

**Desarrollo implementación de laboratorios de Radio Definido por Software  
RDS- para el laboratorio de telecomunicaciones de la UPB**

Juan David Barrero Lizcano

M.Sc. Alex Alberto Monclou Salcedo

UNIVERSIDAD PONTIFICIA BOLIVARIANA  
ESCUELA DE INGENIERÍA  
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA  
FLORIDABLANCA  
2022

**Desarrollo implementación de laboratorios de Radio Definido por Software  
RDS- para el laboratorio de telecomunicaciones de la UPB**

Juan David Barrero Lizcano

Trabajo de Grado para optar por el título de Ingeniero Electrónico

M.Sc. Alex Alberto Monclou Salcedo

UNIVERSIDAD PONTIFICIA BOLIVARIANA  
ESCUELA DE INGENIERÍA  
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA  
FLORIDABLANCA  
2022

Nota de aceptación:

---

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del Jurado

Dedicado a nuestros padres y familia,  
quienes fueron un gran apoyo a lo largo  
de todos nuestros estudios.

## **AGRADECIMIENTOS**

Al Docente Alex Monclou, quien fue mi director de este proyecto y siempre estuvo atento al trabajo que iba realizando, además, de brindar consejos y libros que fueron de mucha ayuda para seguir aportando más conocimiento a este trabajo.

## CONTENIDO

|                                                                         |    |
|-------------------------------------------------------------------------|----|
| 1. Introducción al Radio Definido por Software.....                     | 24 |
| 1.1 Software necesario.....                                             | 24 |
| 1.1.1 Instalando Python.....                                            | 24 |
| 1.1.2 Instalando GNU radio en Windows.....                              | 27 |
| 1.1.3 Instalación de GNU radio en Linux.....                            | 30 |
| 1.1.4 Instalando Audacity.....                                          | 32 |
| 1.1.5 Instalando ZADIG.....                                             | 34 |
| 1.1.6 Instalar SDR# en Windows.....                                     | 35 |
| 1.1.7 Instalando CubicSDR en Linux.....                                 | 36 |
| 1.2 Hardware necesario.....                                             | 38 |
| 1.2.1 RTL-SDR-Dongle.....                                               | 39 |
| 1.2.2 HackRF-One.....                                                   | 41 |
| 1.2.3 Adalm-Pluto.....                                                  | 42 |
| 1.2.4 USRP.....                                                         | 44 |
| 1.3 Primeros pasos en GNU radio.....                                    | 45 |
| 1.3.1 Bloques source y sink.....                                        | 47 |
| 1.3.2 Operaciones aritméticas con GNU radio.....                        | 51 |
| 1.3.3 Filtros GNU radio.....                                            | 54 |
| 1.3.4 Audio file GNU radio.....                                         | 58 |
| 2. Modulación AM utilizando RDS.....                                    | 61 |
| 2.1 Modulación por amplitud portadora suprimida.....                    | 61 |
| 2.1.1 Doble banda lateral portadora suprimida GNU radio simulación..... | 63 |
| 2.1.2 DSB-SC GNU radio SDR transmisión.....                             | 65 |
| 2.1.3 DSB-SC receptor simulado GNU radio.....                           | 72 |
| 2.1.4 DSB-SC recepción con RDS.....                                     | 75 |
| 2.2 Doble banda lateral gran portadora.....                             | 78 |
| 2.2.1 DSB-LC simulación en GNU radio Rx y TX.....                       | 80 |
| 2.2.2 DSB-LC Tx y Rx utilizando RDS.....                                | 82 |
| 2.3 Modulación por cuadratura QAM.....                                  | 84 |
| 2.3.1 Transmisión de señales QAM GNU radio.....                         | 86 |
| 2.3.2 Recepción de señal QAM GNU radio.....                             | 87 |
| 2.3.3 Tx y Rx utilizando RDS QAM.....                                   | 89 |

|       |                                                       |     |
|-------|-------------------------------------------------------|-----|
| 3.    | Modulación FM utilizando RDS.....                     | 91  |
| 3.1   | VCO en GNU radio.....                                 | 92  |
| 3.1.1 | Tx y Rx con modulación de frecuencia simulación ..... | 96  |
| 3.2   | Modulación de frecuencia banda angosta.....           | 98  |
| 3.2.1 | NBFM simulación GNU radio transmisión .....           | 99  |
| 3.2.2 | Transmisión NBFM utilizando RDS .....                 | 102 |
| 3.2.3 | Receptor NBFM utilizando RDS .....                    | 105 |
| 3.3   | Modulación de frecuencia banda ancha.....             | 109 |
| 3.3.1 | GNU radio para verificar funciones Bessel .....       | 112 |
| 3.3.2 | Captando emisoras de FM con RDS .....                 | 114 |
| 3.3.3 | Tx y Rx WBFM con GNU radio .....                      | 116 |
| 3.3.4 | Transmisión WBFM utilizando el RDS.....               | 118 |
| 4.    | Modulaciones digitales utilizando RDS.....            | 120 |
| 4.1   | modulación ASK.....                                   | 121 |
| 4.1.1 | OOK con GNU radio .....                               | 121 |
| 4.1.2 | Transmitiendo modulación OOK con RDS .....            | 123 |
| 4.1.3 | 4-ASK simulación con GNU radio.....                   | 124 |
| 4.1.4 | Transmitiendo 4-ASK con RDS .....                     | 126 |
| 4.2   | modulación FSK.....                                   | 127 |
| 4.2.1 | Modulación binaria FSK GNU radio.....                 | 128 |
| 4.2.2 | Transmisión 2FSK con RDS.....                         | 130 |
| 4.2.3 | Modulación CFSK con GNU radio .....                   | 131 |
| 4.2.4 | Transmisión CFSK utilizando RDS .....                 | 133 |
| 4.2.5 | Modulación GFSK simulación GNU radio.....             | 134 |
| 4.2.6 | Transmisión y recepción GFSK con GNU radio .....      | 135 |
| 4.2.7 | Tx y Rx con GFSK utilizando RDS .....                 | 139 |
| 4.3   | modulación PSK.....                                   | 140 |
| 4.3.1 | 2-PSK simulación con GNU radio.....                   | 141 |
| 4.3.2 | 4-PSK simulación con GNU radio.....                   | 143 |
| 4.3.3 | 8-PSK GNU radio .....                                 | 145 |
| 4.3.4 | Problemas con la comunicación digital.....            | 149 |
| 4.3.5 | Transmisión y recepción 8-PSK GNU radio.....          | 156 |
| 4.3.6 | Rx y Tx PSK con RDS .....                             | 159 |
| 5.    | Televisión digital con RDS .....                      | 161 |

|                                                            |     |
|------------------------------------------------------------|-----|
| 5.1 Transmisión de televisión digital ATSC con SDR.....    | 161 |
| 5.2 Recepción de televisión digital ATSC con SDR .....     | 164 |
| 6. Conclusiones .....                                      | 166 |
| 7. Referencias.....                                        | 167 |
| 8. Anexos.....                                             | 168 |
| 8.1 Links a los códigos y resultados de las practicas..... | 168 |
| 8.2 Modelo para realizar guías de laboratorio .....        | 168 |
| 8.3 Guías de laboratorio.....                              | 172 |

## LISTA DE FIGURAS

|                                                        |    |
|--------------------------------------------------------|----|
| Figura 1.1 Página de fondo .....                       | 25 |
| Figura 1.2 Versiones de Python .....                   | 25 |
| Figura 1.3 Instalando Python en el computador .....    | 26 |
| Figura 1.4 Instalador de Python .....                  | 26 |
| Figura 1.5 Verificar Python.....                       | 27 |
| Figura 1.6 Página de instalación de GNU radio .....    | 28 |
| Figura 1.7 Instalando GNU radio.....                   | 28 |
| Figura 1.8 Wizard de GNU radio .....                   | 29 |
| Figura 1.9 GNU radio .....                             | 29 |
| Figura 1.10 Instalando repositorio .....               | 30 |
| Figura 1.11 Instalado GNU radio Ubuntu .....           | 31 |
| Figura 1.12 GNU radio en Ubuntu.....                   | 31 |
| Figura 1.13 Página de AudaCity.....                    | 32 |
| Figura 1.14 Aplicación de AudaCity .....               | 33 |
| Figura 1.15 Página de Zadig .....                      | 34 |
| Figura 1.16 Aplicación de Zadig .....                  | 35 |
| Figura 1.17 Descarga la versión de SDR .....           | 35 |
| Figura 1.18 Seleccionar la aplicación SDR .....        | 36 |
| Figura 1.19 Aplicación de SDR .....                    | 36 |
| Figura 1.20 Instalando CubicSDR .....                  | 37 |
| Figura 1.21 CubicSDR .....                             | 38 |
| Figura 1.22 Diagrama de bloques básico de un RDS ..... | 38 |
| Figura 1.23 Diagrama de RDS .....                      | 39 |
| Figura 1.24 RTL SDR DAB-FM-DVB-T-FC1001 .....          | 40 |
| Figura 1.25 RTL-SDR.com.....                           | 40 |
| Figura 1.26 HackRF-One .....                           | 41 |
| Figura 1.27 Adalm-Pluto.....                           | 42 |
| Figura 1.28 abriendo GNU radio prompt .....            | 43 |
| Figura 1.29 Instalando Adalm pluto .....               | 44 |
| Figura 1.30 Bloques de adaml pluto .....               | 44 |
| Figura 1.31 USRP .....                                 | 45 |
| Figura 1.32 GNU radio .....                            | 46 |
| Figura 1.33 Bloque signal source .....                 | 48 |

|                                                     |    |
|-----------------------------------------------------|----|
| Figura 1.34 Bloque time sink .....                  | 48 |
| Figura 1.35 Bloque frequency sink .....             | 49 |
| Figura 1.36 Resolución del primer ejercicio .....   | 49 |
| Figura 1.37 resultado del ejercicio 1.....          | 50 |
| Figura 1.38 Operaciones básicas.....                | 51 |
| Figura 1.39 bloque selector .....                   | 52 |
| Figura 1.40 Gui chooser .....                       | 53 |
| Figura 1.41 Resolución del ejercicio propuesto..... | 53 |
| Figura 1.42 Gui range bloque.....                   | 54 |
| Figura 1.43 Parámetros filtro pasa bajo .....       | 55 |
| Figura 1.44 Filtro pasa alto parámetros.....        | 56 |
| Figura 1.45 Filtro pasa banda y rechaza banda ..... | 56 |
| Figura 1.46 Problema tres resuelto .....            | 57 |
| Figura 1.47 Salida espera del problema.....         | 57 |
| Figura 1.48 Grabar audio con Audacity .....         | 58 |
| Figura 1.49 Exportar audio formato wav .....        | 59 |
| Figura 1.50 bloque wav source .....                 | 59 |
| Figura 1.51 Bloque audio sink .....                 | 60 |
| Figura 1.52 bloque wav source .....                 | 60 |
| Figura 1.53 Ejercicio 4.....                        | 60 |
| Figura 2.1 DSB-SC.....                              | 61 |
| Figura 2.2 Convolución grafica.....                 | 62 |
| Figura 2.3 Resultado de la convolución .....        | 62 |
| Figura 2.4 DSB-SC en GNU radio.....                 | 63 |
| Figura 2.5 Modulación de tono DSB-SC .....          | 64 |
| Figura 2.6 Señal de audio DSB-SC.....               | 64 |
| Figura 2.7 Instalando drivers RDS .....             | 65 |
| Figura 2.8 Drivers RDS .....                        | 65 |
| Figura 2.9 Drivers instalados correctamente .....   | 66 |
| Figura 2.10 bloque Osmocom sink.....                | 66 |
| Figura 2.11 DSBSC-RDS_TX.....                       | 67 |
| Figura 2.12 primeros pasos SDR# .....               | 68 |
| Figura 2.13 ajustando SDR# .....                    | 69 |
| Figura 2.14 Ajustar parámetros en SDR# .....        | 69 |

|                                                               |    |
|---------------------------------------------------------------|----|
| Figura 2.15 Sintonizando estaciones de FM .....               | 70 |
| Figura 2.16 Ajustes para recibir señales AM .....             | 71 |
| Figura 2.17 Antes de transmitir la señal .....                | 71 |
| Figura 2.18 Señal recibida DSB-SC .....                       | 72 |
| Figura 2.19 Bloque recepción DSBSC .....                      | 73 |
| Figura 2.20 Resultado de la ecuación 3 .....                  | 73 |
| Figura 2.21 Receptor GNU radio DSBSC .....                    | 74 |
| Figura 2.22 Receptor DSB-SC .....                             | 75 |
| Figura 2.23 Resultados de receptor DSBSC .....                | 75 |
| Figura 2.24 Bloque RTL source.....                            | 76 |
| Figura 2.25 Receptor con el RDS.....                          | 77 |
| Figura 2.26 Señal de modulada .....                           | 77 |
| Figura 2.27 Ecuación DSBLC.....                               | 78 |
| Figura 2.28 Resultado de operación .....                      | 79 |
| Figura 2.29 Diagrama de bloques para transmisión DSB_LC ..... | 79 |
| Figura 2.30 Rx y Tx DSBLC .....                               | 80 |
| Figura 2.31 Bloque AM demod.....                              | 81 |
| Figura 2.32 Resultados DSBLC tono .....                       | 81 |
| Figura 2.33 Señales recuperadas tono y audio .....            | 82 |
| Figura 2.34 Transmisión RDS DSB_LB .....                      | 82 |
| Figura 2.35 Receptor DSB-LC.....                              | 83 |
| Figura 2.36 Resultado de la recepción .....                   | 83 |
| Figura 2.37 Sistema QAM .....                                 | 84 |
| Figura 2.38 Ecuación en el dominio de la frecuencia .....     | 85 |
| Figura 2.39 Resultado al resolver la ecuación.....            | 85 |
| Figura 2.40 Sistema receptor QAM .....                        | 85 |
| Figura 2.41 QAM transmisión.....                              | 86 |
| Figura 2.42 Convierte señales reales a complejas .....        | 86 |
| Figura 2.43 Señales transmitidas .....                        | 87 |
| Figura 2.44 Recepción sistema QAM .....                       | 87 |
| Figura 2.45 Sistema completo Tx y Rx QAM.....                 | 88 |
| Figura 2.46 Señal recuperada QAM.....                         | 88 |
| Figura 2.47 Transmisión QAM con RDS .....                     | 89 |
| Figura 2.48 Señal recibida utilizando SDR#.....               | 89 |

|                                                                     |     |
|---------------------------------------------------------------------|-----|
| Figura 2.49 Receptor de la señal QAM con RDS .....                  | 90  |
| Figura 2.50 Señal de audio recibida .....                           | 90  |
| Figura 3.1 Espectro del ancho de banda .....                        | 92  |
| Figura 3.2 Bloques VCO GNU radio .....                              | 92  |
| Figura 3.3 Parámetros VCO .....                                     | 93  |
| Figura 3.4 Prueba del bloque VCO .....                              | 93  |
| Figura 3.5 Resultado de variar la constante de entrada al VCO ..... | 94  |
| Figura 3.6 Modulación FM de un tono .....                           | 95  |
| Figura 3.7 Modulación FM .....                                      | 95  |
| Figura 3.8 Representación espectro de modulación FM .....           | 96  |
| Figura 3.9 VCO alterno GNU radio .....                              | 96  |
| Figura 3.10 Transmisión con modulación FM .....                     | 97  |
| Figura 3.11 FM demod GNU radio .....                                | 97  |
| Figura 3.12 Recepción de datos modulación FM .....                  | 98  |
| Figura 3.13 Obtención de la información .....                       | 98  |
| Figura 3.142 NBFM diagrama de bloques .....                         | 99  |
| Figura 3.15 bloque NBFM .....                                       | 100 |
| Figura 3.16 parámetros del bloque NBFM .....                        | 100 |
| Figura 3.17 Transmisor NBDM .....                                   | 101 |
| Figura 3.18 Modulación NBFM con señal de audio .....                | 101 |
| Figura 3.19 Modulación NBFM con un tono .....                       | 102 |
| Figura 3.20 Transmisión NBFM con RDS .....                          | 103 |
| Figura 3.21 Banda sin datos .....                                   | 104 |
| Figura 3.22 Cambiando el tipo de modulación .....                   | 104 |
| Figura 3.23 Datos recibido con NBFM .....                           | 105 |
| Figura 3.24 Bloque NBFM receptor .....                              | 105 |
| 2Figura 3.25 Configuración para bloque receptor .....               | 106 |
| Figura 3.26 Diagrama de bloques para receptor NBFM .....            | 106 |
| Figura 3.27 Simulación Tx y Rx NBFM .....                           | 107 |
| Figura 3.28 Señal recibida con NBFM .....                           | 108 |
| Figura 3.29 Receptor NBFM con RDS .....                             | 108 |
| Figura 3.30 Señal recibida utilizando RDS .....                     | 109 |
| Figura 3.31 Coeficientes de Bessel .....                            | 111 |
| Figura 3.32 Espectro ante diferentes índices de modulación .....    | 113 |

|                                                         |     |
|---------------------------------------------------------|-----|
| Figura 3.33 Prueba de Bessel .....                      | 113 |
| Figura 3.34 Valores de coeficientes Bessel .....        | 114 |
| Figura 3.35 Receptor de emisoras .....                  | 114 |
| Figura 3.36 Receptor de emisoras .....                  | 115 |
| Figura 3.37 Receptor de emisora .....                   | 115 |
| Figura 3.38 WBFM transmit.....                          | 116 |
| Figura 3.39 Transmisor WBFM .....                       | 116 |
| Figura 3.40 Tx y Rx WBFM .....                          | 117 |
| Figura 3.41 Resultados WBFM tx y rx.....                | 117 |
| Figura 3.42 Transmisor WBFM con RDS .....               | 118 |
| Figura 3.43 Ajuste SDR.....                             | 118 |
| Figura 3.44 Modulación WBFM .....                       | 119 |
| Figura 3.45 Receptor de señal transmitida con RDS ..... | 119 |
| Figura 4.1 Modulación digitales.....                    | 120 |
| Figura 4.2 Modulación OOK .....                         | 121 |
| Figura 4.3 OOK con GNU radio.....                       | 122 |
| Figura 4.4 resultados de Modulación OOK .....           | 122 |
| Figura 4.5 Transmisión OKK con RDS.....                 | 123 |
| Figura 4.6 Ancho de banda de OOK modulación .....       | 123 |
| Figura 4.7 Generador de Bits .....                      | 124 |
| Figura 4.8 Tren de bits .....                           | 124 |
| Figura 4.9 Empaquetador de bits .....                   | 125 |
| Figura 4.10 Interpolador de datos .....                 | 125 |
| Figura 4.11 modulación 4ASK.....                        | 125 |
| Figura 4.12 resultado de modulación 4-ASK.....          | 126 |
| Figura 4.13 Transmitiendo 4-ASK RDS.....                | 126 |
| Figura 4.14 Ancho de banda .....                        | 127 |
| Figura 4.15 modulación FSK.....                         | 127 |
| Figura 4.16 Receptor FSK.....                           | 128 |
| Figura 4.17 2FSK .....                                  | 128 |
| Figura 4.18 Transmisión y recepción FSK .....           | 129 |
| Figura 4.19 Recuperando bits 2FSK .....                 | 129 |
| Figura 4.20 Transmisión 2FSK.....                       | 130 |
| Figura 4.21 2Fsk transmitida .....                      | 130 |

|                                                              |     |
|--------------------------------------------------------------|-----|
| Figura 4.22 Receptor 2FSK con RDS .....                      | 131 |
| Figura 4.23 CFSK.....                                        | 131 |
| Figura 4.25 MFSK .....                                       | 132 |
| Figura 4.26 MSFK y CFSK.....                                 | 132 |
| Figura 4.27 Cantidad de bits empaquetados.....               | 133 |
| Figura 4.28 Modulación 4-CFSK .....                          | 133 |
| Figura 4.29 Transmisión 4CFSK con RDS.....                   | 134 |
| Figura 4.30 recepción 4CFSK .....                            | 134 |
| Figura 4.31 GFSK .....                                       | 135 |
| Figura 4.32 GMFSK .....                                      | 135 |
| Figura 4.33 GFSK GNU radio bloque.....                       | 135 |
| Figura 4.34 Demodulador GFSK GNU radio .....                 | 136 |
| Figura 4.35 Bloque file source.....                          | 137 |
| Figura 4.36 Parámetros de file source .....                  | 137 |
| Figura 4.37 Bloque Stream to Tagged Stream.....              | 138 |
| Figura 4.38 Transmisión de datos GFSK .....                  | 138 |
| Figura 4.39 File sink .....                                  | 138 |
| Figura 4.40 Receptor GFSK.....                               | 139 |
| Figura 4.41 Transmisión y Recepción GFSK .....               | 139 |
| Figura 4.42 Transmisión RDS GFSK .....                       | 139 |
| Figura 4.43 Receptor GFSK con RDS.....                       | 140 |
| Figura 4.44 PSK demostración.....                            | 140 |
| Figura 4.45 Diagrama de constelación modulación 16-PSK ..... | 141 |
| Figura 4.46 Diagrama de constelación.....                    | 141 |
| Figura 4.47 BPSK constelación.....                           | 142 |
| Figura 4.48 BPSK GNU radio.....                              | 142 |
| Figura 4.49 Constelación BPSK GNU radio .....                | 143 |
| Figura 4.50 Constelación QPSK.....                           | 144 |
| Figura 4.51 Noise Source.....                                | 144 |
| Figura 4.52 QPSK GNU radio .....                             | 145 |
| Figura 4.53 QPSK constelación .....                          | 145 |
| Figura 4.54 Constelación 8PSK .....                          | 146 |
| Figura 4.55 Python block GNU radio.....                      | 147 |
| Figura 4.56 escoger el editor de código .....                | 147 |

|                                                                      |     |
|----------------------------------------------------------------------|-----|
| Figura 4.57 Plantilla código GNU radio .....                         | 147 |
| Figura 4.59 MPSK .....                                               | 148 |
| Figura 4.60 8PSK .....                                               | 149 |
| Figura 4.61 Constelación 8PSK .....                                  | 149 |
| Figura 4.62 Coseno alzado .....                                      | 150 |
| Figura 4.63 PSK mod .....                                            | 150 |
| Figura 4.64 Channel model GNU radio .....                            | 151 |
| Figura 4.65 SIS GNU radio.....                                       | 151 |
| Figura 4.66 Señal distorsionada por la desviación de frecuencia..... | 152 |
| Figura 4.67 polyphase clock.....                                     | 152 |
| Figura 4.68 Recuperando la señal .....                               | 153 |
| Figura 4.69 Señal recuperada .....                                   | 154 |
| Figura 4.70 Ecualizador .....                                        | 154 |
| Figura 4.71 Ecualizador para evitar SIS .....                        | 155 |
| Figura 4.72 Evitando SIS con ecualizador .....                       | 155 |
| Figura 4.73 SIS .....                                                | 156 |
| Figura 4.74 Eliminado problema de frecuencia offset .....            | 156 |
| Figura 4.76 De modulador PSK.....                                    | 158 |
| Figura 4.77 Transmisión 8-PSK con GNU radio.....                     | 158 |
| Figura 4.78 Receptor modulación 8-PSK .....                          | 158 |
| Figura 4.79 Transmisión exitosa .....                                | 159 |
| Figura 4.80 Transmisor PSK RDS.....                                  | 159 |
| Figura 4.81 Ajustando el tipo de transmisión.....                    | 160 |
| Figura 4.82 paquete de bits.....                                     | 160 |
| Figura 4.83 Receptor PSK con RDS .....                               | 160 |
| Figura 5.1 Convertidor de formato de video .....                     | 162 |
| Figura 5.2 Codificación de video tv .....                            | 162 |
| Figura 5.3 Modulador ATSC .....                                      | 163 |
| Figura 5.4 Transmisión de televisión digital .....                   | 163 |
| Figura 5.5 Señal ATSC transmitida .....                              | 164 |
| Figura 5.7 SM player .....                                           | 165 |

## LISTA DE TABLAS

|                                            |     |
|--------------------------------------------|-----|
| Tabla 1 Características de RTL-dongle..... | 40  |
| Tabla 2 Características SDR-RTL-v3 .....   | 41  |
| Tabla 3 Hackrf-one .....                   | 42  |
| Tabla 4 Características Adalm pluto.....   | 43  |
| Tabla 5 Características USRP.....          | 45  |
| Tabla 6 Emisoras FM de Bucaramanga .....   | 103 |
| Tabla 7 Coeficientes de Bessel .....       | 112 |
| Tabla 8 Tabla de verdad QPSK.....          | 143 |
| Tabla 9 8PSK tabla de verdad.....          | 146 |
| Tabla 10 tabla de verdad 8-PSK .....       | 157 |

## **GLOSARIO**

RDS: Radio Definido por Software

USRP: Universal Software Radio Peripheral

AM: Amplitud Modulada

FM: Frecuencia Modulada

ASK: Amplitude Shift Keying

FSK: Frequency Shift Keying

PSK: Phase Shift Keying

Bit: Binary Digit

Byte: Conjunto de 8 bits

Poo: Programación Orientada a Objetos

Poe: Programación Orientada a Eventos

## RESUMEN GENERAL DE TRABAJO DE GRADO

**TITULO:** Desarrollo implementación de laboratorios de Radio Definido por Software RDS- para el laboratorio de telecomunicaciones de la UPB

**AUTOR(ES):** Juan David Barrero Lizcano

**PROGRAMA:** Facultad de Ingeniería Electrónica

**DIRECTOR(A):** Alex Alberto Monclou Salcedo

### RESUMEN

Las comunicaciones hoy en día son algo primordial en todo el mundo, saber como funcionan es un conocimiento esencial para todo ingeniero, por eso en este trabajo de grado se realizaron cinco practicas de laboratorio con ayuda de los dispositivos SDR y software GNU radio, con el fin de llevar la teoría vista en la materias de comunicaciones I y II de la Universidad Pontificia Bolivariana a la practica y de esta forma reforzar el conocimiento que tiene los estudiantes sobre modulación en amplitud, modulación en frecuencia y comunicaciones digitales.

### PALABRAS CLAVE:

SDR,AM,FM,ASK,FSK,PSK.GNU radio



V° B° DIRECTOR DE TRABAJO DE GRADO

## GENERAL SUMMARY OF WORK OF GRADE

**TITLE:** Development and implementation of RDS Software Defined Radio laboratories for the UPB telecommunications laboratory

**AUTHOR(S):** Juan David Barrero Lizcano

**FACULTY:** Facultad de Ingeniería Electrónica

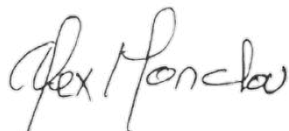
**DIRECTOR:** Alex Alberto Monclou Salcedo

### ABSTRACT

Communications today are essential throughout the world, knowing how they work is an essential knowledge for every engineer, that is why in this degree work five laboratory practices were carried out with the help of SDR devices and GNU radio software, with the In order to put the theory seen in the subjects of communications I and II of the Universidad Pontificia Bolivariana to practice and in this way reinforce the knowledge that students have about amplitude modulation, frequency modulation and digital communications.

### KEYWORDS:

SDR,AM,FM,ASK,FSK,PSK.GNU radio



**V° B° DIRECTOR OF GRADUATE WORK**

## INTRODUCCIÓN

Las comunicaciones hoy en día son algo primordial en todo el mundo, en medio de la pandemia se han destacado como la base de todo tipo de reuniones y transferencia de datos. Entender cómo funciona y saber cómo implementarlas es una de la competencia necesaria para los profesionales que se quieran ir por la rama de telecomunicaciones. Por esta razón es que se quiere hacer énfasis en el entendimiento de nuevas herramientas que existen para poder hacer su uso más entendible y que no todo quede en papel o en simulaciones.

Una de esas herramientas que se ha mencionado anteriormente es el uso de radio definido por software (SDR), los cuales son un sistema de radio comunicaciones que tiene la ventaja de ser programables para poder recibir y transmitir información con distintos tipos de modulaciones y protocolos en distintas bandas del espectro electromagnético, además tienen la ventaja de que no son costosos como los equipos profesionales y un estudiante lo puede adquirir fácilmente y tener en su casa para su aprendizaje.

Para poder realizar la programación de estos dispositivos existen varias herramientas, estos se pueden programar directamente en C++ utilizando ciertas librerías o en Python de igual manera, sin embargo, realizar el trabajo en estas plataformas puede resultar tedioso o pesado dependiendo de la habilidad de programar de cada persona. Pero gracias a GNU radio que es un software de código abierto también se puede programar los SDR en un nivel de programación más alto (diagrama de bloques), este software se encarga de traducir esos bloques a código de Python o C++, para ser interpretados y ejecutados.

Durante este trabajo se tiene la meta de realizar cinco guías de laboratorio donde se pueda apreciar el uso de los SDR en ciertos tipos de enlaces y como es la programación de estos. Todo es esto para poder ser implementados en los laboratorios de la materia de comunicaciones de la carrera de ingeniería electrónica de la Universidad Pontificia Bolivariana.

## **OBJETIVOS**

### **OBJETIVO GENERAL**

- Diseñar e implementar prácticas para el laboratorio de telecomunicaciones de la facultad de ingeniería eléctrica y electrónica de la Universidad Pontificia Bolivariana mediante RDS (Radio Definido por Software).

### **OBJETIVOS ESPECÍFICOS**

- Identificar herramientas software para aplicar en el laboratorio de comunicaciones de la facultad de ingeniería eléctrica y electrónica de la Universidad Pontificia Bolivariana.
- Diseñar códigos y diagramas de bloques sobre cómo utilizar los RDS para su uso en los laboratorios de telecomunicaciones.
- Contrastar el funcionamiento de distintos tipos de modulación en la etapa de transmisión y recepción.
- Desarrollar guías de laboratorio para las prácticas de RDS

## MARCO TEORICO

### Procesado digital de señales analógicas

Al trabajar con un sistema de radio comunicación donde los componentes ahora son remplazados con software es necesario utilizar técnicas que permitan convertir las señales analógicas que viajan por el aire en señales que estos dispositivos puedan leer e interpretar, de igual manera es necesario poder revertir este procedimiento para volver a transmitir las señales nuevamente al aire.

### Filtro de señales para prevenir el solapamiento

Para poder convertir las señales es necesario tener ciertos criterios para no perder información, uno de estos es el conocido como teorema de Nyquist el cual indica que la máxima frecuencia en que es posible recuperar datos de una señal solo es posible si la tasa de muestreo es superior al doble de la frecuencia de su ancho de banda.

$$F_s = 2F_{max}$$

*Ecuación 1 Teorema de Nyquist*

### Conversión analógica a digital

“Los convertidores de aproximaciones sucesivas son de tiempo de conversión constante y son uno de los más utilizados en la actualidad para sistemas de adquisición de datos debido a que permite una velocidad de conversión considerable y una alta resolución a un bajo coste.” (Rios, 2015). Por esta razón son el modelo que más se usa a la hora de implementar un ADC (Conversor analógica a digital)

### Conversión digital a analógica

“Los convertidores digital-analógico tipo sigma-delta, surgieron a partir de la necesidad de poder tratar señales de audio digital con muy buena calidad y a bajo coste. Estos convertidores son muy sencillos y aportan una muy buena linealidad a resoluciones muy elevada” (Rios, 2015)

### Filtros para tratar el ruido producido por lectura o conversiones

Al tener que captar las señales utilizando las antenas de los RDS el ruido es una variable inevitable, por eso es necesario realizar algún tipo de filtrado para que la señal con la que se va a trabajar sea la correcta y poder recuperar toda la

información de manera clara. Para eso hay varios tipos de filtro que se pueden implementar, claramente en los SDR todos estos filtros son totalmente digitales.

## **Modulación y Demodulación para sistemas de comunicaciones**

Para los sistemas de telecomunicaciones es importante tener bien definido cuales son los métodos de modulación que se van a utilizar a la hora de transmitir información ya que de esta forma logran mandar la señal una larga distancia con el menor número de interferencias posibles producidas por el ambiente, entre los sistemas de modulación más usados se encuentran:

### **Modulación de amplitud**

La modulación por amplitud es la más conocida, esta consiste en ir variando la amplitud de la señal moduladora a partir de la amplitud de la portadora, y todo se base de la siguiente ecuación:

$$Am(t) = f(t) * \sin(Wct)$$

*Ecuación 2 Modulación AM*

### **Modulación de frecuencia**

Por la necesidad de disminuir el ruido transmitido por la modulación AM nació este nuevo método que pretende manipular la frecuencia, esto es un gran avance porque ahora la señal de ruido siempre presente en todo sistema posee el mismo valor, por lo que ahora la información que viaja por el aire es más inmune a su efecto. Y en este tipo de modulación la ecuación que representa su comportamiento es:

$$Fm(t) = A\cos(f(t))$$

*Ecuación 3 Modulación FM*

### **Modulación de fase**

Modulación de fase (PM). Es un proceso donde el parámetro de la señal portadora que variará de acuerdo con señal moduladora es la fase, manteniendo la frecuencia y la amplitud constante, es un tipo de modulación exponencial al igual que la modulación de frecuencia. Y se puede describir mediante la siguiente ecuación.

$$Pm(t) = A\sin(\omega t + f(t))$$

*Ecuación 4 Modulación PM*

## **METODOLOGÍA**

Para la elaboración del presente proyecto de grado se consultó sobre los SDR y como poder utilizarlo para realizar modulaciones digitales y analógicas utilizando el software de GNU radio. Para esto se realizó cinco prácticas en donde el estudiante pone a prueba el conocimiento adquirido a lo largo de este libro, el cual explica la teoría detrás de las modulaciones y el uso de los bloques principales que tiene GNU radio para ofrecer.

Durante el desarrollo de la practicas se implementaron cinco pasos con el fin de obtener una ruta de trabajo óptima, el primer capítulo del presente trabajo se enfoca en guiar al estudiante sobre como es el entorno de trabajo de GNU radio antes de pasar al ver la primera modulación, seguido de esto se muestra la teoría de detrás de cada modulación con diagrama de bloques sencillos de entender, el tercer paso busca explicar más fondo lo que hay detrás de estos bloques para que se puede ir avanzado a temas más difíciles.

El cuarto paso pone a prueba el estudiante al dejar de un lado las simulaciones y comenzar a utilizar los SDR para realizar comunicaciones entre dos dispositivos y así enfrentarse a los tipos de problema se enfrente al realizar una comunicación con el ruido del ambiente y los problemas de propagación.

Finalmente se desarrollará cinco guías de trabajo, las cuales se podran implementar en el laboratorio de comunicaciones de la Universidad Pontificia Bolivariana, con el fin de poner en contacto la práctica con la teoría vista en clase.

## **DOCUMENTACIÓN**

- Se hizo la revisión sobre tipos de SDR
- Se estudió sobre modulación de amplitud
- Se estudió sobre modulación de frecuencia
- Se estudió sobre modulación digital (ASK FSK y PSK)
- Se estudió sobre televisión digital

## **SELECCIÓN DE CONTENIDO**

- Teniendo en cuenta la guía que ofrece GNU radio se diseñó el diagrama de bloques para que fuera sencillo de interpretar
- Se realizó una valoración de la información para estructurar el contenido de manera secuencial
- Se escogieron los temas de tal forma que se complementaran a medida que se avanza en la implementación

## **CONSTRUCCIÓN DE LA GUÍAS**

- Se seleccionaron los objetivos a cumplir en cada una de las guías a desarrollar
- Se eligieron los contenidos teóricos relacionados con la aplicación técnica de cada guía

# 1. Introducción al Radio Definido por Software

RDS es “una tecnología de radiocomunicación en la que algunos componentes electrónicos son reemplazados por programas de computadora, el objetivo es proporcionar soluciones flexibles para la innovación en el diseño e implementación de los futuros sistemas de comunicación inalámbricos” (Gareth Montenegro Chaidez, 2016)

Mediante RDS es posible transmitir y recibir información con distintos métodos de modulación, y ayuda a resolver el problema de la actualización de los equipos de radiofrecuencia debido al avance rápido de estas tecnologías y su costo de renovación. Es así como, la combinación hardware y software de los sistemas RDS optimizan y se adaptan a los estándares que van apareciendo.

Estos dispositivos permiten monitorear distintas áreas del espectro radioeléctrico y el rango que pueden cubrir depende del SDR que se esté utilizando y de acuerdo con lo ofrecido en el mercado especializado y con los característicos hardware y software requeridos. Debido a la gran variedad de SDR, los precios no son llevados, lo que permite desarrollar proyectos de radiocomunicaciones de bajo costo y adaptables a cualquier sistema de telecomunicaciones inalámbricas.

La herramienta principal para el uso de estos dispositivos es una computadora y la instalación de programas libres y apropiados que ofrecen al usuario una interfaz amigable para su operación. En estos dispositivos se pueden programar en lenguaje C o Python los diferentes tipos de modulación y el mensaje que se requiera enviar.

## 1.1 Software necesario

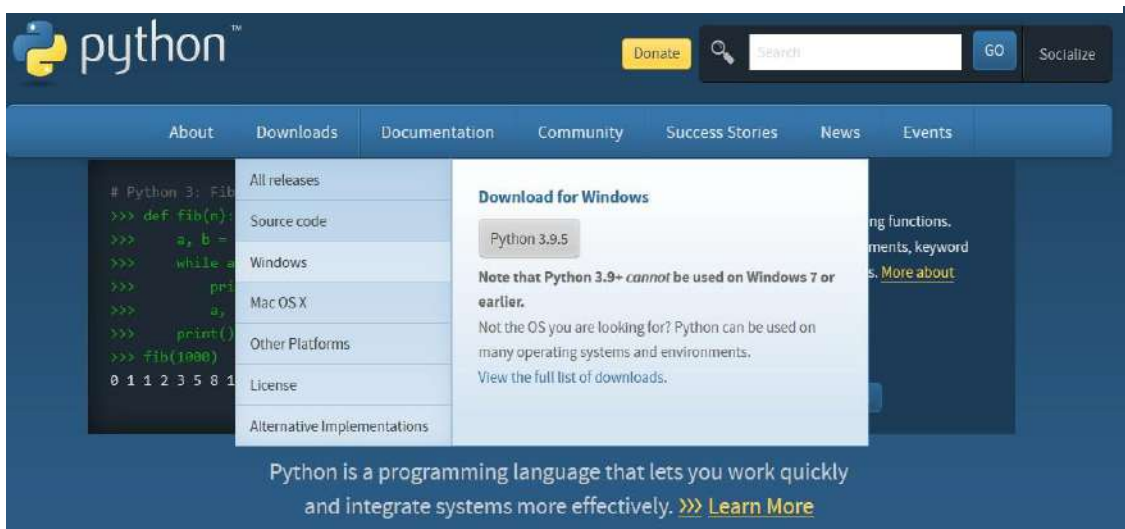
El objetivo del trabajo de grado es presentar como programar los RDS utilizando el lenguaje de programación Python dada su versatilidad y popularidad en el momento y la variedad de librerías que se encuentran, lo cual motiva a que los estudiantes de telecomunicaciones desarrollen sus propios sistemas de modulación.

Se iniciará con la instalación del Python en el computador, base del conocimiento de los sistemas RDS, además se mostrará como instalar las dependencias que se necesitan para poder utilizarlos, y algunas aplicaciones que serán de mucha ayuda para poder explotar todos estos dispositivos tienen que ofrecer

### 1.1.1 Instalando Python

Desde la página oficial de Python [Python link \(https://www.python.org/\)](https://www.python.org/) se accede a la pestaña de descargas y se selecciona el sistema operativo en el que se instalará. La figura 1.1 muestra lo que aparece una vez hecho clic en dicha pestaña. La arquitectura del sistema operativo en la mayoría de los computadores es de 64 bits

Figura 1.1 Página de fondo



Seleccionada la arquitectura que corresponda al computador, se descarga para obtener el archivo ejecutable (.exe) listo para instalar. Se debe tener en cuenta que este trabajo se desarrolló con la versión 3.9.5 de Python. Figura 1.2.

Figura 1.2 Versiones de Python



Una vez el archivo este descargado hay que hacer clic derecho sobre este y escoger la opción de ejecutar como administrador. Tal y como se muestra en la Figura 1.3.

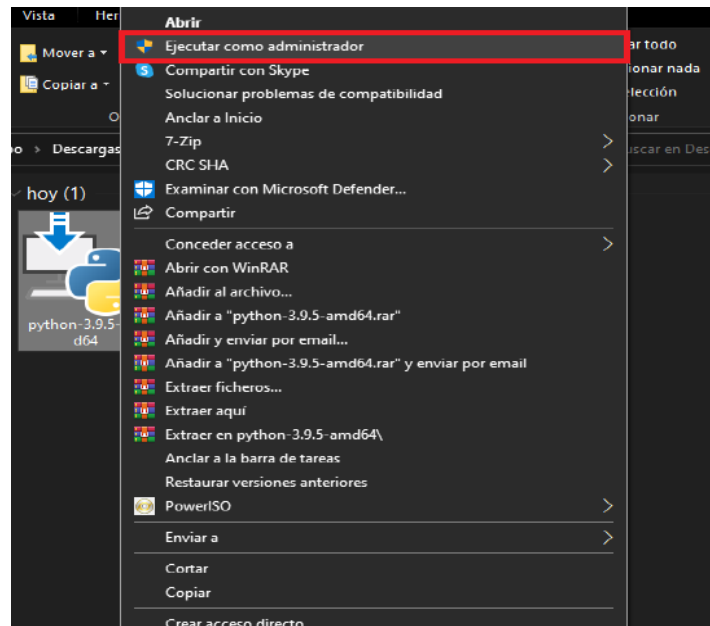


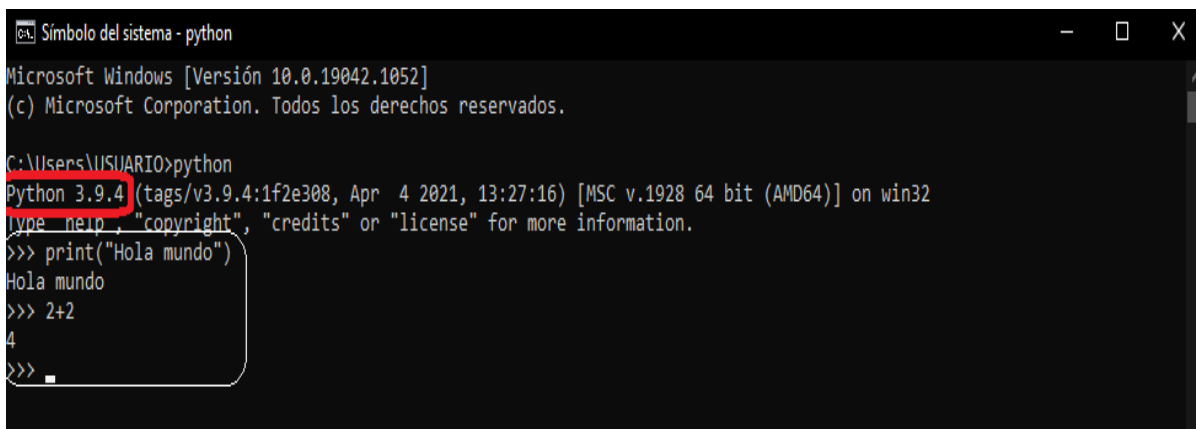
Figura 1.3 Instalando Python en el computador

Ejecutada la acción, en la siguiente pestaña se deberá habilitar el botón que dice "Add python 3.x.x to PATH". Seleccionado esto se hace clic sobre instalar ahora, y se iniciara el proceso de instalación en el computador. Una vez termine se recomienda reiniciar la computadora.



Figura 1.4 Instalador de Python

En este momento se ha instalado satisfactoriamente Python en el computador. Para verificar que se ha instalado correctamente, una opción es abrir la ventana de comandos de Windows también conocido como CMD y escribir Python, y se podrá ver en el terminal unas flechas (>>>) que indican que se puede ejecutar código de Python. Figura 1.5.



```
Símbolo del sistema - python
Microsoft Windows [Versión 10.0.19042.1052]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\USUARIO>python
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 4 2021, 13:27:16) [MSC v.1928 64 bit (AMD64)] on win32
type help, "copyright", "credits" or "license" for more information.
>>> print("Hola mundo")
Hola mundo
>>> 2+2
4
>>> _
```

Figura 1.5 Verificar Python

## 1.1.2 Instalando GNU radio en Windows

*GNU radio Companion* es una herramienta de Python la cual permite codificar utilizando el lenguaje de Python en forma de diagrama de bloques, lo cual facilita la manera en la que se puede escribir un programa. Este ofrece bloques que contienen funciones especializadas para trabajar de manera eficiente con vectores y matrices.

Debido a que esta aplicación fue desarrollada para *sistemas operativos Linux*, el programa en *Windows* presenta fallas o errores que podrían hacer que el programa se cierre de manera espontánea. Por esta razón es importante ir “guardando” para evitar perder el trabajo que se esté desarrollando.

GNU radio está basado en Python el cual es gratuito y de código abierto. Cuenta con una comunidad grande que está dispuesta a mantenerlo actualizado y en constante desarrollo, para que los códigos sean más eficientes y se utilicen de manera más efectiva los recursos del computador durante la ejecución del programa.

Para instalar GNU radio Companion en Windows es necesario ir al enlace [GNU radio](https://wiki.gnuradio.org/index.php/InstallingGR#Windows) (<https://wiki.gnuradio.org/index.php/InstallingGR#Windows>) en donde hay pasos sobre cómo hacer esta instalación en un entorno de anaconda. En este trabajo de grado se instalará utilizando Python que es la forma más sencilla y practica para poder utilizar esta aplicación.

Se debe ir al enlace [Instalador GNU radio Windows](http://www.gcndevlopment.com/gnuradio/index.htm) (<http://www.gcndevlopment.com/gnuradio/index.htm>), página oficial de GNU en donde se encuentra el instalador de GNU radio para Windows en forma de ejecutable. En este curso se usará la versión 3.8 de GNU radio. Y a manera de información, la versión 3.9 ya está libre y podría descargase. Figura 1.6.

## GNURadio Win64 Binaries

[Download Latest GR 3.7 Installer \(GR 3.7.13.5\)](#)

[Download Latest GR 3.8 Installer \(GR 3.8.2.0 py3.9\)](#)

[Download Latest GR 3.9 Beta Installer \(GR 3.9.0.0\)](#)

[Download All Other Binaries](#)

[Documentation](#), [Known Issues](#) / [Change Log](#)

### Background:

GNURadio is a powerful SDR tool, however its biggest limitation has always been that it is a challenge to install, even on Linux systems, because of the large number of shared dependencies. If you are a daily Windows user there is an even larger problem because most of these dependencies are also very Linux focused and difficult to build/install on windows for the average Windows users who is accustomed to simply double-clicking "setup.exe". Using a Linux VM is problematic with SDR because of the high demands that are placed on the USB interface. This leaves the user with dual booting, limiting them to one environment or the other. So the goal of this project was to build an optimized GNURadio binary using the most recent compiler for all dependencies, and packaging the result in an easy-to-use .msi package.

Figura 1.6 Página de instalación de GNU radio

Una vez descargado el instalador se hace clic derecho sobre este y se le da al botón de instalar. Al hacer esto el wizard de instalación aparecerá y se debe oprimir el botón siguiente hasta que complete todo el proceso. Tal y como se puede ver en la Figura 1.7 y 1.8

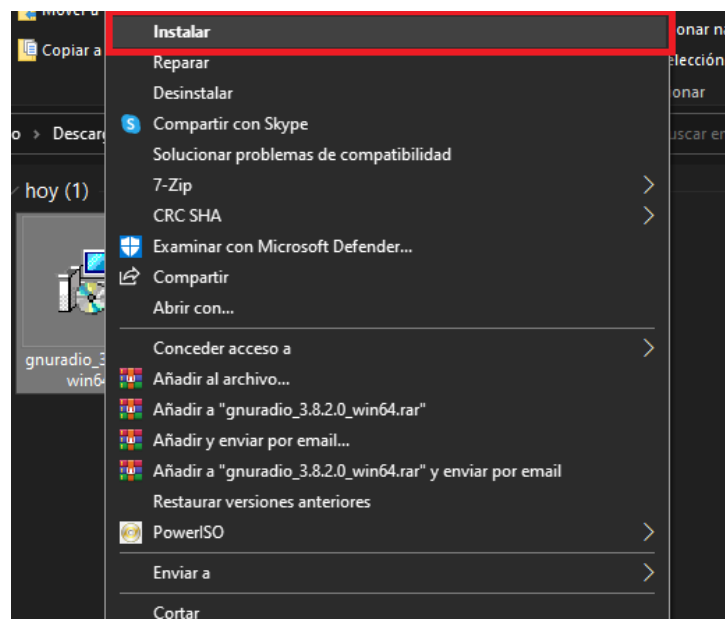


Figura 1.7 Instalando GNU radio

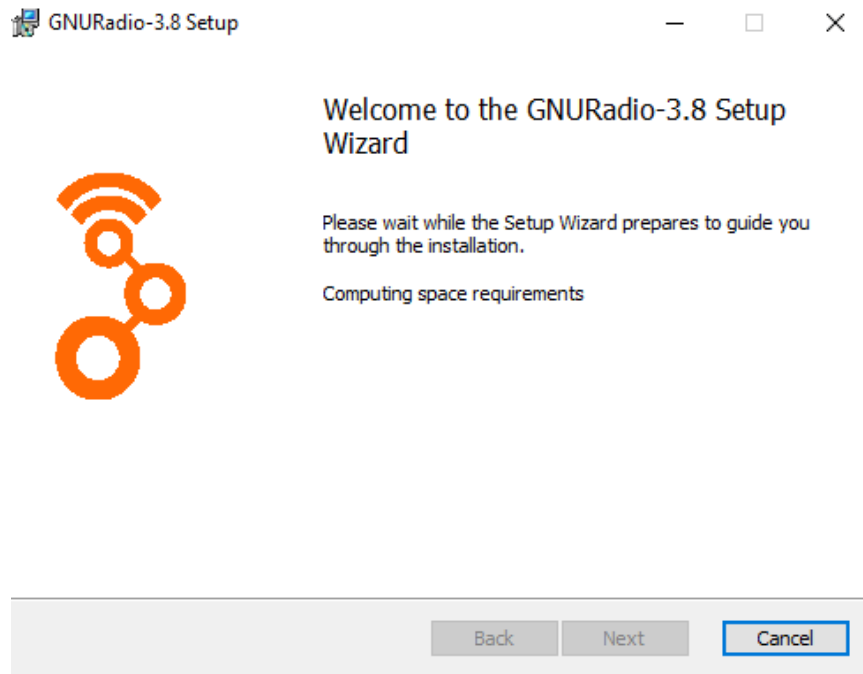


Figura 1.8 Wizard de GNU radio

Ya instalado, en el buscador de Windows deberá aparecer la aplicación lista para ser usada tal que al hacer clic ya se tendrá acceso a todos los bloques y a la programación que se pueda hacer en este entorno. En este momento se puede abrir el programa y debe lucir como la Figura 1.9.

