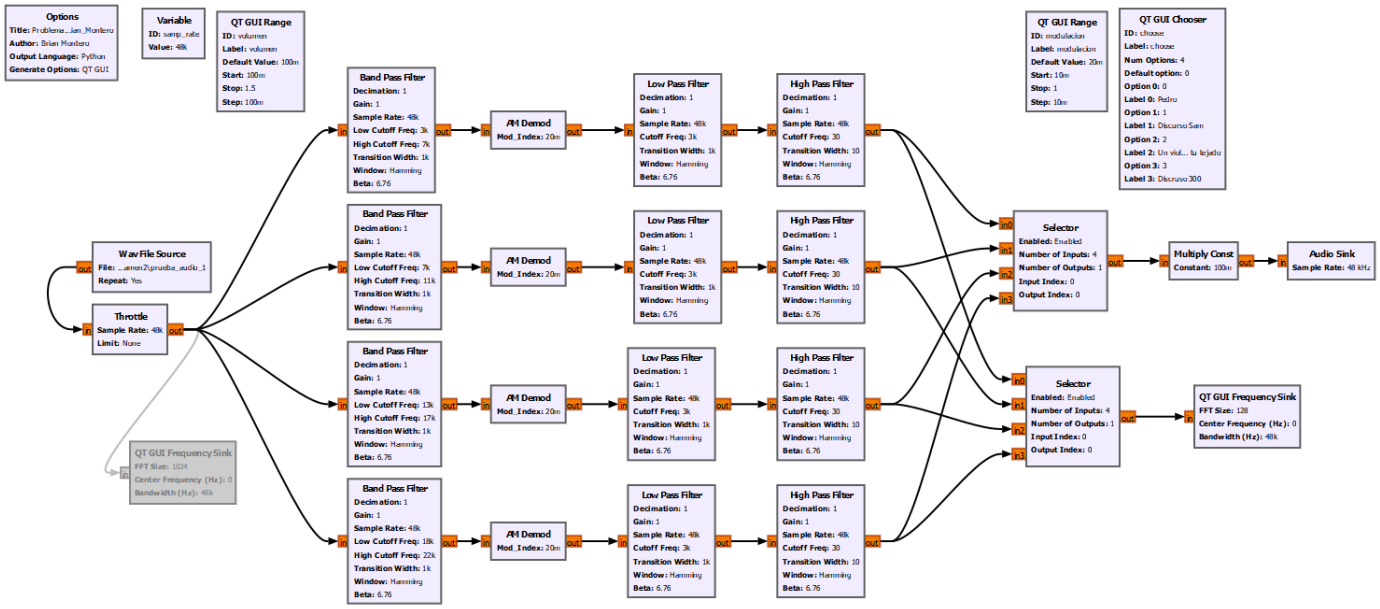



Figura 3: Esquema GNU Radio

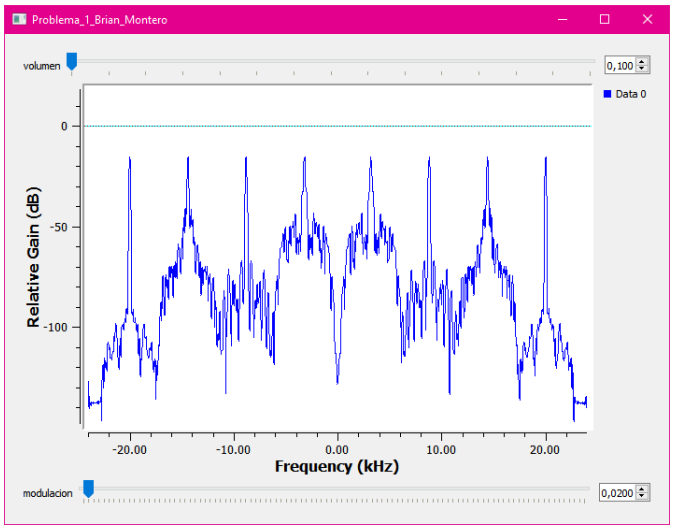



Figura 4: Espectro de la frecuencia para el *prueba_audio_1*

### 1.3. Identifique las señales transmitidas. Nombre los audios como: El audio 1 corresponde a XX canción, El audio 2 corresponde a XX canción.