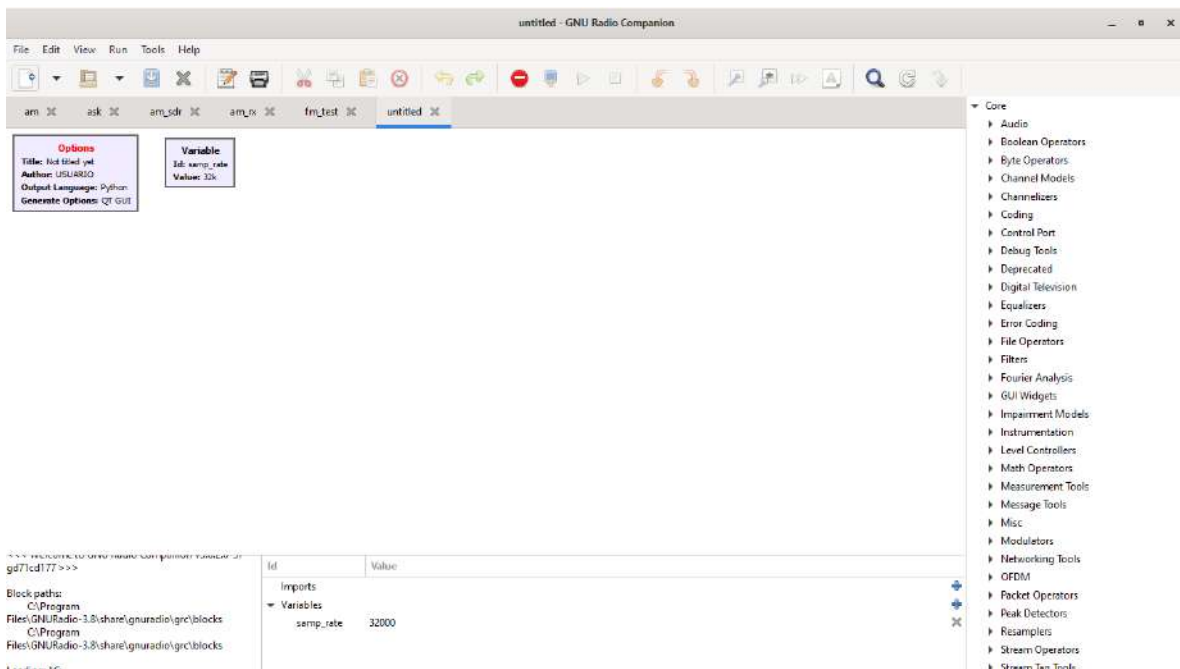


Figura 1.9 GNU radio

### 1.1.3 Instalación de GNU radio en Linux

Como se dijo en la sesión anterior esta plataforma fue desarrollada para ser utilizada en Linux, incluso viene como una aplicación predeterminada en algunas distribuciones de estos como por ejemplo *Radio-start* o *Pentos*. En este trabajo se mostrarán los comandos necesarios para instalar GNU radio en la distribución de Ubuntu la cual es una de las más populares.

Para poder realizar esta instalación es necesario abrir el terminal de Ubuntu esto se puede lograr oprimiendo las teclas `ctrl-alt t`, o con clic derecho y apretar la opción que dice “abrir terminal”, al tener la consola abierta es necesario agregar los siguientes comandos.

- `sudo add-apt-repository ppa:gnuradio/gnuradio-releases-3.8`
- `sudo apt-get update`
- `sudo apt install gnuradio`

Al poner los comandos se observarán los siguientes resultados y se debe escribir la palabra “yes” o la letra “y, cuando pidan para poder dar los permisos de desarrollador, necesarios para que el repositorio se instale sin problemas

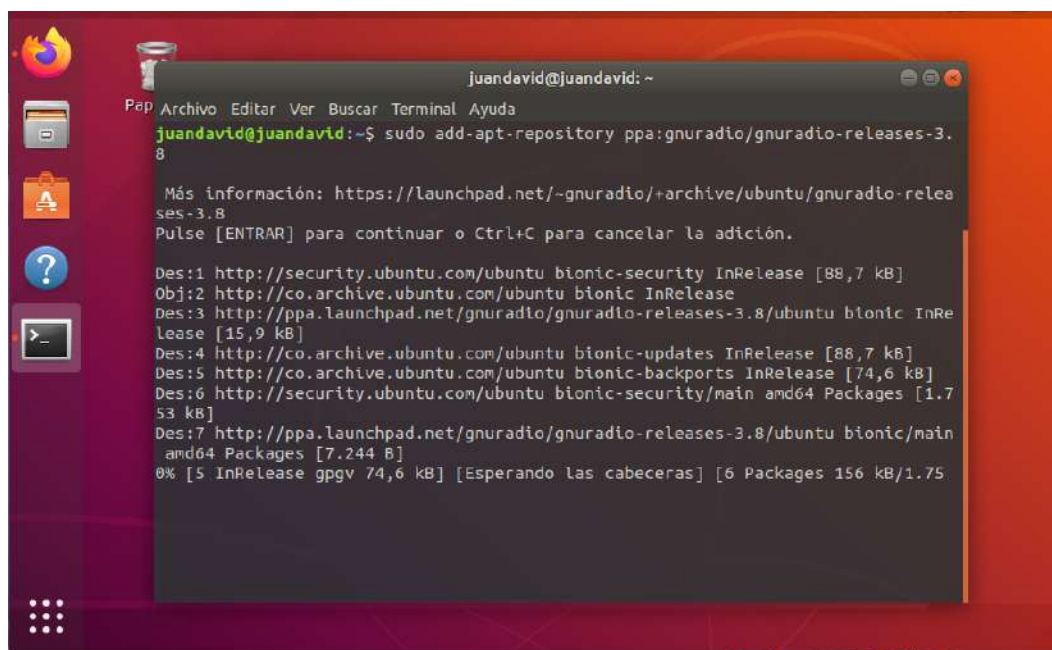


Figura 1.10 Instalando repositorio

Al terminar de ejecutarse el primer comando que es el encargado de instalar el repositorio en el computador, falta hacer la instalación de GNU radio. Y esto se hace con el último comando mostrado en la parte superior tal que se debe obtener lo mostrado en la figura 1.11.



Una vez que esos comandos se ejecuten, se puede utilizar sin ningún problema los bloques de programación de GNU radio y de SDR en el mismo entorno, con estas instalaciones hechas ya se puede seguir el curso al pie de la letra sin ningún tipo de problema.

### 1.1.4 Instalando AudaCity

AudaCity es una herramienta que permite grabar audio utilizando el micrófono del computador a una determinada frecuencia de muestreo que el usuario puede escoger dependiendo de la calidad de sonido que necesite. Para este curso se utilizará para transmitir mensajes personalizados.

Esta aplicación es multiplataforma y en la página oficial se puede descargar el archivo para cualquier sistema operativo (Windows Linux o Mac). El proceso de instalación es tan sencillo como oprimir el botón de instalar y esperar que este termine.

En la Figura 1.12 se puede ver la página oficial de AudaCity y la pestaña donde se puede escoger el sistema operativo en el cual se esté trabajando. Se presenta el enlace para ir a la página principal [Pagina de AudaCity \(https://www.audacityteam.org/\)](https://www.audacityteam.org/).



Figura 1.13 Página de Audacity

Ya instalado, mediante el buscador ubicado en la barra de tareas de Windows se puede abrir la aplicación como se muestra en la figura 1.14

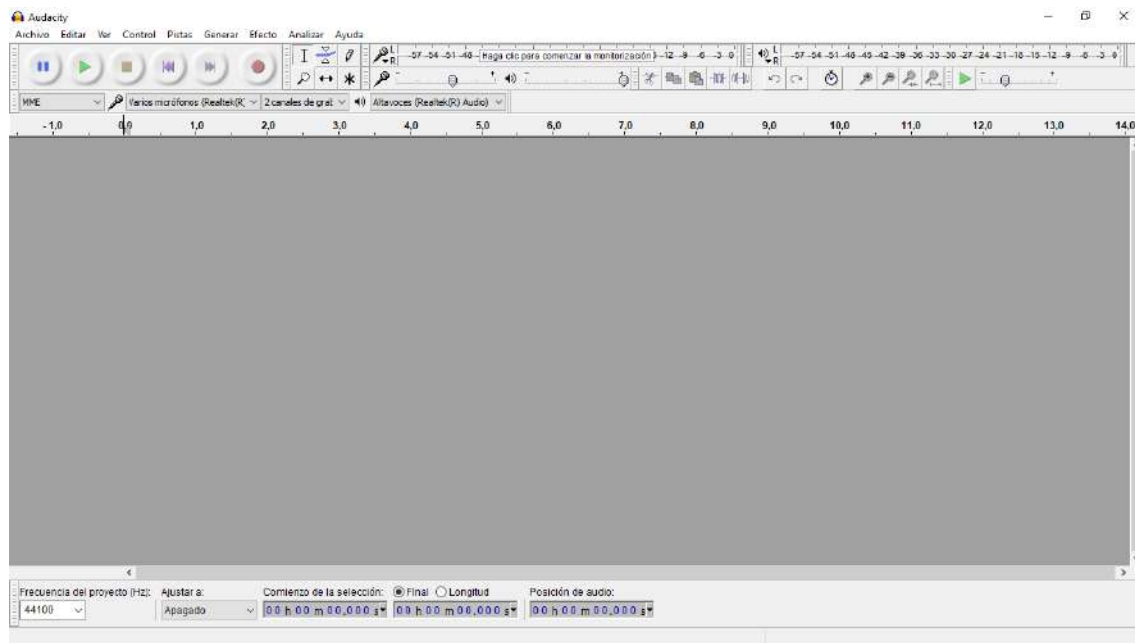


Figura 1.14 Aplicación de Audacity

## 1.1.5 Instalando Zadig

Zadig es una herramienta que permite instalar los drivers que se necesitan para poder comunicar el RDS con el computador. La instalación de esta aplicación se puede hacer de manera rápida solo hay que dirigirse al siguiente [Link Pagina para descargar Zadig](https://zadig.akeo.ie/) (<https://zadig.akeo.ie/> ) y descargar la versión actual para el sistema operativo Windows, ya que para Linux estos drivers se instalan cuando se descarga el repositorio.

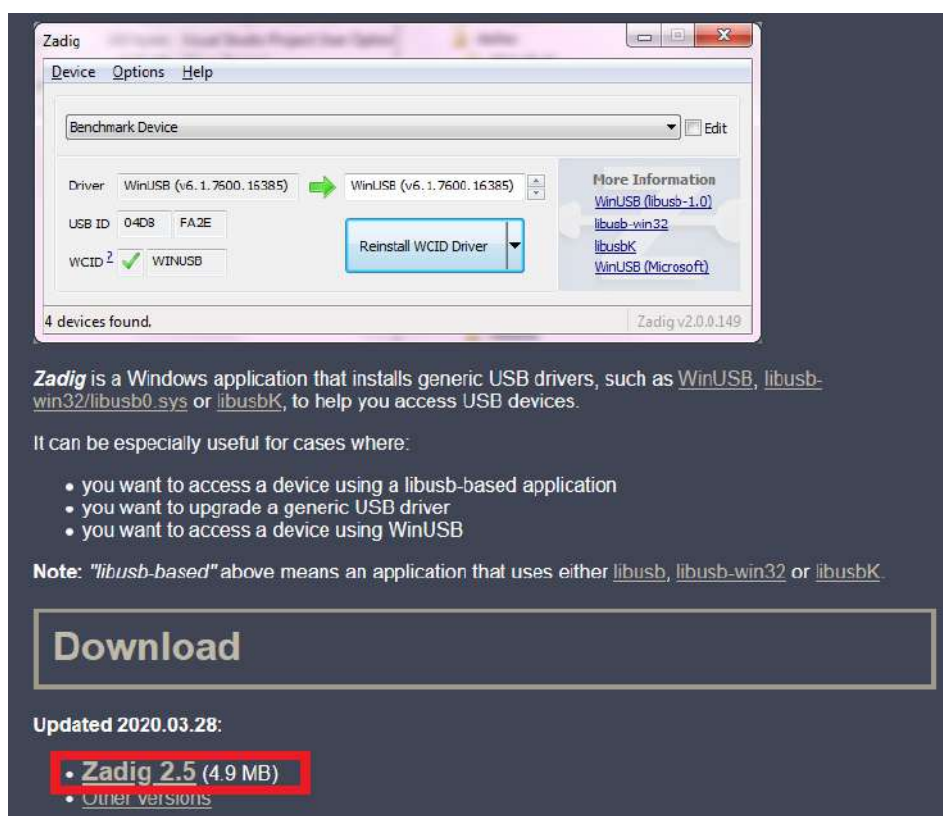


Figura 1.15 Página de Zadig

Como se ve en la Figura 1.15 el link de descarga está en la parte de debajo de la página nombrada anteriormente, con solo hacer clic donde se señala comenzara la descargar de la aplicación, para instalarla es tan sencillo como dar los permisos y oprimir el botón de siguiente hasta que todo esté completo.

Instalada la aplicación ya se puede abrir para verificar que esté funcionando, en esta sección no se mostrara como utilizarla para instalar los drivers de los SDR que se piensan utilizar, esa parte será mostrada durante los primeros pasos en GNU radio. Si todo está bien instalado, se debe observar la interfaz de la figura 1.16.

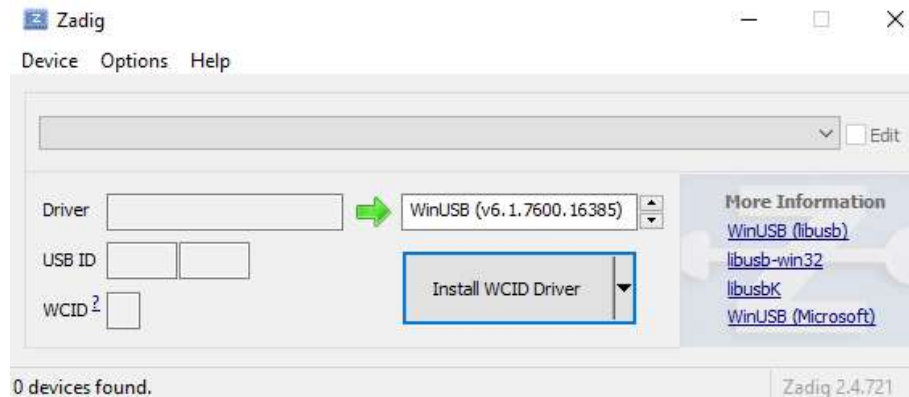


Figura 1.16 Aplicación de Zadig

### 1.1.6 Instalar SDR# en Windows

Esta es una de las aplicaciones más importantes que se va a usar en este curso, ya que con esta se puede de manera sencilla tener acceso a mucho de los recursos que tiene los SDR, como por ejemplo se puede poner a oír las distintas secciones del espectro, ajustar el ancho de banda, ver la intensidad de señales que está recibiendo entre muchas otras ya que también tiene la capacidad de instalar extensiones para hacer de esta una aplicación aún más completa.

SDR# es una aplicación que solo está para el sistema operativo Windows, sin embargo, para Linux también hay varias alternativas que se verán en las siguientes secciones, es menester decir que esta aplicación es gratuita y también es código abierto.

En la página oficial de Air Spy se encuentra información sobre las extensiones que se pueden instalar además de información útil sobre como manejar esta aplicación [Air Spy página](#), sin embargo, es difícil encontrar cual es la versión que mejor se adapte en este caso la versión que se utiliza es la 1664 la cual es la que menos errores ha presentado para utilizar toda la variedad de SDR que hay en el mercado. Para buscar las versiones hay que ir el siguiente enlace [SDR Sharp descarga \(https://www.iz3mez.it/software/SDRSharp/\)](https://www.iz3mez.it/software/SDRSharp/)

|                                    |                  |      |                         |
|------------------------------------|------------------|------|-------------------------|
| <a href="#">SDRSharp_v1727.zip</a> | 2019-10-13 21:22 | 9.6M | GZIP compressed archive |
| <a href="#">SDRSharp_v1700.zip</a> | 2018-12-30 18:13 | 1.7M | GZIP compressed archive |
| <a href="#">SDRSharp_v1672.zip</a> | 2018-09-16 19:43 | 1.7M | GZIP compressed archive |
| <a href="#">SDRSharp_v1666.zip</a> | 2018-09-16 19:43 | 1.6M | GZIP compressed archive |
| <a href="#">SDRSharp_v1664.zip</a> | 2018-09-16 19:43 | 1.6M | GZIP compressed archive |
| <a href="#">SDRSharp_v1631.zip</a> | 2018-01-01 20:15 | 1.6M | GZIP compressed archive |
| <a href="#">SDRSharp_v1583.zip</a> | 2018-09-16 19:42 | 1.1M | GZIP compressed archive |
| <a href="#">SDRSharp_v1570.zip</a> | 2018-09-16 19:42 | 1.6M | GZIP compressed archive |
| <a href="#">SDRSharp_v1525.zip</a> | 2017-03-11 15:37 | 1.5M | GZIP compressed archive |

Figura 1.17 Descarga la versión de SDR

Esto descargara un archivo zip en donde se encuentra la aplicación de SDR #, lo único que hay que hacer es copiar ese archivo a una ubicación que no la perderá

porque siempre que se quiera abrir la aplicación toca volver ahí y abrirla. Cuando se descomprima el paquete se encontrarán muchos archivos y el que ejecutable es el que contiene la aplicación, tal y como se muestra en la Figura 1.18

| Nombre                        | Fecha de modifica... | Tipo                  | Tamaño        |
|-------------------------------|----------------------|-----------------------|---------------|
| sdriq.dll                     | 21/09/2015 3:43      | Extensión de la ap... | 14 KB         |
| SDRSharp.AfedriSDRNet.dll     | 22/09/2020 17:25     | Extensión de la ap... | 21 KB         |
| SDRSharp.BandPlan.dll         | 22/09/2020 17:25     | Extensión de la ap... | 19 KB         |
| SDRSharp.CollapsiblePanel.dll | 22/09/2020 17:25     | Extensión de la ap... | 20 KB         |
| SDRSharp.Common.dll           | 22/09/2020 17:25     | Extensión de la ap... | 13 KB         |
| SDRSharp.Diagnostics.dll      | 22/09/2020 17:25     | Extensión de la ap... | 15 KB         |
| SDRSharp.DNR.dll              | 22/09/2020 17:25     | Extensión de la ap... | 28 KB         |
| <b>SDRSharp</b>               | 22/09/2020 17:25     | <b>Aplicación</b>     | <b>384 KB</b> |
| SDRSharp.exe.config           | 5/06/2021 18:15      | Archivo CONFIG        | 9 KB          |
| SDRSharp.FrequencyEdit.dll    | 22/09/2020 17:25     | Extensión de la ap... | 26 KB         |
| SDRSharp.FrequencyManager.dll | 22/09/2020 17:25     | Extensión de la ap... | 35 KB         |
| SDRSharp.FUNcube.dll          | 22/09/2020 17:25     | Extensión de la ap... | 82 KB         |
| SDRSharp.FUNcubeProPlus.dll   | 22/09/2020 17:25     | Extensión de la ap... | 33 KB         |
| SDRSharp.HackRF.dll           | 22/09/2020 17:25     | Extensión de la ap... | 23 KB         |
| SDRSharp.NoiseBlanker.dll     | 22/09/2020 17:25     | Extensión de la ap... | 16 KB         |
| SDRSharp.PanView.dll          | 22/09/2020 17:25     | Extensión de la ap... | 56 KB         |
| SDRSharp.Radio.dll            | 22/09/2020 17:25     | Extensión de la ap... | 101 KB        |
| SDRSharp.RTLSDR.dll           | 22/09/2020 17:25     | Extensión de la ap... | 30 KB         |
| SDRSharp.RTLTCP.dll           | 22/09/2020 17:25     | Extensión de la ap... | 18 KB         |

Figura 1.18 Seleccionar la aplicación SDR

Al hacer clic sobre el icono señalado en la Figura 18 se abrir la aplicación tal como se muestra en la Figura 1.19, aun no se puede oír nada porque no se han cuadrado los drivers para poder utilizarla con el SDR, eso se hará en el siguiente capítulo.

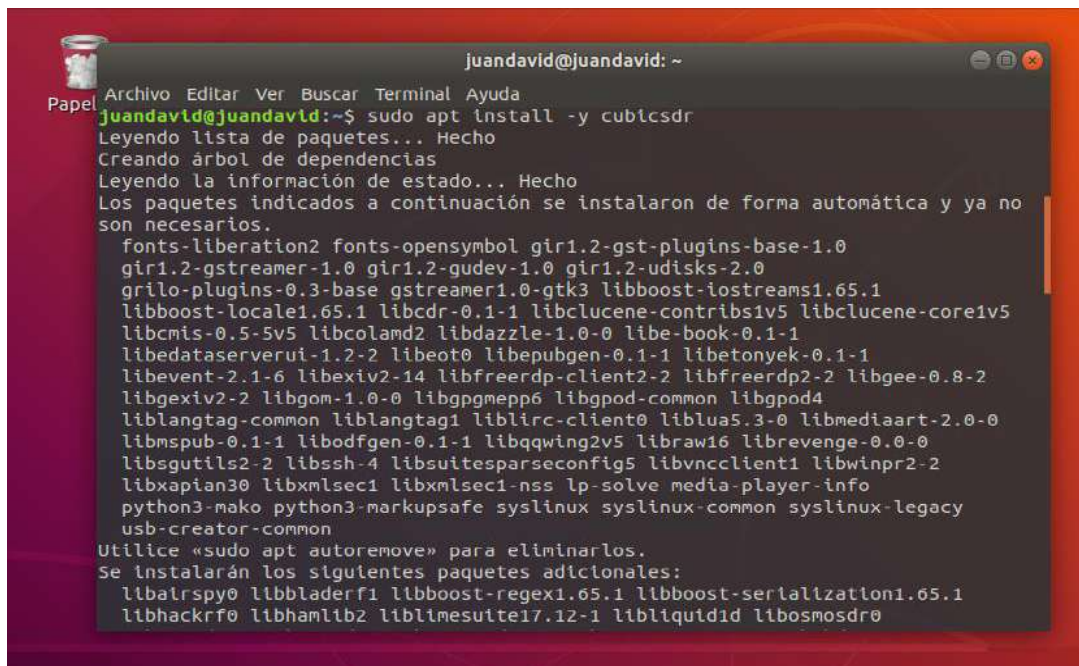


Figura 1.19 Aplicación de SDR

## 1.1.7 Instalando CubicSDR en Linux

CubicSDR es una aplicación hecha para Linux que trata de imitar de la mejor manera a SDR# hecha solo para Windows. Esta cuenta con muchas herramientas que permiten desplazarse por el espectro para poder sintonizar alguna frecuencia en específico también permite cuadrar los recursos del SDR para aumentar o disminuir el ancho de banda para poder captar una señal.

El proceso de instalación es bastante sencillo, lo único que toca hacer es escribir el siguiente comando en la terminal de Ubuntu “sudo apt install cubicsdr” y cuando se pidan los permisos de desarrollador solo es necesarios poner yes. Si se quiere hacer en una sola línea para instalación se puede colocar el siguiente comando “sudo apt install -y cubicsdr”, tal y como se muestra en la Figura 1.20



```
juandavid@juandavid: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
Papel  
juandavid@juandavid:~$ sudo apt install -y cubicsdr  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.  
fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0  
gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0  
grilo-plugins-0.3-base gstreamer1.0-gtk3 libboost-iostreams1.65.1  
libboost-locale1.65.1 libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5  
libcmis-0.5-5v5 libcolamd2 libdazzle-1.0-0 libe-book-0.1-1  
libdataserverui-1.2-2 libeot0 libepubgen-0.1-1 libetonyek-0.1-1  
libevent-2.1-6 libexiv2-14 libfreerdp-client2-2 libfreerdp2-2 libgee-0.8-2  
libgexiv2-2 libgom-1.0-0 libgpgmepp6 libgpod-common libgpod4  
liblangtag-common liblangtag1 liblirc-client0 liblua5.3-0 libmediaart-2.0-0  
libmspub-0.1-1 libodfgen-0.1-1 libqwing2v5 libraw16 librevenge-0.0-0  
libsgutils2-2 libssh-4 libsuitesparseconfig5 libvncclient1 libwinpr2-2  
libxapian30 libxmlseci libxmlseci-nss lp-solve media-player-info  
python3-mako python3-markupsafe syslinux syslinux-common syslinux-legacy  
usb-creator-common  
Utilice «sudo apt autoremove» para eliminarlos.  
Se instalarán los siguientes paquetes adicionales:  
libairspy0 libbladerf1 libboost-regex1.65.1 libboost-serialization1.65.1  
libhackrf0 libhamlib2 liblimesuite17.12-1 libliquid4 libosmosdr0
```

Figura 1.20 Instalando CubicSDR

Una vez finalizado el proceso se puede ejecutar el programa y se debe ver la siguiente aplicación, como en Linux los drivers se instalaron al momento de instalar GNU radio ya se podría probar si todo está funcionando, lo único que toca hacer es conectar el dispositivo al computador y dejar que el seleccionador automático lo detecte, una vez que diga que se ha vinculado con éxito se oprime el botón de “Play” para empezar a escuchar las emisoras el audio que se desea.

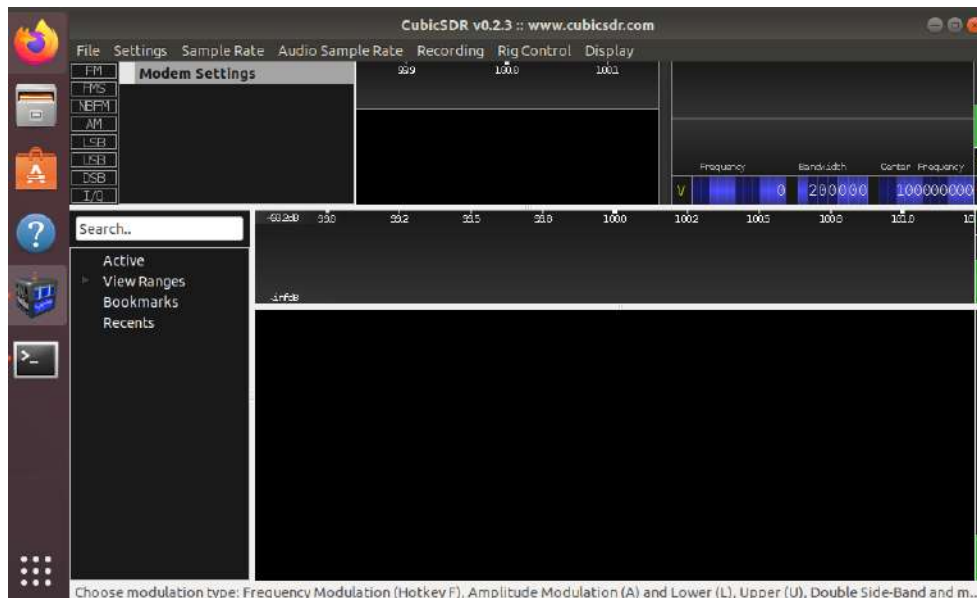


Figura 1.21 CubicSDR

## 1.2 Hardware necesario

Instalado ya el software necesario, es momento de hablar sobre el hardware que se va a utilizar en este proyecto de grado. Como se nombró anteriormente hay gran variedad de estos dispositivos en el mercado, pero todos tienen una arquitectura similar que les permite realizar todo de tipo de acciones tales como “escuchar” el espectro radio eléctrico o transmitir información sobre este.

Para la realización de este trabajo es necesario contar con dos SDR, uno será usado para poder está recibir información y decodificar el mensaje, mientras que el otro será usado para poder transmitir información con distintos tipos de modulación.

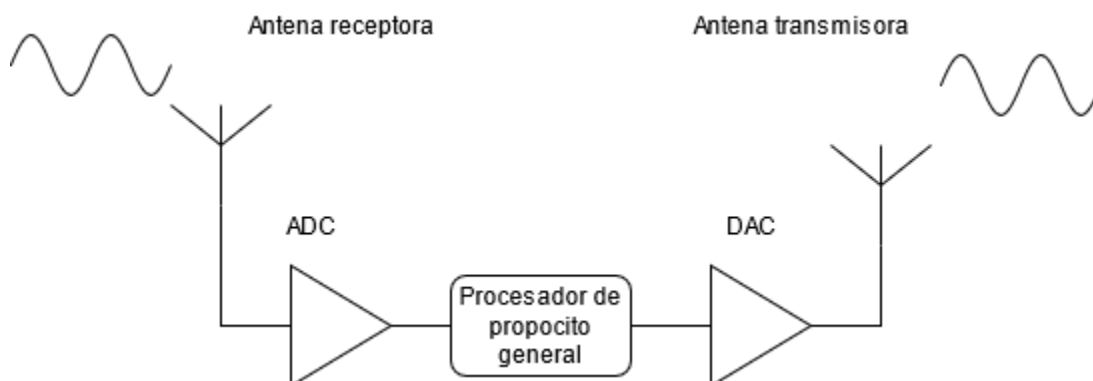
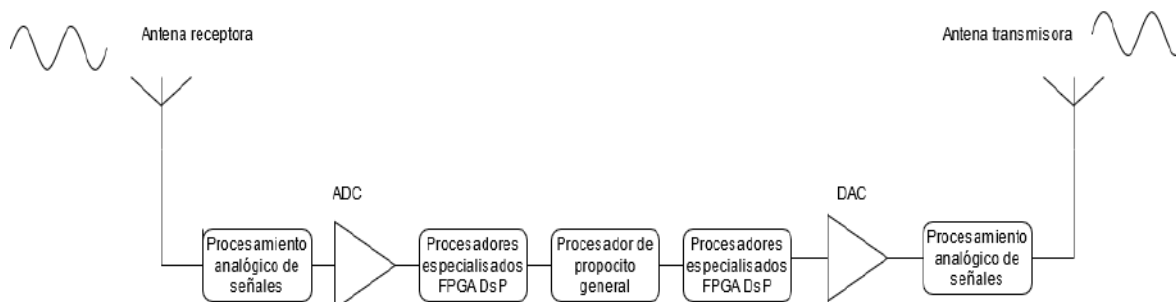


Figura 1.22 Diagrama de bloques básico de un RDS

En la figura 1.22 se observa el diagrama de bloques simplificado de un RDS este cuenta con una o dos antenas que le permiten recibir o transmitir información, a

parte de este cuenta con un ADC (Analog to Digital converter) en español un convertidor de señales analógicas a digitales, siguiente a este tiene una CPU o unidad de procesamiento que le permite manipular, tratar la señal de entrada para luego pasarla a un DAC (Digital to Analog converter) convertidor digital a analógico para nuevamente irradiar la señal al aire.



*Figura 1.23 Diagrama de RDS*

Un diagrama más preciso de cómo es la arquitectura de un RDS se puede ver en la figura 1.23, esta contiene nuevos bloques los cuales son filtros analógicos para procesar las señales directamente cuando son recibidas o transmitidas por las antenas, además cuenta con unidades especiales de procesamiento como lo pueden ser FPGA o DpS, todo esto para poder trabajar de forma rápida y eficiente con vectores y matrices.

## 1.2.1 RTL-SDR-Dongle

El RTL-SDR-dongle es parecido a una memoria USB, pero dentro de esta contiene una pequeña FPGA algunos DpS para procesamiento digital de señales, filtros analógicos y por último tiene un puerto para poder conectar antenas removibles para poder captar en varias frecuencias sin mayores pérdidas. Este dongle no tiene la capacidad de transmitir información, pero tiene la arquitectura para captar todo de tipo de modulaciones.

Este dongle fue originalmente hecho para captar señales de televisión debido a su bajo costo y gran efectividad, con el paso del tiempo los televisores comenzaron a tener este equipo integrado por lo cual, estos dispositivos salieron al mercado para los radios aficionados.

Comercialmente se puede encontrar varios tipos de dongle para trabajar, El precio de esta ronda alrededor de los 30.000\$ y 80.000\$. En este trabajo se trabajará específicamente con el DAB-FM-DVB-T-FC1001, el cual se puede adquirir en el siguiente link [SDR-RTL-Link-compra](#), pero cualquiera de los que se consiga puede servir para seguir este proyecto.



Figura 1.24 RTL SDR DAB-FM-DVB-T-FC1001

En la imagen 1.24 se puede ver dispositivo que se va a utilizar a lo largo de este proyecto, en la siguiente tabla se puede ver las características que este posee

| RTL-Dongle            |               |
|-----------------------|---------------|
| Rango de frecuencia   | 24-1766 [MHz] |
| Ancho de banda máximo | 2 [MHz]       |
| Resolución del ADC    | 8 bits        |
| Transmite             | No            |
| Precio                | 30000         |

Tabla 1 Características de RTL-dongle



Figura 1.25 RTL-SDR.com

En la figura 1.25 se puede apreciar otro tipo de dongle este tiene unas cuantas mejoras con respecto al anterior, la única desventaja es que tiene un precio más elevado, la tabla de especificaciones se puede ver a continuación.

| RTL-Dongle V3         |                              |
|-----------------------|------------------------------|
| Rango de frecuencia   | 24-1766 [MHz]                |
| Ancho de banda máximo | 3 [MHz]                      |
| Resolución del ADC    | 8 bits                       |
| Transmite             | No                           |
| Precio                | 50000                        |
| Extras                | Filtros analógicos más finos |

*Tabla 2 Características SDR-RTL-v3*

## 1.2.2 HackRf-One

El HackRf-One es otro SDR con una arquitectura similar al visto anteriormente, pero con equipo mucho más especializado y con una comunidad grande que se ha encargado de actualizarlo y mejorarlo con el paso del tiempo. Este equipo es más costoso que los dongles, porque además tiene la capacidad de transmitir y sus características son más eficiente.



*Figura 1.26 HackRF-One*

En la siguiente tabla se puede observar las características del HackRf One, en esta se puede evidenciar lo superior que es respecto frente a los dongles y se justifica su precio elevado también.

| HackRf-One            |                                                      |
|-----------------------|------------------------------------------------------|
| Rango de frecuencia   | 1 a 6000[MHz]                                        |
| Ancho de banda máximo | 20 [MHz]                                             |
| Resolución del ADC    | 8 bits                                               |
| Transmite             | Half-duplex                                          |
| Precio                | 280000                                               |
| Extras                | Filtros analógicas más finos, mayor calida de sonido |

*Tabla 3 Hackrf-one*

### 1.2.3 Adalm-Pluto

Este es otro tipo de SDR el cual es más económico que el HackRF-One, sin embargo, este cuenta con un rango de frecuencia menor, pero lo suficientemente grande como para ir a la frecuencia de Wifi. A demás cuenta con una característica única la cual es que su sistema de transmisión el Full-duplex esto quiere decir que puede transmitir y recibir información al mismo tiempo, lo que permite hacer sistemas de pruebas con solo un equipo.



*Figura 1.27 Adalm-Pluto*

Como se puede ver en la figura que este cuenta con dos antenas incorporada una para transmisión y otra para recepción, además de ser un tamaño portable para poder trabajar en distintos entornos, en la siguiente figura se puede ver las características que este posee.

| Adalm Pluto                |                   |
|----------------------------|-------------------|
| Rango de frecuencia        | 325-3800[MHz]     |
| Ancho de banda máximo      | 20 [MHz]          |
| Resolución del ADC         | 12 bits           |
| Precio                     | 300000            |
| Transmisión                | Full duplex       |
| Características especiales | Filtro analógicos |

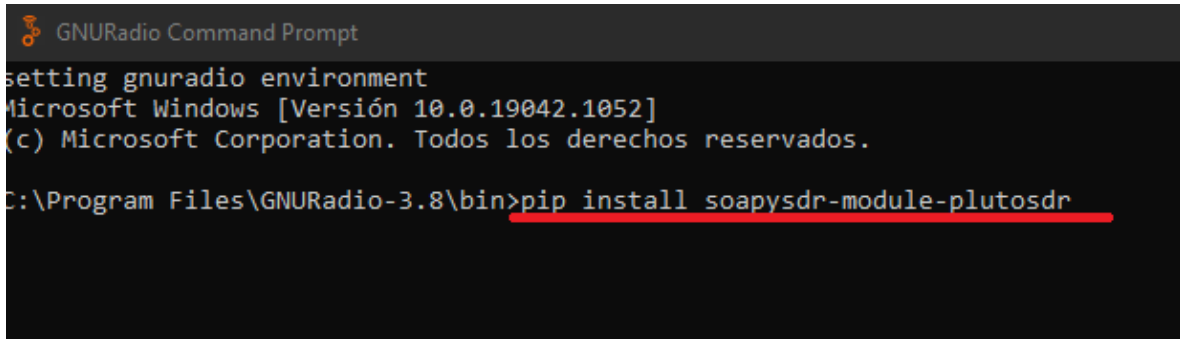
*Tabla 4 Características Adalm pluto*

Este SDR no viene instalador por defecto en GNU radio para poder utilizarlo es necesario colocar los siguientes comandos en el prompt de GNU radio en Windows, el cual se puede encontrar en el buscado de este sistema operativo como se ve en la figura 1.28.



*Figura 1.28 abriendo GNU radio prompt*

Al abrir el prompt se debe colocar el comando “pip install soapysdr-module-plutosdr” tal y como se ve en la figura 1.29



```
GNURadio Command Prompt
setting gnuradio environment
Microsoft Windows [Versión 10.0.19042.1052]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Program Files\GNURadio-3.8\bin>pip install soapysdr-module-plutosdr
```

Figura 1.29 Instalando Adalm pluto

Una vez ejecutado el comando en GNU radio debe aparece el bloque de sink (transmisión) y el bloque source (recepción) para ser usados.

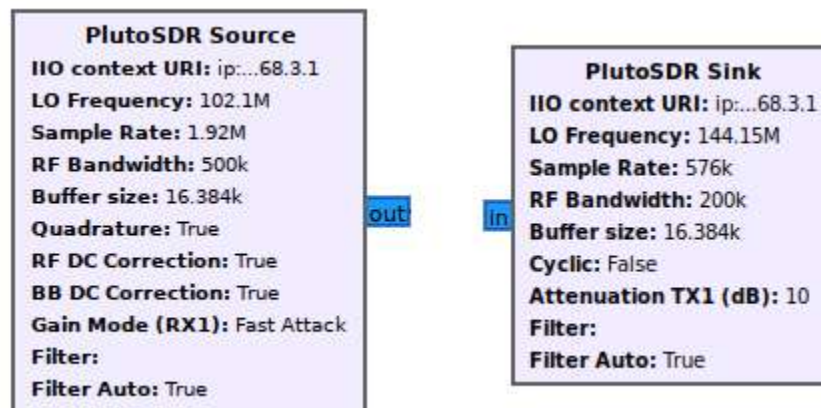


Figura 1.30 Bloques de adaml pluto

## 1.2.4 USRP

A diferencia de los demás SDR mencionados anteriormente este fue desarrollado comercialmente para un uso profesional este se puede comprar igualmente pero el precio es más elevado ya que tiene componentes de puntas que son sensible a los cambios más mínimos, por lo cual lo hace una herramienta excelente para todo profesional.



Figura 1.32 USRP

Este equipo no viene instalado por defecto en GNU radio para poder utilizarlo es necesario seguir los mismos pasos vistos en la figura 1.28 y 1.29 solo toca colocar el siguiente comando “pip install soapysdr-module-uhd”, con esto ya está instalado para ser usado.

| USRP                       |                          |
|----------------------------|--------------------------|
| Rango de frecuencia        | 70-6000[MHz]             |
| Ancho de banda máximo      | 50 [MHz]                 |
| Resolución del ADC         | 10 bits                  |
| Precio                     | 3000000                  |
| Transmisión                | Full duplex              |
| Características especiales | Spartan 6 XC6SLX150 FPGA |

Tabla 5 Características USRP

### 1.3 Primeros pasos en GNU radio

Como se mencionó anteriormente GNU radio es una aplicación que permite escribir código en forma de diagrama de bloques y este lo transforma en código Python el cual es el lenguaje de programación que se usara en este trabajo de grado, ya que este es uno de los lenguajes con los cuales se puede programar los SDR vistos en la sección anterior.

Acá se verá cuáles son los bloques más usados para trabajar y como configurarlos para poder trabajar de mejor manera, además de buscar cómo hacer una estructura limpia y ordenada para que otro usuario pueda leerlo y entenderlo sin dificultades, además de cómo adicionar comentarios para hacer un código entendible.

El primer paso es abrir la aplicación de GNU radio en donde se encontrar un menú como este donde se puede ver la siguiente capa.

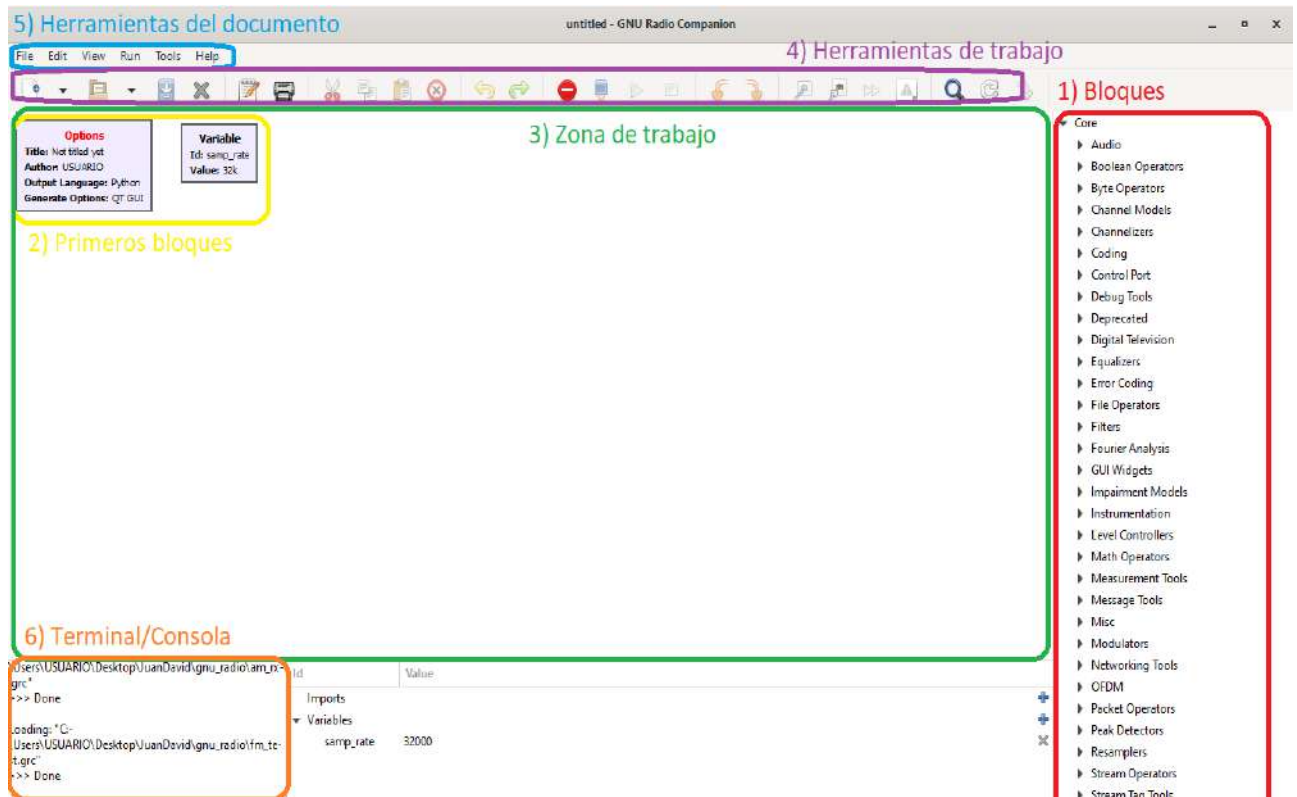


Figura 1.32 GNU radio

En la figura 1.32 se puede observar cómo se ha separado en varios bloques los componentes principales de GNU radio para analizarlos y detallar de manera rápida lo que se puede hacer en cada una de esas secciones.

- **Bloques:** La sección encerrada con el color rojo contiene todas las librerías que se tiene instaladas en el entorno, y dentro de cada una se puede encontrar los bloques que se necesitan para cada trabajo, un ejemplo de esto es buscar en la librería "boolean operator" y ahí se podrá encontrar los bloque AND, OR, XOR entre otros. Y en lo demás se encontrar los respectivos bloques de acuerdo con la librería.
- **Primeros bloques:** La segunda parte resaltada con amarillo son los bloques por defecto que se tienen al inicio de cada proyecto en esta parte se puede elegir el nombre el código además de autor el tamaño del canvas en donde se trabajara y el lenguaje de programación que se usará (por defecto esta Python pero este se puede pasar a c++), y el otro bloque que aparece ahí dice la frecuencia de muestreo con que se va a trabajar, este se puede cambiar por cualquier número dependiendo de las señales que se quieran y siempre debe ser un número entero.

- **Zona de trabajo:** La zona verde es el área de trabajo acá se colocarán todos los bloques, comentarios, variables necesarias para realizar el código, esta pantalla crecerá dependiendo de los bloques que se vayan colocando.
- **Herramientas de trabajo:** La zona con el color morado tiene las herramientas con los cuales se puede ejecutar el código, detenerlo, además de funciones como copiar pegar y buscar, esta última se usa mucho para buscar los bloques de forma más rápida.
- **Herramientas de documentos:** Estas herramientas las cuales están encerradas con el color azul contiene opciones de abrir guardar los códigos, además de editar que permite cambiar algunos parámetros de la aplicación como habilitar una cuadrícula en el canvas, además se tiene la opción de view y help que es donde se puede buscar ayudar sobre la versión de GNU radio en la que se esté trabajando.
- **Terminal/Consola:** Acá se puede ver la salida del código final, acá también se mostrará los errores que puedan suceder a la hora de ejecutar el código.

### 1.3.1 Bloques source y sink

Estos bloques son los que permiten tener señales de entrada y salida, además algunos de los bloques sink permiten graficar en el dominio del tiempo o en el dominio de la frecuencia, cada bloque sink cuenta con una breve descripción donde se detalla lo que esta puede hacer.

Para poder entender eso de una mejor manera, se propone generar tres señales periódicas un seno otra triangular y por último una cuadrada, y se debe poder ver en una gráfica las tres señales y además por separado se debe poder observar los armónicos de dichas señales.

Utilizando la herramienta de búsqueda poner “source signal” este bloque permite crear una señal a la cual se le debe seleccionar de que tipo será (byte int float complex), luego identificar el tiempo de muestreo el cual ya está definido como una variable global al inicio de cada programa, además se puede escoger la forma de la onda en la casilla que dice “waveform”. Finalmente, también se puede seleccionar la frecuencia, amplitud, el nivel dc y fase de esta. Tal y como se ve en la figura 1.33

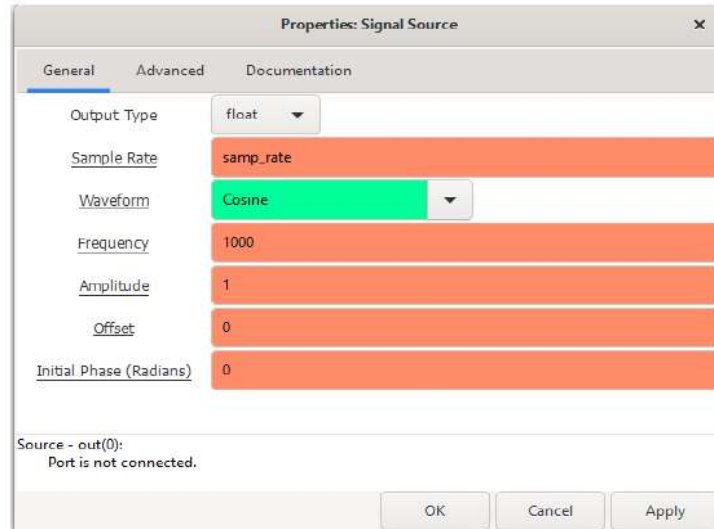
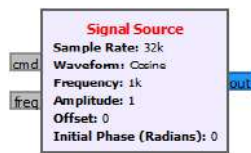


Figura 1.33 Bloque signal source

Con este bloque se puede seleccionar las tres señales que piden en el ejercicio anteriormente nombrado. Una vez seleccionado los tres bloques es momento de buscar el bloque “Time sink” y “Frequency sink” los cuales muestran en un GUI la función en el dominio del tiempo y los armónicos respectivamente.

Como se observa en la figura 1.33 los parámetros más importantes que toca poner son el tipo de datos que va a recibir, el nombre de los ejes, la frecuencia de muestreo y finalmente el número de puntos, este valor debe ser siempre un múltiplo de dos para tener un buen rendimiento.

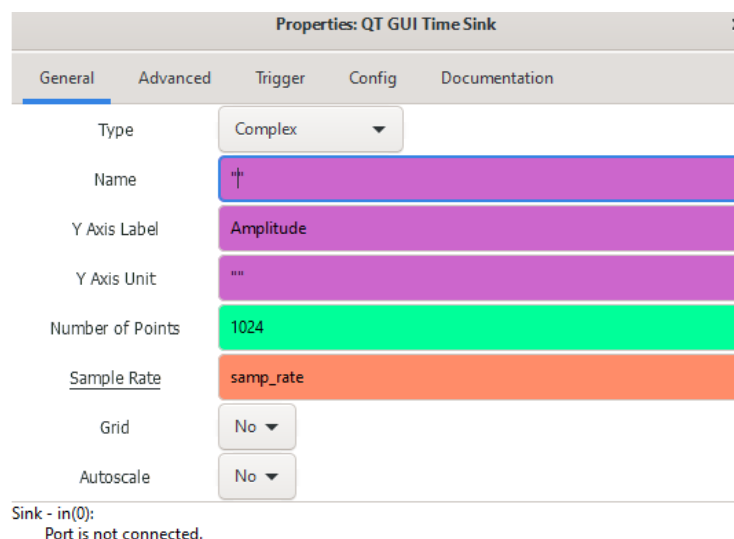
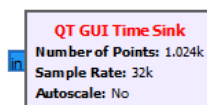


Figura 1.34 Bloque time sink

Finalmente, solo falta por añadir el bloque frequency sink el cual se encargará de mostrar los armónicos de esta señal, en la figura 1.35 se puede observar los componentes que necesita este bloque para funcionar.