Se utilizan variables como chooser, el volumen y el índice de modulación para poder identificar (escuchar) de mejor forma las cuatro señales, las cuales corresponden a:

- Señal 1: Canción Pedro

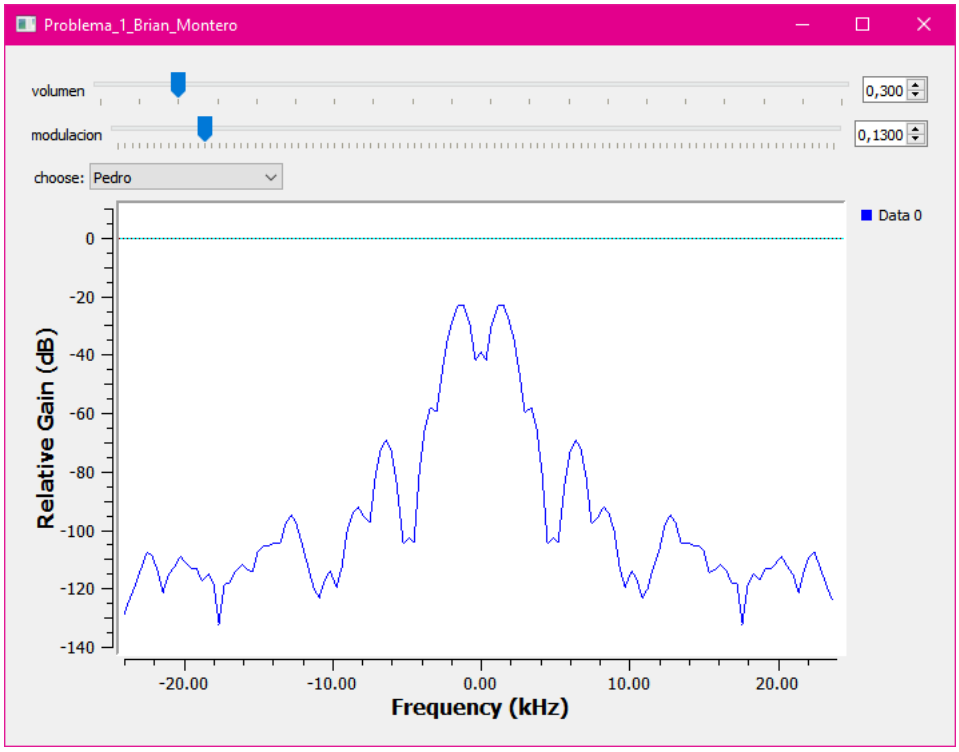



Figura 5: Espectro de la frecuencia de la señal 1

- Señal 2: Discurso Sam El señor de los Anillos