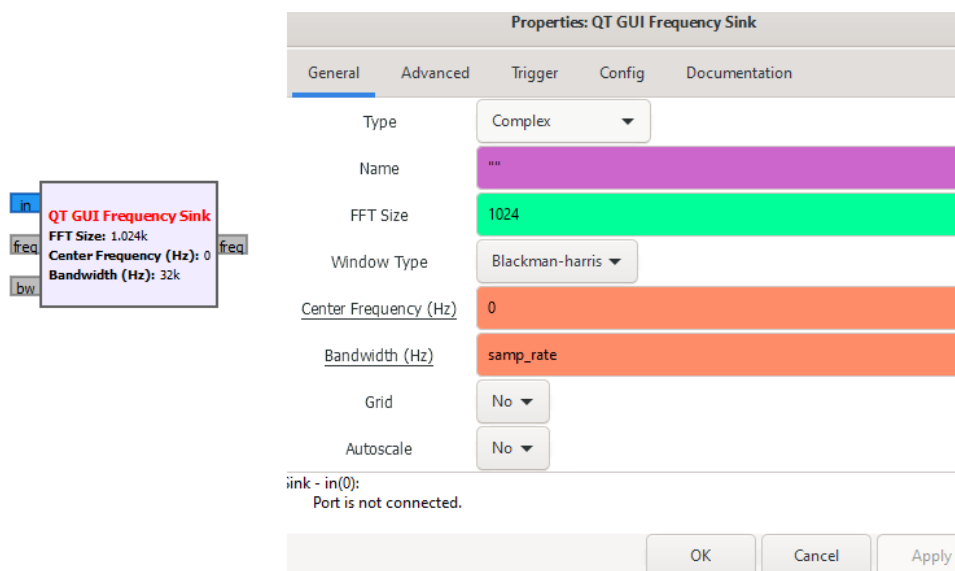


Figura 1.35 Bloque frequency sink

De este bloque es importante decir que el tipo de ventana que se utilice puede ayudar a contrarrestar el ruido que se pueda producir y da una gráfica más limpia, pero entre más compleja sea la ventana más costo computacional se deberá pagar.

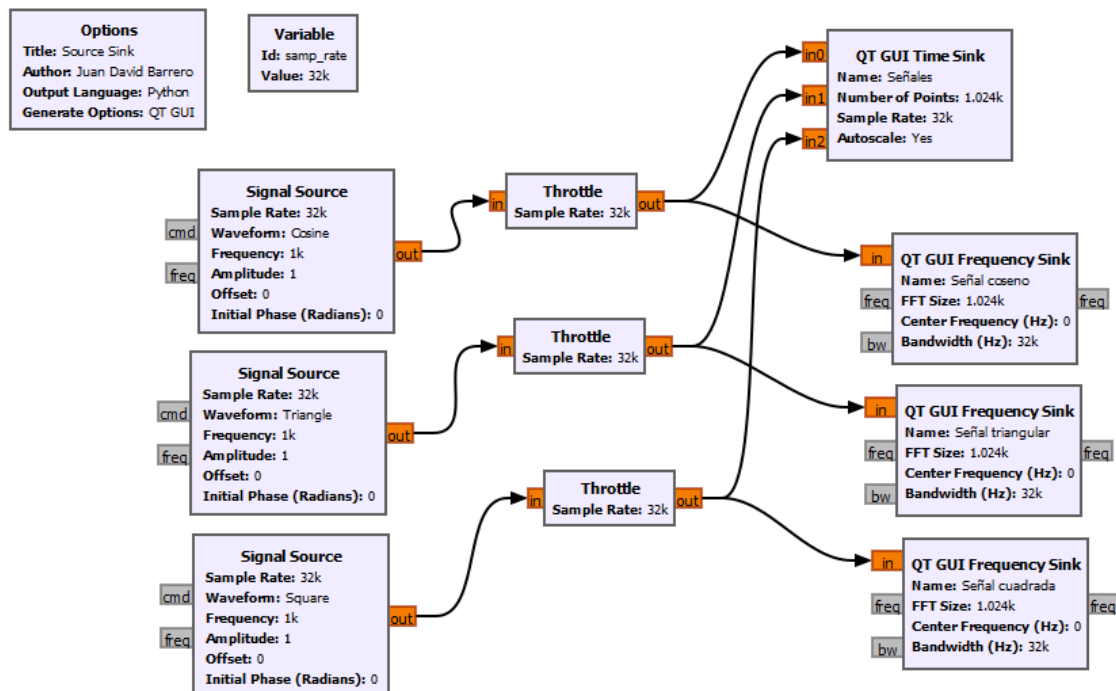


Figura 1.36 Resolución del primer ejercicio

Como se puede ver en la figura 1.37, el problema nombrado anteriormente esta resuelto, acá se observa un nuevo bloque llamado “Throttle” esta es un bloque sencillo que evitar que el computador de recursos de más durante la simulación es recomendable ponerlos siempre que se vaya a simular para no sobre cargar el procesador del computador. Este bloque solo se debe quitar cuando ya se esté usando los SDR.

Al ejecutar el programa se debe obtener el siguiente resultado.

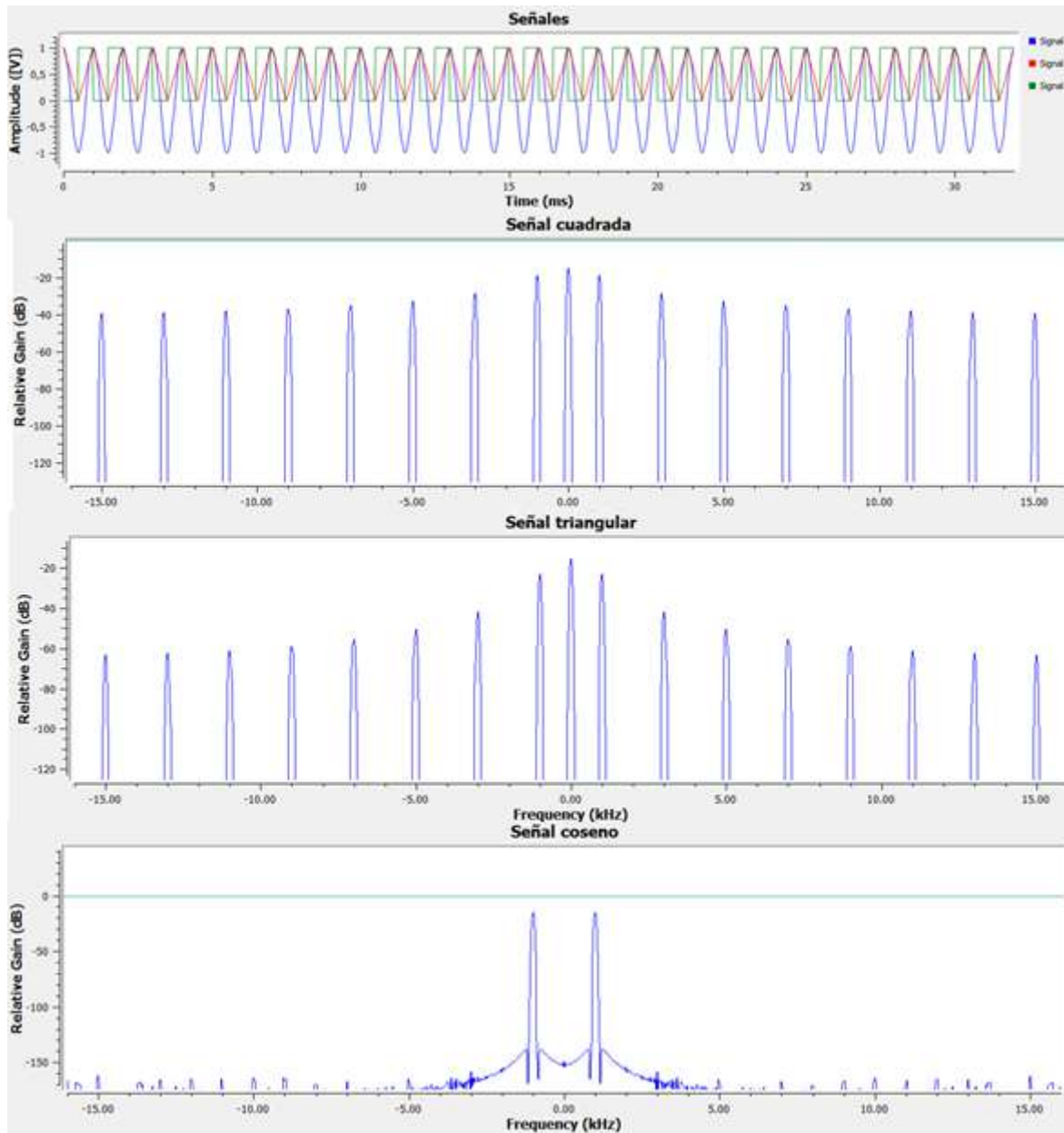


Figura 1.37 resultado del ejercicio 1

Para descargar el diagrama de bloques haga clic en el siguiente enlace (<https://github.com/JuanDavidBarrero/GNURadio/tree/main/Practice> )

### 1.3.2 Operaciones aritméticas con GNU radio

Durante el tratamiento de señales es necesario tener la capacidad de operarlas mezclarlas, para esto GNU radio ofrece una variedad de bloques con los que se pueden sumar, multiplicar y dividir señales periódicas de manera sencilla y eficiente gracias a una librería llamada Volk Wolf la cual esta optimizada para realizar operaciones aritméticas con vectores de gran tamaño.

En esta sección se resolverá el siguiente ejercicio en el cual se pide realizar las operaciones matemáticas básicas de dos señales cuales quiera, pero en la salida solo debe ver dos graficas uno de la señal en el dominio del tiempo y otra en el dominio de la frecuencia. Para hacer esto se recomienda utilizar el bloque selector, el cual funciona como un multiplexor para poder seleccionar la operación que se desea.

Para iniciar esta práctica hay que seleccionar dos fuentes de señales tal y como se vio en la sección anterior, además se debe colocar los bloques “sink time” y “sink frequency” para poder ver las respectivas operaciones que se piden.

Para la segunda parte es necesario buscar en la librería que se encuentra en la parte izquierda de la aplicación una llamada “Math operators”, aca se pueden buscar todas las operaciones que se pueden realizar, o utilizando el buscador nombrado en la primera sección se puede escribir el nombre de la operación que se quiere realizar y arrastrarlo a la zona de trabajo.

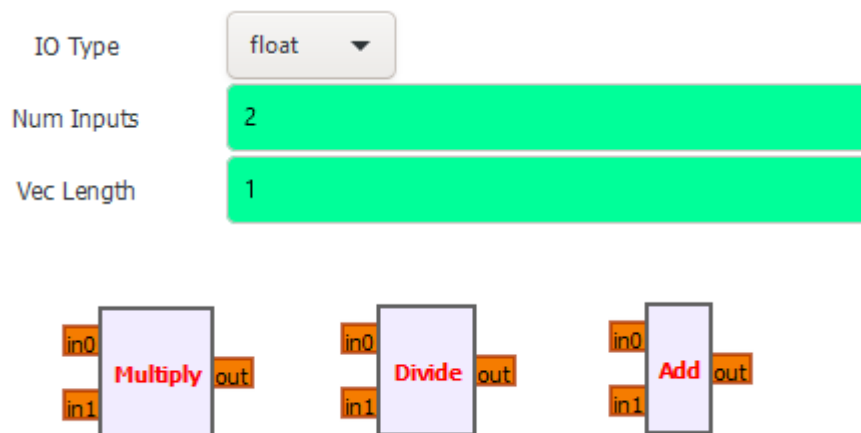


Figura 1.38 Operaciones básicas

Como se puede ver en la figura 1.38 la configuración más importante que hay que realizar es seleccionar el tipo de variable que van a ser la entrada y el número de entradas que va a operar.

El siguiente bloque que se debe colocar es el “Selector” este se encuentra en la librería misc, igualmente se puede utilizar el buscador para encontrarlo, este bloque permite mostrar una salida o varias dependiendo de la señal que se le mande y como se configure, en la siguiente imagen se puede ver como poder configurar dicho bloque.

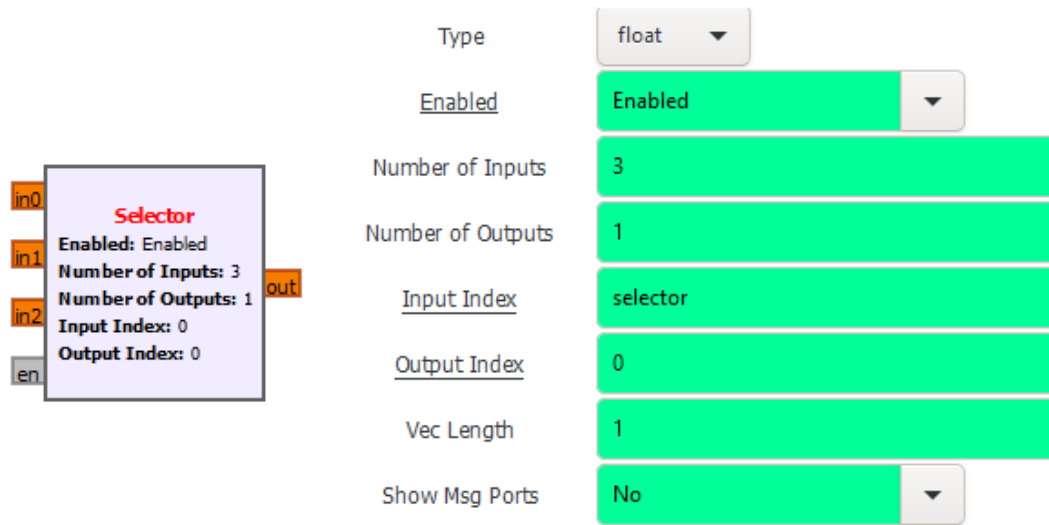


Figura 1.39 bloque selector

Como se ve la figura 1.39 el primer paso es identificar el tipo de variable que se va a utilizar, luego es necesario seleccionar el número de salidas y de entradas, el siguiente parámetro es el índice de la entrada, para esto es necesario utilizar un nuevo bloque de variable llamada “Gui chooser”.

“Gui chooser” es un bloque que permite tener varias opciones de selección de acuerdo con el número de entradas o salidas que se necesiten, En la configuración de este bloque es necesario indicar el id el cual debe ser único entre todos los bloques que se encuentra en la zona de trabajo, lo siguiente es configurar el nombre de las opciones para poder identificarlas fácilmente. Lo último es colocar el ID del este nuevo bloque en la sección de selector llamada “Input index”

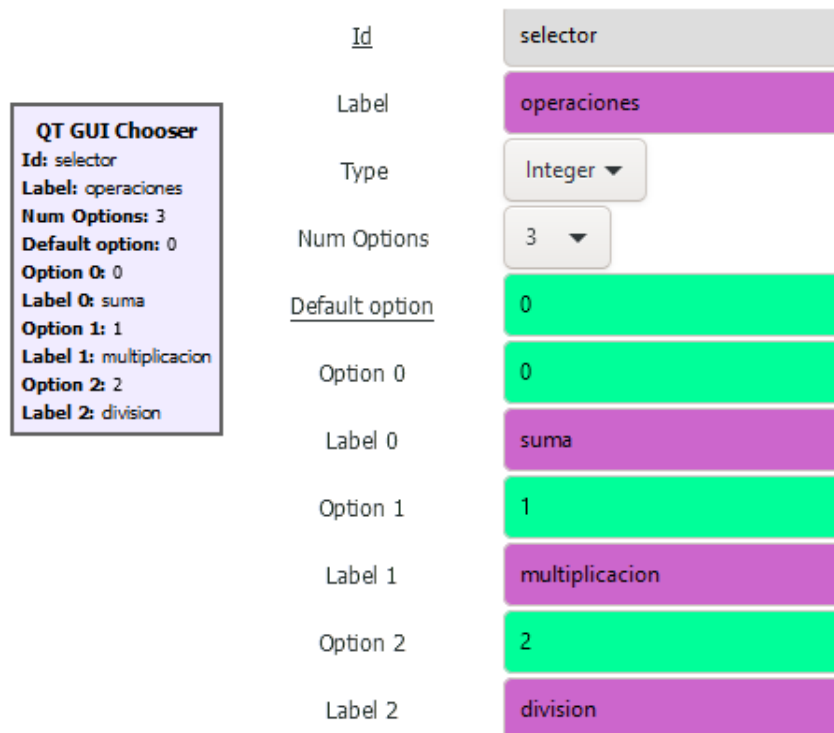


Figura 1.40 Gui chooser

Finalmente, con los bloques vistos ya se pueden realizar las conexiones y configuraciones necesarias para realizar el ejercicio propuesto al comienzo de esta sección.

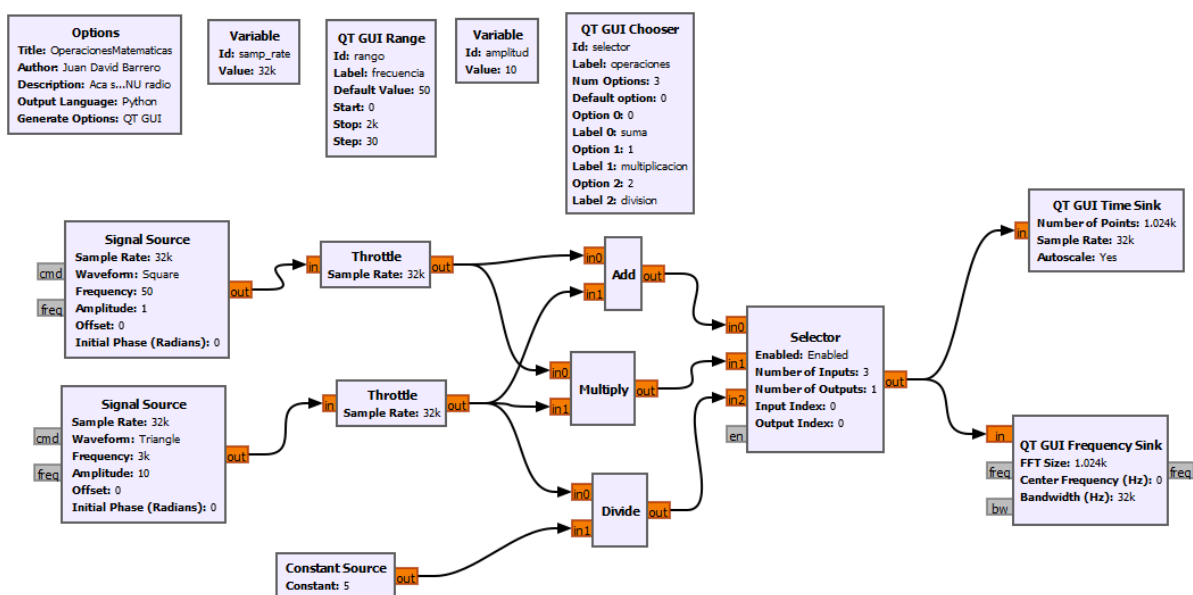


Figura 1.41 Resolución del ejercicio propuesto

En la figura 1.41 se observan un bloque nuevo que no se nombró durante la sección llamado “GUI range” el cual permite tener generar una barra de progreso, esta se usara para cuadrar la frecuencia de una de las señales, esta será una herramienta importante para futuras secciones.

En la siguiente figura se puede observar la configuración para hacerlo funcionar, lo más importante es seleccionar el tipo de datos que va a tener que procesar.

|                      |                    |
|----------------------|--------------------|
| <u>Id</u>            | rango              |
| Label                | frecuencia         |
| Type                 | float ▼            |
| <u>Default Value</u> | 50                 |
| Start                | 0                  |
| Stop                 | 2e3                |
| Step                 | 30                 |
| Widget               | Counter + Slider ▼ |

Figura 1.42 Gui range bloque

En el siguiente enlace se puede descargar el diagrama de bloques donde está la resolución del ejercicio propuesto en esta sección (<https://github.com/JuanDavidBarrero/GNURadio/tree/main/Practice> ).

### 1.3.3 Filtros GNU radio

Los filtros son una parte primordial en todo tratamiento de señales, estos se pueden definir como una caja negra la cual tiene como objetivo atenuar o resaltar las características de la señal de entrada. Estas sirven para evitar que pasen ciertas frecuencias de una señal.

Existen varios tipos de filtros disponibles en GNU radio y todos se pueden encontrar en la librería “Filters” que ya viene instalada por defecto en la aplicación. En esta parte del curso se mostrará 4 de los filtros más importantes que se usan (filtro pasa bajo, pasa alto, pasa banda y rechaza banda), y como poder configurarlos para poder separar señales que previamente se han mezclado.

El ejercicio que se realizará en esta sección se trata de poder filtrar tres señales que previamente se han mezclado. Para esto se necesita tres generadores de señales y un bloque de suma, a la salida de esto se colocará filtros los cuales recuperaran las características deseadas.

Para comenzar es necesario buscar los filtros, para esto se puede buscar el filtro pasa bajo en el buscador (Este se usa para no dejar pasar las frecuencias más altas a una frecuencia de corte), como se ven en la siguiente figura esos son los parámetros que se necesitan para configurar el bloque y poder recuperar la señal de baja de frecuencia.

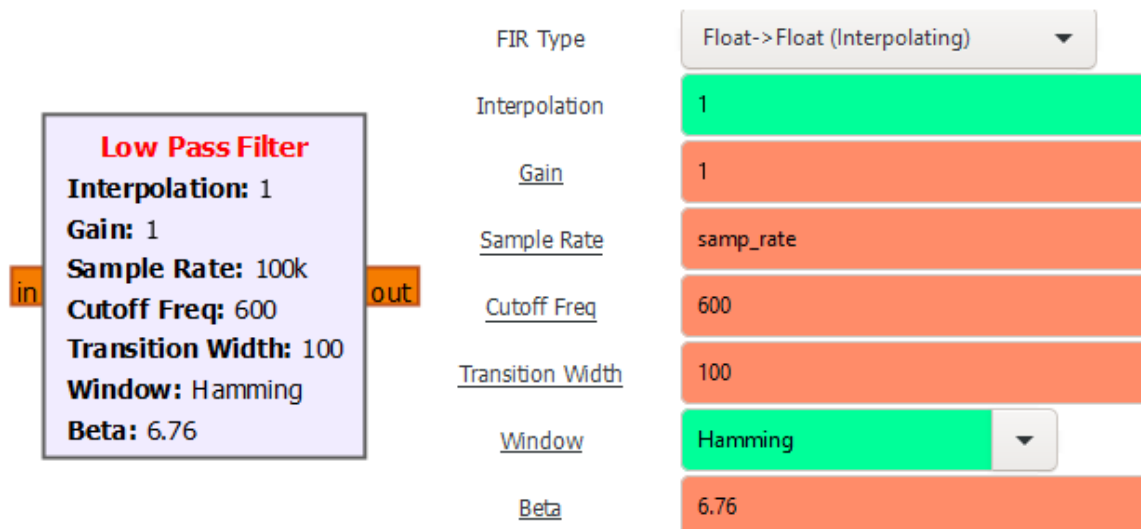


Figura 1.43 Parámetros filtro pasa bajo

El primer paso es escoger el tipo de datos que va a manejar el filtro, además se da la opción de escoger si este va a interpolar o diezmar el número de muestras de la señal de entrada, el siguiente parámetro es la ganancia del filtro, finalmente se coloca la frecuencia del corte y el ancho de la ventana de transición (este puede ser más grande dependiendo del orden que se quiera el filtro).

El tipo de ventana que se puede usar depende de comportamiento que se quiera en el filtro. El valor beta solo se aplica si se usa la ventana tipo Kaiser.

El siguiente filtro del que se nombrara es el filtro pasa alto que tiene como función solo dejar pasar las señales que superen la frecuencia de corte, este se configura de la misma manera que el filtro pasa bajo visto en la figura 1.43, en la siguiente imagen se observa cómo se ve el bloque y la configuración que tiene para el ejercicio propuesto.

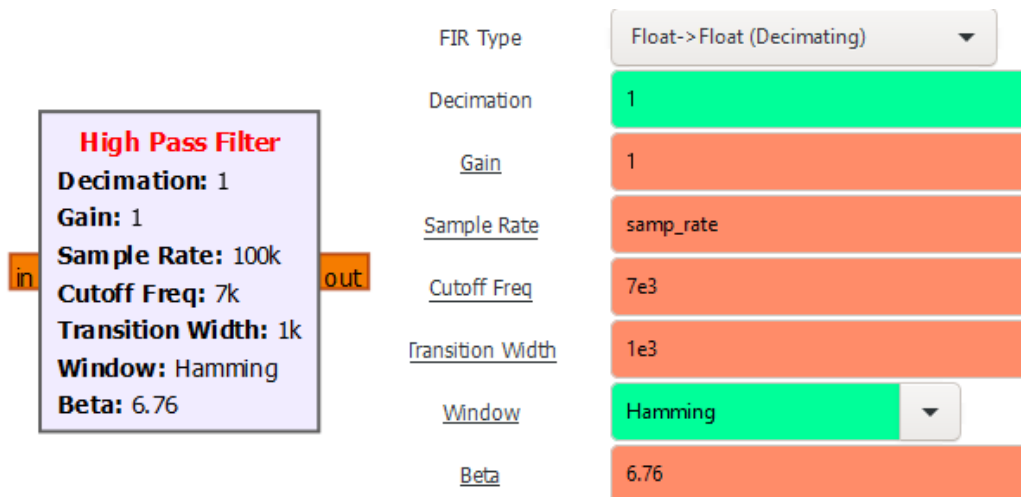


Figura 1.44 Filtro pasa alto parámetros

Los siguientes filtros son la pasa banda y rechaza banda estos poseen dos frecuencias de corte una para las frecuencias bajas y otra para las frecuencias altas. Estos filtros dejaran pasar o rechazar la frecuencia que se encuentren en el rango propuesto. Como se muestra en la siguiente figura.

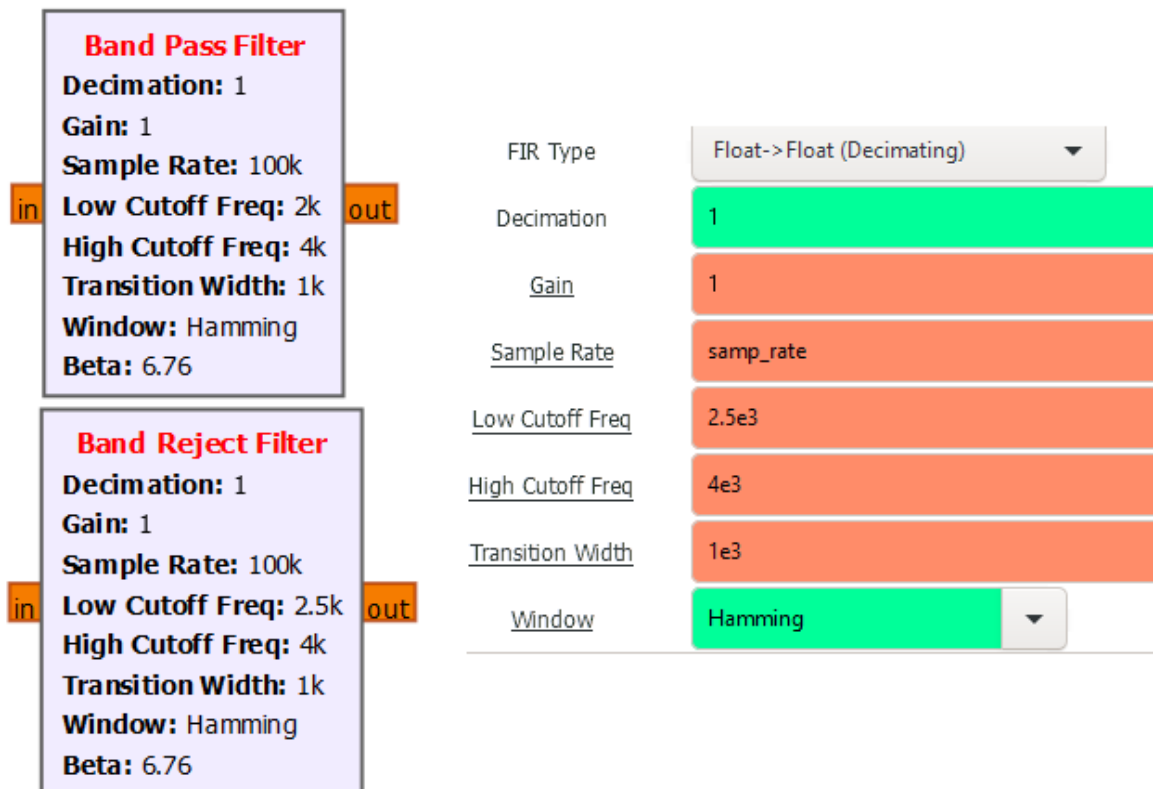


Figura 1.45 Filtro pasa banda y rechaza banda

Con los filtros vistos, ya se puede resolver el problema propuesto, una solución posible para esta ve en la figura 1.46 y 1.47 donde se observa los diagramas de bloques y las salidas de estos respectivamente.

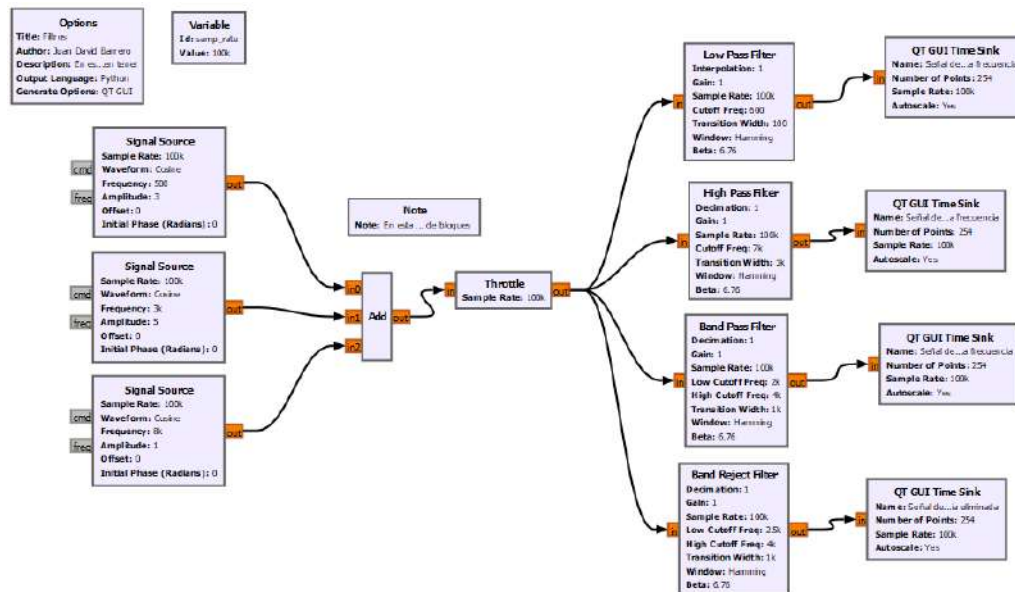


Figura 1.46 Problema tres resuelto

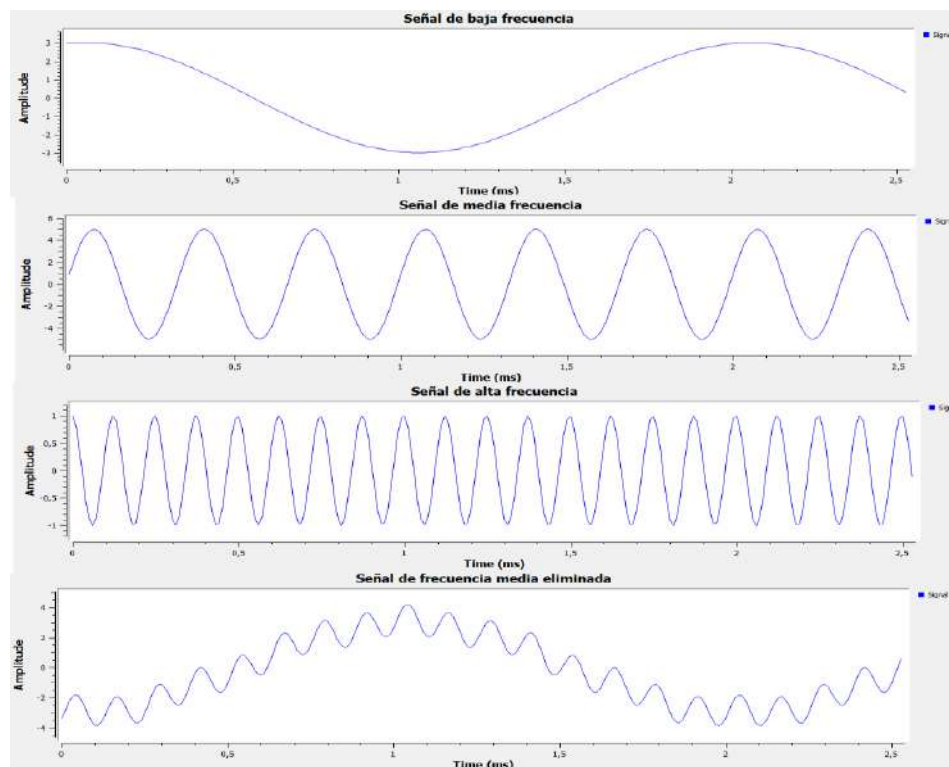


Figura 1.47 Salida espera del problema

El código fuente de esta sección se puede encontrar en el siguiente enlace (<https://github.com/JuanDavidBarrero/GNURadio/tree/main/Practice>).

### 1.3.4 Audio file GNU radio

GNU radio cuenta también con la habilidad de poder utilizar el micrófono del computador para poder grabar lo que este capte, y también puede usar los parlantes para reproducir el tono o audio que se esté generando. Esto será de mucha ayuda para poder transmitir señales de audio más complejas que los tonos vistos en las secciones anteriores.

Para poder hacer uso de este bloque hay que utilizar la librería de audio integrada en el entorno de trabajo, allí se encuentra el bloque de “audio sink” el cual permite tener acceso a parlantes y el bloque “audio source” el cual habilita el micrófono y la señal que capten podrá ser usada por GNU radio.

En esta sección se propone el siguiente ejercicio el cual consiste en grabar un audio utilizando el software Audacity, y pasarlo por alguno de los filtros vistos en la sección anterior, para poder observar el comportamiento en una señal real.

Para comenzar hay que abrir Audacity tal y como se vio en la figura 1.14 luego es necesario escoger la frecuencia de muestreo, por defecto esta es 44100 pero se puede ajustar (hay que tener en cuenta hay ciertas frecuencias que GNU radio no puede trabajar), una vez ya fijada la frecuencia es necesario pulsar botón rojo señalado en la figura 1.48, de esta forma ya se puede grabar audio.

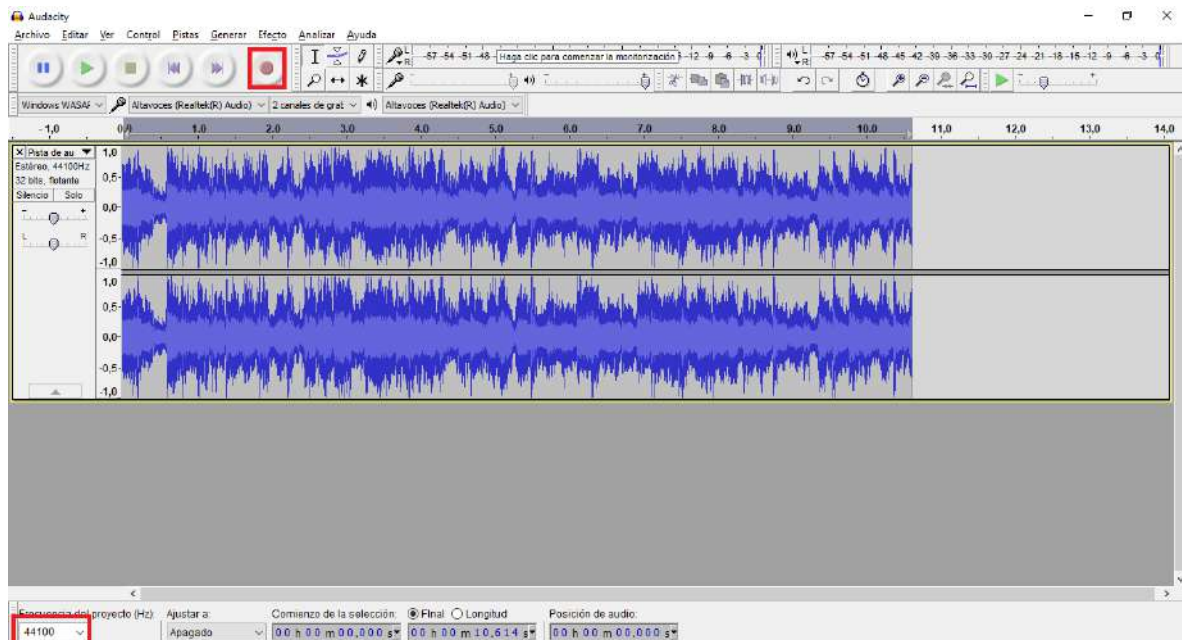


Figura 1.48 Grabar audio con Audacity

Una vez satisfecho con el audio grabado, toca exportar el archivo en formato .wav el cual es el recomendado para trabajar en GNU radio tal como se muestra en la figura 1.49.

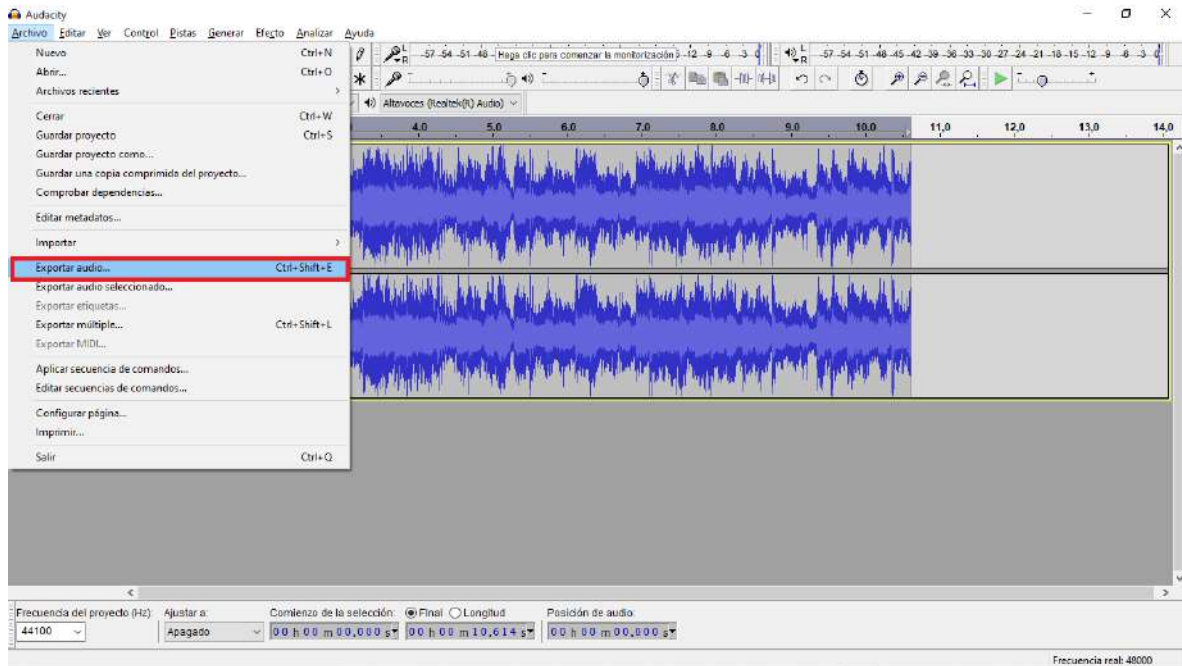


Figura 1.49 Exportar audio formato wav

Para utilizar la señal de audio que recién se ha grabado es necesario utilizar el bloque “wav source” disponible en GNU radio, la configuración de este es simplemente colocar la dirección donde se encuentra y si se desea repetir el audio cada vez que este termine.

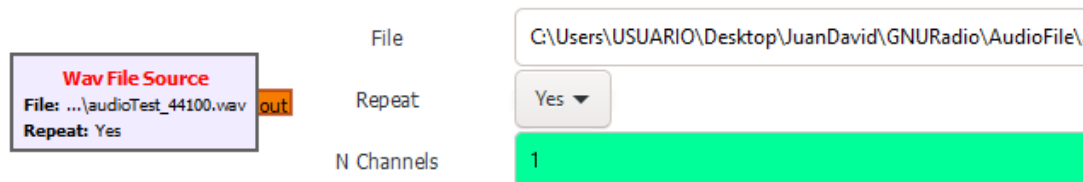


Figura 1.50 bloque wav source

El siguiente bloque que se debe utilizar para realizar el ejercicio propuesto es el “audio sink”, este se debe configurar con la misma frecuencia de muestreo que el audio guardado, si esto no se hace así puede haber interferencias o problemas a la hora de volver a reproducirlo.



Figura 1.51 Bloque audio sink

Como se puede ver en la figura 1.51 estas son frecuencias de muestreo admitidas por GNU radio, ya que estas son las predeterminadas por la mayoría de los computadores. Si se desea guardar el audio modificado se utilizar el bloque “wav sink” como se ve en la siguiente figura.

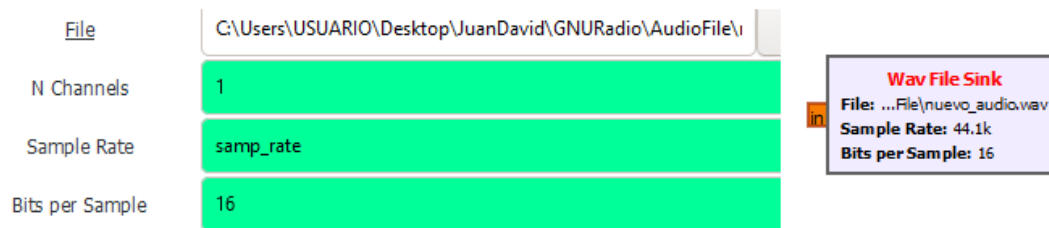


Figura 1.52 bloque wav source

Con los bloques ya vistos se puede realizar el trabajo propuesto, una solución para este se ve a continuación

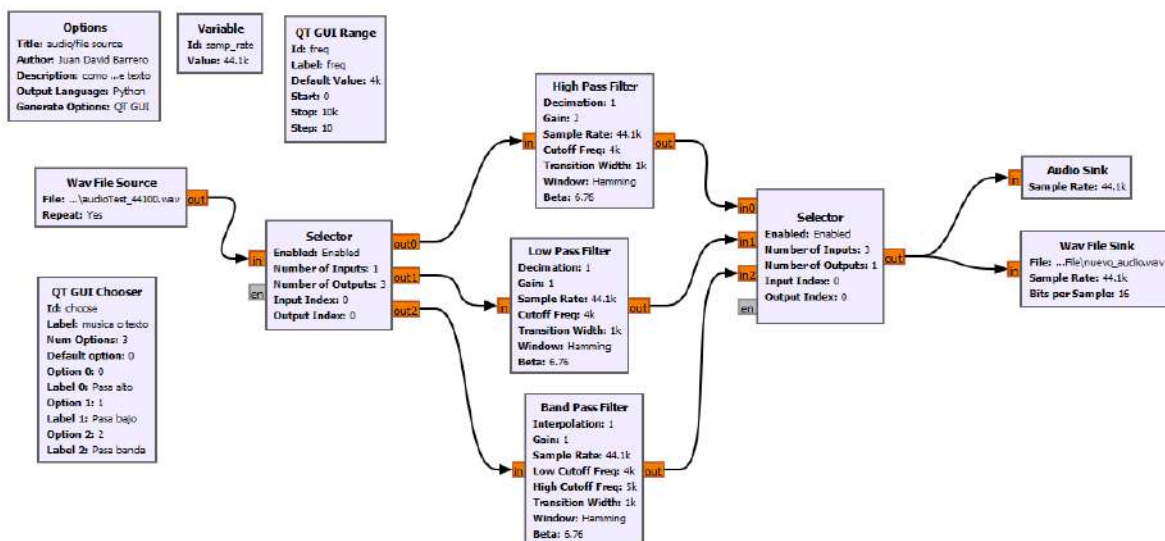


Figura 1.53 Ejercicio 4

El código final de esta sección se puede encontrar en el siguiente enlace (<https://github.com/JuanDavidBarrero/GNURadio/tree/main/Practice> )

## 2. Modulación AM utilizando RDS

En el mundo de las telecomunicaciones las modulaciones son necesarias para poder transmitir a larga distancia todo tipo de información. Este sistema funciona con un sistema de dos señales, la que tiene el mensaje o data llamada moduladora ya que esta es la que modula o envuelve a la portadora y le da su forma característica, y la otra señal como se dijo anteriormente es la portadora ya que esta tiene una frecuencia superior a la moduladora y le añade la propiedad de poder recorrer grandes distancias.

La modulación por amplitud se logra “al variar la amplitud de una señal sinusoidal con frecuencia y fase fija en proporción a una señal dada, esto altera a la señal, trasladando sus componentes de frecuencia a frecuencias más altas” (Introducción a los sistemas de comunicación, 2000).

### 2.1 Modulación por amplitud portadora suprimida

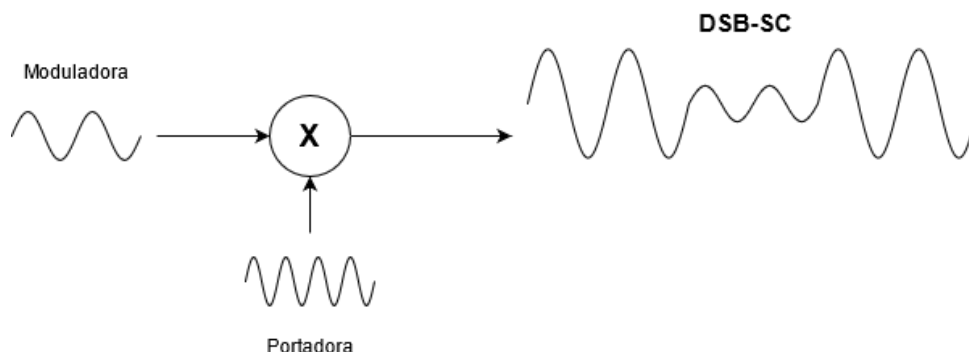
Esta es una modulación lineal que consiste en modificar la amplitud de la señal portadora en función de las variaciones de la señal de información o moduladora. La modulación en doble banda lateral equivale a una modulación AM, pero sin reinserción de la portadora.

La ecuación general de una señal sinodal puede describirse con la siguiente ecuación:

$$\phi_{Am} = a(t) \cos(2\pi ft)$$

*Ecuación 3 DSB-SC*

Esta ecuación se puede representar en el siguiente diagrama de bloques.



*Figura 2.1 DSB-SC*

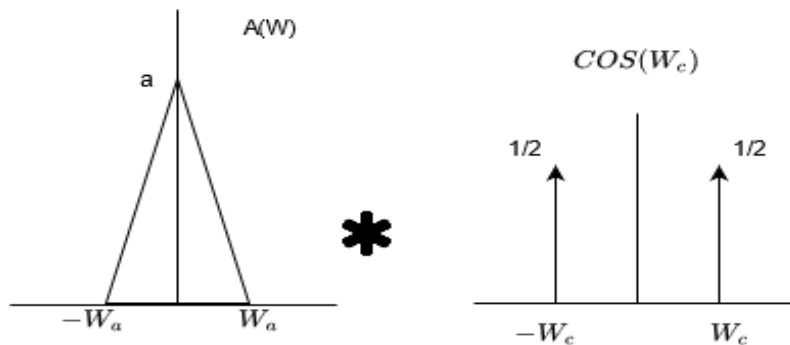
Al hacer un análisis matemático en el dominio de la frecuencia se puede observar la razón por la cual este tipo de modulación se llama doble banda lateral portadora suprimida (DSB-SC), para poder ver esto hay que tomar la ecuación 3 y

transformarla en su equivalente en el dominio de la frecuencia. Tal y como se puede ver en el siguiente ecuación.

$$\phi_{AM} = A(w) * \frac{1}{2} [ \delta(w - w_c) + \delta(w + w_c) ]$$

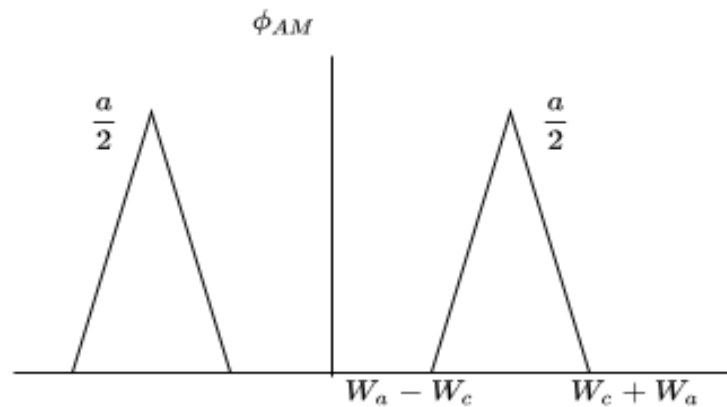
*Ecuación 4 DSB-SC Fourier*

Esta ecuación se puede expresar como la convolución de la señal moduladora con la señal portadora y gráficamente se ve de la siguiente forma.



*Figura 2.2 Convolución grafica*

Al realizar la operación se obtiene la siguiente respuesta



*Figura 2.3 Resultado de la convolución*

En el resultado de la operación que se observa en la figura 2.3 se ve claramente que el ancho de banda de la señal de salida es el doble que, al comienzo, y que la amplitud ahora se ha reducido a la mitad. Además se observa que la señal de la portadora ha desaparecido, por este motivo es que a este tipo de modulación se le conoce como DSB-SC.

## 2.1.1 Doble banda lateral portadora suprimida GNU radio simulación

Es momento de pasar a ver como poder generar esta modulación en GNU radio, en este punto ya se debe tener el nivel suficiente para poder generar la señal y poder ver en el analizador de espectro si lo que se ha mostrado en la parte de teoría se cumple.

Para esto se armar el siguiente diagrama de bloques en GNU es necesario volver ver la figura 2.1. con la diferencia de que ahora se pondrá dos tipos de señal moduladora, una es un tono al cual se le puede variar la frecuencia y la otra es una señal de audio previamente grabada.

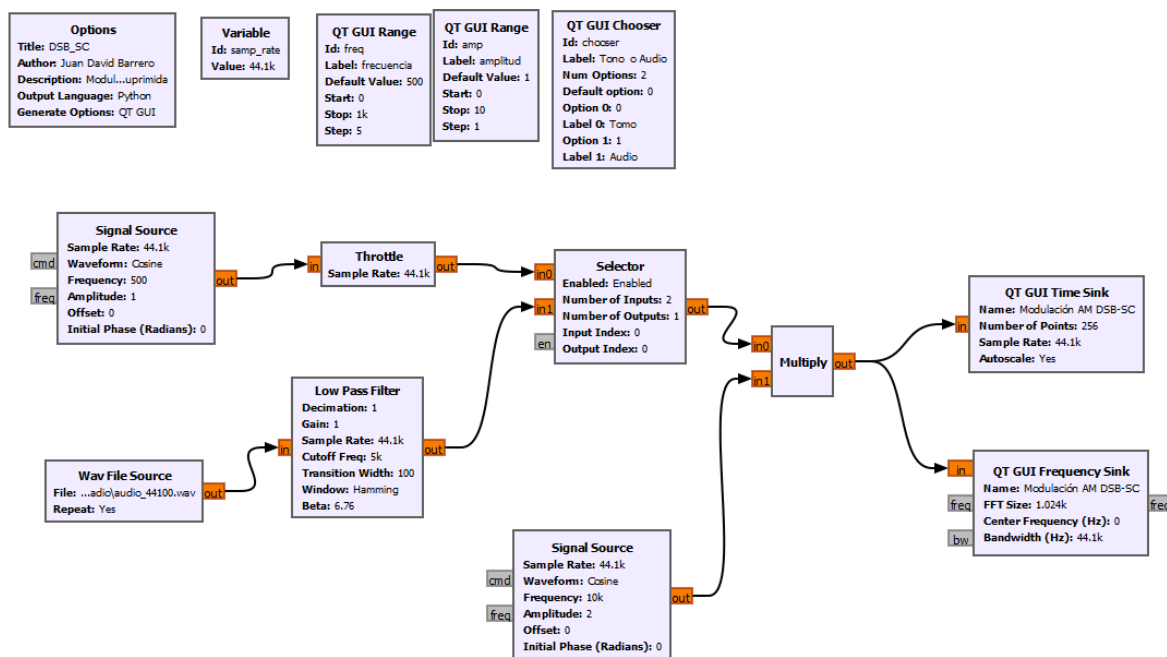


Figura 2.4 DSB-SC en GNU radio

Al ejecutar el código se puede observar como la señal de audio modula a la señal portadora, también se ve como ahora ancho de banda se hace dos veces más grande (10 KHz), y si se usa el tono se puede ver como al variar la frecuencia de la señal moduladora el ancho de banda aumenta. Si se quiere se puede añadir un bloque final de audio sink para escuchar la señal AM desde el computador.

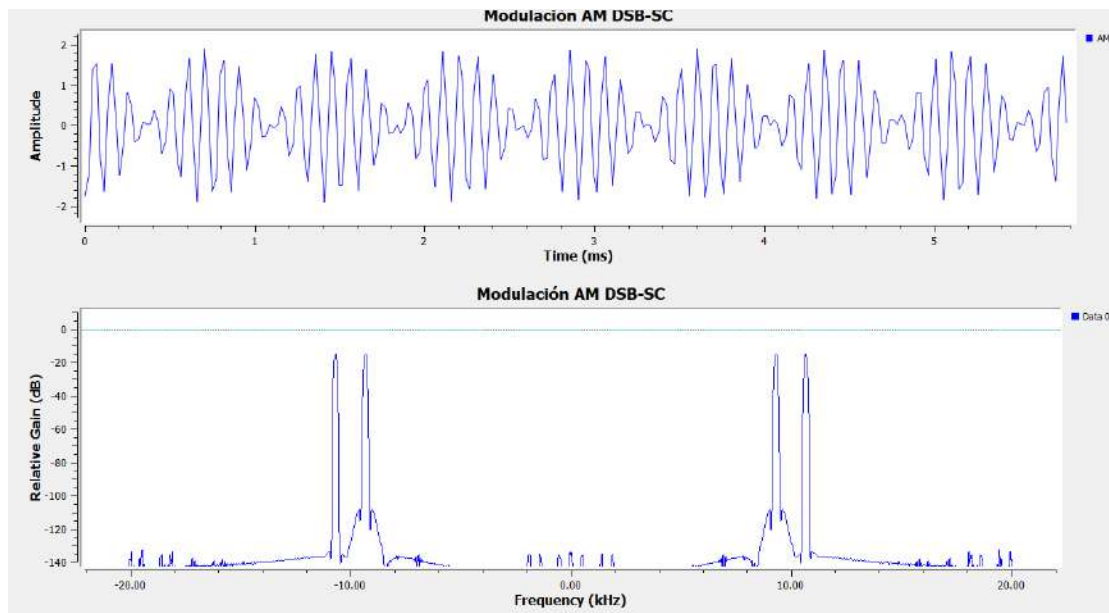


Figura 2.5 Modulación de tono DSB-SC

Como se observa en la figura 2.5 el tono produce una señal modulada en amplitud sin portadora, pero es evidente que la portadora se encuentra en lo 10KHz, la señal de moduladora se puede medir y esta se encuentra en lo 500Hz por lo que el ancho de banda es de 1KHz.

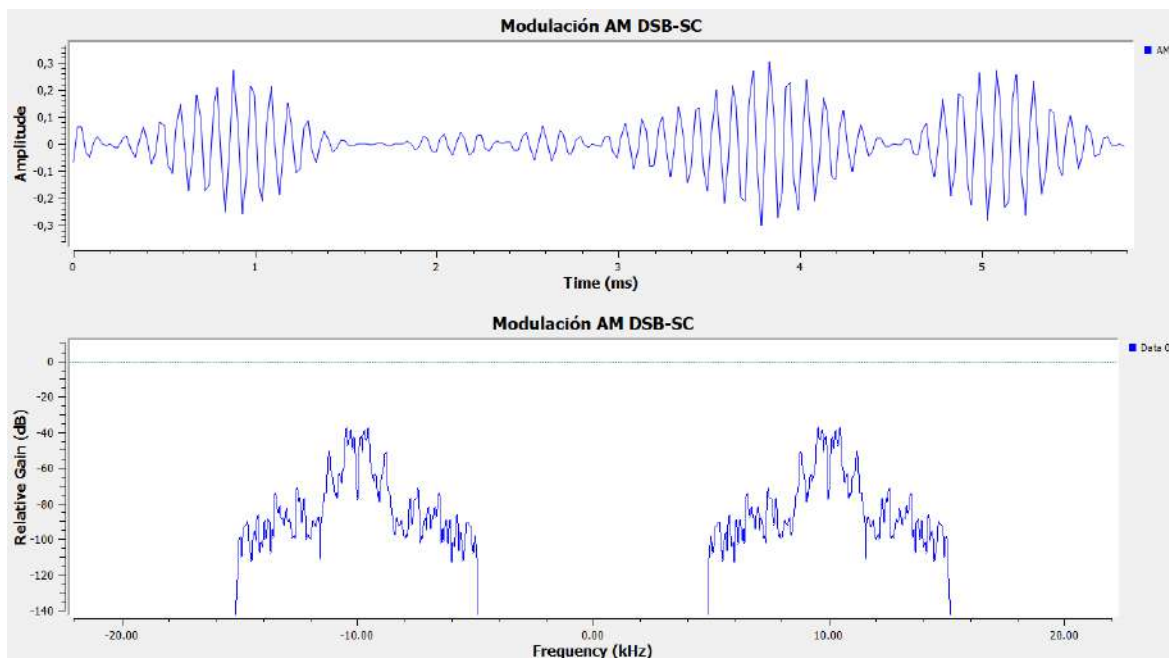


Figura 2.6 Señal de audio DSB-SC

A diferencia de la figura 2.5, en la figura 2.6 se observa como ahora en vez de tener dos armónicos de una señal seno, se obtiene todos los armónicos de la voz por lo que se ve una la forma de onda tal y como se calculó en la figura 2.3.

## 2.1.2 DSB-SC GNU radio SDR transmisión

En este punto se mostrará como transmitir el primer mensaje utilizando el SDR (HackRf) mostrado en la sección de dos de este trabajo, pero antes de esto hay que instalar los drivers necesarios para que el computador se pueda comunicar con el sin ningún tipo de problema.

Para poder hacer esto se debe utilizar el software Zadig tal y como se puede ver en la figura 16 y seguir la siguiente lista de pasos.

1. Conectar el RDS al puerto serial del computador
2. Abrir el software de Zadig (Con el dispositivo ya conectado)
3. Dar clic en opciones y seleccionar todos los dispositivos

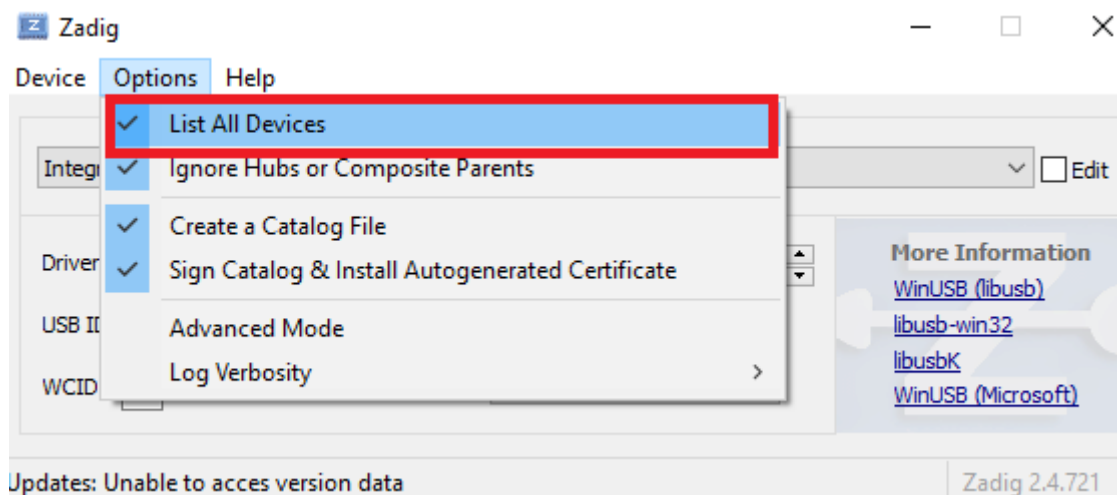


Figura 2.7 Instalando drivers RDS

4. Escoger la opción que dice "Bulk-in, Interface (Interface 1)"

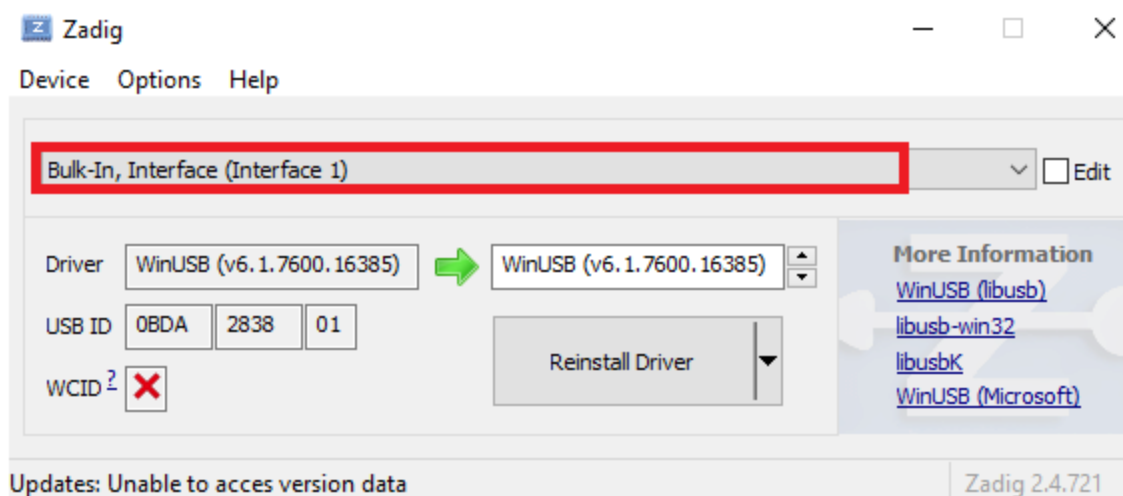


Figura 2.8 Drivers RDS

5. Finalmente se hace clic en la opción instalar drivers
6. Si todo se realizó correctamente debe aparece el siguiente letrero.

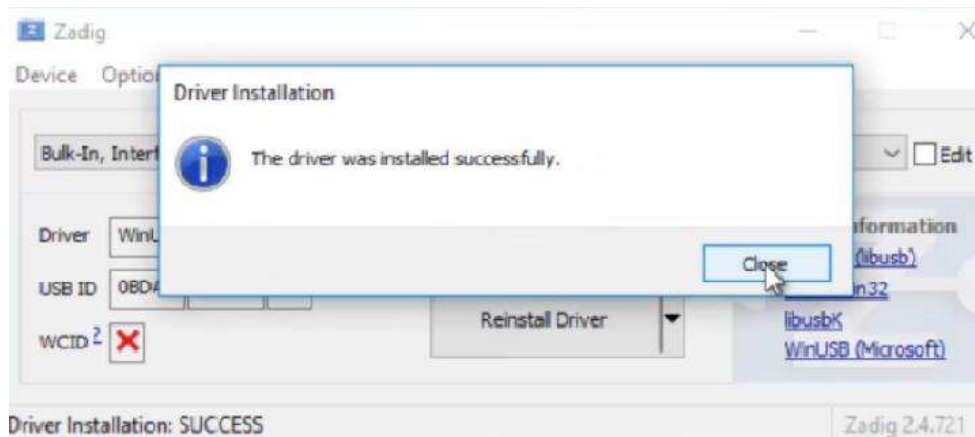


Figura 2.9 Drivers instalados correctamente

7. Repetir ese proceso con cada uno de los RDS que se tenga.

Con los drivers ya instalados se puede utilizar los bloques de GNU radio para programar los distintos RDS. Es importante recordar que durante este trabajo se utilizara el **RTL-SDR dongle** y el **HackRf-ONE**, cuyas librerías ya se encuentran instaladas en la aplicación.

En la librería de Osmocom se encuentra el bloque que permite utilizar los recursos internos del HackRf-ONE, si se quiere transmitir es necesario utilizar el bloque “Osmocom sink”, el cual se puede ver en la siguiente figura.



Figura 2.10 bloque Osmocom sink

Los parámetros que se deben modificar para poder utilizar son los siguientes:

- **Device argument:** Acá se debe colocar el nombre el dispositivo o el id que aparece en dispositivos conectados, si se deja en blanco este se conectara al primer RDS que encuentre disponible.
- **Number channels:** Si se quiere enviar más de una señal al mismo tiempo, se puede habilitar más número de canales.
- **Samp\_rape:** Este parámetro define cual es la frecuencia de muestreo con la que se va a enviar la señal, una recomendación es que debe ser elevada para tener una calidad de audio decente.
- **Ch0 Frequency:** Esta es la frecuencia de la portadora en la que se sube la señal que se desea transmitir, esta debe estar en un rango donde no interfiera con las otras señales.
- **RF gain:** Esta es la ganancia de la parte real de la señal.
- **IF gain:** Es la ganancia de la parte imaginaria de la señal.
- **BB gain:** ganancia banda base del dispositivo.
- **Bandwith:** Es el ancho de banda que ocupa la señal al ser enviada.

Las señales que el RDS utiliza son complejas, así que para poder convertir las señales flotantes vistas en todos los bloques anteriores hay que usar un convertidor, el cual también se encuentra disponible en la librerías instaladas en GNU radio.

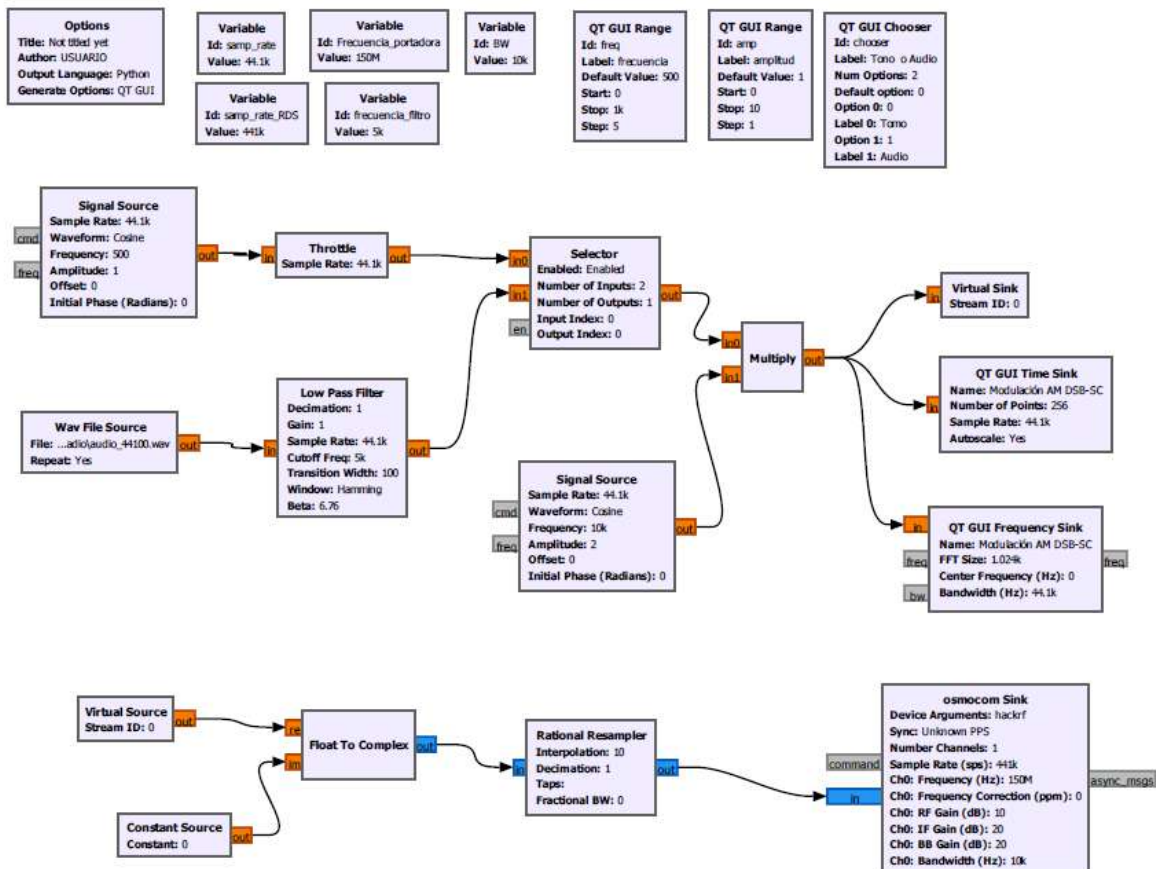


Figura 2.11 DSBSC-RDS\_TX

En la figura 2.11 se muestra dos bloques nuevos, el primero es uno llamada “virtual sink” y “virtual source”, este bloque permite conectar dos diagramas de bloques separados en el mismo diagrama de flujo, lo cual hace que este sea más ordenado y fácil de entender. El código se puede encontrar en el siguiente enlace ([https://github.com/JuanDavidBarrero/GNURadio/tree/main/AM\\_modulation](https://github.com/JuanDavidBarrero/GNURadio/tree/main/AM_modulation) ).

El otro bloque nuevo que ve es el “Rational Resampler” este permite interpolar o diezmar el número de muestras de una señal, esto se vuelve muy útil a la hora de poder trabajar con distintas señales, ya que en el dominio discreto siempre se tiene que trabajar con el mismo número de muestras.

Al ejecutar el código la señal se está transmitiendo en la frecuencia de la portadora que se asignó al RDS, esta podría estar en cualquier rango del espectro, pero para no tener problemas se usan frecuencias altas en los rangos libres del espectro, para este ejercicio esa frecuencia es 156 MHz.

Aún queda el problema de cómo saber si de verdad se está transmitiendo la señal y con las características especificadas, ya que aún no se ha diseñado el receptor para esto se utilizará el software SDR # el cual permite sintonizar todo tipo de frecuencias y además cuenta de moduladores para poder recuperar la señal original.

El primer paso es abrir el software y seleccionar el RDS que se utilizara, para poder captar la señal se utilizara el RTL SDR Dongle, entonces ese es el que se va a debe buscar, luego se cuadra a la frecuencia que se quiera sintonizar, una vez hecho esto ya se puede dar en el botón play para comenzar a escuchar que se encuentra en ese rango.

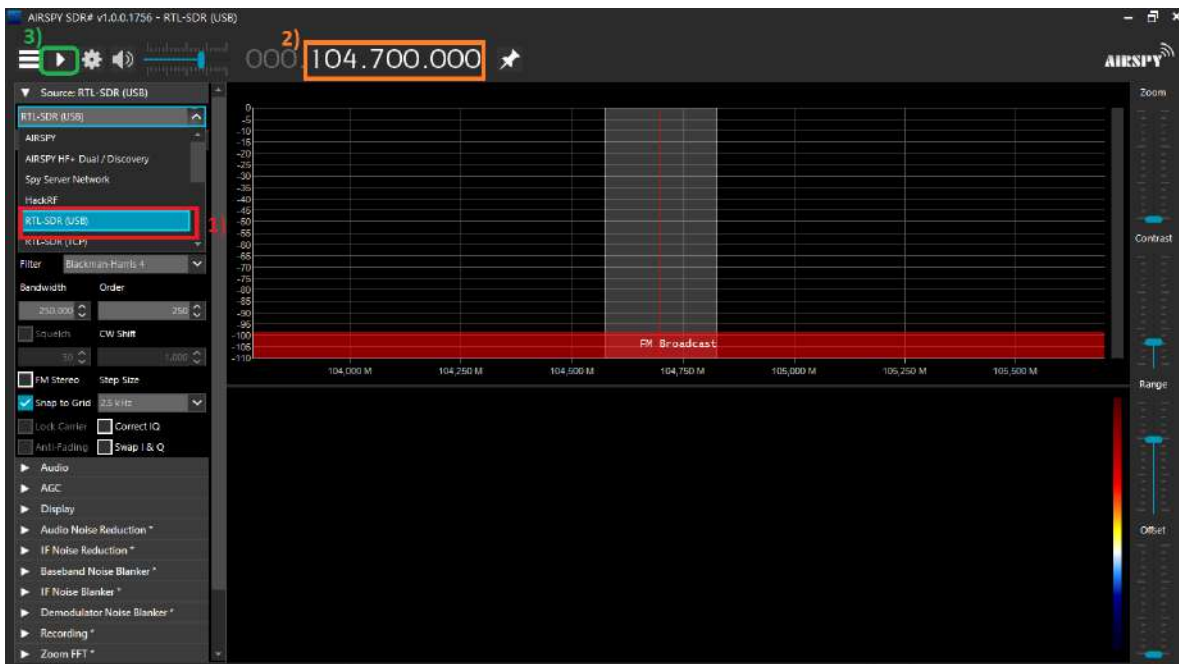


Figura 2.12 primeros pasos SDR#

Para comprobar que el dispositivo esté funcionando correctamente es recomendable sintonizar alguna emisora conocida, si este está trabajando de

manera correcta por los parlantes del computador se debe escuchar la información que se está transmitiendo, además se puede utilizar el mouse para ir cambiando en tiempo real la frecuencia la cual se está sintonizando.

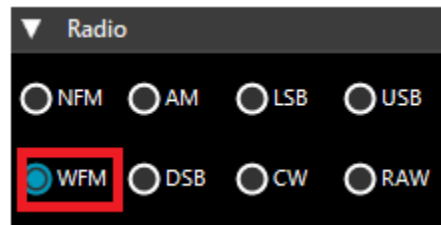


Figura 2.13 ajustando SDR#

Por defecto el radio viene ajusta para sintonizar FM, así que no se debe hacer ninguna modificación para escuchar este tipo de emisoras, si este no está así hay que ajustarlo o no se podrá decodificar correctamente la información y solo se escuchar ruido.

Si se quiere ajustar los parámetros del RDS con el que se está trabajando es necesario hacer clic donde se muestra a continuación.

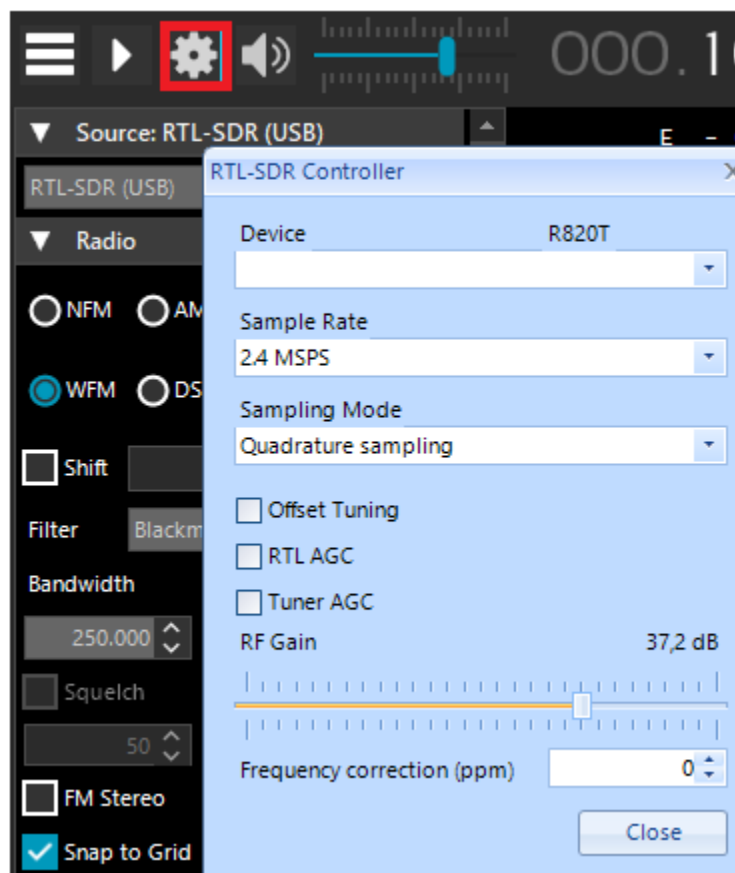
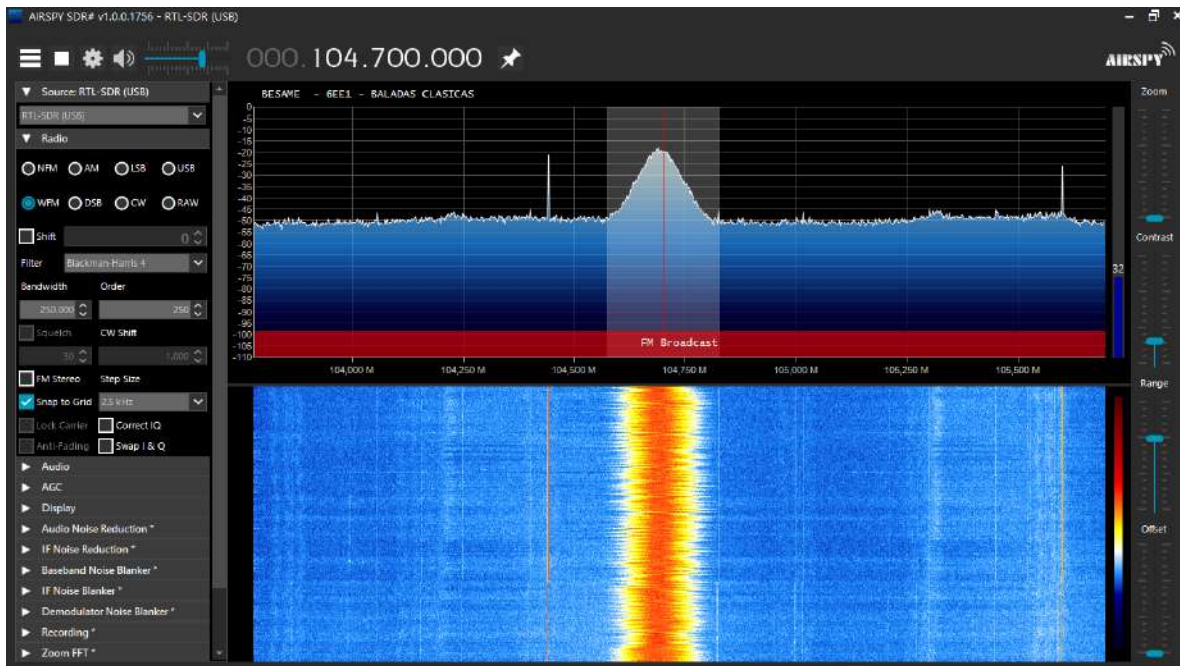


Figura 2.14 Ajustar parámetros en SDR#



*Figura 2.15 Sintonizando estaciones de FM*

Si todo se encuentra en orden se debe poder sintonizar todo tipo de emisoras FM y ser escuchadas por los parlantes del equipo, además se observa el espectro de la señal y la intensidad de esta en el diagrama de cascada visto en la zona de abajo en el diagrama de cascada.

Confirmado que todo esté funcionando correctamente con SDR#, ya se puede ejecutar el código visto en la figura 2.11 en un computador y si se quiere desde otro PC se puede conectar el RTL SDR Dongle para poder oír la información que se está transmitiendo desde el HackRf-ONE. Como se va a trabajar son señales AM se debe colocar el radio en AM para que esta aplicación puede decodificar la información correctamente. Y la frecuencia central debe ser la misma de la portadora colocada el bloque Osmocom sink.

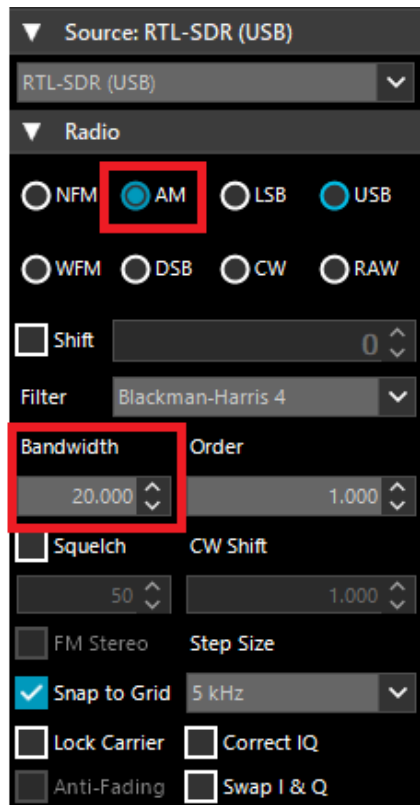


Figura 2.16 Ajustes para recibir señales AM

El ancho de banda se debe ajustar para captar mejor la señal cuando se esté de codificando, éste es un parámetro importante a tener en cuenta o si no se puede captar el ruido que se encuentra en el ambiente y en elementos del dispositivo.

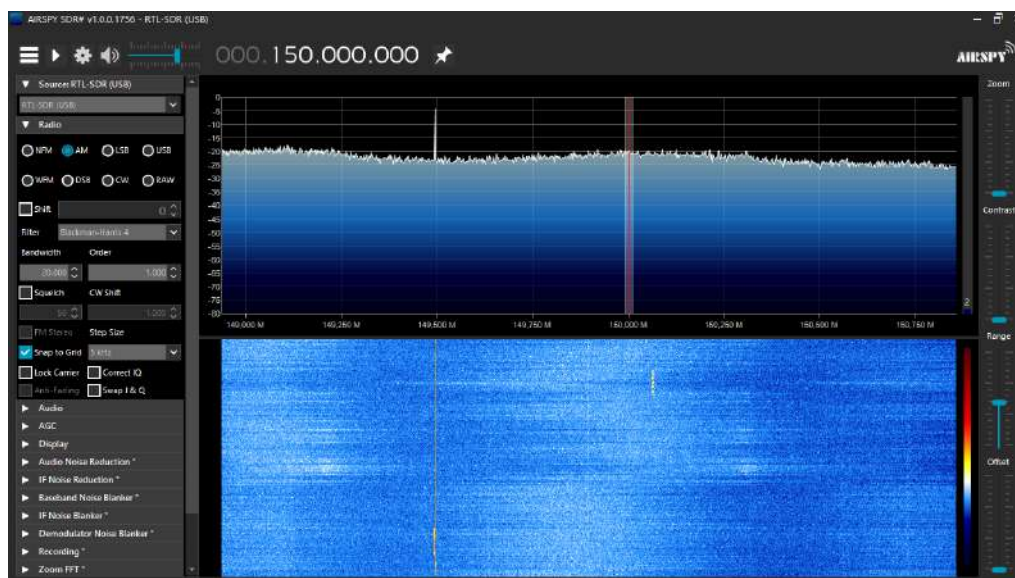
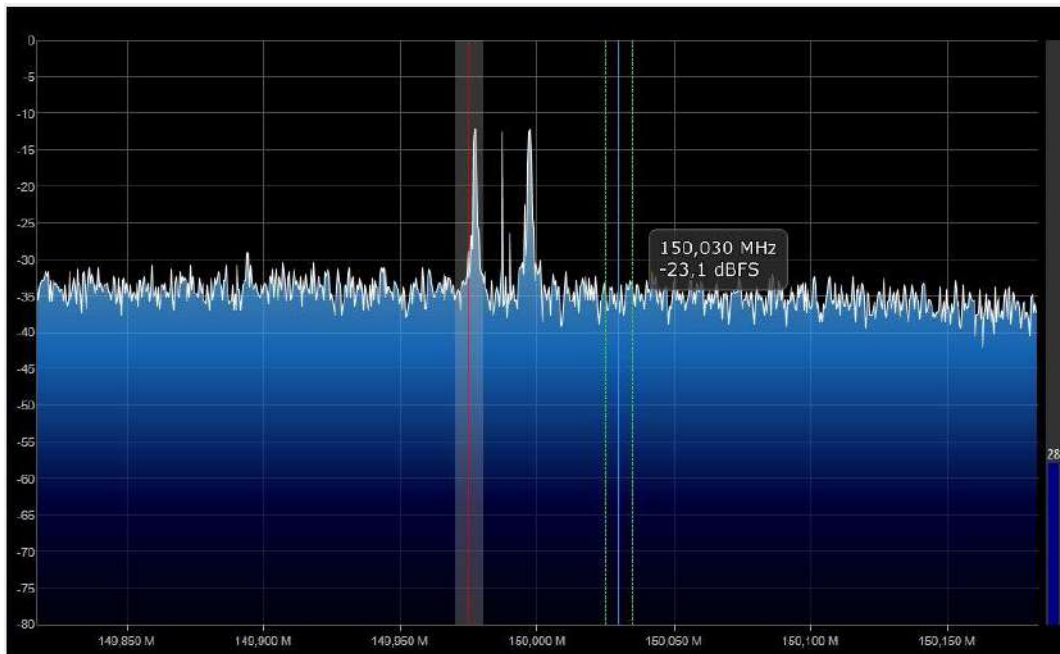


Figura 2.17 Antes de transmitir la señal

Con todo esto visto ya se puede probar que todo funcione, en la figura 2.17 se observa que no se está transmitiendo nada, pero cuando se ejecuta el código se ven los siguientes resultados, además en los parlantes del PC se puede escuchar con claridad el mensaje recibido.

Al ejecutar el código y sintonizar la frecuencia de la portadora en SDR Sharp se puede observar que ahora hay información en ese rango del espectro, tal y como se observa en la en la figura 2.18, en la próxima sección se mostrará como recibir la señal de información utilizando GNU radio.



*Figura 2.18 Señal recibida DSB-SC*

### **2.1.3 DSB-SC receptor simulado GNU radio**

Para poder recibir la información es necesario tener la misma señal de la portadora a la misma frecuencia y a la misma amplitud, en GNU radio hay muchas formas de identificar la portadora de una señal, pero en capítulos más avanzados se mostrará más al respecto.

En esta sección se utilizará el receptor mostrado en el siguiente diagrama de bloques y posteriormente se realizará un análisis en el dominio de la frecuencia para entender como este sistema funciona. Finalmente se pondrá a prueba este en una simulación y se contrastará el resultado.

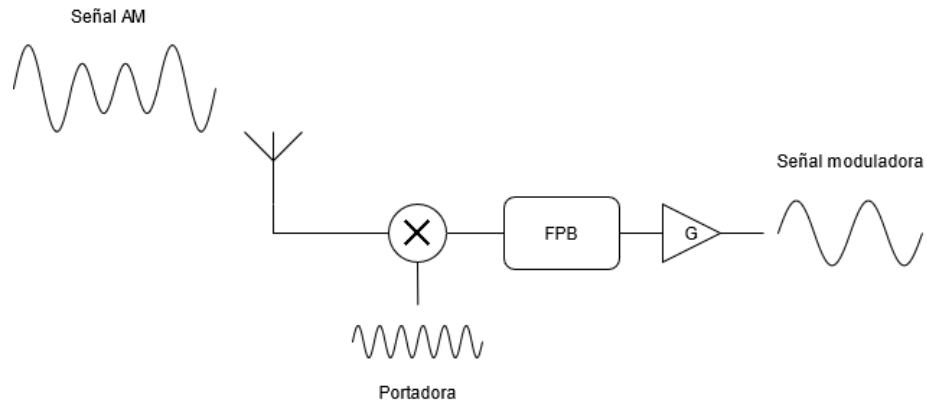


Figura 2.19 Bloque recepción DSBSC

Como se ve la señal antes de llegar al filtro pasa bajos se puede expresar matemáticamente de la siguiente forma.

$$y(t) = a(t) \cos(\omega t) \cdot \cos(\omega t)$$

Ecuación 5 Señal antes del filtro

Si se analiza la señal en el dominio de la frecuencia se observa el siguiente comportamiento.

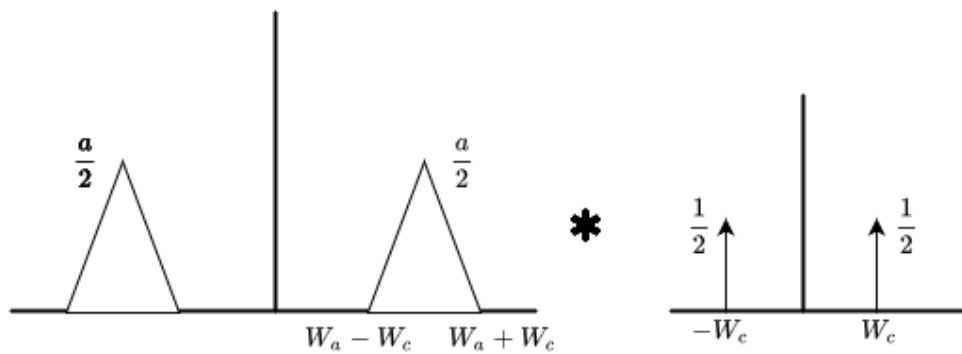


Figura 8 Ecuación 3 en el dominio de la frecuencia

Al realizar la operación se obtiene el siguiente resultado

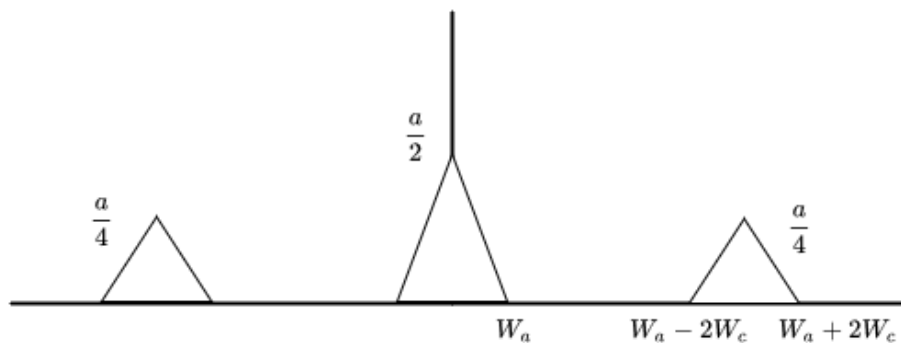


Figura 2.20 Resultado de la ecuación 3

Como se ve en la figura 2.20 la señal ha vuelto a la frecuencia original y su amplitud se ha disminuido un factor de 2, además hay una señal en alta frecuencia, así que para corregir este resultado es necesario utilizar un filtro pasa bajo y colocar una ganancia de 2 para tener la señal original.

En el siguiente diagrama de bloque hecho en GNU radio se va a recrear el visto en la figura 2.19, en donde se mostrará los análisis matemáticos calculados en las ecuaciones de arriba.

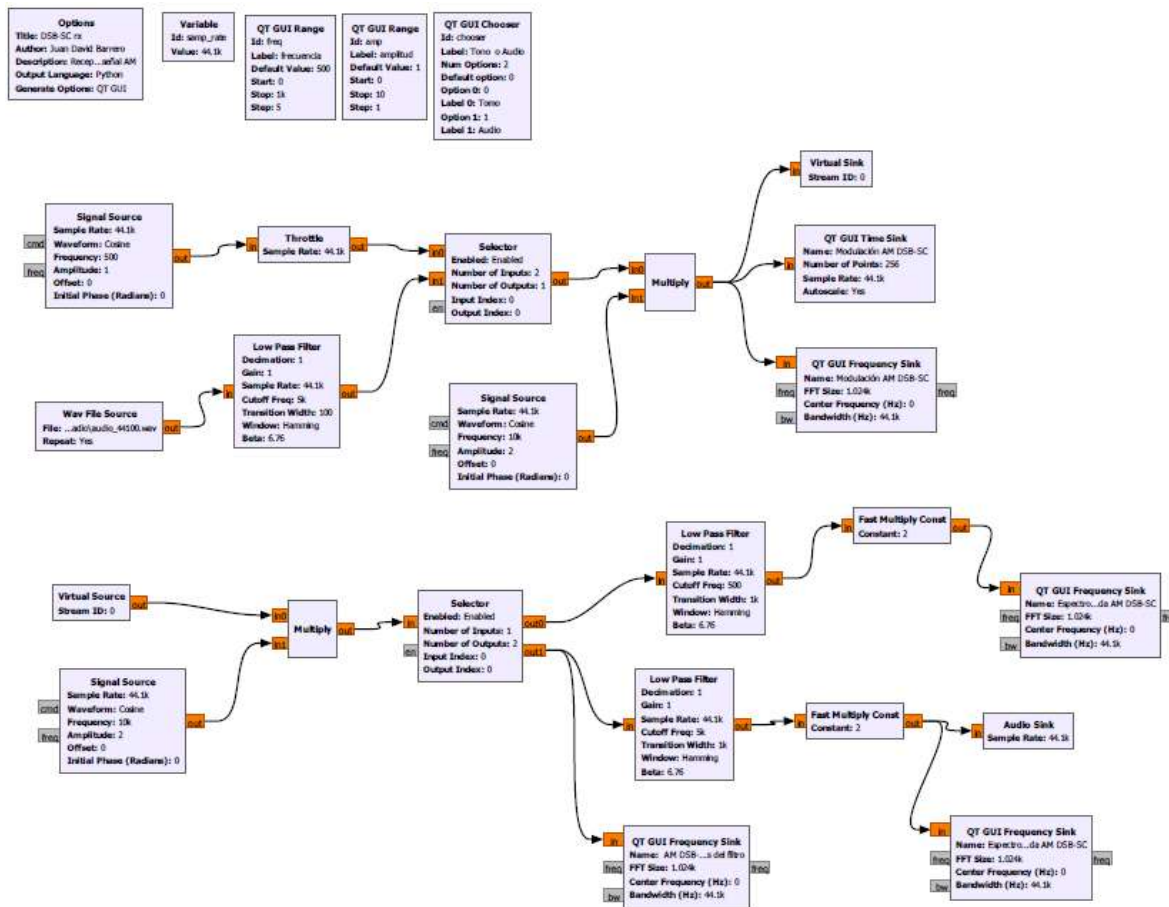


Figura 2.21 Receptor GNU radio DSBSC

Se observa que los bloques ubicados en la parte de arriba representan el transmisor visto en la sección anterior. Los bloques de la parte de abajo representan el diagrama de bloques del receptor tal y como se vio en la figura 2.19, para poder ver este de una manera más clara se mostrara en una imagen separada solo esa sección de todo el diagrama en bloque.

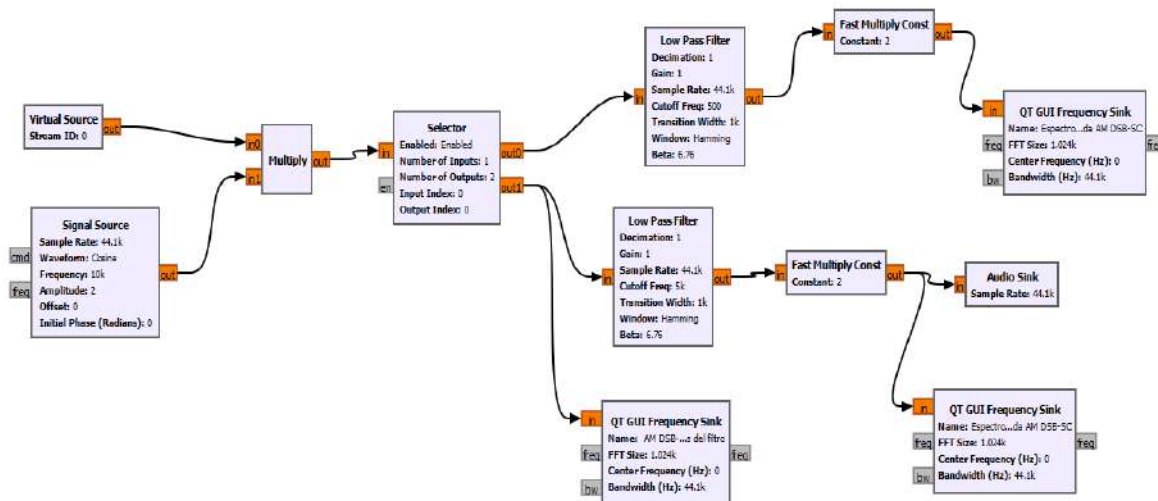


Figura 2.22 Receptor DSB-SC

Tal y como se esperaba los resultados coinciden con la descripción matemática explicada anteriormente.

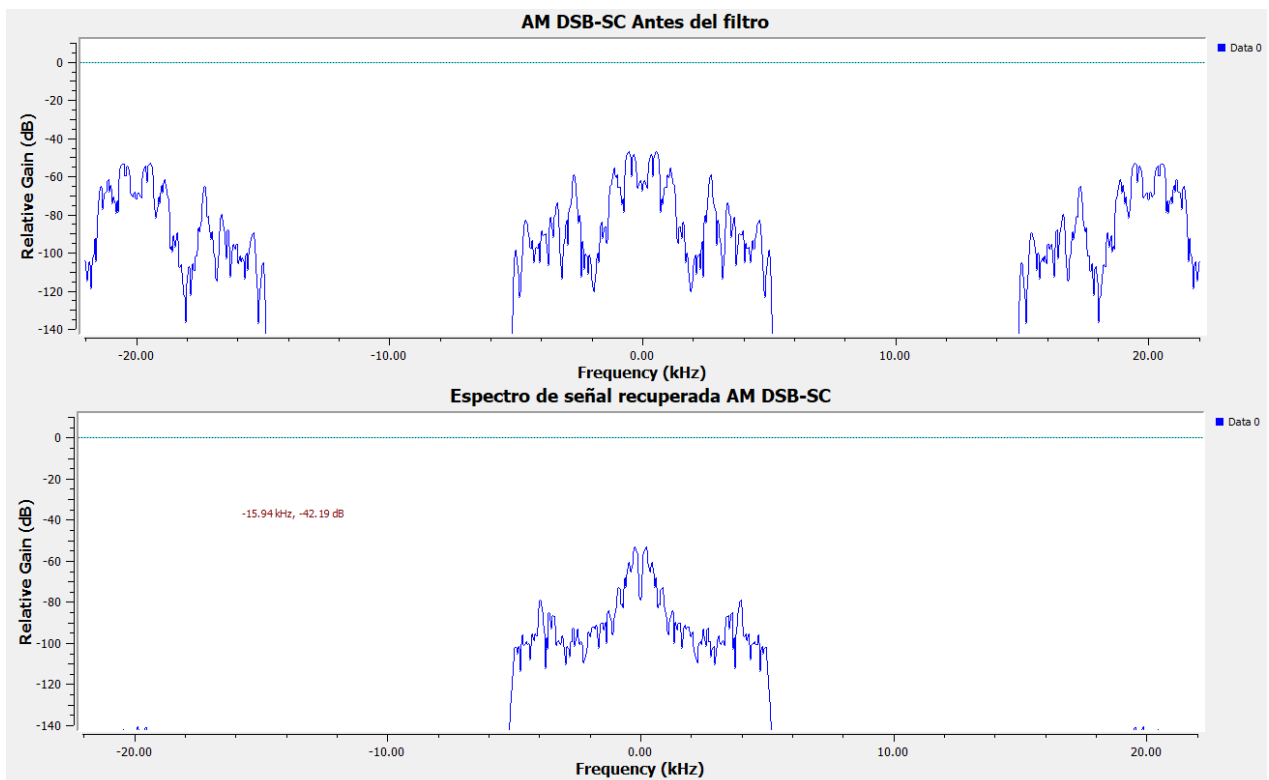


Figura 2.23 Resultados de receptor DSBSC

## 2.1.4 DSB-SC recepción con RDS

Para poder recibir la señal enviada por RDS (HackRF\_one) debe utilizar el bloque RTL source este, en este bloque se debe tener en cuenta que el periodo de muestreo del transmisor coincida con el del receptor, si esto no es así el mensaje enviado estará distorsionado.

Este bloque se encuentra en la librería de Osmocom, tal y como se muestra a continuación

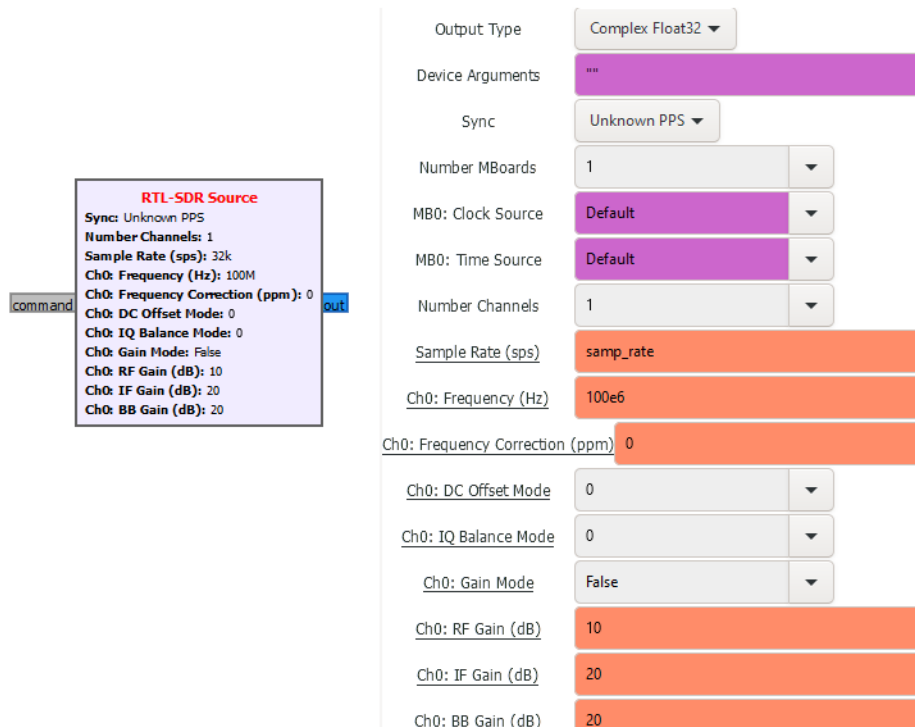


Figura 2.24 Bloque RTL source

Los siguientes parámetros son los que deben ser configurados para el correcto uso del bloque.