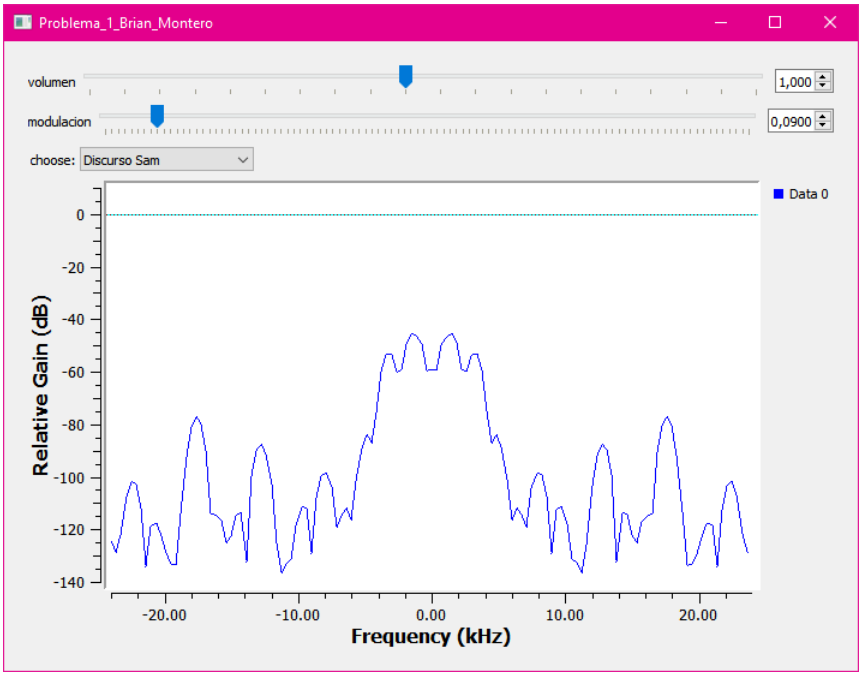



Figura 6: Espectro de la frecuencia de la señal 2

- Señal 3: Canción Un violinista en tu tejado

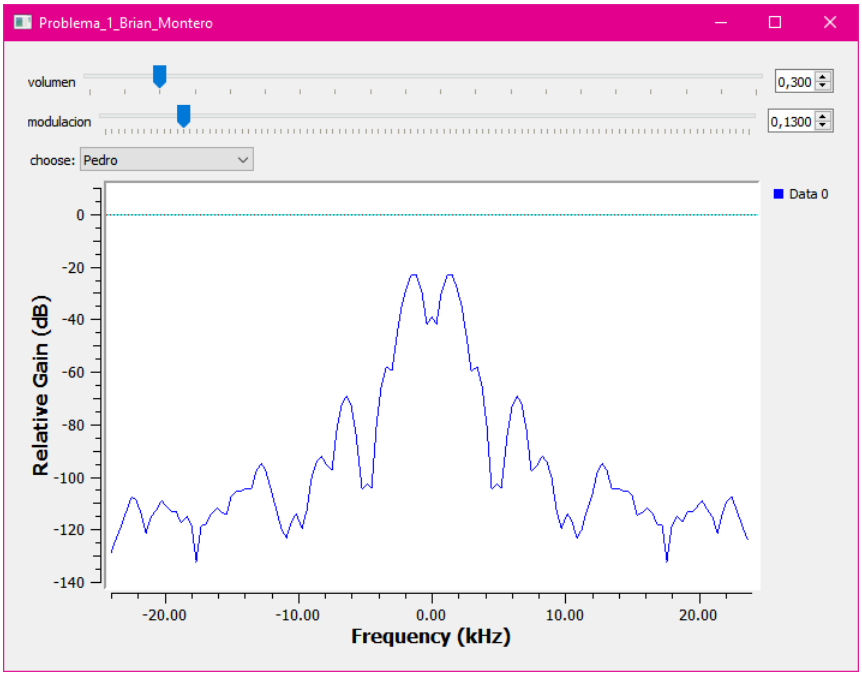



Figura 7: Espectro de la frecuencia de la señal 3

- Señal 4: Discurso Dilios película 300

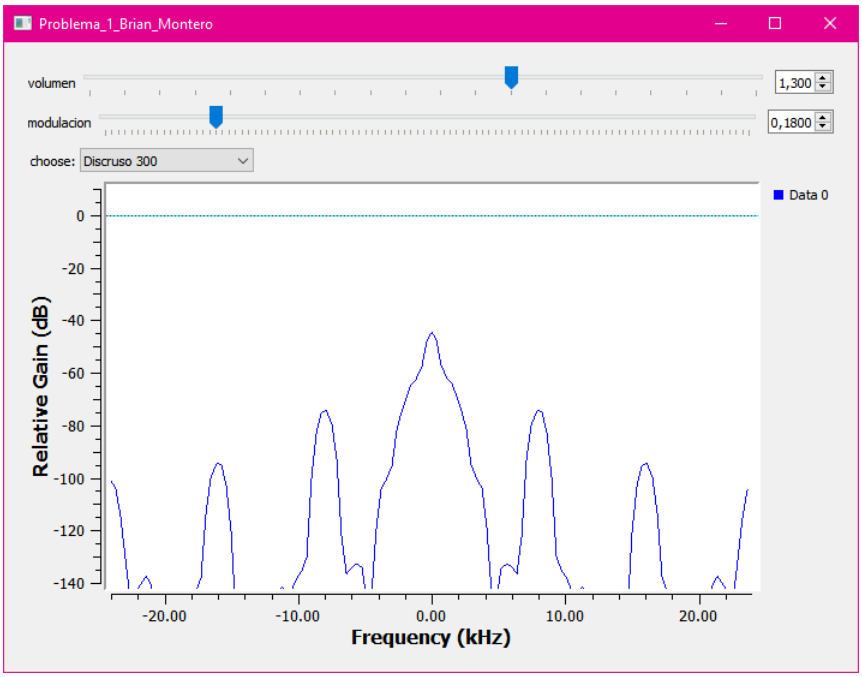
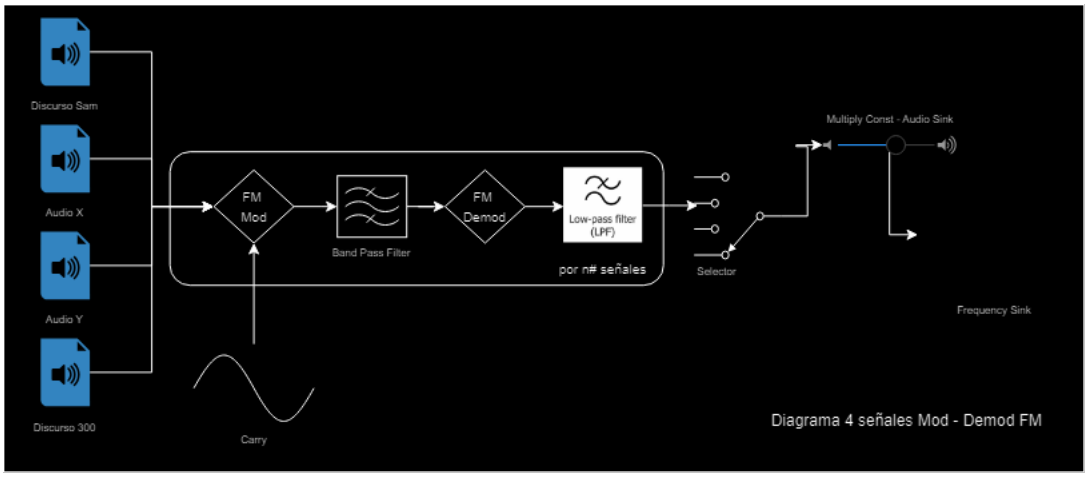