- **Device Argument:** Acá se coloca el ID puesto por el computador para identificar al RDS, si se deja en blanco este se conectará al primer dispositivo que encuentre por el puerto serial.
- **Number Channels:** Si se va a recibir en más de un rango de frecuencia se puede seleccionar el número de señales que se van a recibir.
- **Ch0 Frequency:** Acá se selecciona la frecuencia en la se va a estar sintonizando
- **Ch0 RF gain:** La ganancia de la parte real de la señal recibida
- **Ch0 IF gain:** La ganancia de la parte imaginaria de la señal recibida
- **Ch0 BB gain:** La ganancia de banda base.

Con los parámetros bien configurados se puede hacer todo el sistema de recepción que se vio anteriormente para poder captar la señal y ser de-modulada

correctamente para poder escucharla en los parlantes del equipo o ser almacenada en un archivo con formato .wav.

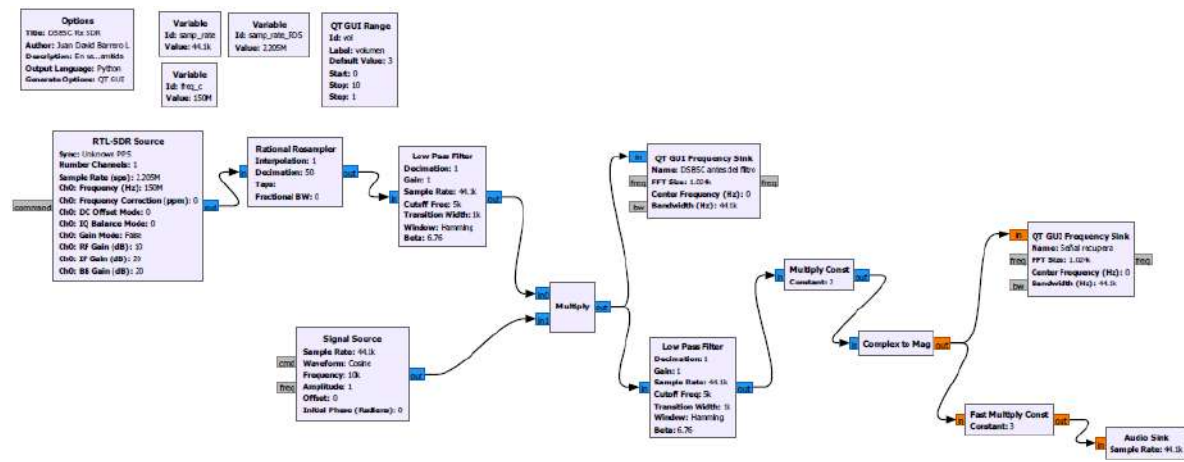


Figura 2.25 Receptor con el RDS

En el diagrama de bloques mostrado en la figura 2.25 se observa los mismos bloques mostrados para de modular y captar la señal de AM enviada, Este código se puede ejecutar desde otro computador o en el mismo, al ejecutarse se debe observar el siguiente resultado.

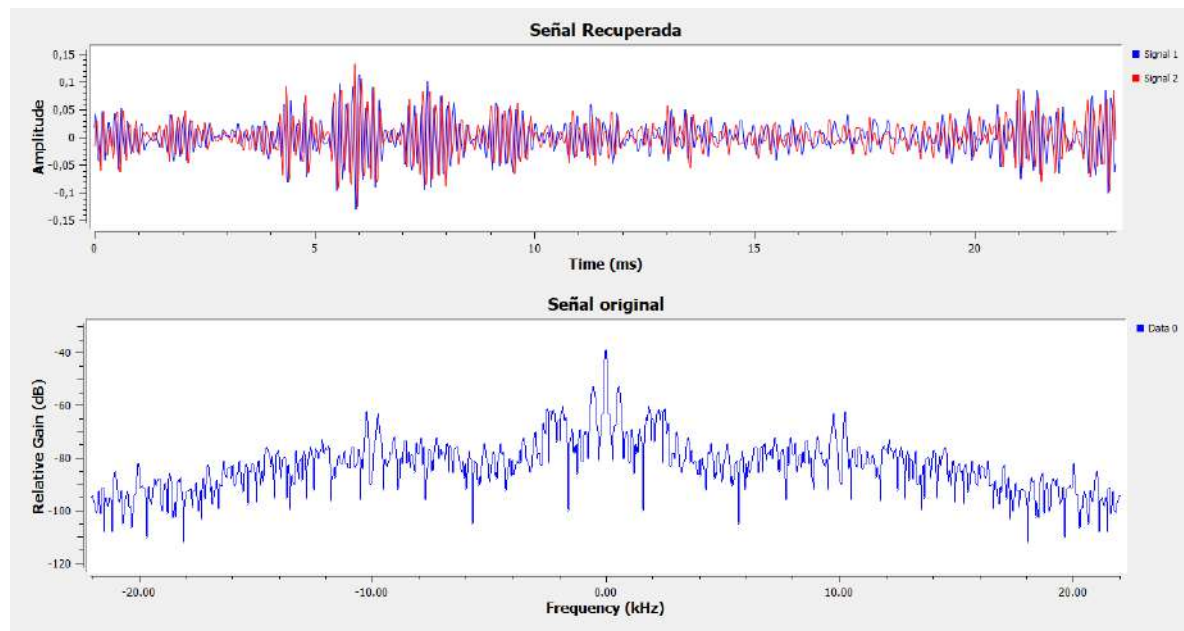


Figura 2.26 Señal de modulada

## 2.2 Doble banda lateral gran portadora

Al utilizar el sistema de transmisión sin portadora la sincronización es un problema, por lo que añadir la señal portadora proporcionara la información necesaria para sincronizar la señal del receptor y poder recuperar nuevamente la señal modulada,

Sin embargo existen ciertos problemas la hacer esto ya que la frecuencia a bajas frecuencia se puede deteriorar por lo que este sistema no conviene para todos los casos.

Este tipo de señal se puede expresar con la siguiente ecuación

$$\phi_{AM} = f(t) \cos(\omega t) + A \cos(\omega t)$$

*Ecuación 6 DSB\_LC*

A su vez esta expresión se puede simplificar en la siguiente:

$$\phi_{AM} = (f(t) + A) \cos(\omega t)$$

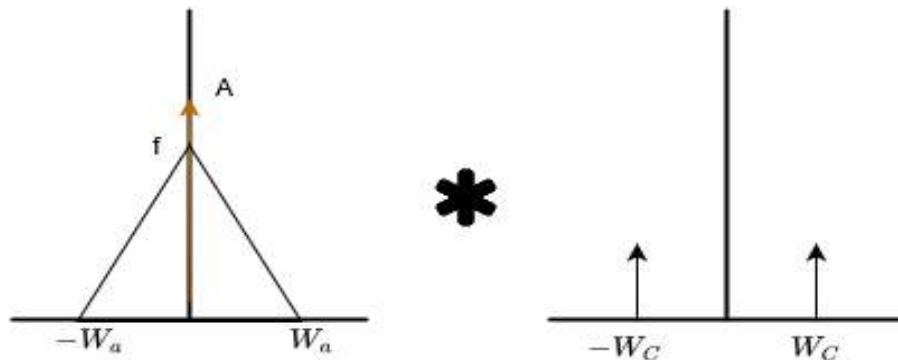
*Ecuación 7 DSB\_LC simplificada*

Al hacer el análisis de estas ecuaciones en el dominio de la frecuencia se obtiene el siguiente resultado:

$$\phi_{AM} = (F(W) + A\delta(W)) * \frac{1}{2}(\delta(W - W_c) + \delta(W + W_c))$$

*Ecuación 8 DSB\_LC dominio de la frecuencia*

Gráficamente se puede ver de esta forma



*Figura 2.27 Ecuación DSBLC*

Al realizar la convolución de la operación vista en la figura 81 se observa como ahora la señal esta desplazada a la frecuencia de la portada y a su vez se puede ver que en su posición esta una función delta que indica donde esta se encuentra ubicada.

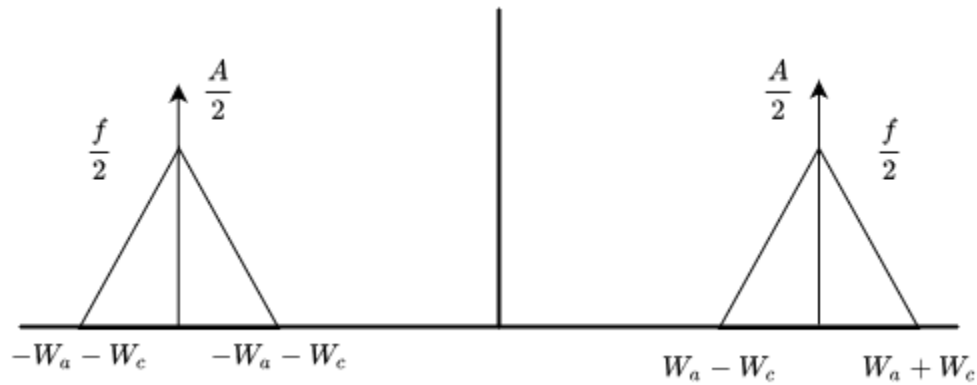


Figura 2.28 Resultado de operación

Estas ecuaciones y este comportamiento se pueden resumir en el siguiente diagrama de bloques.

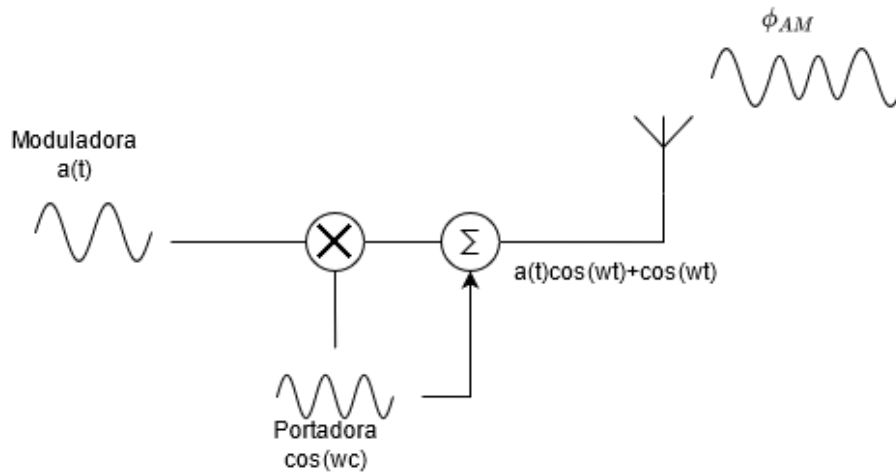


Figura 2.29 Diagrama de bloques para transmisión DSB<sub>LC</sub>

Un requisito fundamental para que se puede recobrar de manera sencilla y exitosa el mensaje de la portadora es:

$$A \geq |\min \{f(t)\}|$$

Ecuación 9 Requisito para recuperar la señal moduladora

Para poder controlar la relación entre las bandas laterales y la portadora se define el valor  $m$  el cual se puede expresar como

$$m = \frac{\text{Amplitud pico DSB} - \text{SC}}{\text{Amplitud pico de la portadora}}$$

Ecuación 10 Índice de modulación

Con la ecuación 8 se puede concluir que la ecuación de AM quede de la siguiente manera

$$\phi_{AM} = A(1 + m\cos(Wmt))\cos(Wct)$$

La desventaja que aparece al usar esta modulación es que gran parte de la energía se pierde al tener que enviar la portadora, pero gracias a la portadora detectar la señal moduladora se vuelve un trabajo más sencillo.

## 2.2.1 DSB-LC simulación en GNU radio Rx y TX

En esta sección se mostrará cómo realizar el diagrama de bloques en el software GNU radio además se mostrará cómo realizar el sistema de recepción el cual puede ser el mismo que se vio en la figura 2.29.

Acá también se mostrará si el comportamiento visto en la matemática de arriba corresponde a lo que se espera ver en la vida real, con esto también se podrá constatar si la recepción funciona de igual manera.

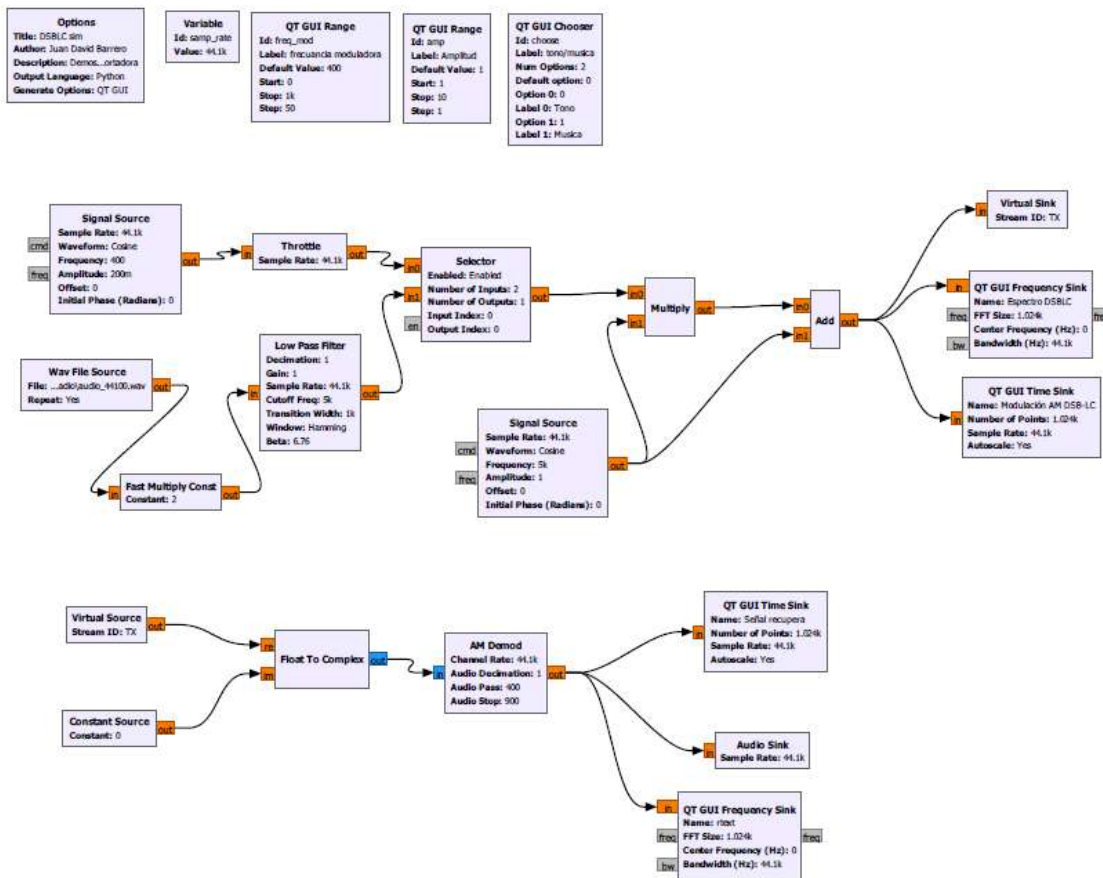


Figura 2.30 Rx y Tx DSBLC

Gracias a que ahora se está transmitiendo la portadora en la misma señal modulada la información ahora es la envolvente de la señal por lo que recuperarla es más simple y esto se puede lograr con el bloque "AM demod" el cual tiene el trabajo de extraer la señal moduladora.

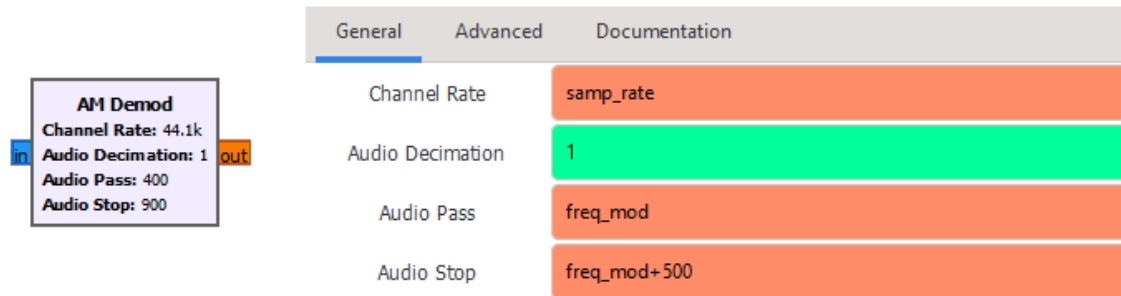


Figura 2.31 Bloque AM demod

Al ejecutar el código generado en la figura 84 cuyo enlace de descarga se puede encontrar en el siguiente ([https://github.com/JuanDavidBarrero/GNURadio/tree/main/AM\\_modulation](https://github.com/JuanDavidBarrero/GNURadio/tree/main/AM_modulation)) se obtiene el siguiente resultado en donde se ve el comportamiento esperado y predicho con la ecuación anteriormente mencionada.

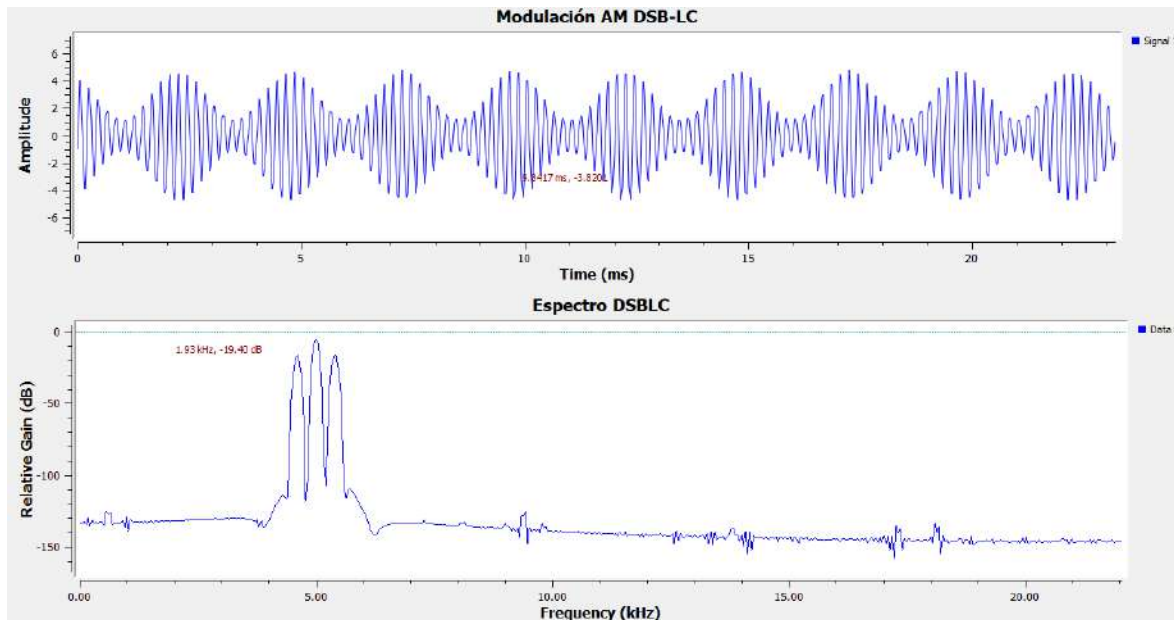


Figura 2.32 Resultados DSBLC tono

En el mismo código también se muestra grafica donde se muestra la señal recuperada y tiene comentado el bloque de audio sink, si se quiere se puede habilitar para poder escuchar el tono o el audio que se está simulando transmitir, en la siguiente figura se puede el resultado de las señales.

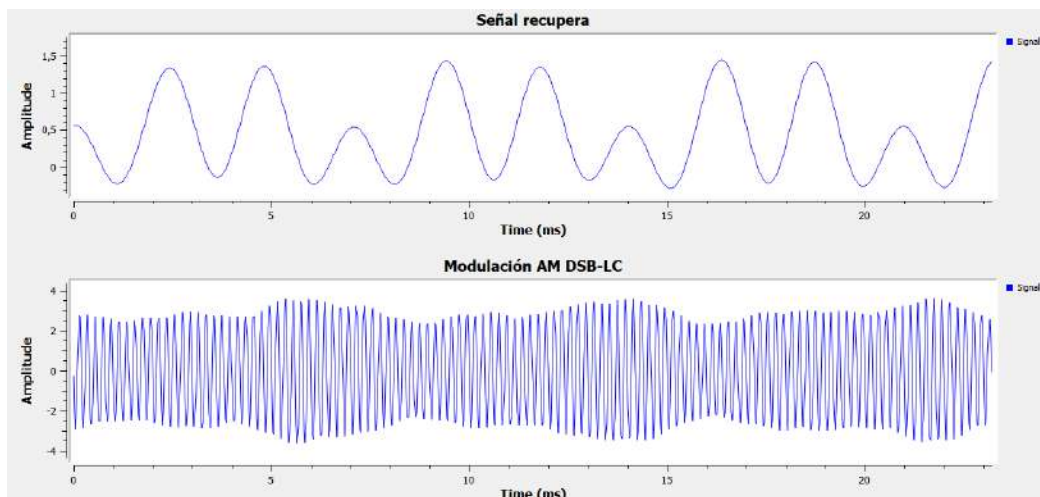


Figura 2.33 Señales recuperadas tono y audio

## 2.2.2 DSB-LC Tx y Rx utilizando RDS

Para poder transmitir utilizando esta tecnología es necesario utilizar los bloques vistos en la sección anterior, esto permite utilizar la arquitectura interna de estos dispositivos para colocar la señal modulada en una nueva portadora y así irradiarlo al aire.

En el siguiente diagrama de bloques se muestra como armar el diagrama de bloques para utilizar el RDS para poder reconstruir una señal modulada en amplitud con gran portadora.

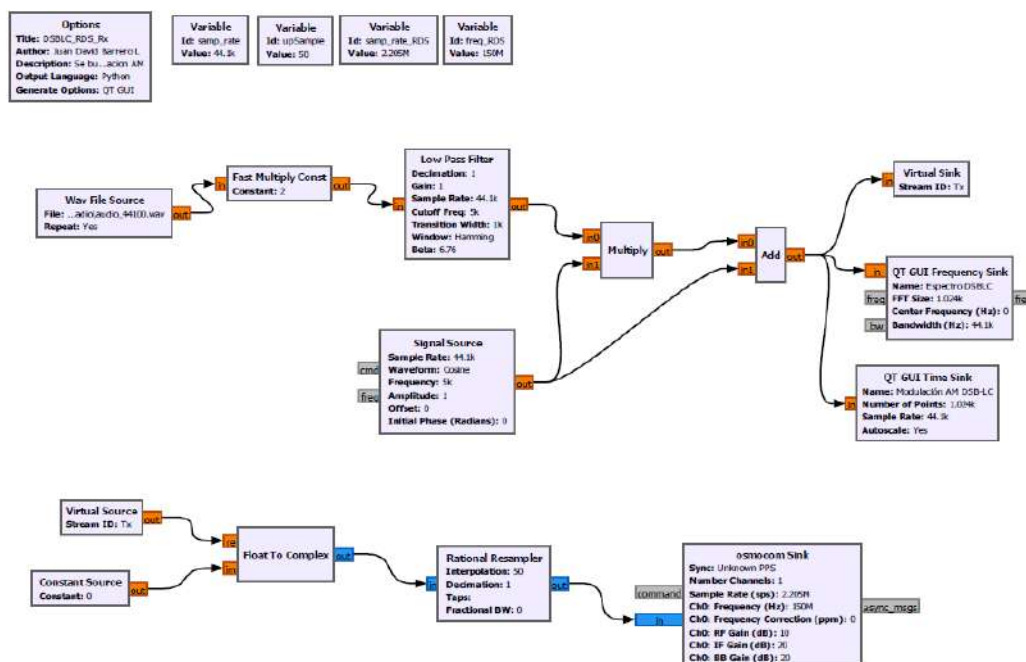


Figura 2.34 Transmisión RDS DSB\_LB

Para poder recibir la señal se utiliza los siguientes bloques ya que se está mandando la señal portadora la demodulación de esta se hace de manera más sencilla.

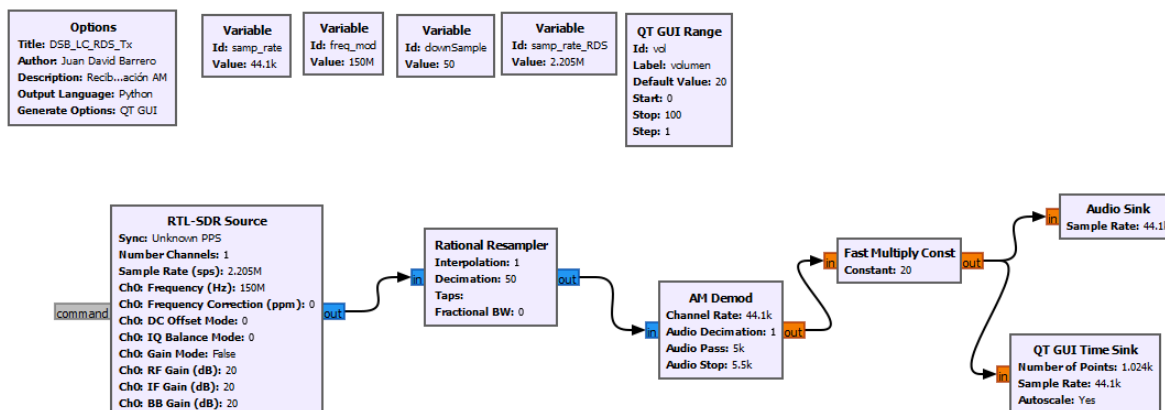


Figura 2.35 Receptor DSB-LC

Como se puede observar en la figura 2.35 la información llega de manera correcta y la forma de detectarla es más sencilla gracias a que se está enviando la información de la portadora en la señal modulada.

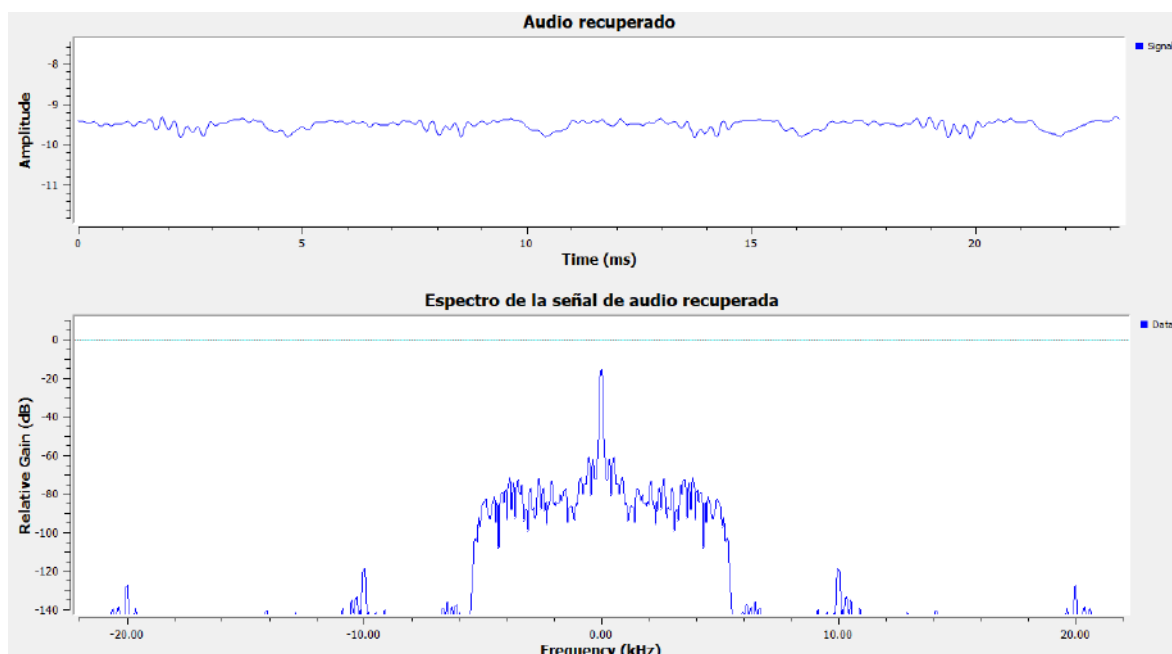


Figura 2.36 Resultado de la recepción

De esta forma se puede enviar y recibir información utilizando la modulación AM y los RDS los códigos de esta sección se pueden encontrar en el siguiente ([https://github.com/JuanDavidBarrero/GNURadio/tree/main/AM\\_modulation](https://github.com/JuanDavidBarrero/GNURadio/tree/main/AM_modulation)).

## 2.3 Modulación por cuadratura QAM

Como se ha visto hasta el momento solo se ha enviado una señal a la vez, pero esto se puede lograr “Aprovechando la ortogonalidad de senos y cosenos es posible transmitir y recibir dos señales de manera simultánea en la misma frecuencia portadora” (Introducción a los sistemas de comunicación, 2000), en la siguiente figura se muestra como poder ensamblar este tipo de sistema.

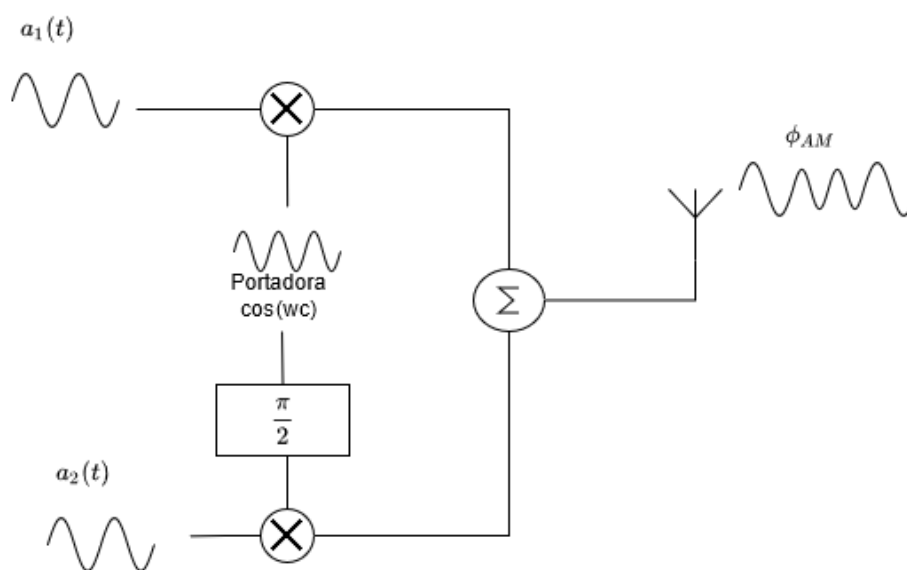


Figura 2.37 Sistema QAM

El diagrama de bloques se puede expresar matemáticamente en la siguiente ecuación

$$\phi_{AM} = a_1(t) \cos(wct) + a_2 \sin(wct)$$

Ecuación 12 Modulación QAM

Para poder analizar este sistema lo más conveniente es estudiarlo en el dominio de la frecuencia por lo cual la ecuación 10 queda de la siguiente manera

$$\phi_{AM} = A_1(W) * \frac{1}{2}(\delta(W - Wc) + \delta(W + Wc)) + A_2(W) * \frac{1}{2j}(\delta(W - Wc) + \delta(W + Wc))$$

Ecuación 13 QAM en el dominio de la frecuencia

Como se puede ver en la ecuación 11 el seno es ortogonal al coseno por esta razón es que en la misma frecuencia pueden estar dos señales al mismo tiempo, gráficamente se puede ver de la siguiente forma.

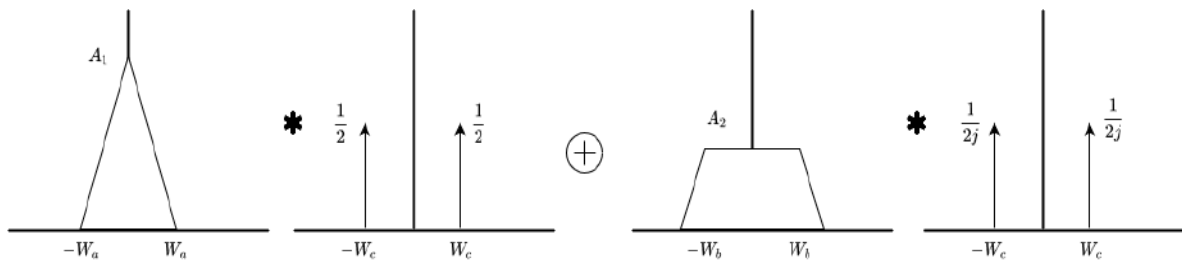


Figura 2.38 Ecuación en el dominio de la frecuencia

Al realizar la operación se obtiene el siguiente resultado

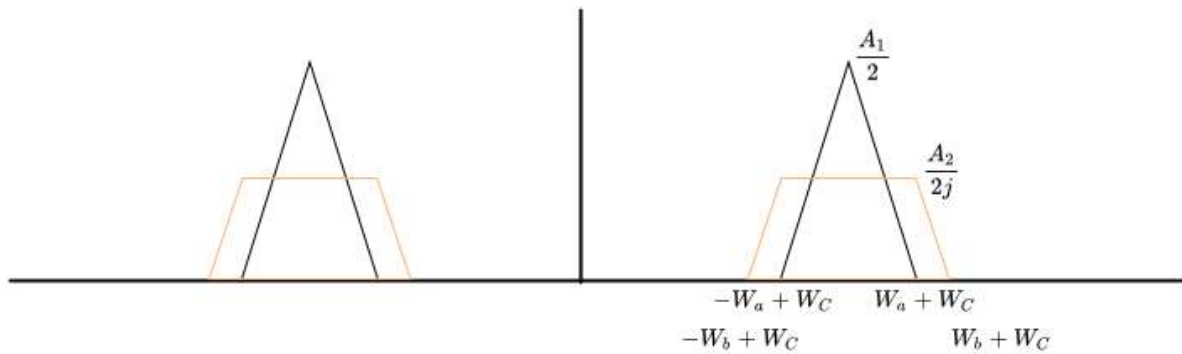


Figura 2.39 Resultado al resolver la ecuación

Como las señales son ortogonales no se interfieren entre sí, por lo que al trasmitirlas cada una estará desplazada 90 grados de la otra por lo que recuperarlas es posible con el sistema correcto.

El sistema para recuperar la señal se muestra a continuación

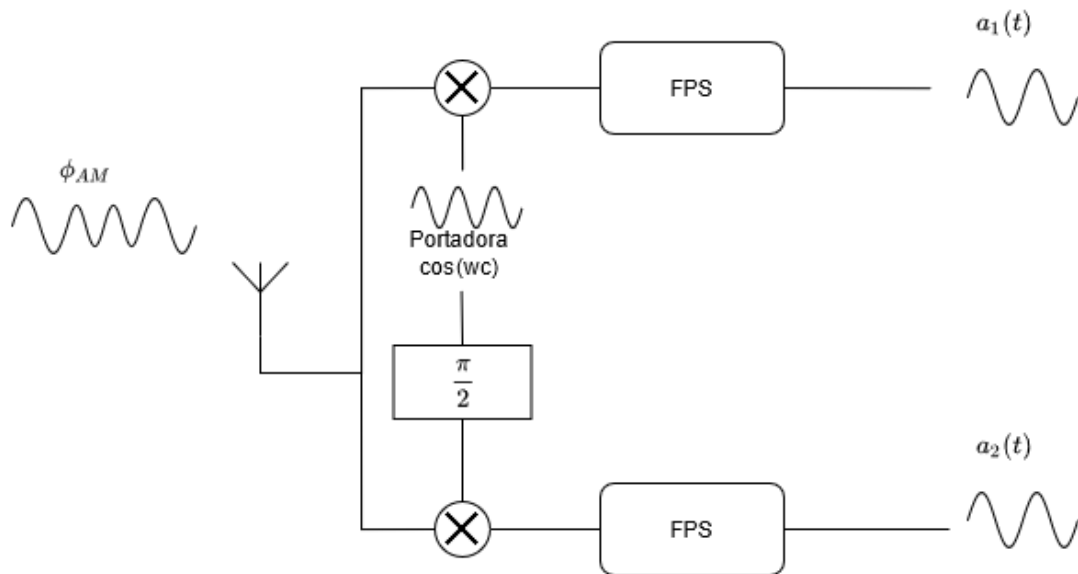


Figura 2.40 Sistema receptor QAM

El diagrama de bloques para poder recibir mostrado en la figura 2.40 muestra como poder recuperar las señales enviadas, el resultado para obtener la información es el mismo visto la figura 73 y 74 de doble banda lateral portadora suprimida.

### 2.3.1 Transmisión de señales QAM GNU radio

Para realizar la simulación se utiliza el siguiente diagrama de bloques, para poder hacer la recepción más sencilla se utiliza la forma de modulación doble banda lateral gran portadora.

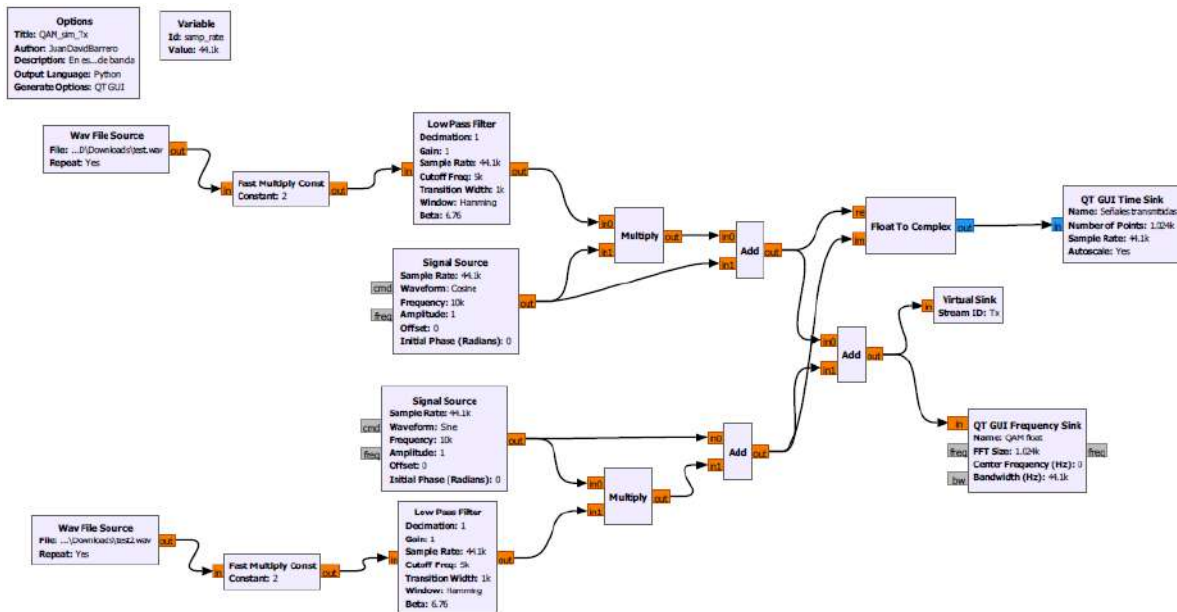


Figura 2.41 QAM transmisión

Para observar como la señales no interfieren la una con la otra se utiliza el bloque “time sink” pero esta vez el tipo de datos que va a recibir son tipo flotante, por eso se colocó el bloque “Float to complex”. El cual se puede ver continuación.



Figura 2.42 Convierte señales reales a complejas

Los resultados se pueden ver a continuación:

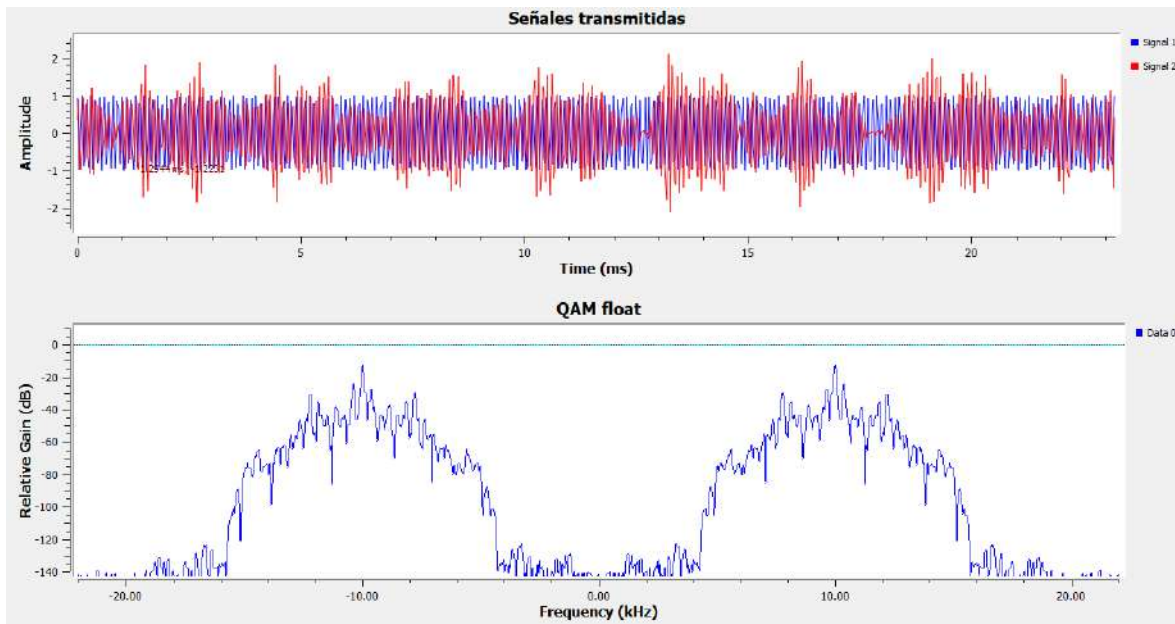


Figura 2.43 Señales transmitidas

En la figura 2.43 se observa solo un ancho de banda de una señal, ya que la otra es ortogonal no se puede observar, si se quiere ver ambas hay que colocar dos “Frequency sink” para cada una de las señales. El código de descarga se puede encontrar en el siguiente ([https://github.com/JuanDavidBarrero/GNURadio/tree/main/AM\\_modulation](https://github.com/JuanDavidBarrero/GNURadio/tree/main/AM_modulation) )

## 2.3.2 Recepción de señal QAM GNU radio

El sistema de recepción puede ser el mismo mostrado en la figura 2.40, pero con fines educativos se realizó el diagrama de bloques mostrado en la figura 2.44 ya que este tiene como finalidad separar las dos señales, y de esta manera se podrán reproducir para verificar que el sistema funcione correctamente.

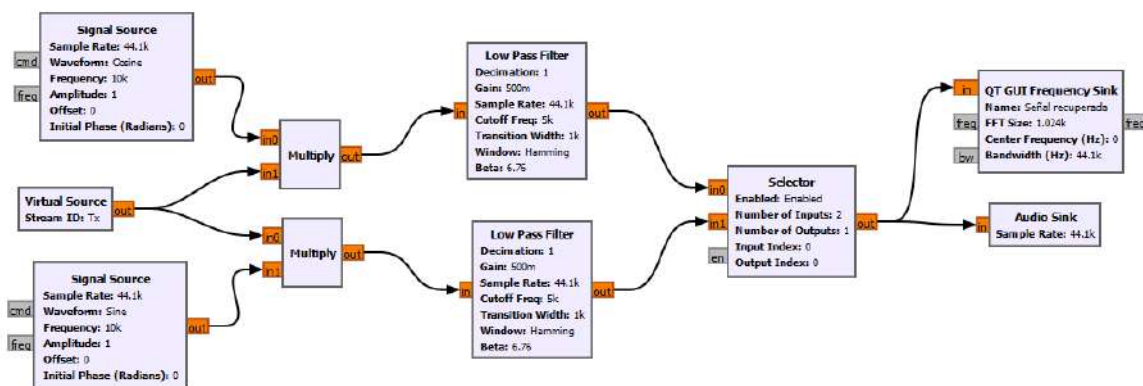


Figura 2.44 Recepción sistema QAM

En la figura 2.44 se muestra solo el diagrama de bloques para realizar la recepción, pero para que esto funcione, debe haber una señal para transmitir, todo el diagrama de bloques se muestra en la siguiente ilustración.

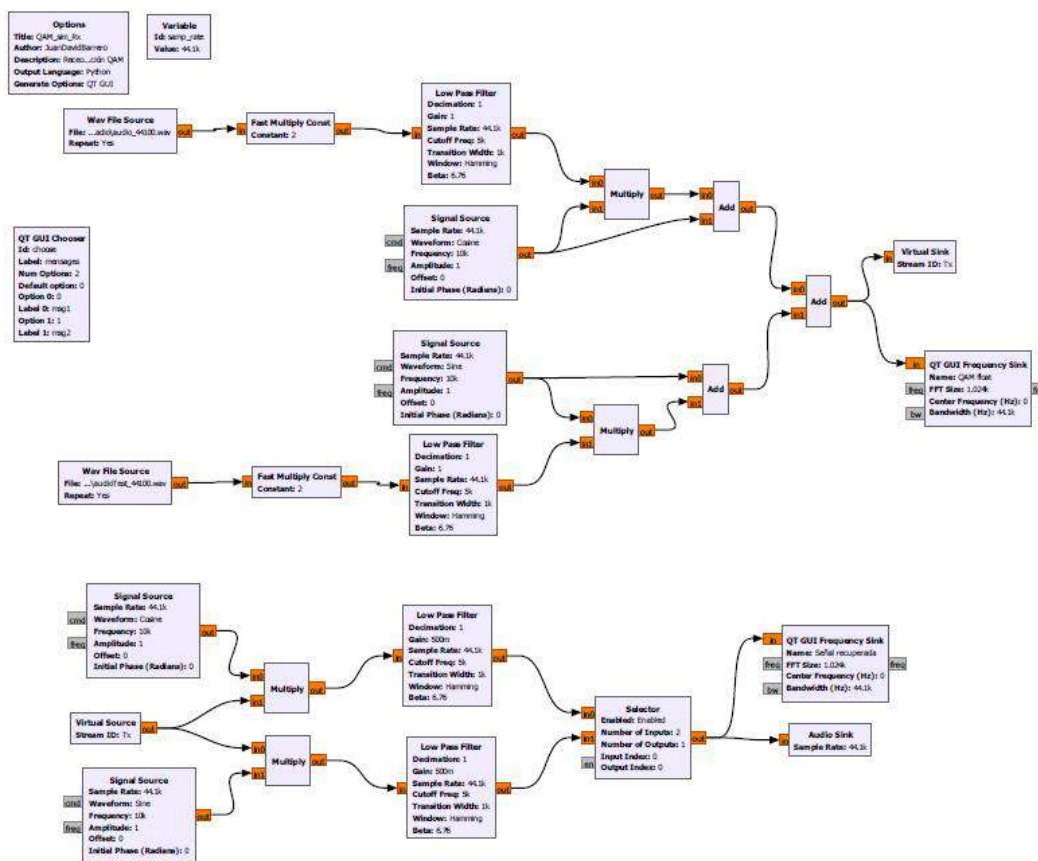


Figura 2.45 Sistema completo Tx y Rx QAM

Los resultados se muestran a continuación, también está habilitado el audio para poder oír la señal y de esta forma contrastar que la información se ha recuperado correctamente.

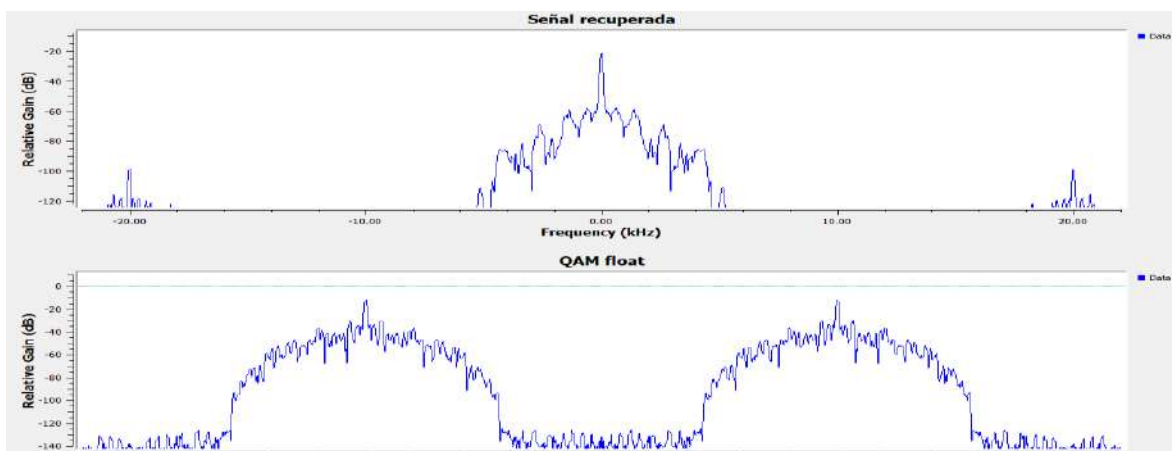


Figura 2.46 Señal recuperada QAM

### 2.3.3 Tx y Rx utilizando RDS QAM

En el siguiente diagrama se muestra cómo realizar el proceso de transmisión de dos señales distintas utilizando la ortogonalidad de las señales seno y coseno, acá se puede ver como se realiza la transformación de datos reales a complejos para poder utilizar el RDS

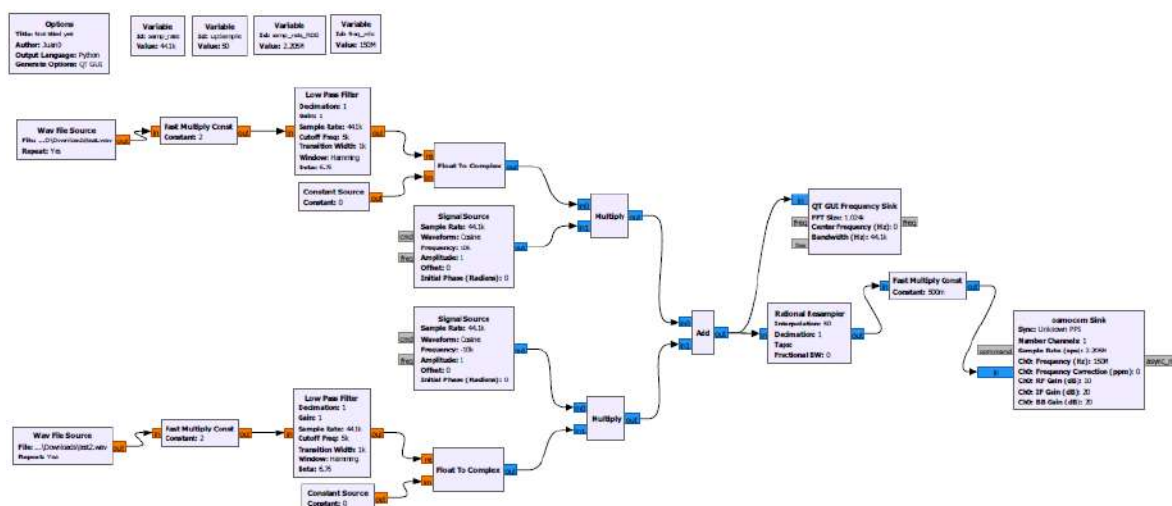


Figura 2.47 Transmisión QAM con RDS

Antes de realizar el diagrama de bloques del receptor se utilizará el software de SDR# para verificar que ambas señales se estén transmitiendo correctamente, para esto hay seguir los mismos pasos vistos en la figura 69. Para descargar este diagrama de bloques ir a la siguiente ([https://github.com/JuanDavidBarrero/GNURadio/tree/main/AM\\_modulation](https://github.com/JuanDavidBarrero/GNURadio/tree/main/AM_modulation))

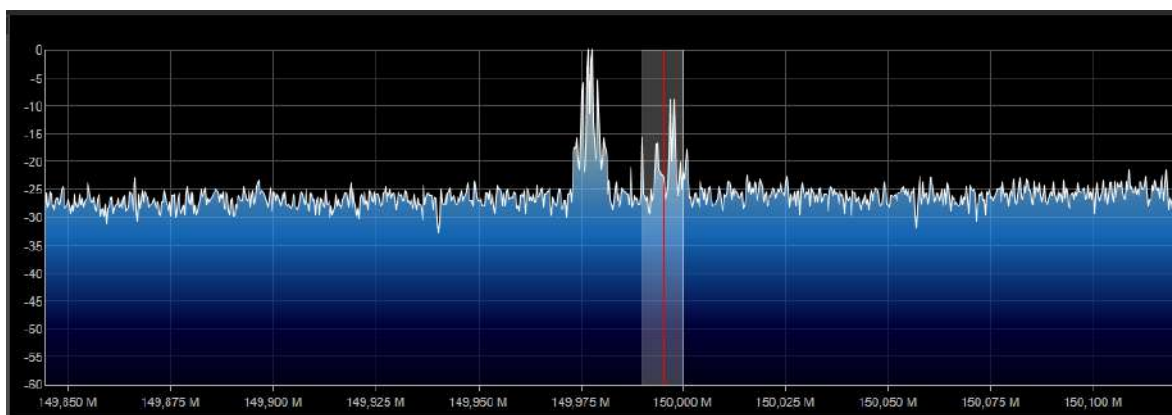


Figura 2.48 Señal recibida utilizando SDR#

Como se observa las señales enviadas se reciben y se escuchan correctamente, por lo que ahora se puede pasar el diagrama de bloques de la recepción, mostrando que el sistema funciona.