Figura 8: Espectro de la frecuencia de la señal 4

# 2. Problema 2: Modulación en frecuencia

## 2.1. Dibuje un diagrama de bloques en el que explique cómo enviará cuatro señales de audio de forma simultánea moduladas en frecuencia. Explique su estrategia y especifique qué valores debe utilizar para modular los audios.

Para enviar cuatro señales de audio al mismo tiempo en un solo archivo, utilizaremos una estrategia basada en el proceso de modulación en frecuencia (FM) de cada señal de audio y su combinación con una portadora generando asi una única señal compuesta. Esto se puede observar claramente en el siguiente diagrama de bloques:




En el diagrama de bloques, se observan los siguientes componentes: **FM Mod + Band Pass Filter + FM Demod + Low Pass Filter**

- *Audio Source:* Se utiliza como entrada para las señales X e Y de las cuatro señales de audio. Cada fuente de audio proporciona la señal que será modulada. Las otras dos señales vienen del problema 1 que corresponderían al Discurso de Sam y al Discurso de Dilios.
- *FM Mod:* Cada señal de audio se modula en frecuencia utilizando un bloque de Python. Este bloque toma la señal de audio y la combina con una portadora para producir una señal modulada en frecuencia. Los parámetros a configurar son:
  - *Frecuencia de Portadora 1:* 5 kHz
  - *Frecuencia de Portadora 2:* 10 kHz
  - *Frecuencia de Portadora 3:* 15 kHz
  - *Frecuencia de Portadora 4:* 20 kHz
- *Band Pass Filter:* Después de la modulación, se utilizan filtros de paso de banda para asegurar que cada señal modulada en frecuencia esté en el rango adecuado de frecuencia para evitar interferencias entre señales. Los parámetros para los filtros son:
  - *Filtro 1:* 3 kHz - 7 kHz
  - *Filtro 2:* 8 kHz - 12 kHz
  - *Filtro 3:* 13 kHz - 17 kHz
  - *Filtro 4:* 18 kHz - 22 kHz

- *Low Pass Filter:* Este filtro elimina las componentes de alta frecuencia que resultan de la demodularon amplitud, permitiendo que solo las frecuencias bajas (la señal de interés) sean pasadas a la siguiente etapa.

La estrategia es modular cada señal de audio con una portadora distinta, filtrar las señales moduladas y así poder recuperar la señal inicial.

## 2.2. Guarde una señal de audio que contenga las cuatro señales obtenidas del archivo prueba_audio_1, pero esta vez debe ser modulada en frecuencia y el orden debe ser: audio 4, audio X, audio Y, audio 2. Los audios X e Y corresponden a cualquier señal de audio de su elección que tenga por lo menos la duración mínima de todas las señales de audio, es decir, que su tiempo sea mayor a 3 minutos y menor a 5 minutos.

Para esto utilizaremos los bloques de Virtual Sink y Virtual Source, por lo cual podemos dividir el diagrama de bloques de GNU Radio en dos partes, la primera donde juntamos los 4 audios y se modulan en FM, utilizamos los mismos dos audios que se ocuparon en el Problema 1, en conjunto a los otros dos audios como se observan en la figura a continuación:

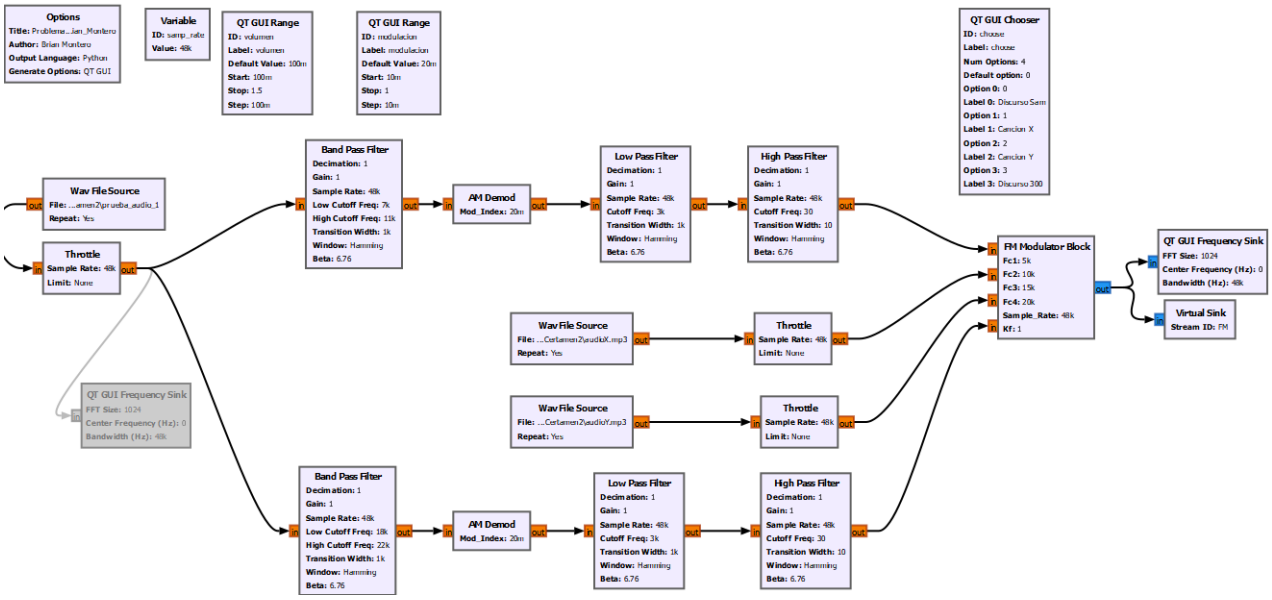



Figura 9: Esquema de modulación FM.

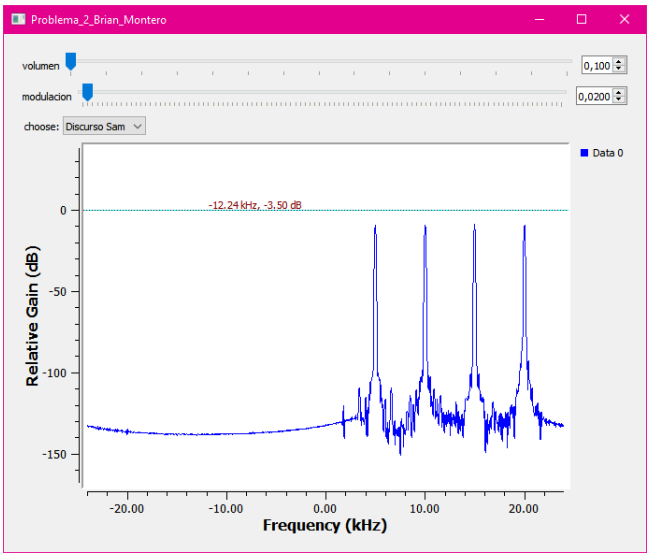



Figura 10: Espectro de Frecuencia para el archivo con las 4 señales moduladas en FM.