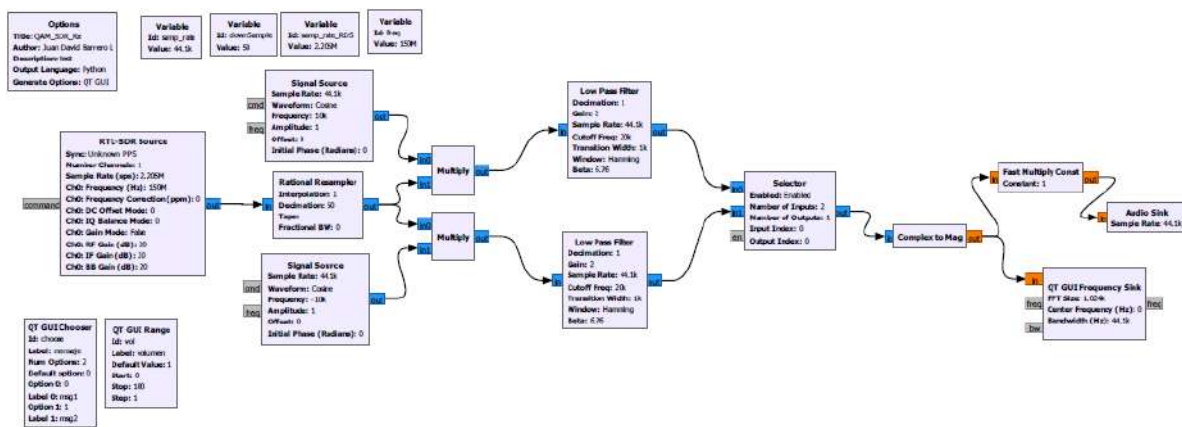


Figura 2.49 Receptor de la señal QAM con RDS

Si todo está ajustado correctamente se deben oír los mensajes enviados por los parlantes del computador. Además, se añadió una slider en la parte superior para poder cuadrar el volumen de la señal que se está recibiendo.

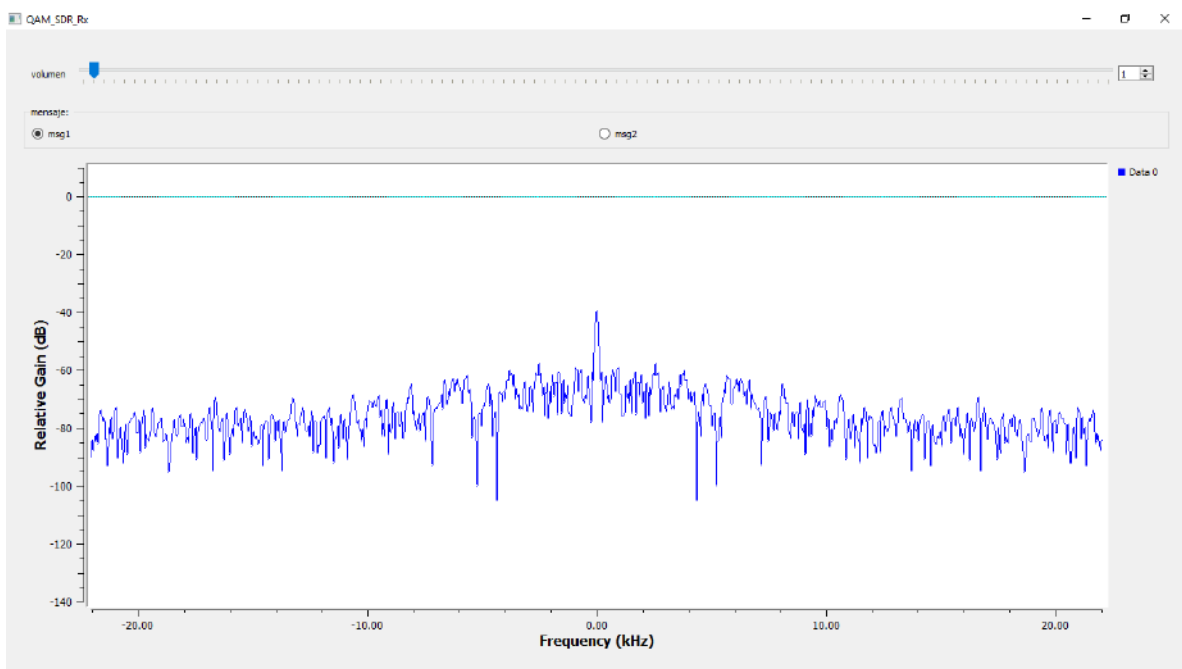


Figura 2.50 Señal de audio recibida

Con esto tipos de modulación se puede ver que el ruido juega un papel crítico a hora de enviar información, ya que como la amplitud de la señal va aumentando el ruido también lo hace y esto supone un problema para hacer de este un sistema de

transmisión 100% confiable. Por otro lado, tiene la ventaja de poder recorrer grandes distancias sin ser perturbado por los distintos elementos del ambiente.

También es importante agregar que para un sistema de modulación Am comercial las antenas son grandes, por esta razón durante toda la sección la señal que se quería enviar fue puesta en una portadora de 150 MHz (Este valor puede más grande o pequeño depende del usuario), ya que esta requiere una antena de poco tamaño.

### 3. Modulación FM utilizando RDS

FM significa frecuencia modulada; es una técnica que permite transmitir información a través de una onda portadora, variando su frecuencia. Al hacer esto se elimina el problema de ruido ya que este no se suma a la señal modulada, haciendo que el proceso de removerla sea más sencillo.

En la frecuencia modulada comercial, el ancho de banda comercial es de 20 MHz y este va desde 88 a 108 MHz y tiene un ancho de banda de cada emisora de 200 KHz, esto se debe a que el audio (señal moduladora) tiene un ancho de banda de 15Khz y la desviación de frecuencia utilizada comercialmente es de 75 KHz, por lo que utilizando la ecuación de Carsson (Ecuación 14) se obtiene que.

$$\Delta B = 2(\Delta f + f_m)$$

*Ecuación 14 Ancho de banda FM*

$$2(75 + 15) = 180 [Khz]$$

*Ecuación 15 Ancho de banda FM comercial*

A este resultado se le debe agregar 10 [Khz] a cada lado para dar una banda de seguridad de esta forma se obtiene el ancho de banda de un FM comercial.

Por otro lado, FM tiene un alcance más limitado que el AM ya que “las señales de frecuencia modulada tienen una longitud de onda menor al de la amplitud modulada. Esto se debe a que la frecuencia modulada se transmite entre 88 y 108 MHz, es decir, en frecuencias muy altas, cuyas ondas pueden medir entre un metro (300 MHz) y diez metros (30 MHz). Este tipo de ondas, además, tienen longitudes considerablemente pequeñas, de modo que se desplazan en línea recta y se atenúan rápidamente” (Significados, 2015)

En la siguiente figura se puede observar el espacio del espectro electromagnético donde se ve el ancho de banda de la modulación AM y FM comercial y la cantidad de bandas que tiene cada uno

Figura 3.1 Tomada de (Nave, 2010)

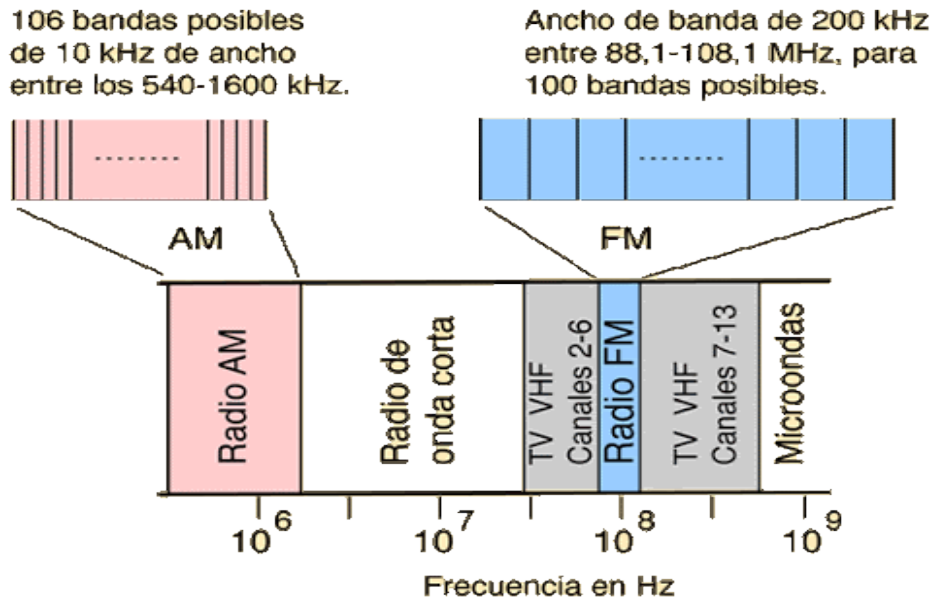


Figura 3.1 Espectro del ancho de banda

### 3.1 VCO en GNU radio

El corazón de la modulación por frecuencia se logra con la ayuda de un VCO (Voltage Controller Oscillator) Oscilador controlado por voltaje, este crea una señal senoidal la cual va variando su frecuencia dependiendo de la amplitud y de la frecuencia de la señal de entrada.

En los sistemas analógicos “un oscilador consiste esencialmente de dos componentes: una parte activa la cual actúa como amplificador, y una red de retroalimentación que provee retroalimentación positiva al sistema. La red de retroalimentación generalmente contiene un elemento de reactancia variable (osciladores LC) para controlar la frecuencia” (Ascencio, 2010)

Pero como se está trabajando con la tecnología de RDS estos VCO pueden ser programados, por lo que ahora son más sencillos de aplicar a este tipo de modulación, por suerte GNU radio cuenta con este bloque ya instalado por defecto y se puede ver en la siguiente imagen.

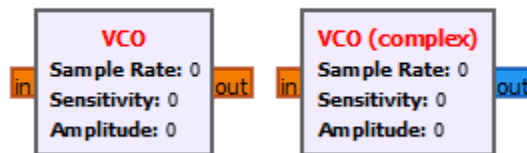


Figura 3.2 Bloques VCO GNU radio

Como se observa en la figura 3.2 existen dos tipos de bloques de VCO estos tienen las mismas características, salvo una pequeña diferencia la cual es que uno de estos bloques a su salida entrega una señal compleja, la cual será muy útil para poder acoplar la señal modulada al RDS al momento de ser transmitida.

Para poder configurar estos bloques toca ajustar los siguientes parámetros



Figura 3.3 Parámetros VCO

**Sample rate:** es la frecuencia de muestreo que espera el bloque de la señal de entrada si estos no coinciden la modulación se va a descuadrar y no se podrá sintonizar correctamente.

**Sensitivity:** A este parámetro se le puede conocer como la desviación de frecuencia y debe ser expresado en radianes por segundo, esta muestra el cambio de frecuencia dependiendo de la amplitud de la señal de entrada.

**Amplitude:** Es la amplitud de la señal de salida del oscilador.

Estos parámetros son los mismos para ambos bloques vistos en la figura 3.2, se debe tener en cuenta que VCO complex en su salida brinda una señal real y otra imaginaria, lo cual es perfecto ya que es el tipo de señal que espera el bloque de RDS para poder ser transmitida.

En el siguiente diagrama de bloques se muestra como la frecuencia de la señal va variando, dependiendo del valor de una señal constante, con fin de entender el comportamiento de bloque VCO.

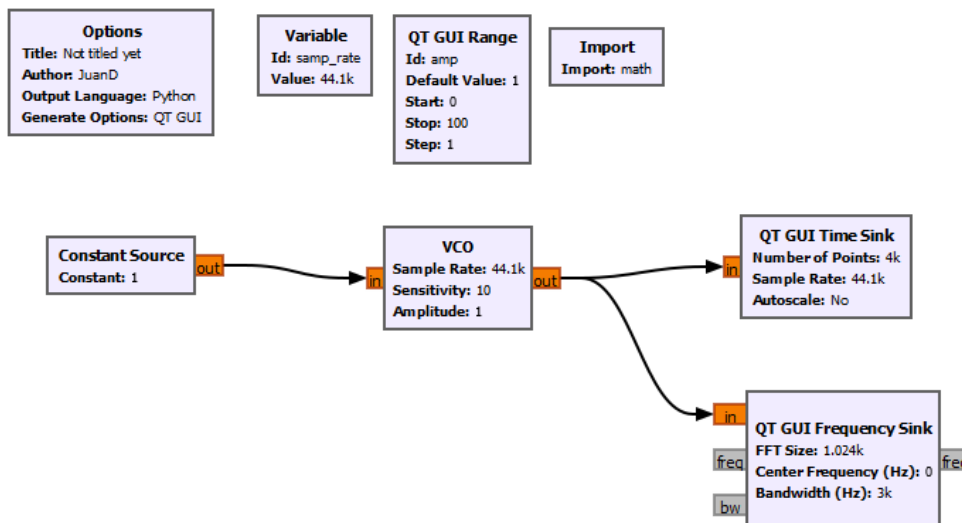
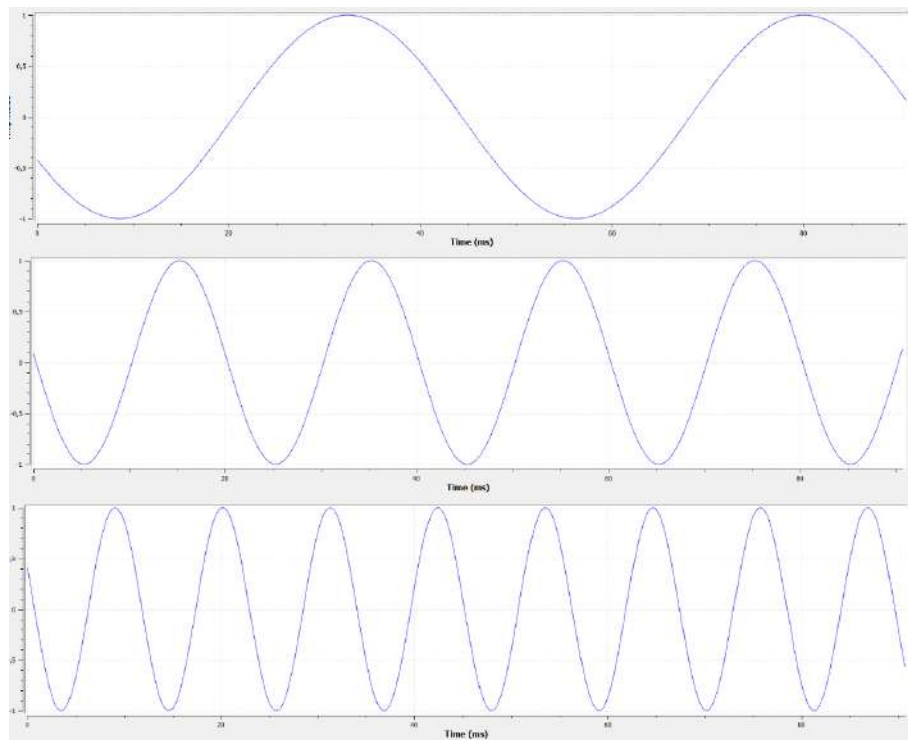


Figura 3.4 Prueba del bloque VCO

En la siguiente figura 3.5 se puede ver como la frecuencia de la señal va aumentando a medida que la constante puesta a la entrada del bloque VCO sube, de esta forma se puede hacer una modulación en frecuencia. Si ahora la entrada se remplace con una señal que tenga un frecuencia fija, como lo podría ser una señal de audio o un tono.



*Figura 3.5 Resultado de variar la constante de entrada al VCO*

Para poder analizar el comportamiento de una modulación en frecuencia es necesario utilizar una señal con una frecuencia fija en la entrada de bloque VCO, también hay que adicionar un offset ya que la el VCO no realiza ninguna operación con valores negativos y por último añadir la posibilidad de variar la amplitud para ajustar la señal de la mejor manera posible.

Para realizar esto se presenta el siguiente diagrama de bloques, el cual cumple con todas las condiciones mencionadas anteriormente, además de eso cuenta con un osciloscopio para mostrar la señal modulada en frecuencia y el espectro de esta.

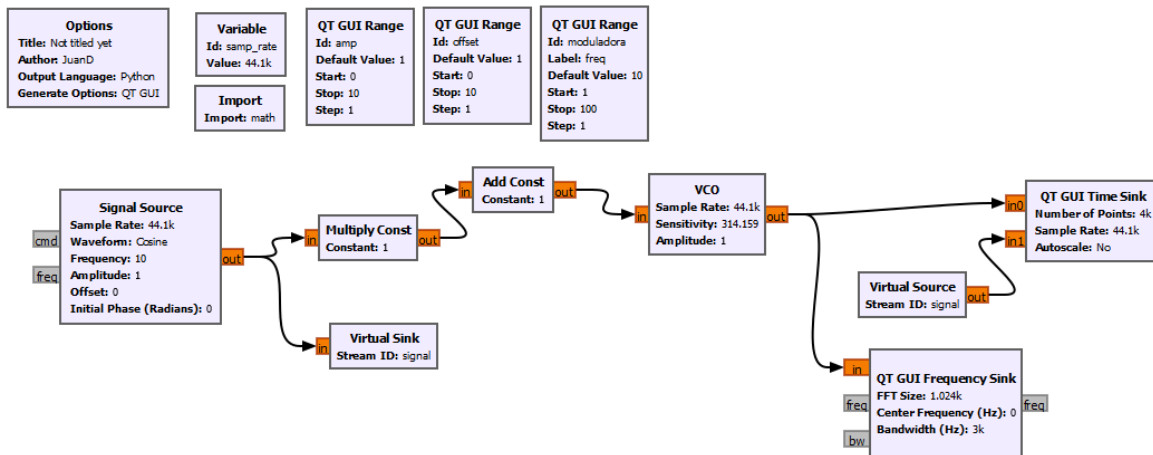


Figura 3.6 Modulación FM de un tono

En la siguiente figura se muestra el resultado obtenido al ejecutar el código mostrado en la parte de arriba, en esta se observa como la señal moduladora (roja) controla la frecuencia de la señal modulada (azul) a medida que la su amplitud cambia, en la parte de abajo se observa el espectro de la señal modulada, en la siguiente sección se entrara más en detalle cómo se interpreta los datos vistos.

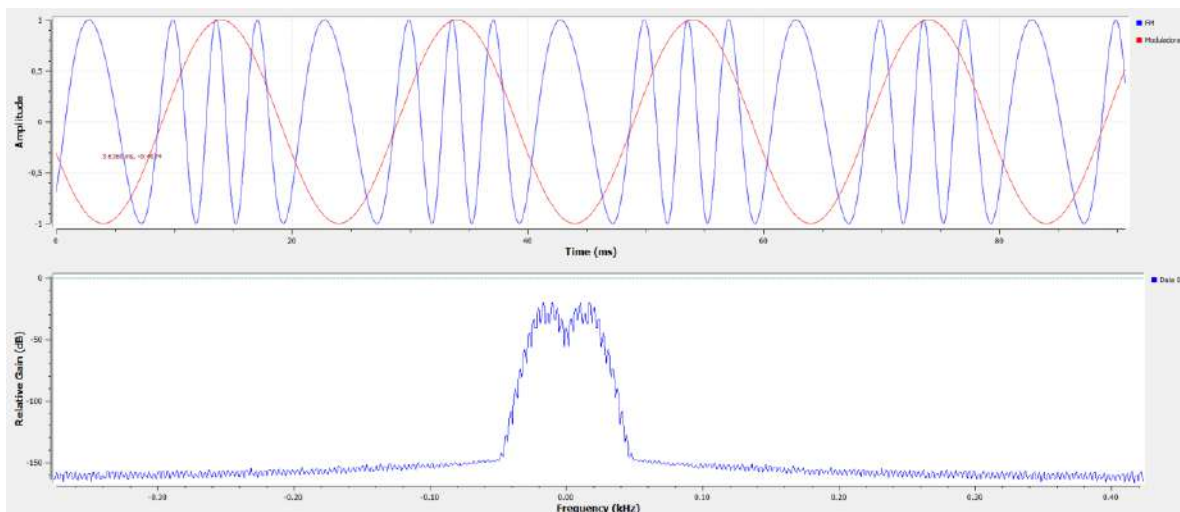


Figura 3.7 Modulación FM

Con este programa se pueden sacar las siguientes conclusiones

- Al aumentar la amplitud de la señal moduladora el ancho de banda aumenta.
- Al hacer variar el valor dc (offset) la señal se desplaza por el eje x
- Al aumentar el valor de la frecuencia moduladora la aparecen más picos en el espectro de la señal modulada.

Esto se debe a que ahora más de una frecuencia está apareciendo y ocurre tan rápido que parece como si fuera un ancho de banda constante tal y como se puede ver en la siguiente figura.

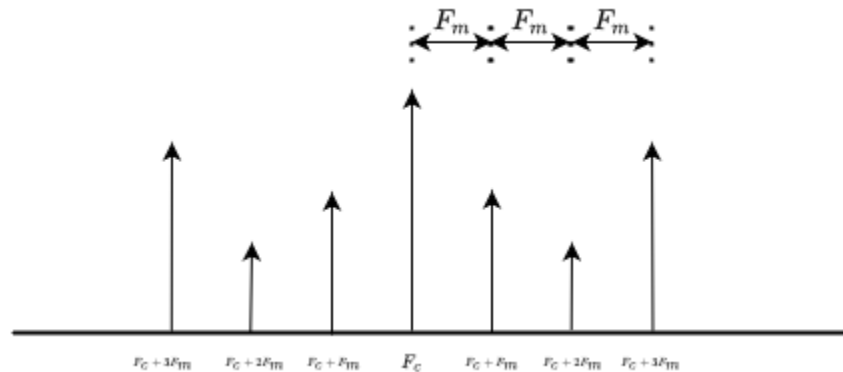


Figura 3.8 Representación espectro de modulación FM

La forma que esta toma esta dada por los coeficientes de Bessel, los cuales se verán un poco más adelante en esta misma sección cuando se hable de modulación FM banda ancha.

### 3.1.1 Tx y Rx con modulación de frecuencia simulación

Con el conocimiento adquirido de cómo utilizar un VCO, es momento de hacer la primera transmisión y recepción de información utilizando el método de modulación FM. Para esto se debe utilizar GNU radio y armar una estructura similar a la vista anteriormente para enviar información

Para lograr el objetivo se implementó el siguiente diagrama de bloques, con una nueva herramienta que al igual que VCO transforma la señal de entrada y una señal senoidal que varía la frecuencia, este bloque se llama “Frequency mod”, y se puede ver a continuación.

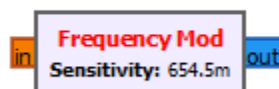


Figura 3.9 VCO alterno GNU radio

Este bloque a diferencia del VCO visto anteriormente, cuenta con un solo parámetro para definir y este es el coeficiente de sensibilidad. Y según los desarrolladores de este bloque este se puede calcular de la siguiente forma.

$$K = 2\pi \frac{\Delta F}{F_s}$$

Ecuación 16 Coeficiente de sensibilidad

Una vez colocada el coeficiente con él se esté cómodo ya se puede realizar la transmisión de la información tal y como se ve en el siguiente diagrama de bloques.

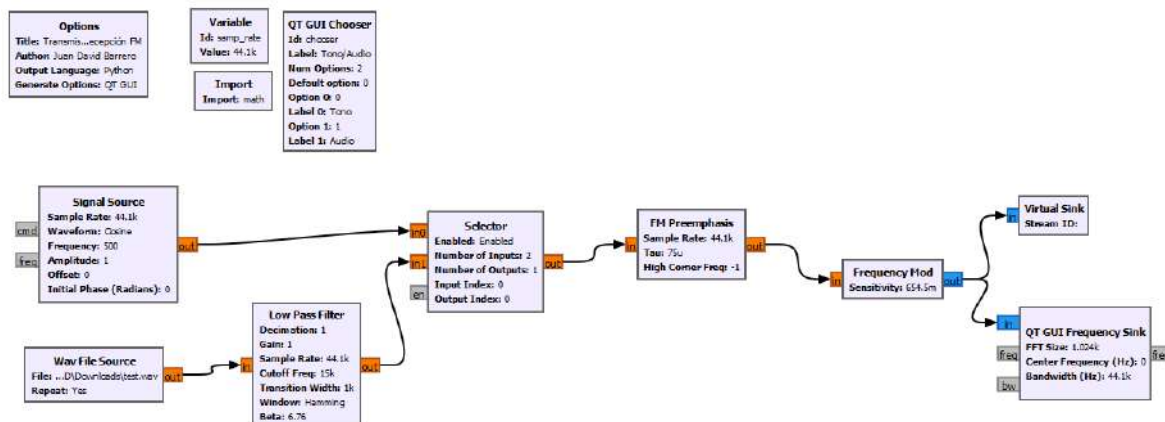


Figura 3.10 Transmisión con modulación FM

Para recibir la información también es necesario utilizar el bloque “FM demod” el cual cuenta con la lógica de recobrar la información que ha sido manipulada durante el proceso de modulación, este bloque tiene ciertas características que se verán a continuación.

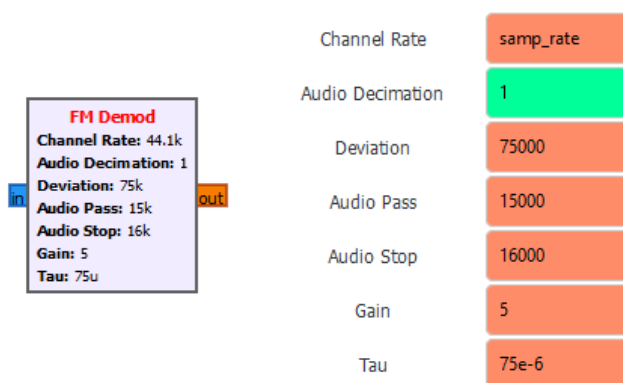


Figura 3.11 FM demod GNU radio

**Audio Pass:** es el parámetro de la frecuencia de corte de un filtro pasa bajo que evitara que pase frecuencias superiores a este

**Audio Stop:** Es la frecuencia de corte en la cual frecuencia más altas de este parámetro no pasan.

**Tau:** Es la desviación de la frecuencia esta debe coincidir con los parámetros establecidos durante la fase de transmisión.

Con lo parámetros bien establecidos se puede hacer el siguiente diagrama para poder recibir los datos, este cuenta con el bloque audio sink, para poder escuchar la información enviada.

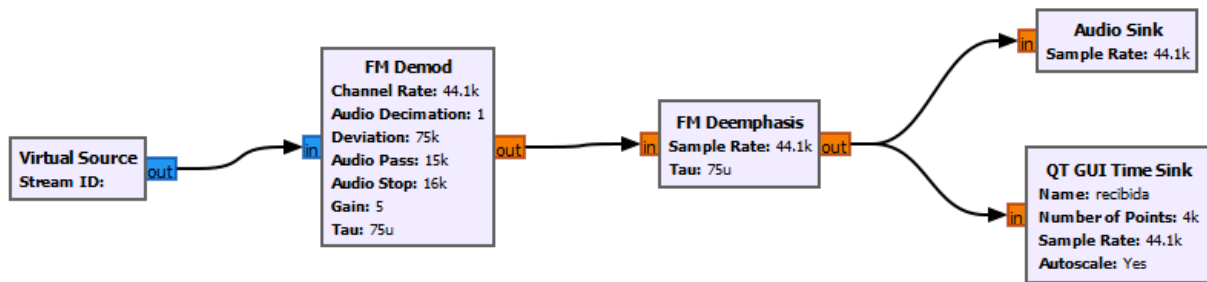


Figura 3.12 Recepción de datos modulación FM

A continuación, se muestra los resultados del diagrama de bloques funcionando, en este se logra observar como la señal se reconstruyo exitosamente, además de que se puede oír el audio desde los parlantes del equipo sin ningún problema.

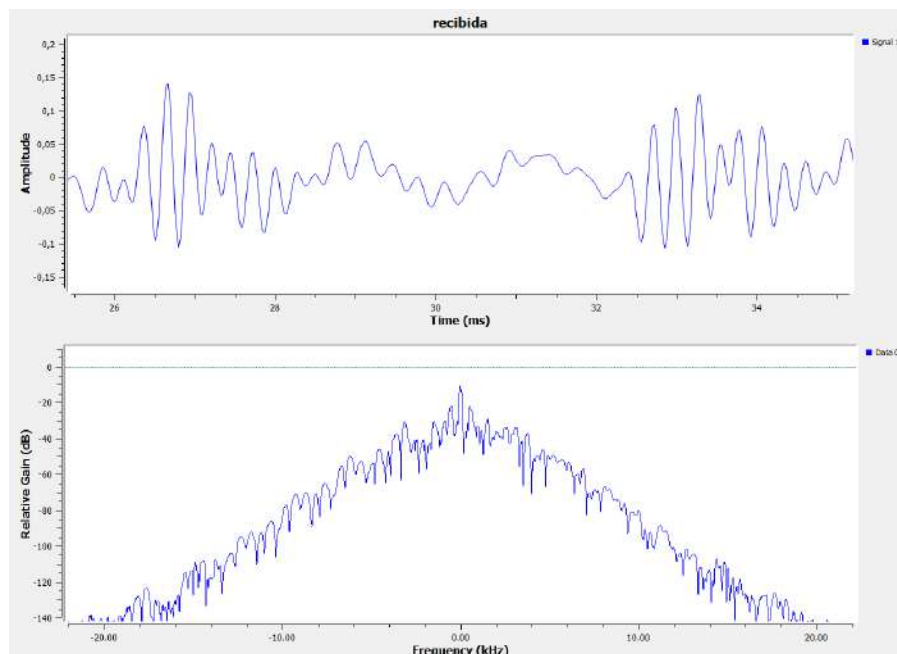


Figura 3.13 Obtención de la información

## 3.2 Modulación de frecuencia banda angosta

En la modulación de frecuencia banda angosta se busca la mejor calidad y se puede expresar en la siguiente ecuación:

$$\varphi_{FM} = A \cos(\omega_c t + \beta \sin(\omega_m t))$$

*Ecuación 17 (FM banda angosta)*

Al usar una identidad trigonométrica se puede llegar a la siguiente ecuación

$$\varphi_{FM} = A \cos(W_c t) \cos(\beta \text{Sen}(W_m t)) - A \text{Sen}(W_c t) \text{Sen}(\beta \text{Sen}(W_m t))$$

*Ecuación 18 Extensión de la ecuación 14*

En el momento sigue siendo una ecuación complicada de analizar por lo que es necesario utilizar las series de Taylor para aproximar la ecuación si  $\beta$  es cercano a 0

$$\begin{aligned} \cos(\beta \text{Sen}(W_m t)) &\approx 1 \\ \text{Sen}(\beta \text{Sen}(W_m t)) &\approx \beta \text{Sen}(W_m t) \end{aligned}$$

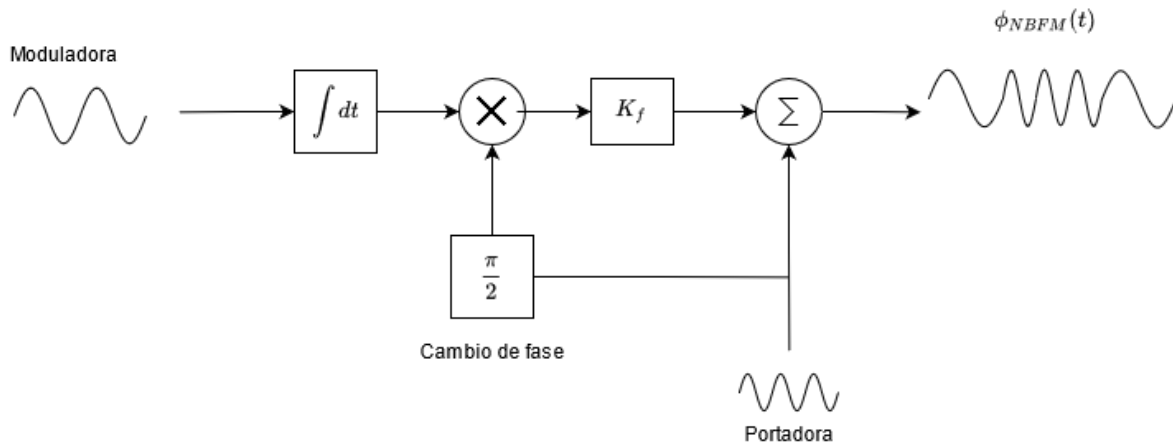
*Ecuación 19 Aproximación de Taylor*

Gracias a esto la ecuación de banda angosta queda de la siguiente manera.

$$\phi_{NBFM} = A \cos(W_c t) - A \beta \text{Sen}(W_c t) \text{Sen}(W_m t)$$

*Ecuación 20 FM ancho de banda angosto*

Esta ecuación se puede representar en el siguiente diagrama de bloques, con esto cuando se pase a la simulación en GNU radio se puede hacer de manera rápida y eficiente.



*Figura 3.142 NBFM diagrama de bloques*

### 3.2.1 NBFM simulación GNU radio transmisión

GNU ofrece la alternativa de utilizar el bloque especializado que tiene integrado para poder transmitir y también recibir la señal moduladora, En esta sección se mostrara como configurar este bloque para transmitir un tono y una señal de audio, además de poder observar el comportamiento de esta en el osciloscopio integrado y los armónicos producidos como resultado.

Para poder utilizar este bloque es necesario buscarlo en la librería “modulator” ahí se encuentra y se llama “NBFM Transmit”, este se puede observar en la siguiente figura.

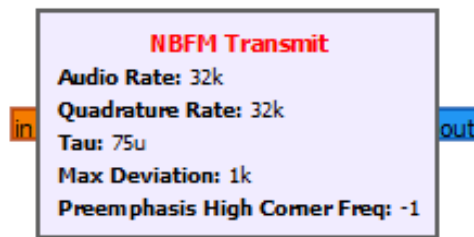


Figura 3.15 bloque NBFM

A este bloque se puede dejar configurado de la manera predeterminada ya que está diseñado para modular la señal en la banda de FM comercial, sin embargo, en esta sección se mostrará lo que se puede hacer la modificar tres de los cuatro parámetros principales

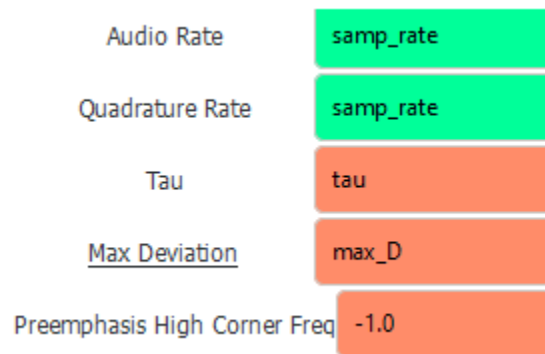


Figura 3.16 parámetros del bloque NBFM

**Audio Rate:** Es el encargado de decir cual es la frecuencia de muestreo de la señal de audio que se va a modular esta debe coincidir con la frecuencia del tono o la frecuencia a la que fue muestreada la señal de audio

**Quadrature Rate:** La tasa de cuadratura es la encargada de cuadrar la información de manera correcta, si se hizo algún tipo de diezmación o interpolación a la señal antes de entrar al bloque, es importante que este sea un valor entero y múltiplo de la tasa de muestreo.

**Tau:** esta cuadrada por defecto y está la variación de la desviación de frecuencia utilizada para la modulación por frecuencia comercial.

**Max Deviation:** Es la desviación máxima que va a tolerar para modular la señal, esta viene predeterminada al comienzo para la banda comercial, pero este valor se puede cambiar por uno más grande o pequeño.

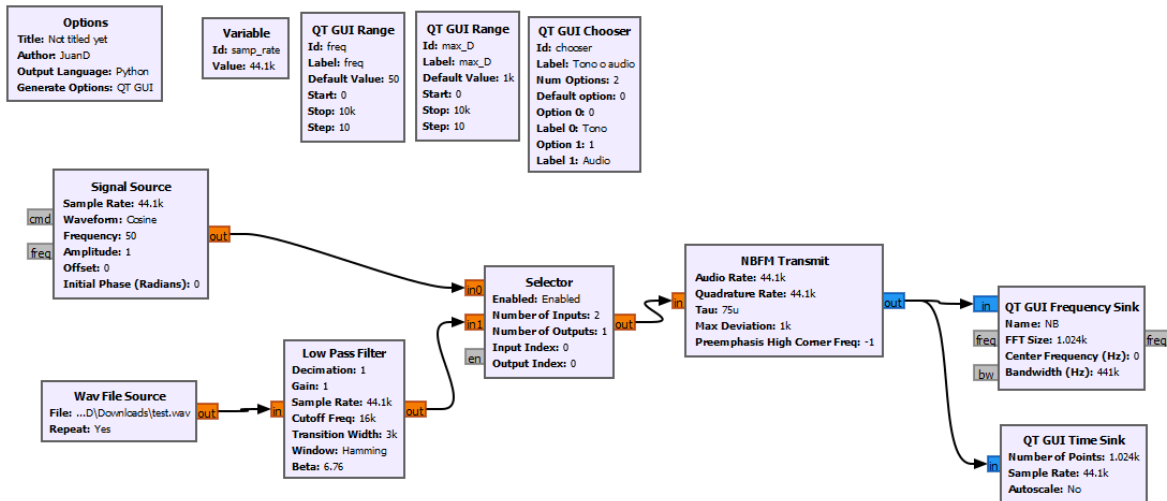


Figura 3.17 Transmisor NBFM

Como FM permite tener un ancho de banda más grande el filtro que se le coloca a la fuente de audio se agrando a una frecuencia de corte de 15[KHz] a esto se le conoce como High Fidelity en español alta fidelidad de audio, en la siguiente imagen se puede apreciar el resultado de utilizar este bloque.

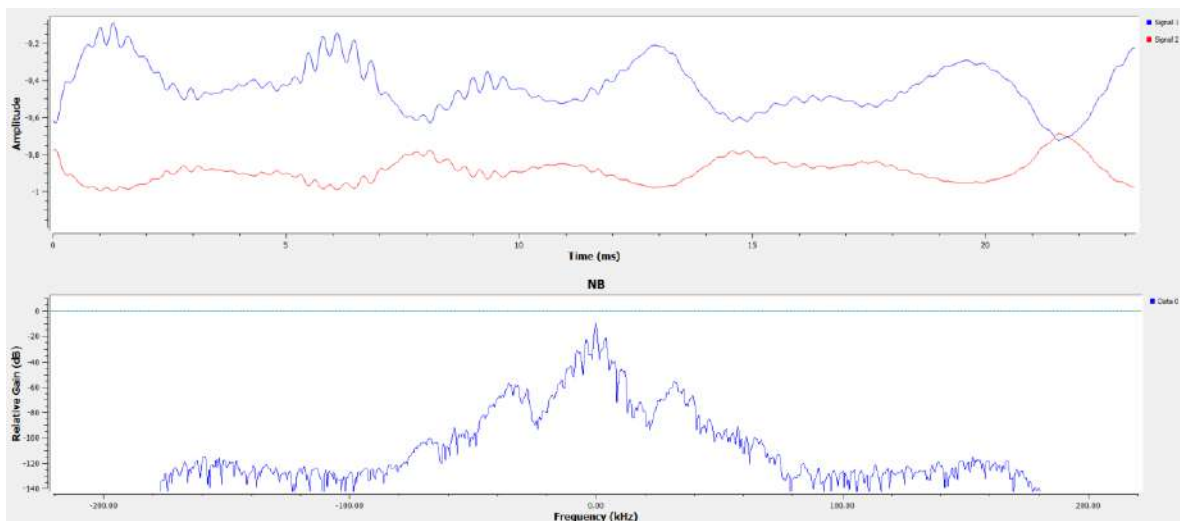
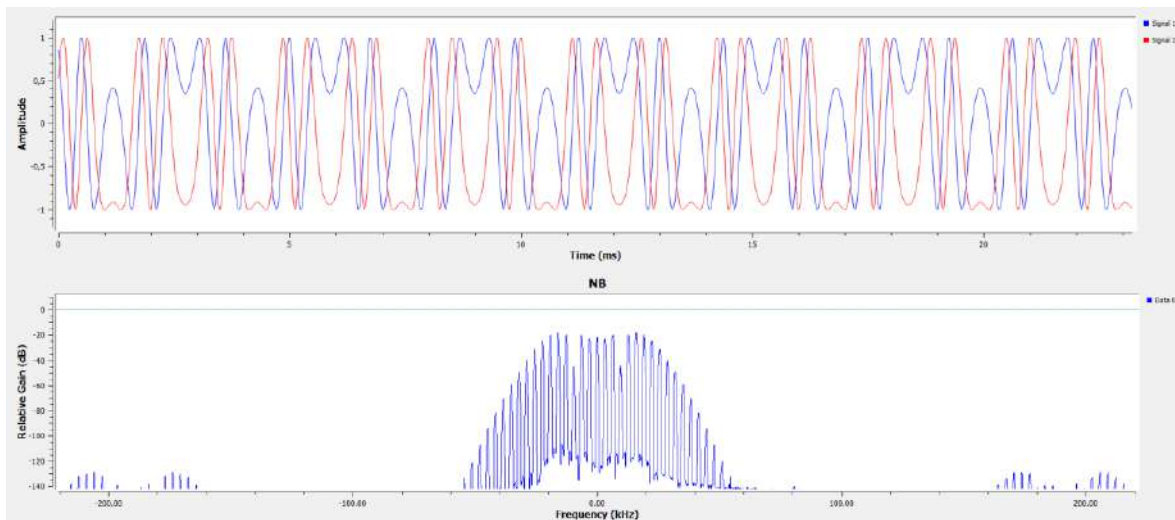


Figura 3.18 Modulación NBFM con señal de audio

Si se usa el tono se puede observar como si la portadora fuera saltando de un lugar a otro dando la impresión de que hay de una portadora a lo largo del ancho de banda establecido en el bloque.



*Figura 3.19 Modulación NBFM con un tono*

## 3.2.2 Transmisión NBFM utilizando RDS

En esta sección se utilizará el RDS conocido como HackRF-one para transmitir una señal de audio a una frecuencia determinada, y para poder percibir que la señal se ha transmitido correctamente se utilizara el software SDR # que se ha usado en lecciones anteriores para poder escuchar el mensaje.

Para esta transmisión la frecuencia a la que se a transmitir será en el rango comercial de FM, como el HackRF-one no tiene una potencia muy alta para transmitir esto no supone ninguna interferencia a las demás estaciones, la idea de hacer esto en este rango es para poder utilizar cualquier radio disponible incluso el teléfono para poder escuchar el mensaje que se quiera transmitir.

En la siguiente tabla se puede observar cuales son las emisoras que se encuentran en el área de Bucaramanga. Esta se tendrá en cuenta para escoger la frecuencia a la que se transmitirá el mensaje, y así evitar interfieran a la hora de escuchar (ya que estas tienen mucha potencia opacarían el mensaje)

| Frecuencia | Cadena                              | RDS | en vivo | Nombre                              |
|------------|-------------------------------------|-----|---------|-------------------------------------|
| 90,7       | Caracol Radio                       | si  | si      | W radio                             |
| 91,7       | Policía Nacional de Colombia        | no  | si      | Radio Policía Nacional              |
| 92,3       | Sistema de medio públicos           | si  | si      | Radio Nacional de Colombia          |
| 92,9       | Ejército Nacional                   | si  | si      | Colombia Estéreo                    |
| 93,4       | Comunitaria                         | no  | si      | La Brújula FM                       |
| 95,7       | Caracol Radio                       | si  | si      | Tropicana                           |
| 96,2       | Universidad Santo Tomas             |     | si      | Radio USTA                          |
| 96,9       | UIS                                 | no  | si      | UIS FM                              |
| 97,7       | ORO                                 | si  | si      | Olímpica Estéreo                    |
| 99,2       | Caracol Radio                       | si  | si      | Caracol Radio                       |
| 99,7       | RCN radio                           | si  | si      | La FM                               |
| 100,7      | Instituto de cultura de Bucaramanga | no  | no      | Instituto de cultura de Bucaramanga |
| 101,7      | UTS                                 | no  | si      | UTS radio                           |
| 102,5      | RCN radio                           | si  | si      | La Mega                             |
| 103,7      | RCN radio                           | no  | si      | El sol                              |
| 104,7      | Caracol Radio                       | si  | si      | Bésame                              |
| 106,7      | RCN radio                           | si  | si      | Radio Uno                           |
| 107,7      | UIS-UDES                            | si  | no      | La U radio                          |

Tabla 6 Emisoras FM de Bucaramanga

Al ver esta table se ve que puede enviar una señal en la frecuencia 105.7[MHz] y esta no interferirá con ninguna otra estación y de paso se podrá escuchar claramente el mensaje que se quiera transmitir.

Para poder realizar esta operación es necesario utilizar el siguiente diagrama de bloques creado en GNU radio. Al igual que se vio cuando se transmitía señales en AM hay que aumentar el número de muestras antes de enviarlas al bloque del RDS para que la calidad de audio sea aceptable.

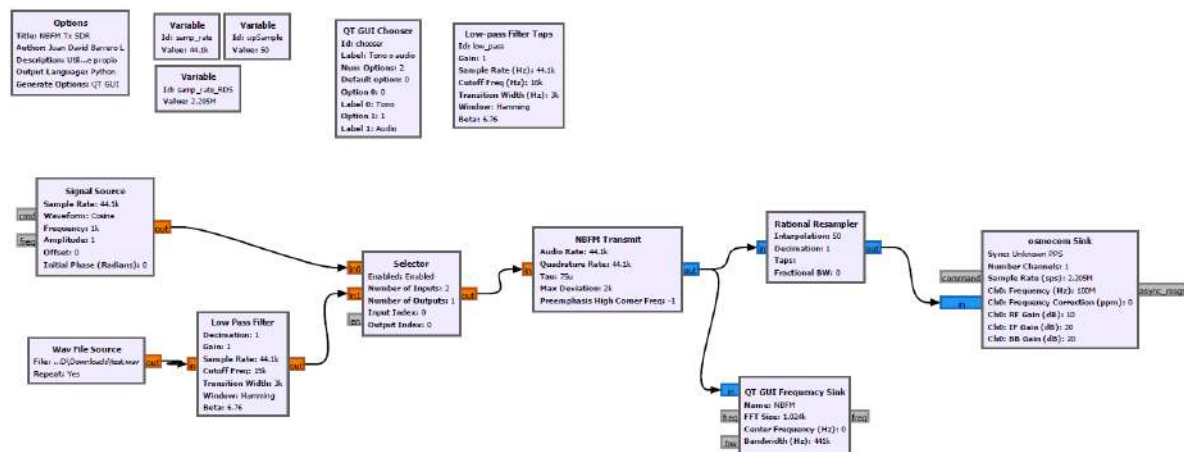


Figura 3.20 Transmisión NBFM con RDS

Al ejecutar el código se abrirá una ventana donde se mostrar el espectro de la señal que se está transmitiendo, así se sabrá cuando la señal ya esté siendo irradiada por el RDS.

Si no se ejecuta el código se observa que no hay ningún tipo de señal en la frecuencia seleccionada, solo se puede ver el espectro de las otras emisoras que están cercanas la frecuencia de transmisión.

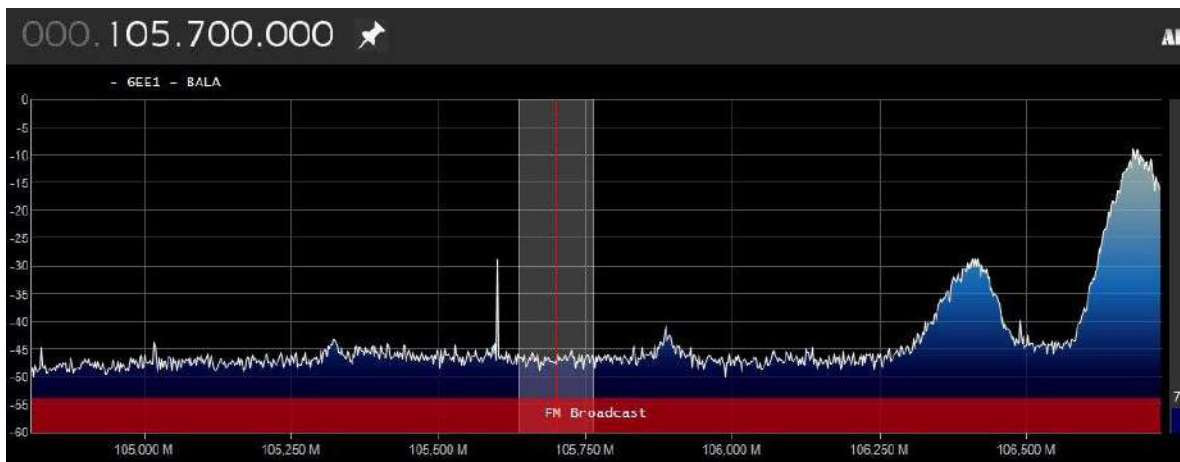


Figura 3.21 Banda sin datos

Es importante seleccionar la opción de NFM para que el software puede de modular la señal de maanera correcta y se puede escuchar por los parlantes de la maquina que se utilizando, de no hacer esto solo se podra escuchar el ruido producido por una mal decodificación de la informaición.

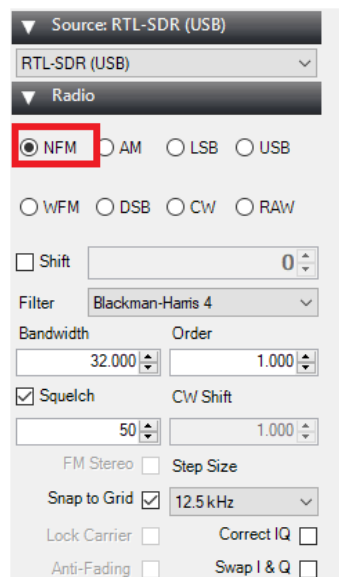


Figura 3.22 Cambiando el tipo de modulación

Una vez ejecutado el código y viendo que la la ventan con el espectro aparecio como se nombro anteriormete, se puede vovler a sintonizar la frecuencia y esta vez si se obvera el tipo de señal que esta enviando. Incluso se puede oir el mensaje transmitido claramente.

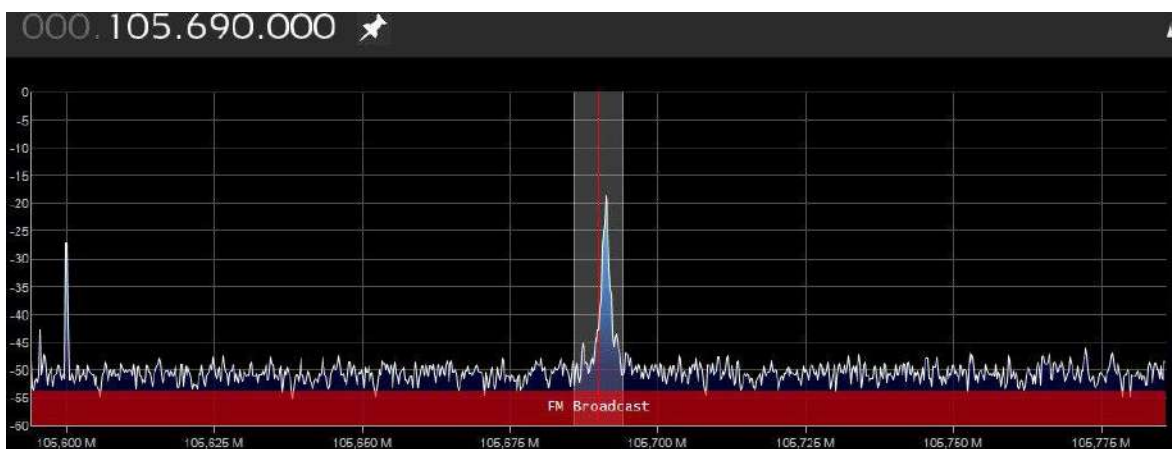


Figura 3.23 Datos recibido con NBFM

### 3.2.3 Receptor NBFM utilizando RDS

Antes de utilizar el RTL-SDR-Dongle para captar la señal transmitida es necesario primero captarla utilizando el software de GNU radio, esta manera se podrá entender que hace cada bloque se va a utilizar, una buena noticia es que el este programa en la librería “Modulators” ya tiene incluido el bloque de desmodulación de NBFM.

En la siguiente figura se puede observar el bloque nombrado anteriormente, este está programado de tal manera que puede de-modular la señal enviada con NBFM, este debe tener los mismos parámetros establecidos del transmisor para que todo se desarrolle de manera correcta.

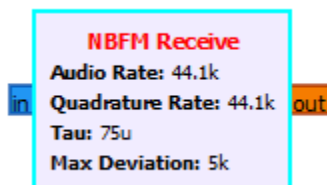


Figura 3.24 Bloque NBFM receptor

Las configuraciones necesarias para poder utilizar el bloque visto en la figura 3.24 se muestran a continuación.

|                 |           |
|-----------------|-----------|
| Audio Rate      | samp_rate |
| Quadrature Rate | samp_rate |
| Tau             | 75e-6     |
| Max Deviation   | 5e3       |

Figura 3.25 Configuración para bloque receptor

De esta manera se puede captar la señal enviada y ser convertida a una señal de audio que el computador puede interpretar y colocar en los parlantes del equipo para poder escucharla.

En el siguiente diagrama de bloques se muestra como se ha colocado los bloques para poder recibir la información, además se adiciono el bloque de osciloscopio y analizado de espectro para verificar que la señal haya llegado correctamente.

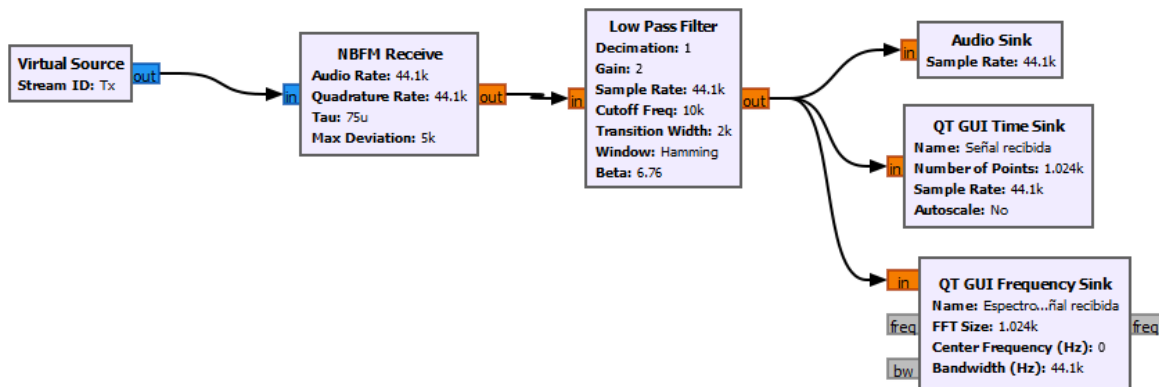


Figura 3.26 Diagrama de bloques para receptor NBFM

En la siguiente figura se muestra todo el diagrama de bloques utilizando, porque aún no se puesto alguna fuente para transmitir. Esta sección se muestra el mismo transmisor utilizado en la figura 3.20

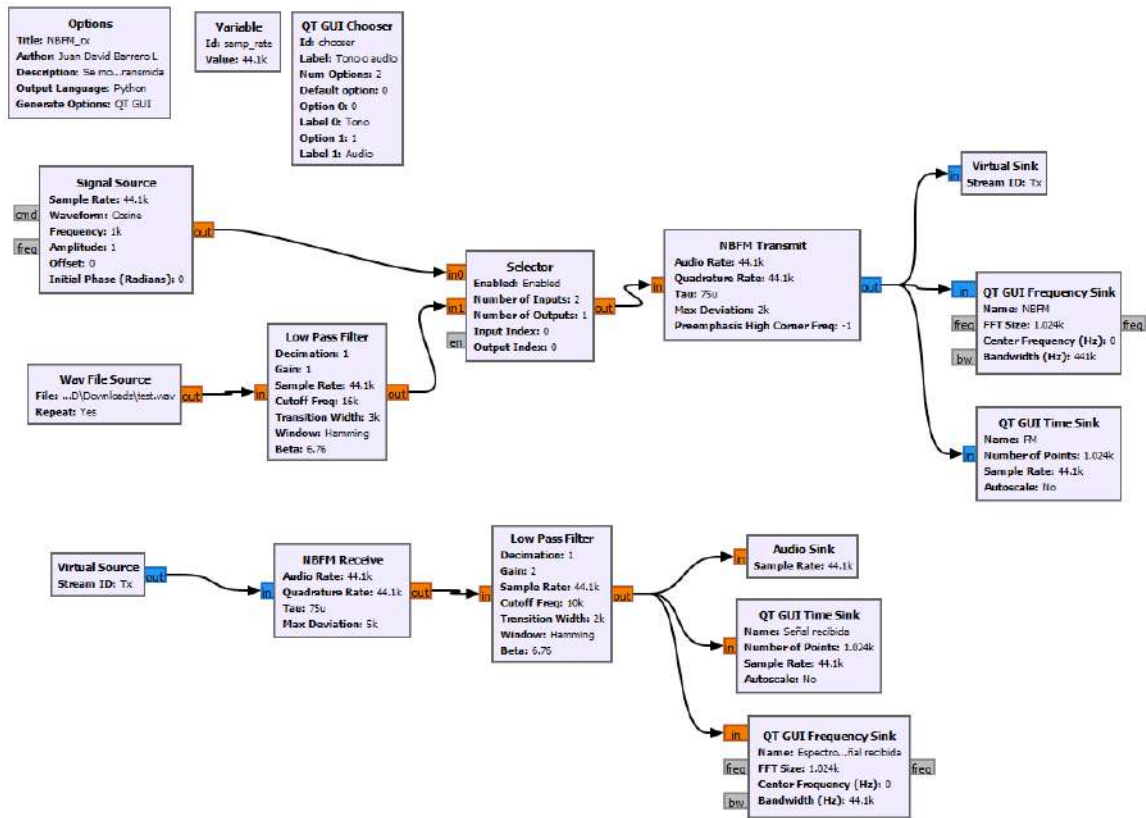


Figura 3.27 Simulación Tx y Rx NBFM

A continuación, se muestran los resultados, en estos se ve claramente que la señal ha llega de manera correcta.

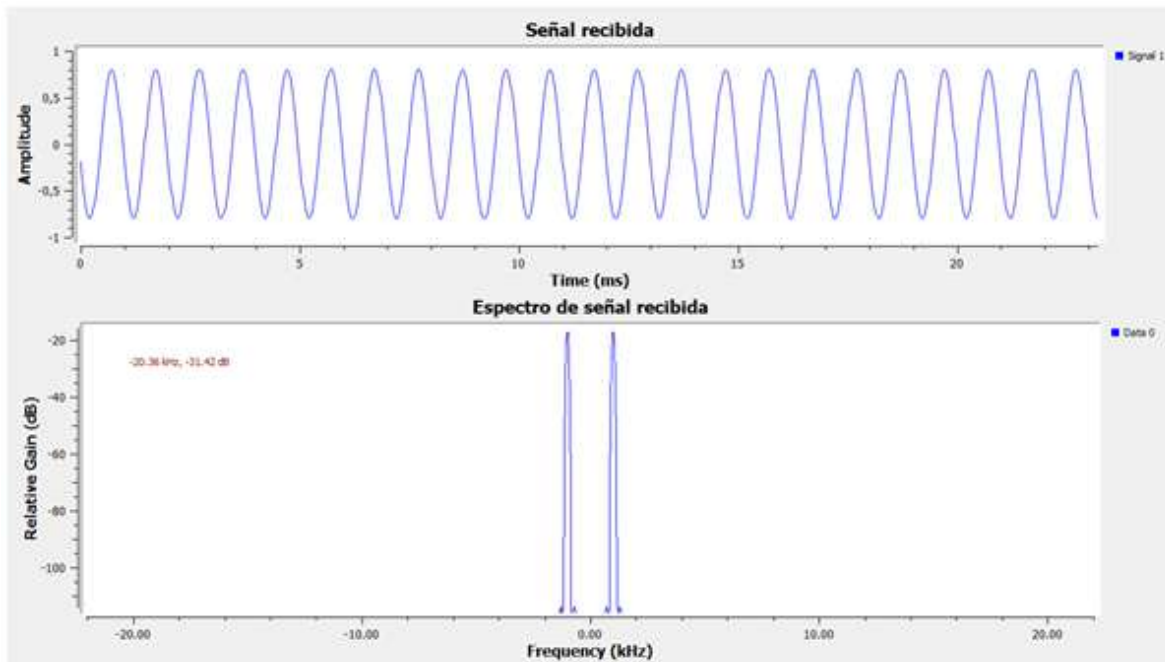


Figura 3.28 Señal recibida con NBFM

De la misma forma que se realizó el receptor simulado, se va a construir el receptor con el RDS RTL-SDR-Dongle para esto solo toca hacer unos cambios y agregar unas variables para que el código se fácil de leer y de interpretar. Este se puede ver a continuación en la siguiente figura.

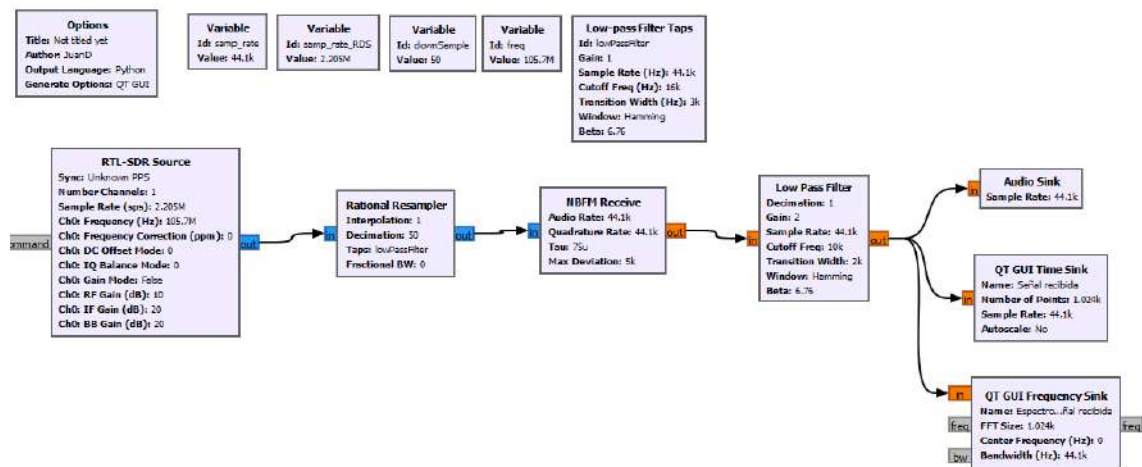


Figura 3.29 Receptor NBFM con RDS

Al observar la figura 3.30 la señal llega y se desmodula de forma clara, por lo que el mensaje transmitido ha sido enviado con éxito. En los parlantes de la maquina se debe poder oír la información que fue enviada.

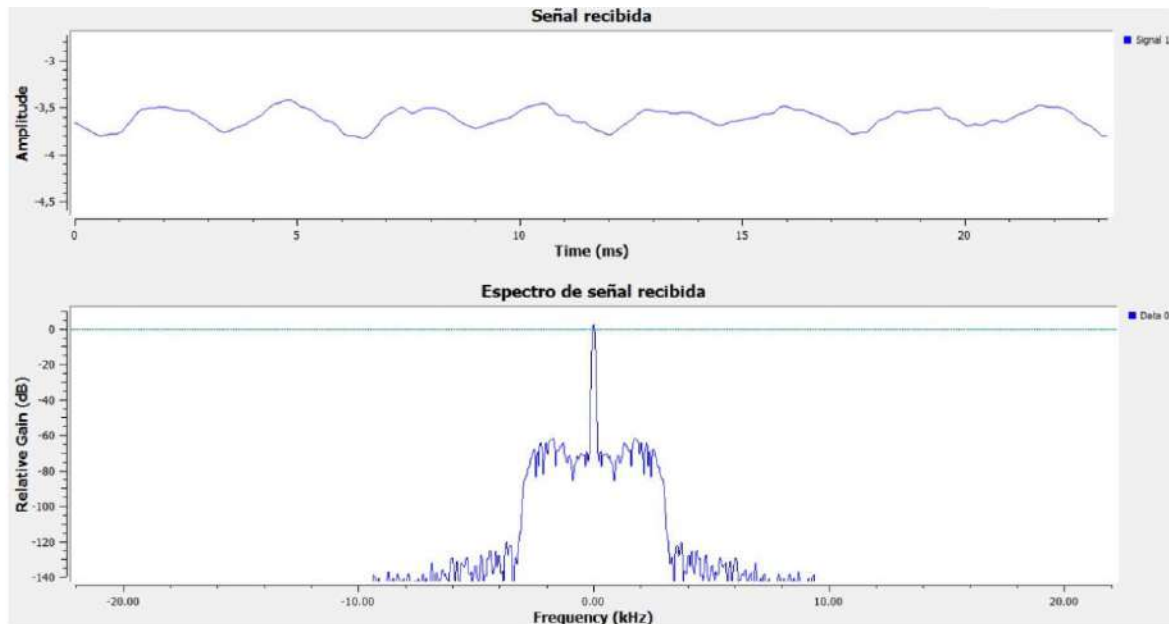


Figura 3.30 Señal recibida utilizando RDS

### 3.3 Modulación de frecuencia banda ancha

Cuando no se puede aproximar utilizando las series de Taylor, es evidente que el ancho de banda de FM es infinito, sin embargo, esto no es así en la práctica se debe considerar un valor llamado desviación de frecuencia ( $\Delta f$ ), el cual se utiliza para poder dar un valor estimado del ancho de banda de la señal modulada.

De lo dicho anteriormente es lógico decir que el espectro de modulación FM es mucho mayor al ancho de banda producido por una señal modulada en amplitud, y hay varios parámetros que caracterizan a este tipo de modulación como lo son:

**El coeficiente de sensibilidad:** el cual es el número de veces que es necesario amplificar la señal moduladora para lograr el índice de modulación deseado y se puede expresar de la siguiente forma

$$K_f$$

Ecuación 21 Coeficiente de sensibilidad

**Desviación de frecuencia:** El cual es el cambio en la frecuencia que sucede que ocurre en la señal portadora cuando hay un cambio en la amplitud de la señal moduladora. Este se puede expresar con la siguiente ecuación

$$\Delta F = K_f A_m$$

*Ecuación 22 Desviación de frecuencia*

**Ancho de banda:** Es el espacio que ocupa la señal modulada en el espectro el electromagnético, y esta se puede expresar con la siguiente ecuación, las cuales son una aproximación válida para calcularlo, ya que, en este tipo de modulación el ancho de banda es teóricamente infinito.

$$B_w = 2\Delta F$$
$$B_w = 2(\Delta f + f_m)$$

*Ecuación 23 Ancho de banda para FM*

**Índice de modulación:** Es la relación entre la desviación de frecuencia y la frecuencia de modulación, tal y como se ve en la siguiente ecuación

$$\beta = \frac{\Delta F}{f_m} = \frac{K_f A_m}{f_m}$$

*Ecuación 24 Índice de modulación*

Usando una notación compleja se puede expresar la ecuación de modulación de frecuencia de la siguiente forma

$$\phi_{Fm} = \text{Re}\{ A e^{j\omega_c t} e^{j\beta \text{Sen}(\omega_m t)} \}$$

*Ecuación 25 Ecuación compleja FM*

La segunda función exponencial, se puede expandir como una serie en el dominio de la frecuencia y se puede expresar de la siguiente forma

$$e^{j\beta \text{Sen}(\omega_m t)} = \sum_{n=-\infty}^{\infty} F_n e^{jn\omega_m t}$$

*Ecuación 26 Ecuación dominio de la frecuencia*

Donde

$$F_n = \frac{1}{T} \int_{-T/2}^{T/2} e^{j\beta \text{Sen}(\omega_m t)} e^{-jn\omega_m t} dt$$

*Ecuación 27 aproximación del segundo termino*

Haciendo el siguiente cambio de variable  $\zeta = \omega_m t$  se obtiene

$$F_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{j(\beta \text{Sen}(\xi) - n\xi)} d\xi$$

*Ecuación 28 Reducción con cambio de variable*

Para poder resolver esta integral la manera más sencilla de lograrlo es utilizando las series de Bessel y se obtiene el siguiente resultado.

$$e^{j\beta \text{Sen}(W_m t)} = \sum_{n=-\infty}^{\infty} J_n(\beta) e^{jnW_m t}$$

*Ecuación 29 aproximación series de Bessel*

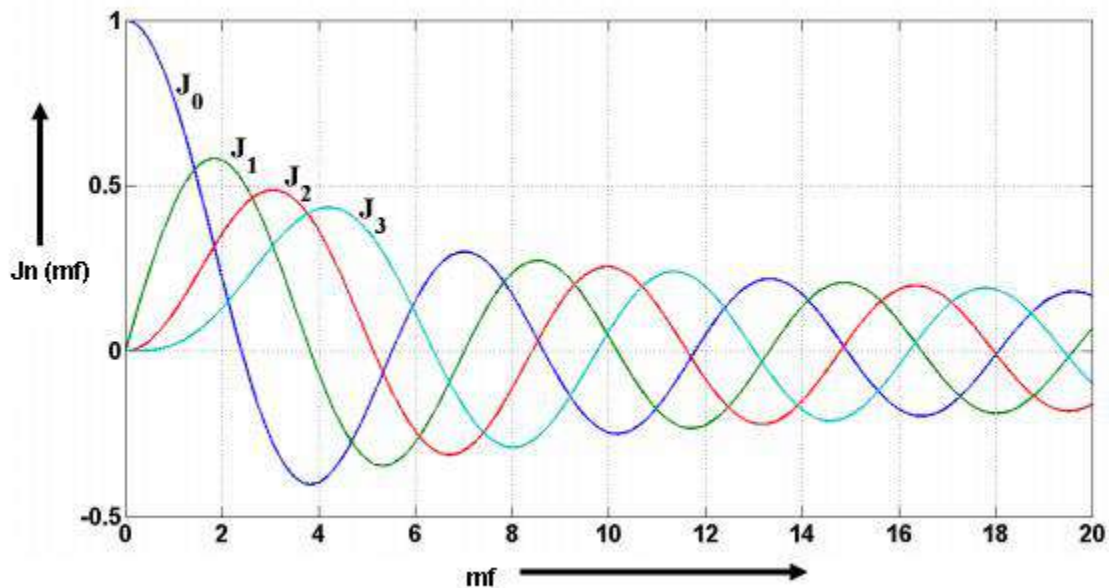
Remplazando la ecuación obtenida en la ecuación 22 se obtiene la siguiente expresión para poder expresar la modulación en frecuencia.

$$\phi_{Fm} = A \sum_{n=-\infty}^{\infty} J_n(\beta) \text{Cos}((W_c + nW_m)t)$$

*Ecuación 30 Ecuación FM*

De manera gráfica la ecuación 27 se puede expresar como una secuencia de funciones senoidales tal y como se observa en la siguiente figura.

*Figura 9 Imagen tomada de (Andreotti, 2015)*



*Figura 3.31 Coeficientes de Bessel*

En la siguiente tabla se puede el valor puede tener los coeficientes de Bessel a medida que cambia el valor del índice de modulación

| $\beta$ | J0   | J1   | J2   | J3   | J4   | J5   | J6   | J7   | J8   | J9   | J10  | J11  |
|---------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0       | 1    | -    | -    | -    | -    | -    | -    | -    | -    | -    | -    | -    |
| 0,25    | 0,98 | 0,12 | -    | -    | -    | -    | -    | -    | -    | -    | -    | -    |
| 0,5     | 0,94 | 0,24 | 0,03 | -    | -    | -    | -    | -    | -    | -    | -    | -    |
| 1       | 0,77 | 0,44 | 0,11 | 0,02 | -    | -    | -    | -    | -    | -    | -    | -    |
| 1,5     | 0,51 | 0,56 | 0,23 | 0,06 | 0,01 | -    | -    | -    | -    | -    | -    | -    |
| 2       | 0,22 | 0,58 | 0,35 | 0,13 | 0,03 | -    | -    | -    | -    | -    | -    | -    |
| 2,5     | 0,05 | 0,5  | 0,45 | 0,22 | 0,07 | 0,02 | -    | -    | -    | -    | -    | -    |
| 3       | 0,26 | 0,34 | 0,49 | 0,31 | 0,13 | 0,04 | 0,01 | -    | -    | -    | -    | -    |
| 4       | 0,4  | 0,07 | 0,36 | 0,43 | 0,28 | 0,13 | 0,05 | 0,02 | -    | -    | -    | -    |
| 5       | 0,18 | 0,33 | 0,05 | 0,36 | 0,39 | 0,26 | 0,13 | 0,05 | 0,02 | -    | -    | -    |
| 6       | 0,15 | 0,28 | 0,24 | 0,11 | 0,36 | 0,36 | 0,25 | 0,13 | 0,06 | 0,02 | -    | -    |
| 7       | 0,3  | 0    | 0,3  | 0,17 | 0,16 | 0,35 | 0,34 | 0,23 | 0,13 | 0,06 | 0,02 | -    |
| 8       | 0,17 | 0,23 | 0,11 | 0,29 | 0,1  | 0,19 | 0,34 | 0,32 | 0,22 | 0,13 | 0,06 | 0,03 |

*Tabla 7 Coeficientes de Bessel*

### 3.3.1 GNU radio para verificar funciones Bessel

Utilizando el software se generará el siguiente diagrama de bloques para poder verificar si los coeficientes vistos en la tabla 7 corresponden con la teoría nombrada anteriormente.

Antes de realizar el código se muestra cómo debería ser el diagrama de Bessel, según los diferentes valores del índice de modulación, la siguiente imagen fue tomada de (Introducción a los sistemas de comunicación, 2000)

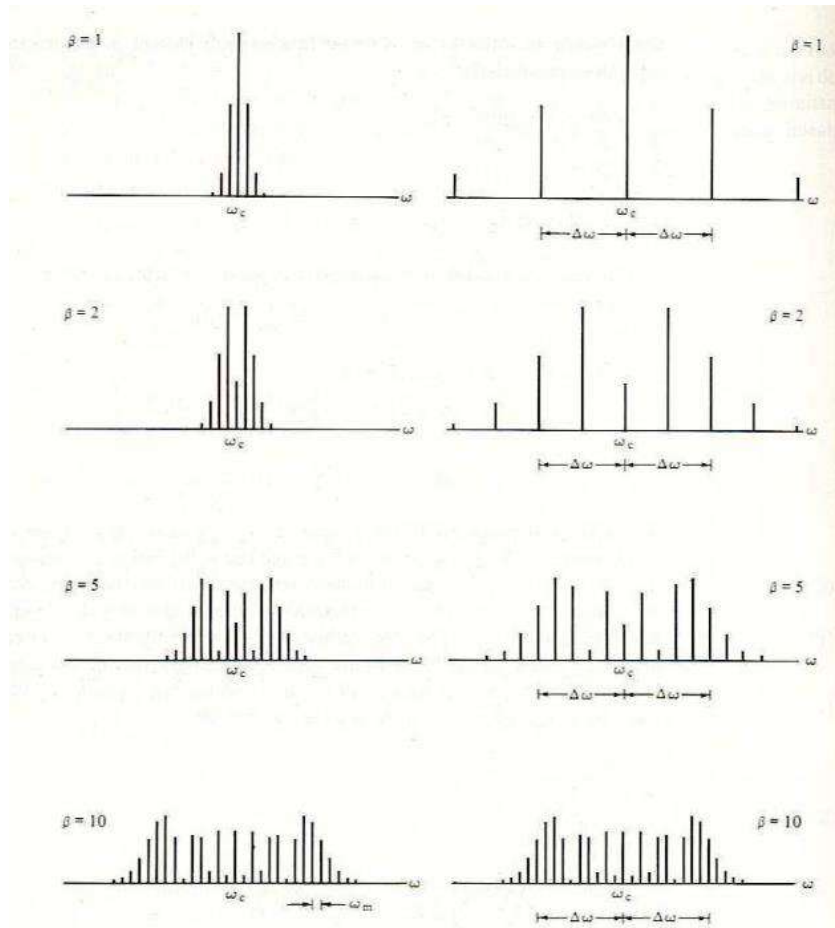


Figura 3.32 Espectro ante diferentes índices de modulación

Para poder comprobar el comportamiento del espectro de la modulación FM se construyó el siguiente diagrama de bloques en GNU radio, esta vez se utilizó el bloque VCO complex ya que este brinda una mejor resolución del resultado que es está esperando ver al final en el analizador de espectro.

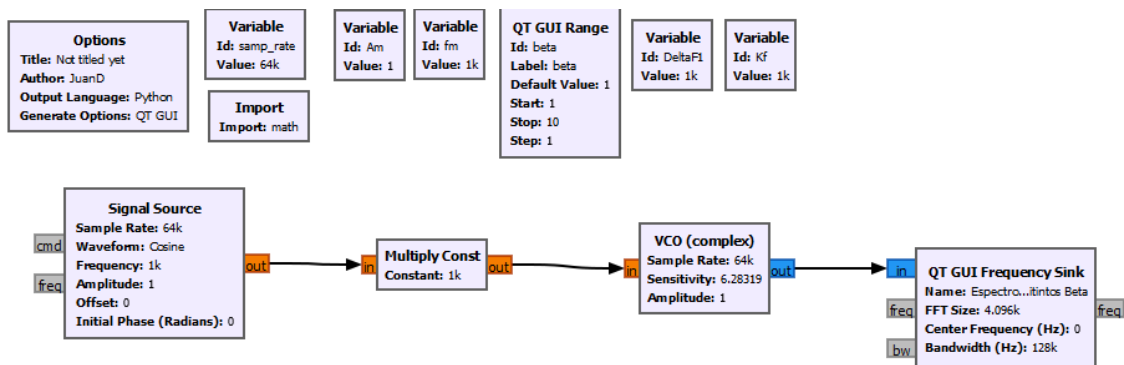


Figura 3.33 Prueba de Bessel

Acá se puede apreciar el resultado ante distintos valores de beta, y como se observa estos concuerdan con los valores vistos en la figura 3.32, y también con los mostrados en la tabla 7

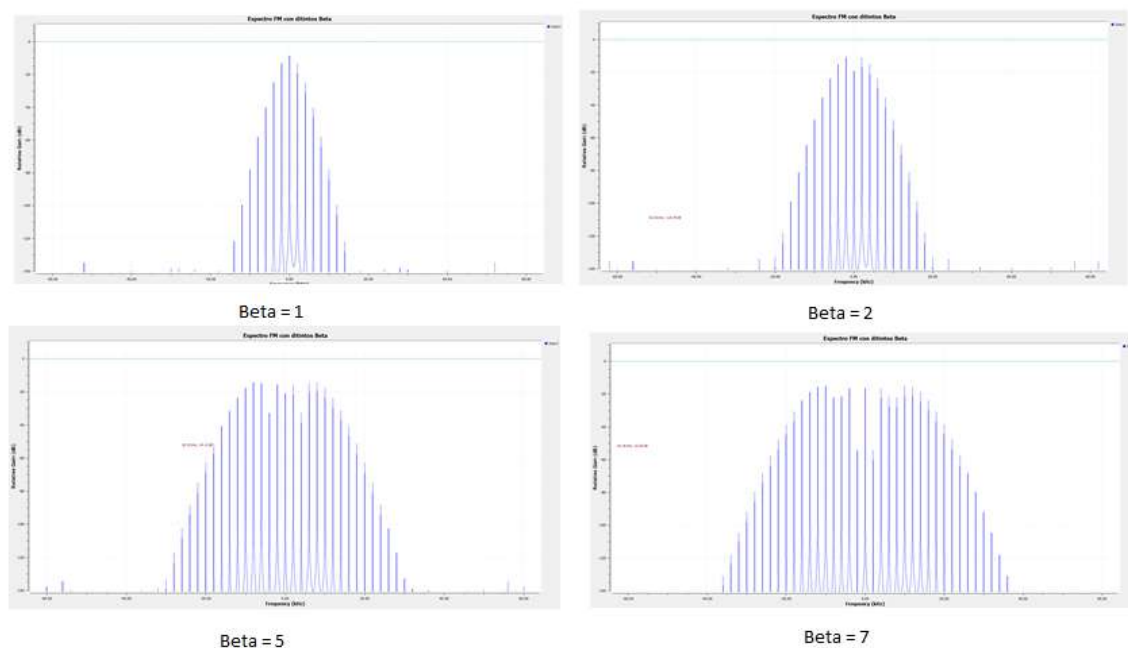


Figura 3.34 Valores de coeficientes Bessel

### 3.3.2 Captando emisoras de FM con RDS

Una de las grandes ventajas de utilizar RDS es poder interceptar todo de tipo de señales que se encuentran en el aire. En la sección anterior se mostros como el protocolo de WBFM el cual es el mismo que utiliza las emisora de FM en el mundo, con esta información se puede usar esta información para captar las emisoras de radio utilizando GNU radio.

Para poder lograr este radio es necesario utilizar el bloque que viene con GNU radio para decodificar señales WBFM, el cual tiene dos parámetros que debe ser puesto el primero es la cuadratura y este debe coincidir con el número de muestra que están llegando de la entrada del bloque, en la siguiente imagen se puede ver como es el bloque.



Figura 3.35 Receptor de emisoras

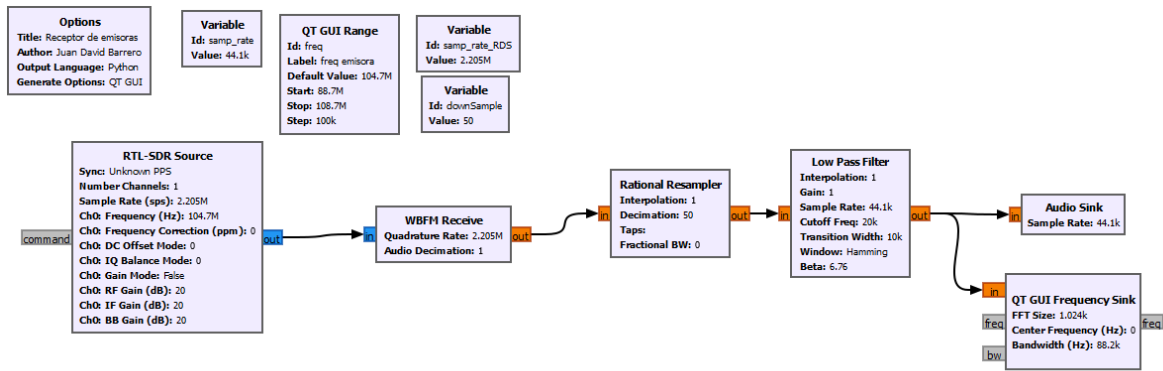


Figura 3.36 Receptor de emisoras

De esta manera se puede captar las emisoras de radio que se encuentran en emisión, el código cuenta con un deslizador para captar la emisora que desea, además se puede ver en tiempo real el espectro de la frecuencia, para poder observar la emisora captada. Tal y como se muestra en la siguiente figura.

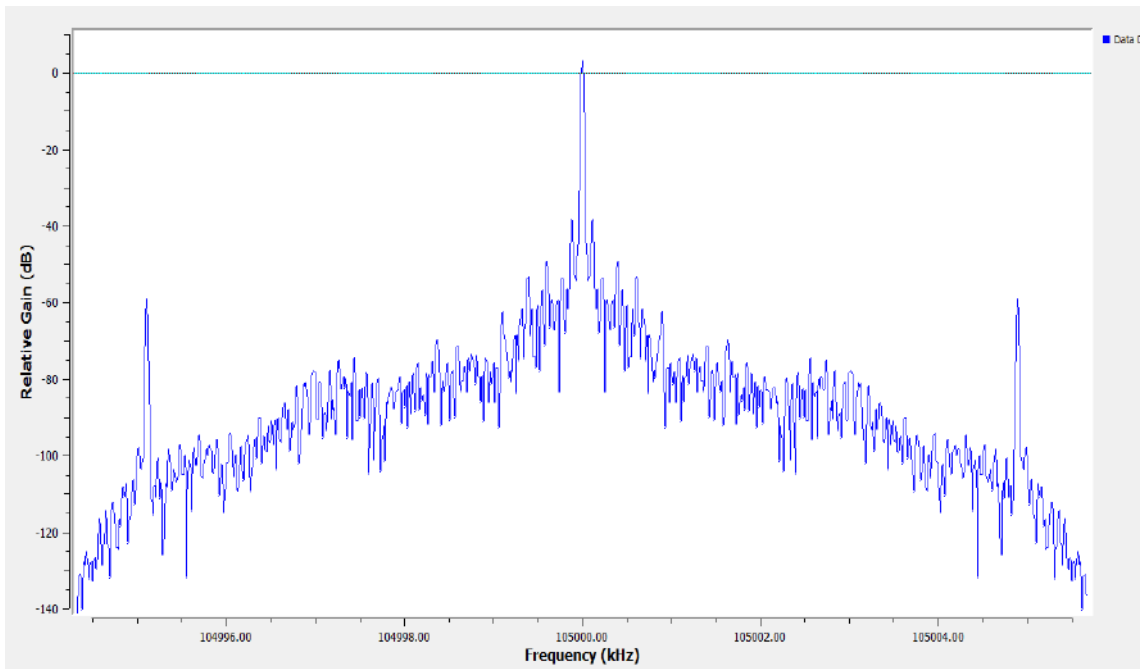


Figura 3.37 Receptor de emisora

Como se observa en la figura 3.37 la señal es captada correctamente, y por los parlantes del equipo se puede escuchar la emisora que se sintonizo. Una anotación importante que debe hacerse es que algunas emisoras tienen poca potencia dependiendo de la ubicación y se puede oír ruido.

### 3.3.3 Tx y Rx WBFM con GNU radio

En la sección anterior se mostró como poder hacer la recepción de emisoras utilizando el bloque “WBFM receive”, para transmitir la señal es necesario utilizar el bloque “WBFM transmit”, este genera la señal modulada, en la siguiente figura se muestra los parámetros que deben ser colocados para hacerlo funcionar.

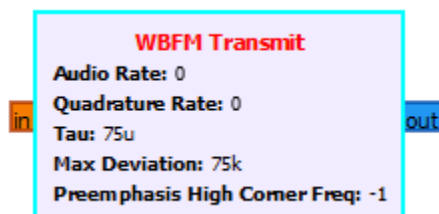


Figura 3.38 WBFM transmit

**Audio rate:** es la tasa de muestro que va a tener la señal de entrada si no coinciden la señal modulada no se podrá reconstruir.

**Quadrature rate:** Es la tasa a la que se desea tener la señal de salida, este debe un múltiplo del audio rate

**Max deviation:** es la desviación de frecuencia, este valor y el de tau están puestos por defecto, y están configurados para ajustarse a los parámetros de las emisoras comerciales

Al tener los parámetros claros ya se puede hacer el transmisor de la señal que se quiera, para la prueba se pondrá un tono a una frecuencia constante y una señal de audio. Esto se puede ver en la siguiente figura.

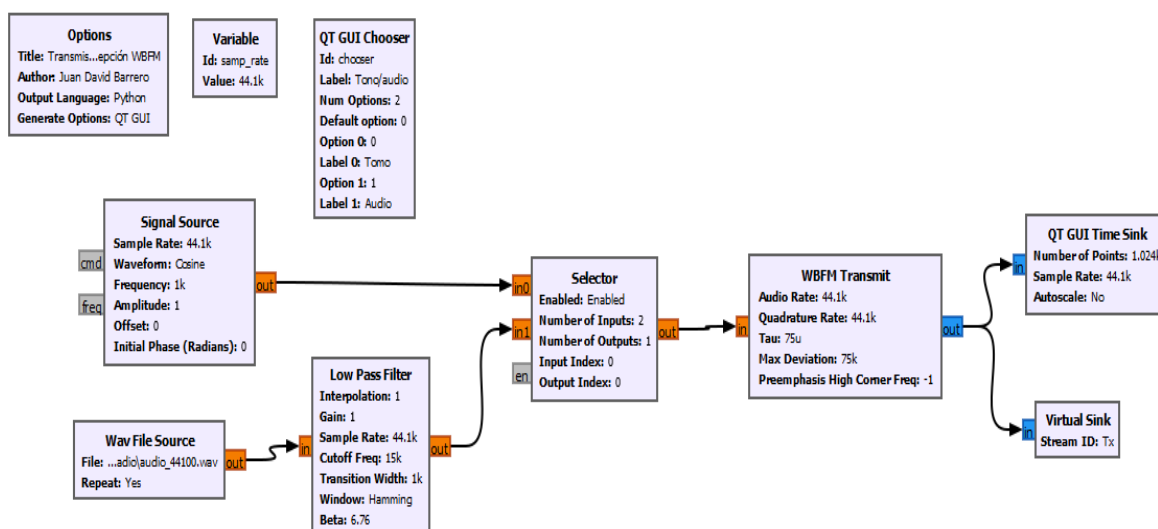


Figura 3.39 Transmisor WBFM

Para poder probar que todo esté funcionando se colocó el mismo receptor visto en la sección anterior, ya que se ha probado que funciona para emisora de radio, por esta razón se decidió colocar el mismo. En la siguiente figura se muestra el diagrama de bloques completo.

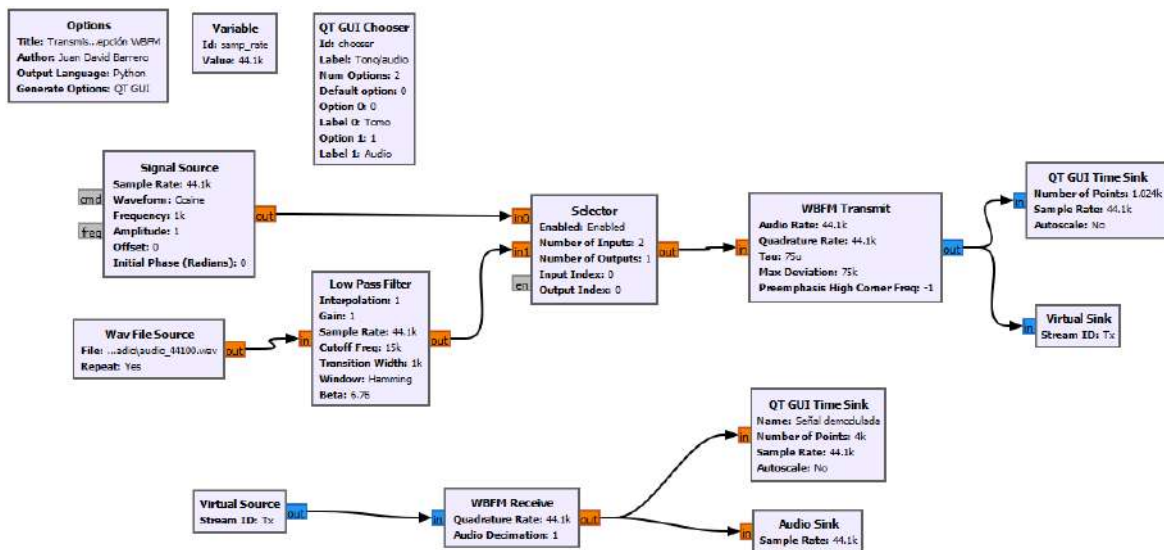


Figura 3.40 Tx y Rx WBFM

Al ejecutar el código se aparece la señal de información de modulada, y la señal modulada, además por los parlantes del equipo se podrá escuchar el mensaje enviado o el tono, se puede escoger cuál de los se quiere transmitir y oír

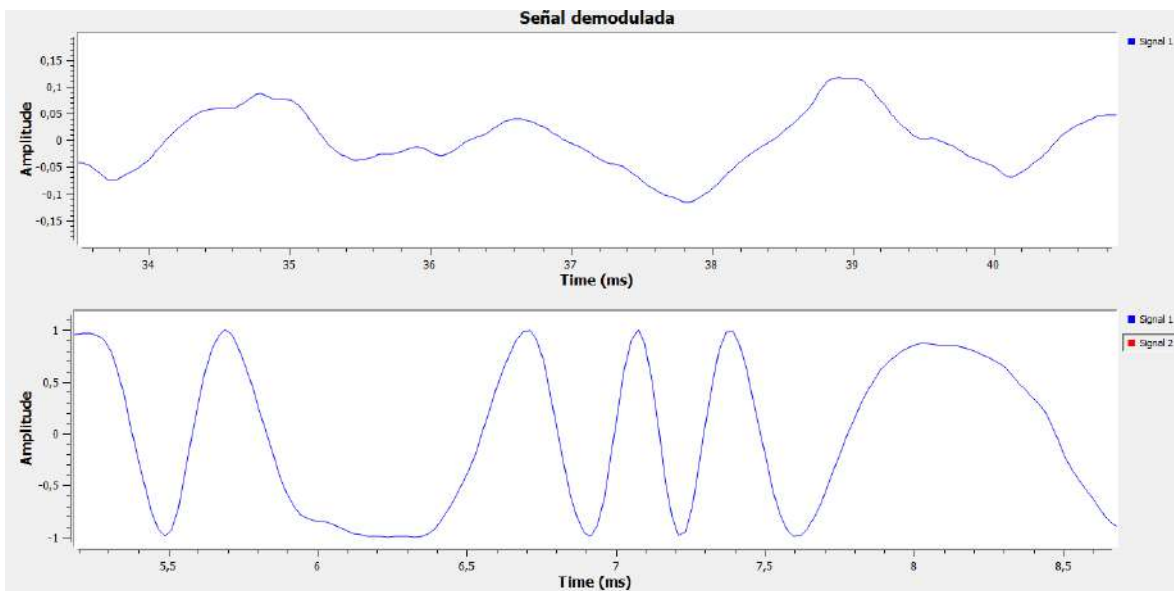


Figura 3.41 Resultados WBFM tx y rx

### 3.3.4 Transmisión WBFM utilizando el RDS

Con el siguiente diagrama de bloques se puede lograr la transmisión y al igual que hizo en la sección de modulación de frecuencia banda agosta, se escogerá una frecuencia de transmisión en la banda FM para de esta manera poder escuchar por un radio o por un celular que pueda captar señales de FM.

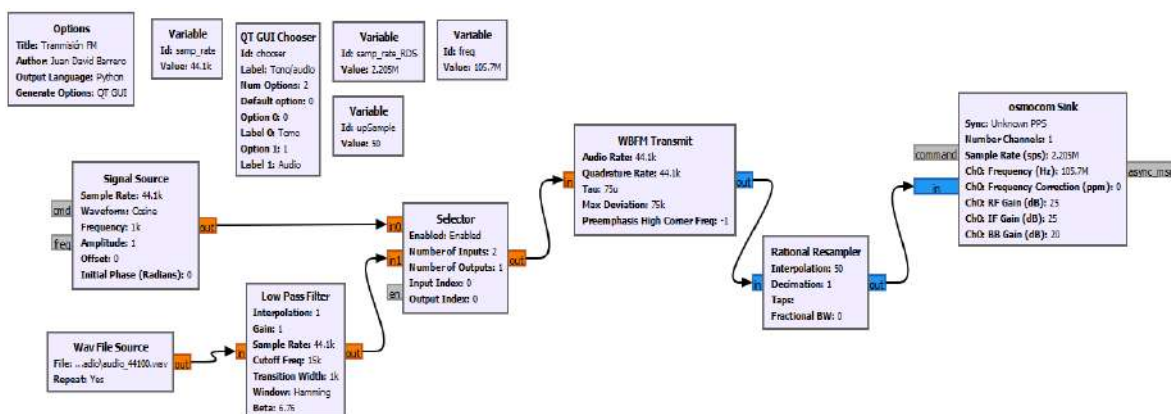


Figura 3.42 Transmisor WBFM con RDS

Antes de realizar el receptor con el RDS, se utilizará el software SDR# para comprobar que la señal enviada corresponda con el espectro de las señales de las emisoras (Las cuales también están modulada con WBFM).

Antes hacer esto hay que dar un vistazo a la tabla 6 y escoger el valor de la frecuencia que mejor se ajuste a las necesidades. En este caso se manda la información en una portadora con una frecuencia de 105.7 [MHz].

Para poder escuchar la señal hay que ajustar el siguiente parámetro en SDR#, una vez hecho esto ya se puede ajustar la frecuencia a la que se fijó la portadora y hacer clic sobre el botón reproducir.

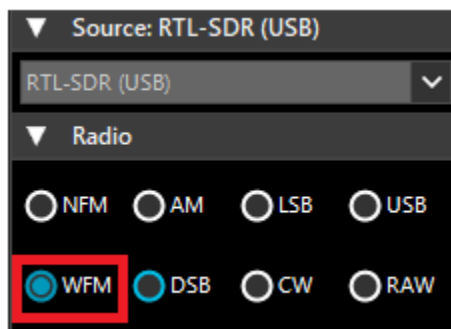


Figura 3.43 Ajuste SDR

En la siguiente figura se puede ver que en la frecuencia seleccionada aparece un espectro de la misma forma, que el de la otra señal de la emisora con esto se puede contrastar que la enviada por el RDS fue modulada correctamente.

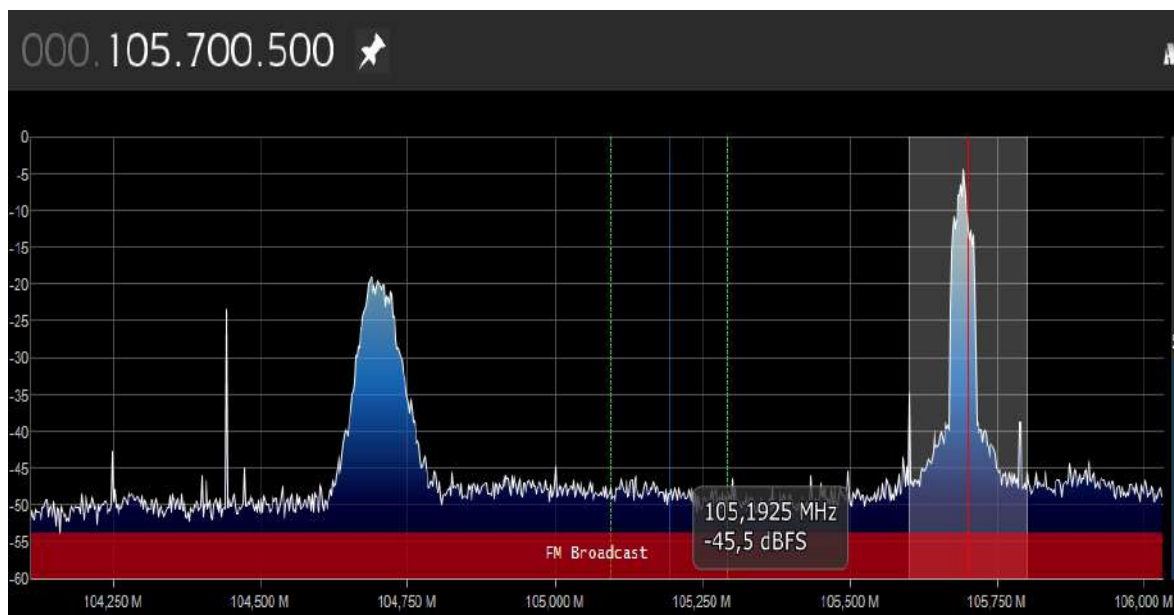


Figura 3.44 Modulación WBFM

Incluso si se coloca un radio y se sintoniza la frecuencia se puede oír el mensaje enviado. Para poder recibir la información utilizando el software de GNU radio se puede utilizar el mismo diagrama de bloques usado en la sección anterior. A continuación se muestra el espectro captado utilizando el receptor de emisoras.