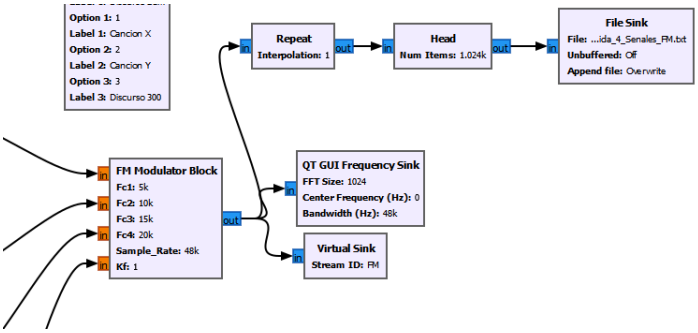



Figura 11: Bloques GNU Radio para guardar la nueva señal

**2.3. Genere un archivo en GNU Radio que actúe como demodulador de las señales. Al ejecutar el archivo, se debe escuchar el mensaje transmitido. Utilice como base el esquema disponible en la figura 1. El archivo debe tener una forma de seleccionar el audio que se está escuchando mientras se ejecuta.**

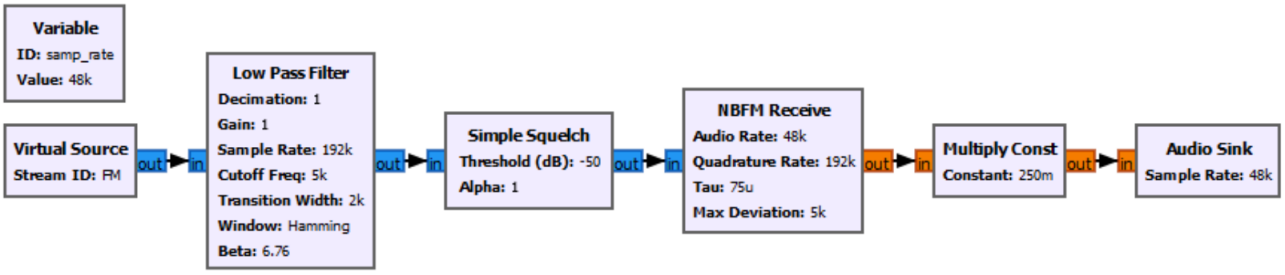



Figura 1: Esquema básico de demodulación FM

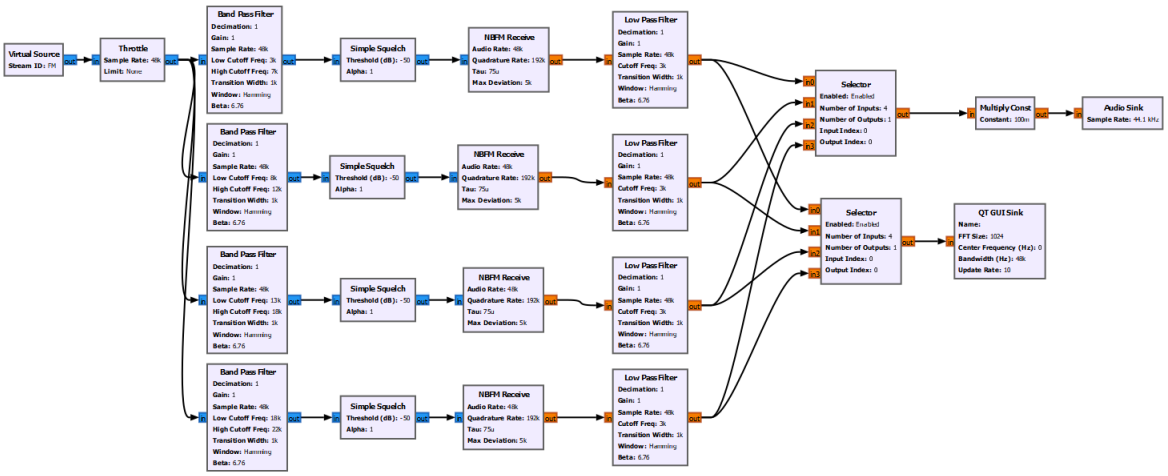


Figura 12: Esquema de demodulación FM

Siguiendo el ejemplo dado arriba solo que incluiremos un *Low Pass Filter* al final del *NBFM Receiver* y el formato para seleccionar sera al igual que la Pregunta 1 a través del selector, con esto podemos seleccionar cual queremos analizar tal como se muestra a continuación:

- Señal 1: Discurso Sam El señor de los Anillos


Figura 13: Espectro de la frecuencia de la señal 1

- Señal 2: Canción X Lucid Dreams

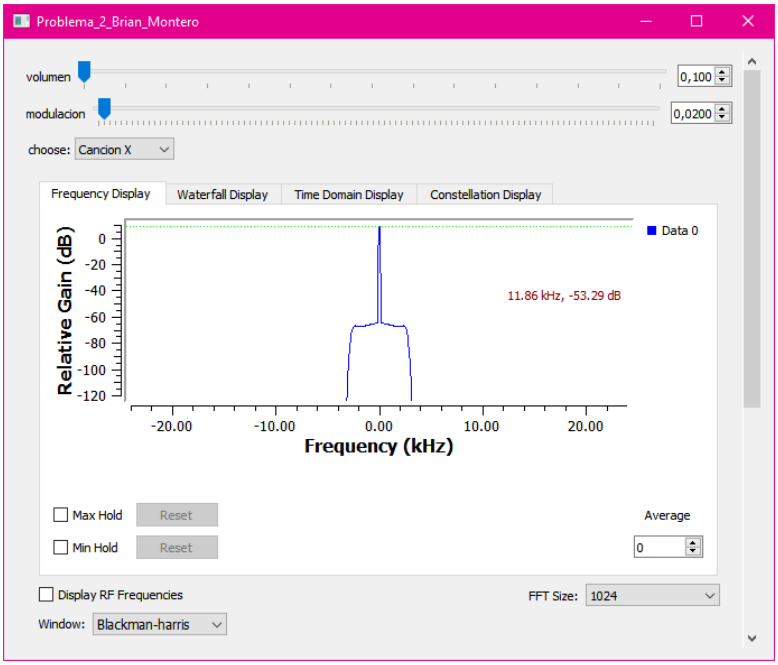



Figura 14: Espectro de la frecuencia de la señal 2

- Señal 3: Canción Y Goosebumps

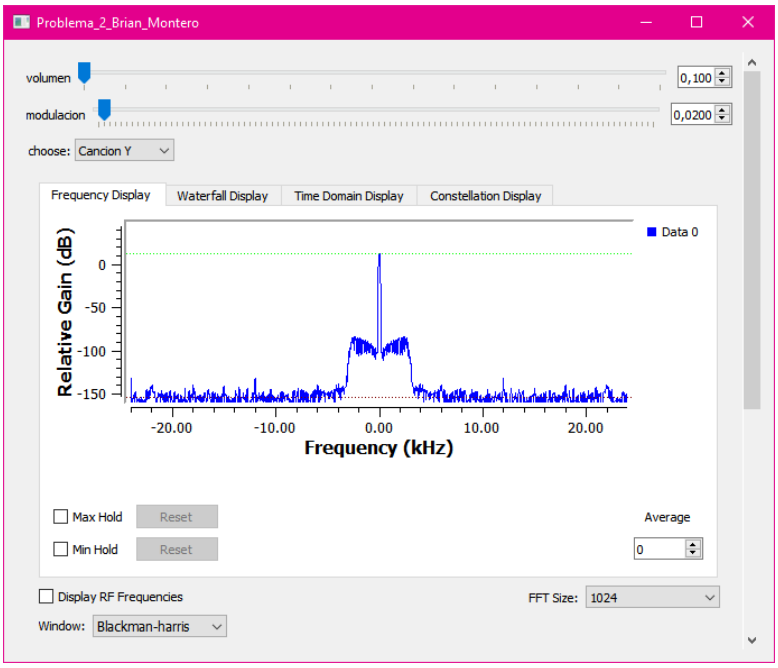



Figura 15: Espectro de la frecuencia de la señal 3

- Señal 4: Discurso Dilios película 300


Figura 16: Espectro de la frecuencia de la señal 4

# 3. Referencias

**Carpeta con los desarrollos para este trabajo** Enlace al códigos GNU, Audios y Bloques de Python