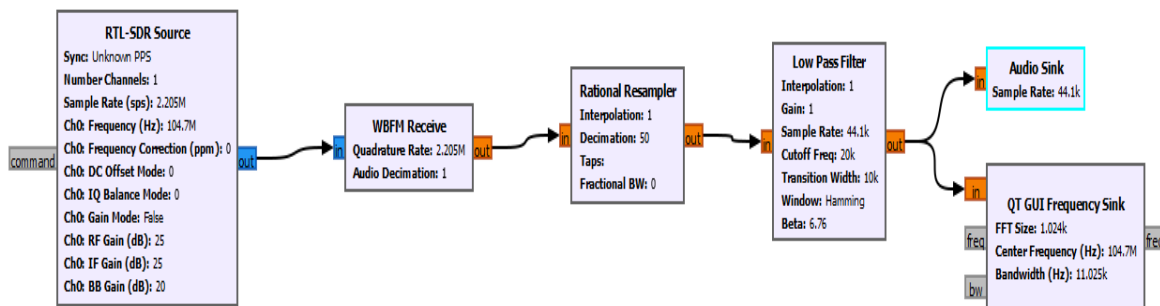


Figura 3.45 Receptor de señal transmitida con RDS

## 4. Modulaciones digitales utilizando RDS

A lo largo de este curso se ha mostrado como utilizando como señal moduladora señales reales, en esta sección se hablará sobre la combinación de señales digitales (moduladora) y analógicas (portadora). A demás se aborda las ventajas y los inconvenientes que se tiene a la hora de trabajar con estos tipos de modulaciones.

### Desventajas

- Menor posibilidad de irradiar la información
- Menor eficiencia en el uso del espectro electromagnético

### Ventajas

- Inmunidad frente al ruido
- Facilidad de operar
- Codificación y encriptación
- Modulación y demodulación con DPS

Dentro de este tipo de modulaciones se puede considerar entre las importantes ASK(Amplitud Shift Keying), FSK(Frequency Shift Keying), PSK(Phaser Shift Keying) y QAM la cual es la combinación de ASK con PSK.

*Figura 10 Imagen tomada de (Rodrigurz, s.f.)*



*Figura 4.1 Modulación digitales*

A lo largo de esta sección se mostrará como utilizando GNU radio generar y transmitir esta información con el RDS, también como poder captarla y de decodificarla para obtener de mensaje que se este enviando. Finalmente observar el comportamiento de la señal en el dominio de la frecuencia para contrastar el funcionamiento con la teoría.

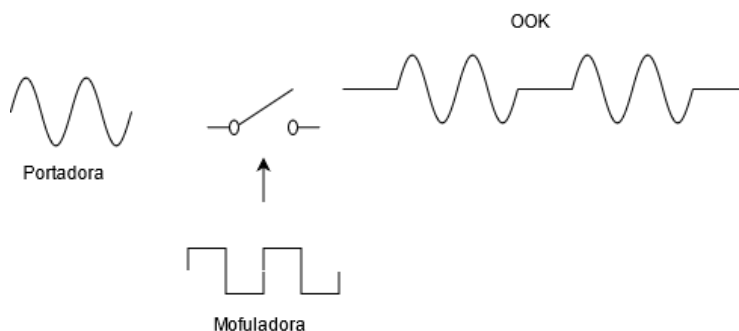
## 4.1 modulación ASK

Amplitud Shift Keying es la implementación digital de la modulación AM, en esta se utiliza como portadora una señal digital, esta puede ser una señal bit generada por el computador o algún tipo de documento (archivo de texto o imagen) que se quiera transmitir.

Este tipo de modulación es bastante usado por su facilidad de implementar, codificar y decodificar además cuenta con unas modificaciones que permiten hacer este tipo de transmisión la idea para comunicaciones de bajo consumo y de poca seguridad como lo controles remotos para televisores o juguetes.

### 4.1.1 OOK con GNU radio

Para comenzar es necesario ver el diagrama de bloques del modulador OOK el cual solo deja pasar una señal de una amplitud fija a determinado tiempo y la retiene por un periodo, de esta forma se consigue modular la información. Dejándola lista para ser enviada. tal y como se ve en la siguiente figura.



*Figura 4.2 Modulación OOK*

La gran ventaja que tiene esta modulación es su simplicidad y el poco consumo energético que esta tiene, el mayor inconveniente que se presenta se es la gran cantidad de ancho de banda que requiere, por esta este tipo de modulación se usa para mandar información rápida y que no requiera ser enviada constantemente.

En la siguiente figura se puede observar cómo se puede producir una señal modulada con OOK, utilizando como base la configuración vista en la figura 155, este hace con la intención de ver la cantidad de ancho da banda que consume enviar con este tipo de modulación.

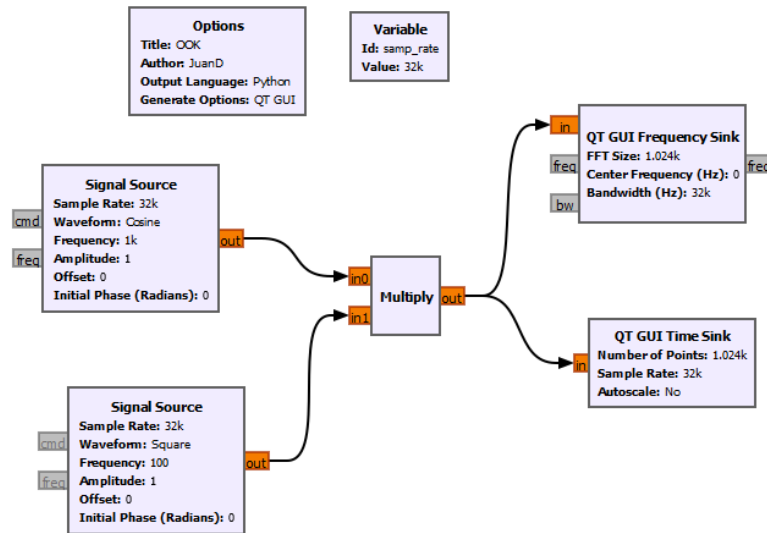


Figura 4.3 OOK con GNU radio

Al ejecutar el código se puede observar en la figura el resultado espera, como la señal cuadrada tiene infinitos armónicos el ancho de banda se extiende por todo el analizador de espectro sin un fin aparente, para evitar esto es necesario poner algún tipo de filtro ya que no se puede consumir todo ese espacio para enviar a la hora de transmitir información.

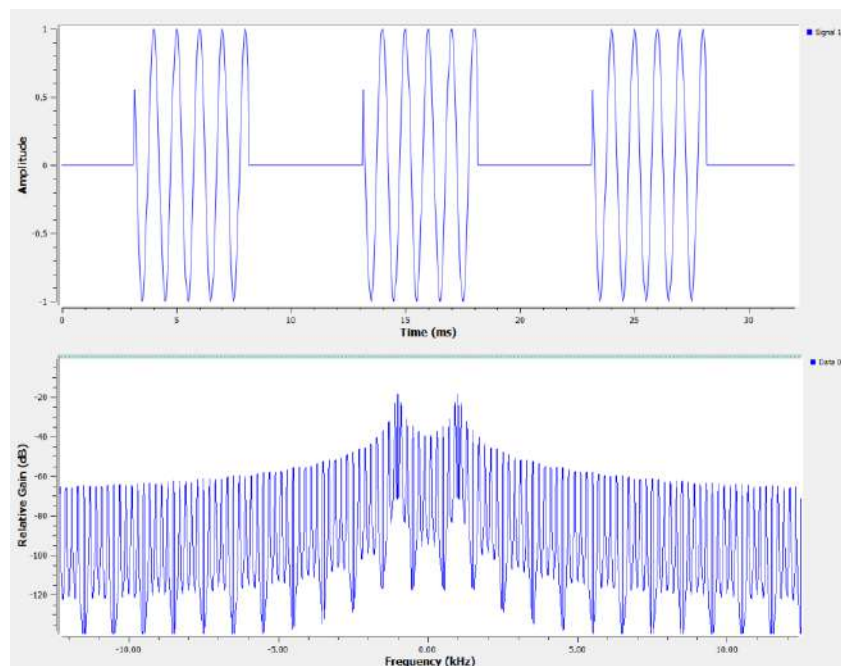


Figura 4.4 resultados de Modulación OOK

## 4.1.2 Transmitiendo modulación OOK con RDS

Con la intención de observar mejor el ancho de banda utilizado por este tipo de modulación es que se utilizara el RDS para transmitir data al aire y poder cuanto ancho de banda este ocupa. Para esto se construye el siguiente diagrama de bloques.

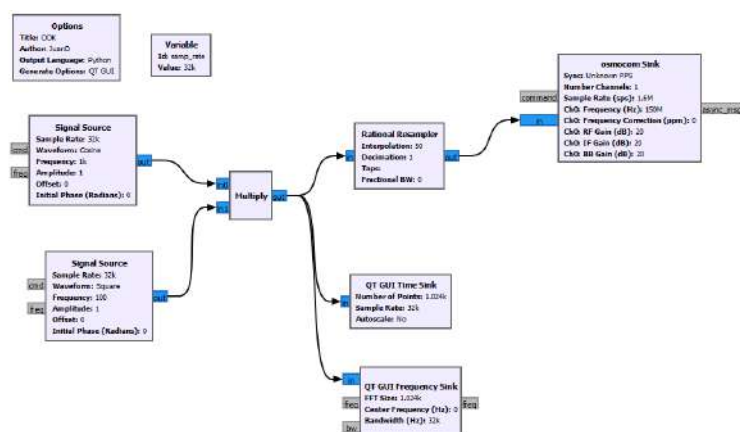


Figura 4.5 Transmisión OOK con RDS

Con la ayuda de del software SDR# se puede observar el ancho de banda total que ocupa enviar este tipo de modulación, lo único que hay que hacer es sintonizar la frecuencia a la cual se está transmitiendo.

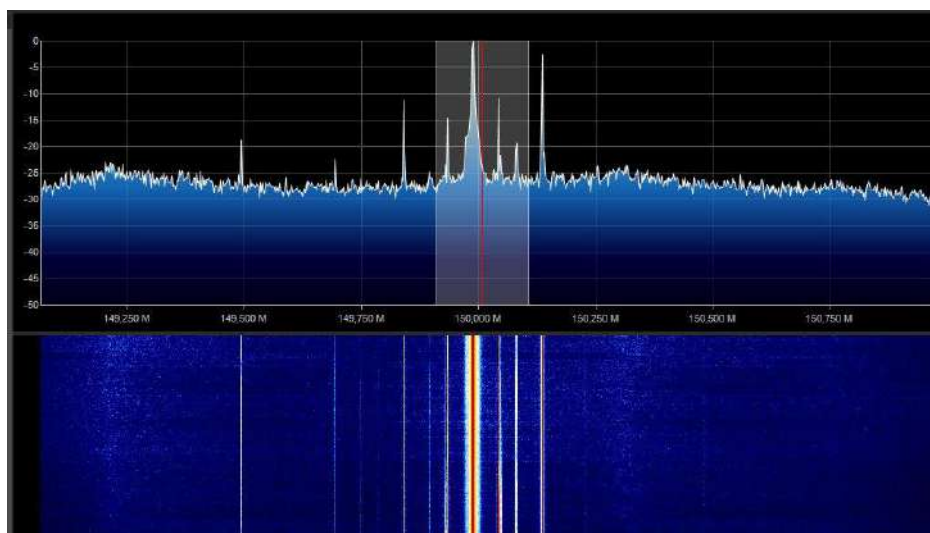


Figura 4.6 Ancho de banda de OOK modulación

En la figura 159 se observa que el ancho de banda se extiende aproximadamente 500[KHz], mucho grande que el FM comercial permitido, por esta razón es que en este tipo de modulación se debe enviar paquetes cortos de información.

### 4.1.3 4-ASK simulación con GNU radio

Con el objetivo de no tener que enviar dos señales totalmente diferentes como lo son un 0 y una señal seno como resultado, nace la modulación 4-ASK la cual consiste en tomar 2 bits de la señal moduladora para generar dos amplitudes similares y de esta forma reducir el ancho de banda necesario para transmitir la información.

Para hacer la prueba primero se genera una señal de bits aleatoria con el siguiente bloque:

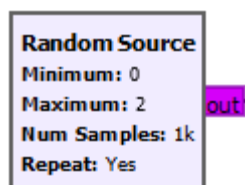


Figura 4.7 Generador de Bits

Este bloque generará una un tren de bits con valor entre 0 y 1, este será tipo de señal se puede ver en la siguiente figura. Estos tienen una rata de bits de 32 [KHz] (este valor puede ser modificado)

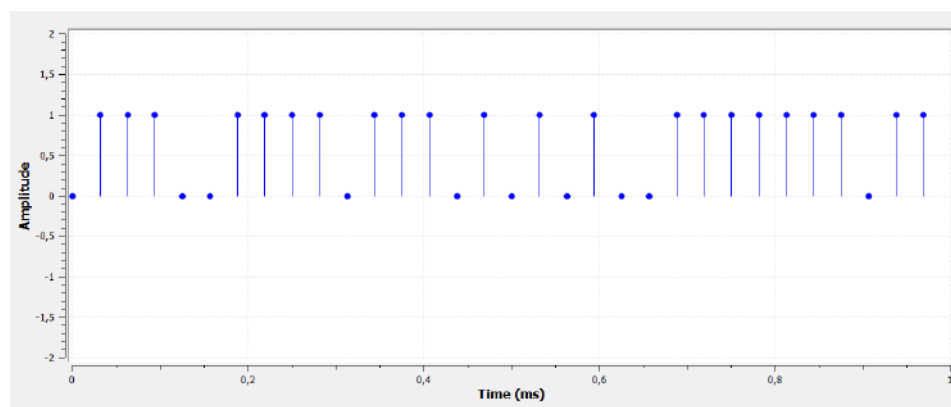


Figura 4.8 Tren de bits

Para poder agrupar el número de bits deseados se utiliza el siguiente bloque el cual sumada el valor de los bits y generara una nueva salida. Es importante tener en cuenta que ahora la rata de bits disminuye el valor de bits que se tome de la muestra

$$R_s = \frac{R_b}{N_{bs}}$$

Ecuación 31 Rata de símbolos

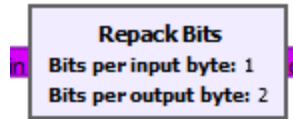


Figura 4.9 Empaquetador de bits

De esta manera ya se logra tener una señal de impulsos dos valores de amplitud diferentes para poder modular la señal portadora. Sin embargo, como estos son valores de bits no se pueden operar con los demás tipos de valores, por este se debe hacer una interpolación con filtro FIR para convertirlos en valores flotantes.

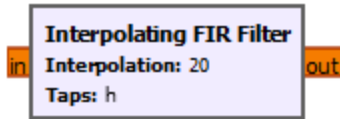


Figura 4.10 Interpolador de datos

A la salida de este bloque ya se obtiene una señal con la se puede trabajar para modulara la señal portadora, en la siguiente figura se observa el diseño completo de diagrama de bloques para producir una señal 4-ASK

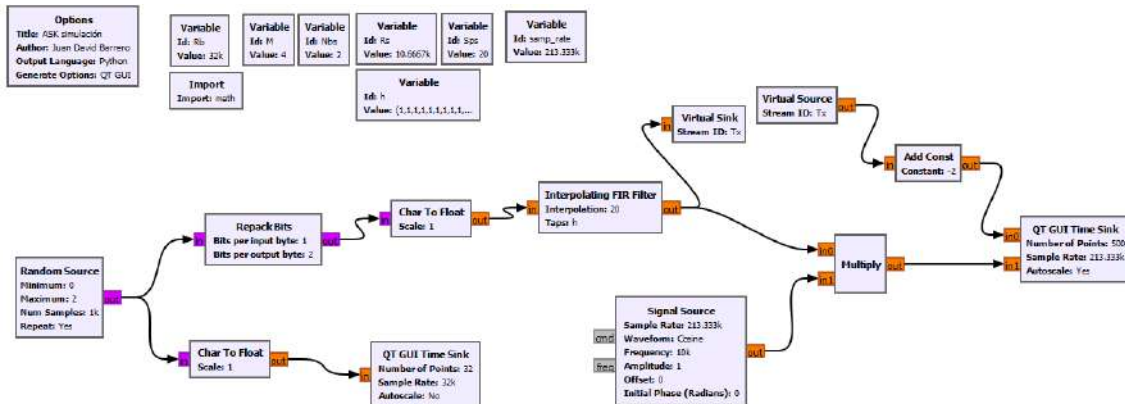
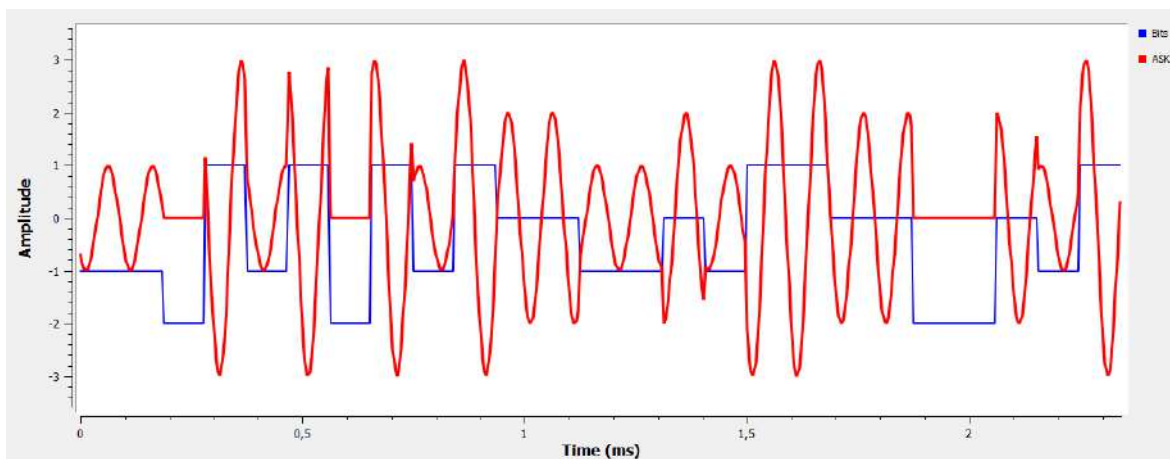


Figura 4.11 modulación 4ASK

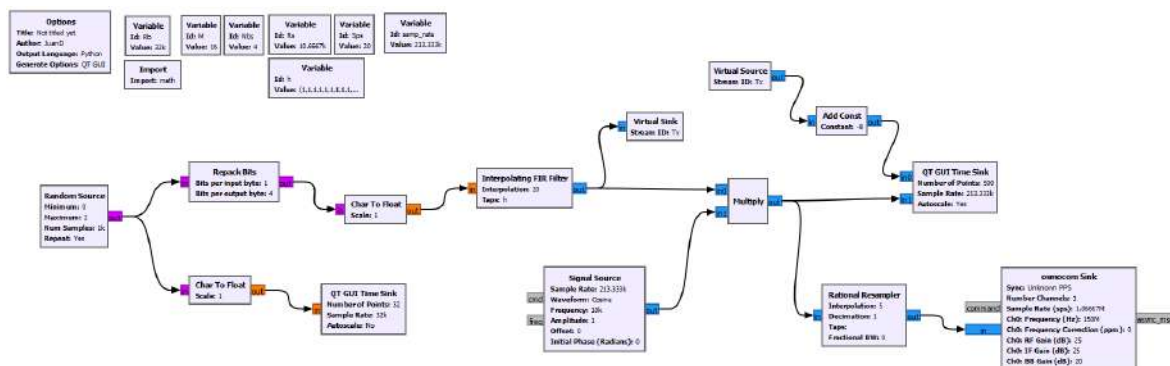
Al ejecutar el código se obtiene el siguiente comportamiento



Con esto se logra reducir el ancho de banda necesario para poder enviar la información, entre mas pequeño sea el cambio en la altura de la señal de bits menor ancho de banda ocupara la señal modulara.

#### 4.1.4 Transmitiendo 4-ASK con RDS

Para comprobar el ancho de banda ocupado al usar la modulación 4-ASK se transmitirá la información utilizando el HackRF-ONE, de esta forma se hará la comparación entre la figura 159 y el resultado obtenido con el siguiente diagrama de bloque.



Al correr el código el ancho de banda que ocupa la nueva señal es

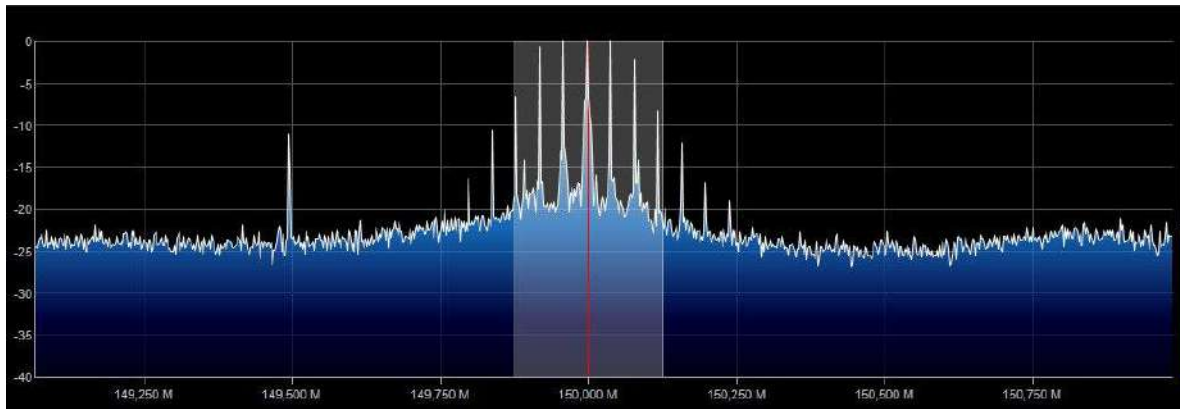


Figura 4.14 Ancho de banda

Como se observa el ancho de banda consumido para transmitir esta señal es de 250[KHz], la mitad de lo que gasto utilizando el sistema de OOK.

## 4.2 modulación FSK

FSK (Frequency-shift keying), es una modulación de frecuencia donde la señal moduladora es digital. Al igual que en ASK existen varios tipos de configuraciones para este proceso la más simple, consiste en dos valores binarios se representan con dos frecuencias diferentes ( $f_1$  y  $f_2$ ) próximas a la frecuencia de la señal portadora  $f_b$ . En el siguiente diagrama de bloque se muestra como es la configuración básica de un diagrama de bloques para modulación FSK.

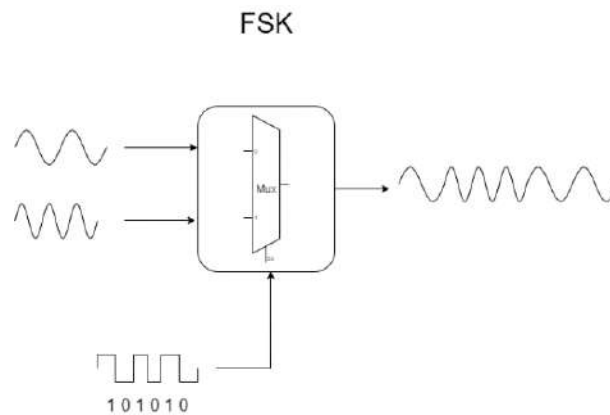
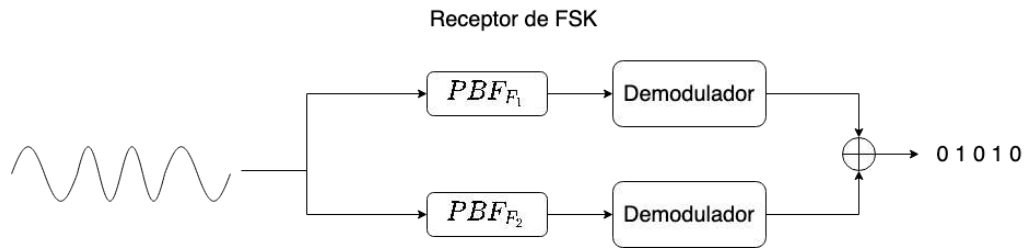


Figura 4.15 modulación FSK

Para poder recuperar la señal modulada se utiliza el siguiente diagrama de bloques. Este funciona de la siguiente manera en el receptor se recibe la señal con ambas frecuencias y se pasan por un filtro pasa banda cada uno de estos las separa respectivamente, luego la señal se demodula para posteriormente sumarla y tener la información transmitida.

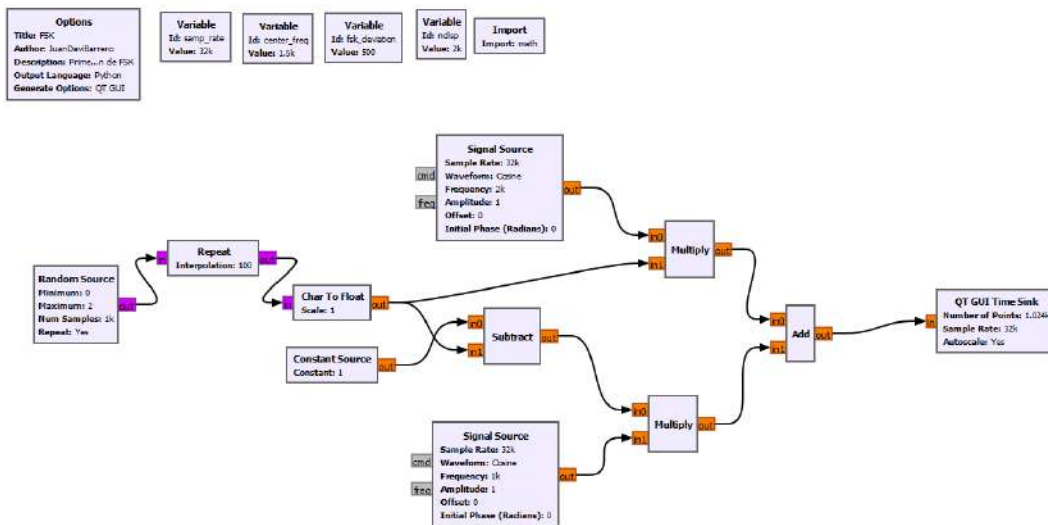


*Figura 4.16 Receptor FSK*

Entre más similares sean esta frecuencia menor será el ancho de banda que se necesita para transmitir la información. Durante esta sección se verá varios métodos de FSK y se podrá determinar cuál de estos es más eficiente para la transmisión de información es importante tener en mente que cada uno de estos, tiene su propias ventajas y desventajas, que se aprovechan según la necesidad.

## 4.2.1 Modulación binaria FSK GNU radio

Con los diagramas de bloques vistos, se puede hacer la modulación FSK binaria en GNU radio, para esto es necesario construir el siguiente diagrama de bloques. Pero como no existe un multiplexor en las librerías implementadas es importantes recrear esta lógica para que la modulación ocurra.



*Figura 4.17 2FSK*

Para la recepción de datos se le agrego al diagrama anterior los siguientes filtros vistos en la figura 169, para así obtener los valores de la señal de entrada. La cual esta generas número de bits aleatorios.

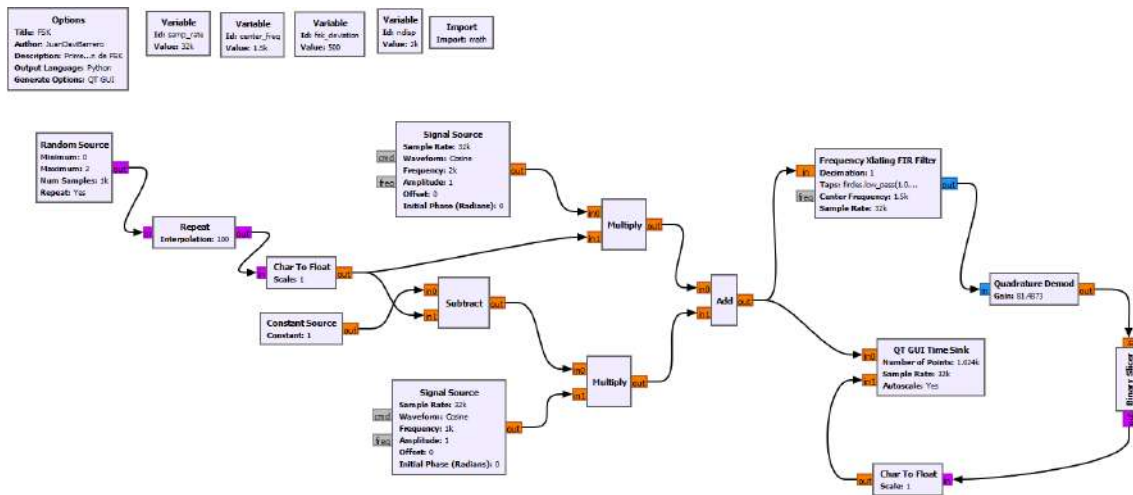


Figura 4.18 Transmisión y recepción FSK

Al ejecutar el código se tiene el siguiente resultado, en este se puede ver como la señal bits se recuperó correctamente, además se puede ver el ancho de banda que ocupa la señal modulada, este se compara luego con los demás resultados de los diferentes métodos de modulación FSK.

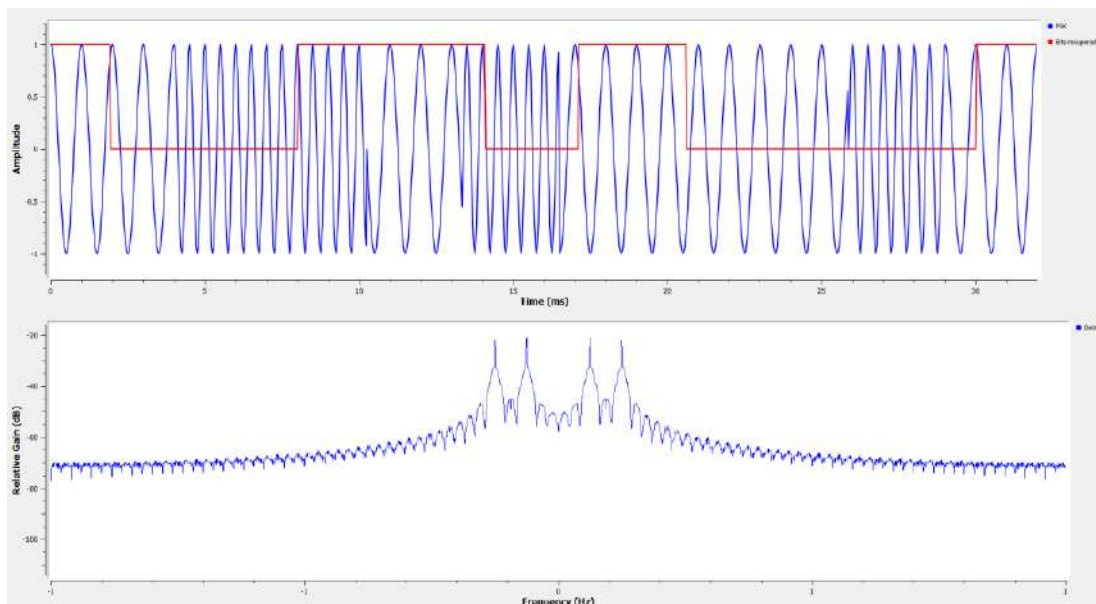


Figura 4.19 Recuperando bits 2FSK

Como se observa en la figura 172 aparecen dos picos los cuales indican la frecuencia de la señal portadora, este es el comportamiento de las modulaciones de frecuencia, la aparición de varias frecuencias, en este caso solo se esperaban ver dos.

## 4.2.2 Transmisión 2FSK con RDS

Con la teoría vista se mostrar cómo hacer la transmisión de bits utilizando este tipo de modulación, para esto solo hay que realizar unos cambios al diagrama de bloques mostrado anteriormente, este ahora debe lucir de la siguiente manera.

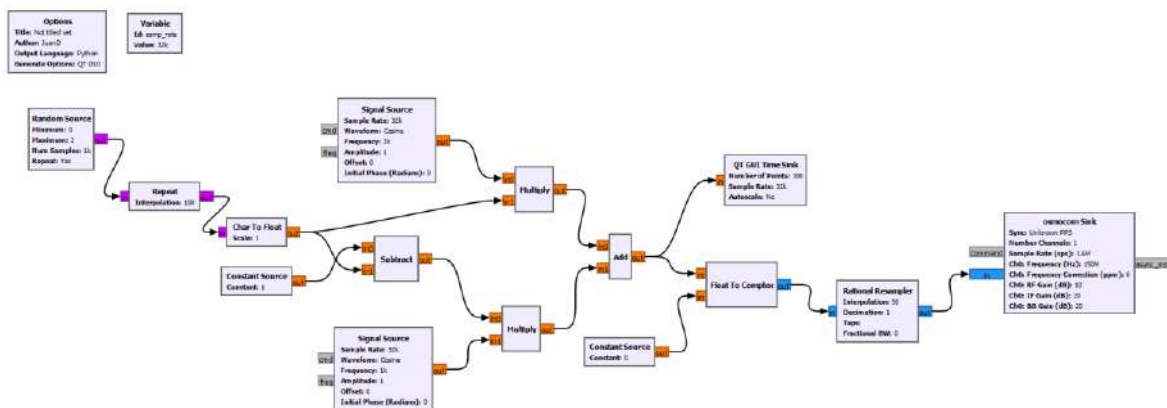


Figura 4.20 Transmisión 2FSK

Para verificar la información se este enviando correctamente se utilizo el software SDR# sintonizado a la frecuencia de transición y se obtuvo el siguiente resultado

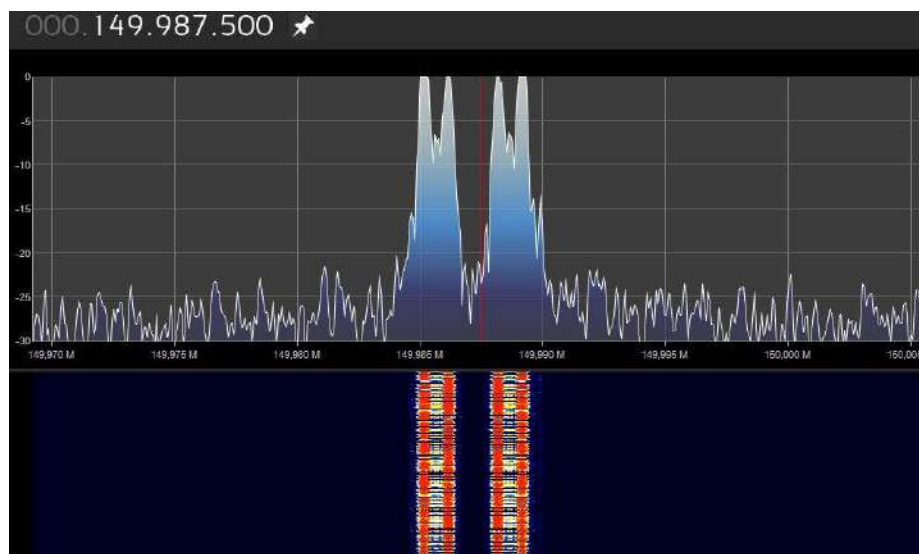


Figura 4.21 2Fsk transmitida

Justo como se esperaba solo aparecen dos señales de frecuencia en el espectro de la señal transmitida, si quiere recibir la información transmitida se puede utilizar el siguiente diagrama de bloques, el cual decodifica la información y se obtiene nuevamente los bits enviados.

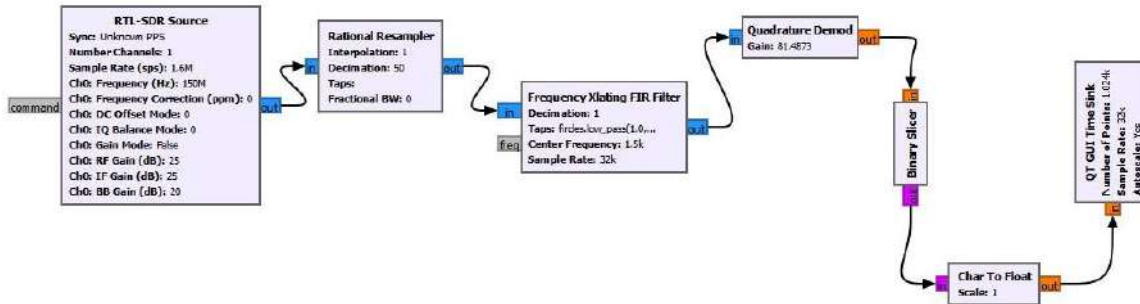


Figura 4.22 Receptor 2FSK con RDS

## 4.2.3 Modulación CFSK con GNU radio

La gran mejora que tiene el CFSK con respecto sobre el FSK es que ya no posee cambios bruscos cada vez que el conmutador se activa, esto se traduce en un mejor ancho de banda. Esto se logra gracias un dispositivo llamado VCO (oscilador controlador por voltaje), con ese aparato la señal que se produce siempre va a estar en fase cuando se haga el cambio y gracias a esto no van a haber picos en los cambios y el ancho de banda no se va a volver teóricamente infinito.

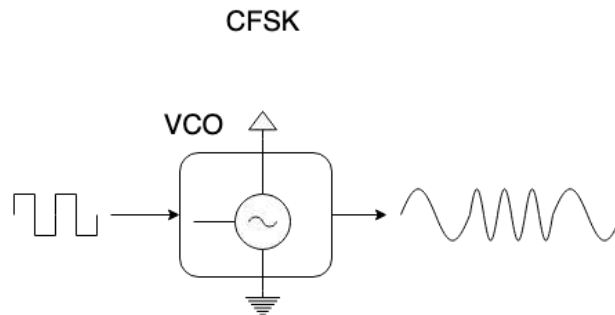


Figura 4.23 CFSK

Similarmente existe el MFSK el cual utiliza el mismo concepto que el CFSK, pero esta vez hay mínimo teórico de cuan diferente deben ser las ondas para que el ancho de banda se reduzca a su mínima expresión, es decir que si se logra que la frecuencia del oscilador se reduzca solo hasta la mitad se tendrá la modulación y de paso una señal con un ancho de banda más angosto, y aun se garantiza que la señal se puede recibir.



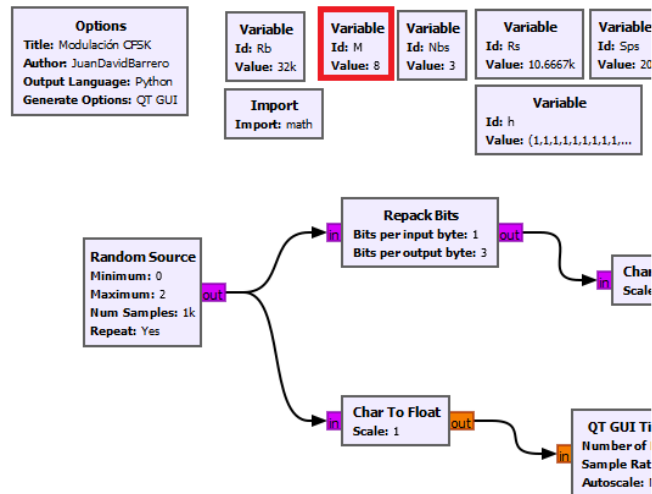


Figura 4.27 Cantidad de bits empaquetados

Al ejecutar el código se obtiene el siguiente resultado.

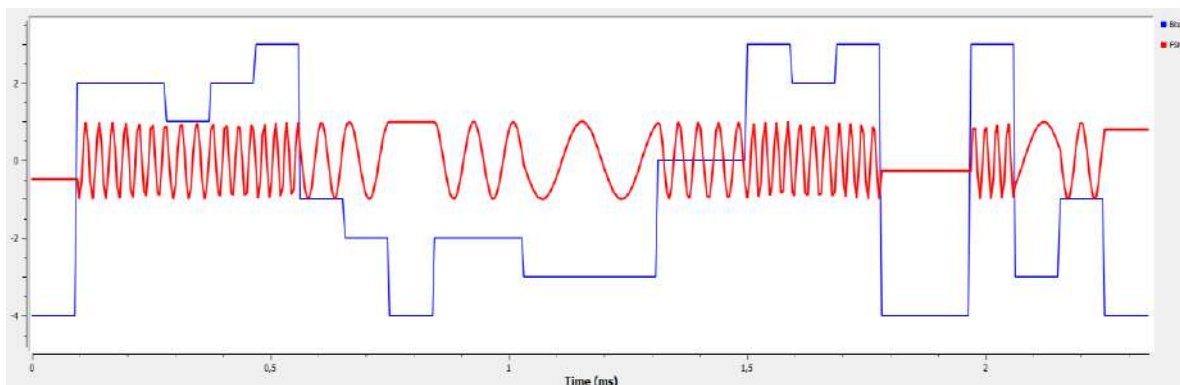


Figura 4.28 Modulación 4-CFSK

## 4.2.4 Transmisión CFSK utilizando RDS

En la sección anterior se vio en la figura 178 que había unos bloques comentados, estos son los necesarios para poder realizar la transmisión de información utilizando este tipo de modulación. Si quiere transmitir basta con descomentar esos bloques utilizando la tecla E (o clic derecho y descomentar bloque) y comentar los siguientes bloques, esto se puede hacer con la tecla D (o clic derecho y comentar bloque).

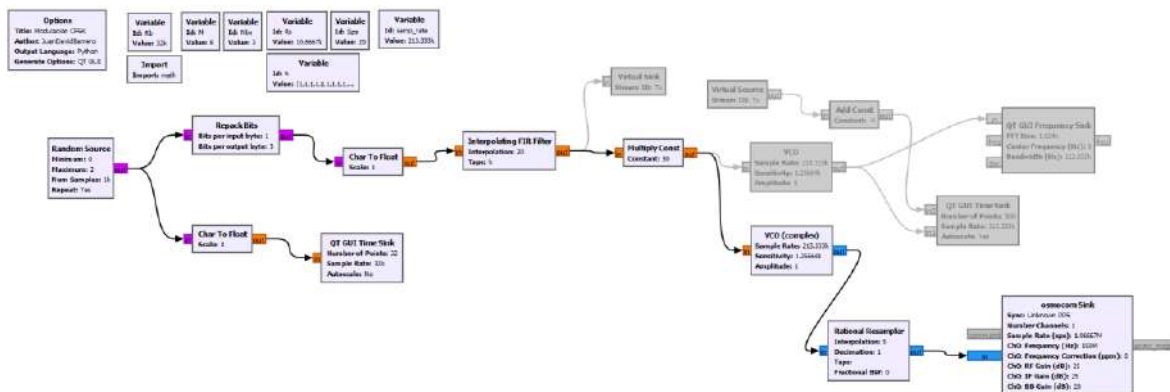


Figura 4.29 Transmisión 4CFSK con RDS

Y justo como se espera la señal recibida tiene 4 señales de frecuencia distintas, esto se puede ver en la siguiente figura.

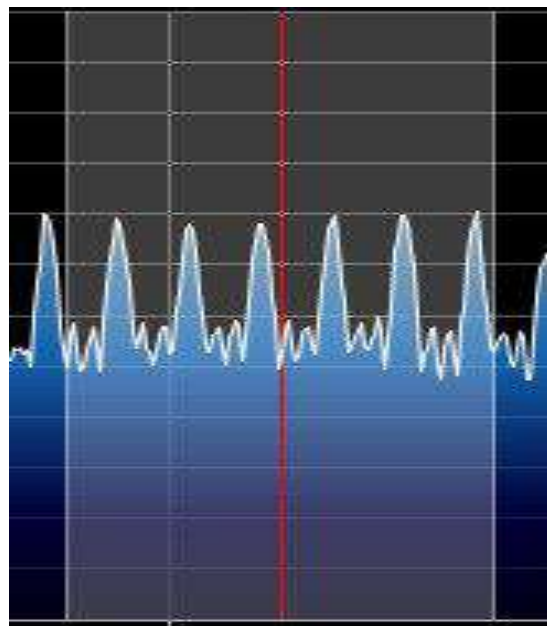


Figura 4.30 recepción 4CFSK

## 4.2.5 Modulación GFSK simulación GNU radio

Los ingenieros de Bluetooth se preguntaron cómo reducir más el ancho de banda lo más posible y dieron con una solución brillante, poniendo altos y bajos en el VCO aun los cambios era demasiados bruscos por lo que decidieron aplicar un filtro Gaussiano a la señal de entrada por lo que los valores ya no irían de 0 a 1 bruscamente si no irían un tiempo más controlador permitiendo que la señal se module gradualmente y al cambiar de manera lenta y controlada el ancho de banda disminuye.

En el siguiente diagrama de bloques se muestra cómo se puede implementar esta solución la hora de construir esta nueva y más eficiente modulación.

#### GFSK

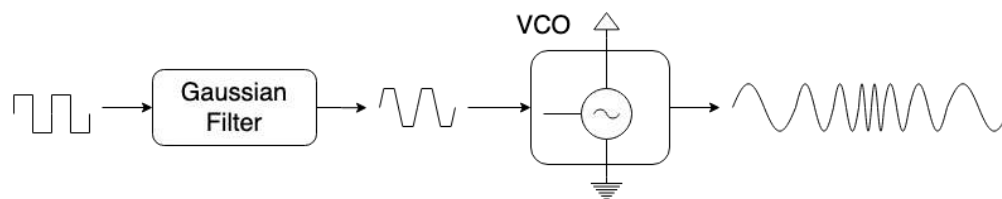
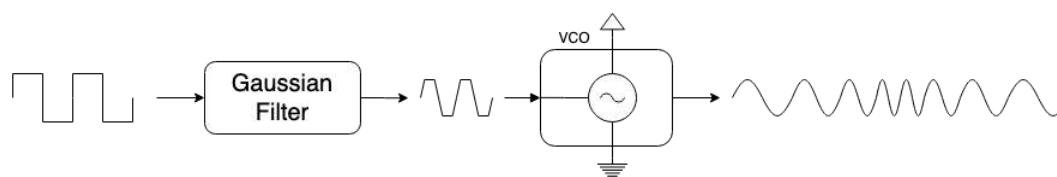


Figura 4.31 GFSK

La combinación de MFSK con GFSK aplicar el filtro Gaussiano a la señal de información para que esta baje lentamente hasta la mitad de la segunda frecuencia lo que reduce aún más el ancho de banda.

#### GMFSK



$$F1 = 2 \cdot F2$$

Figura 4.32 GMFSK

## 4.2.6 Transmisión y recepción GFSK con GNU radio

En GNU radio ya viene instalada por defecto el bloque de modulación GFSK, este cuenta con uno VCO interno por lo que no es necesario adicionar otro a la salida de este, como se observa en la siguiente figura, también cuenta para ajustar el valor de filtro que se le aplicara.

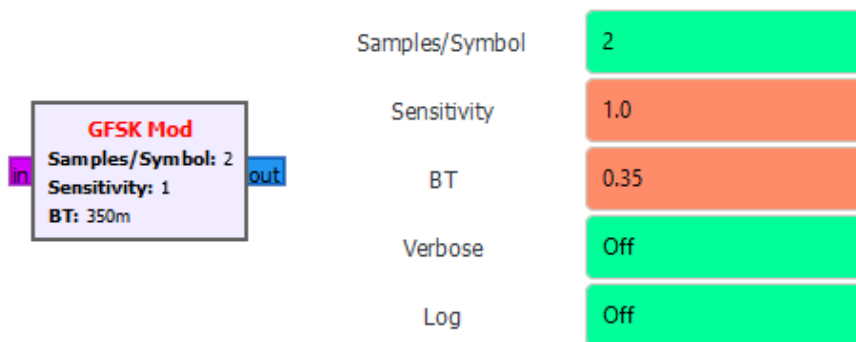


Figura 4.33 GFSK GNU radio bloque

A continuación, se mostrará como ajustar los valores de este para poder realizar la codificación según las especificaciones que se requieran.

**Samples/Symbols:** Este parámetro le indica a bloque cuantas muestras van a venir de la entrada, este debe coincidir con el valor que del tamaño de los bits que se le vaya a dar.

**Sesntivity:** Este es el valor de desviación de frecuencia, del cual se ha hablado anteriormente.

**BT:** Este es el parámetro que se debe cambiar para modificar el tipo de filtro Gaussiano que va a utilizar antes de hacer la modulación. Si este valor es 1 no se generar ningún tipo de filtro y se realizara una modulación FSK normal

**Verbose:** si se activa muestra información acerca del modulador.

**Log:** muestra la información modulada si se activa.

Siguiendo con la sección es momento de poder recuperar la señal y la información que esta contiene, para realizar esto hay que hacer uso del bloque “GFSK demod”, el cual ya se encuentra instalado en la librería de GNU radio. Acá se puede ver como poder configurarlo para decodificar correctamente la data enviada.

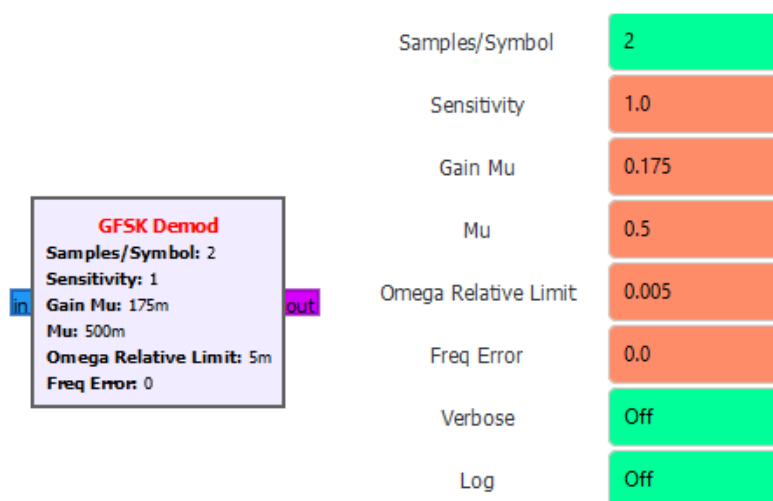


Figura 4.34 Demodulador GFSK GNU radio

**Samples/Symbols:** Este parámetro le indica a bloque cuantas muestras van a venir de la entrada, este debe coincidir con el valor con valor indicado en el modulador, previamente visto.

**Sesntivity:** Este es el valor de desviación de frecuencia, del cual se ha hablado anteriormente.

**Gain Mu:** controla la rata de ajuste de mu

**Mu:** Es el retraso que tiene de la entrada del filtro con la señal de moduladora

**Omega Relative limit:** Determina la máxima variación de omega

**Freq Error:** Este es un factor de error que se espera en la rata de bits.

**Verbose:** si se activa muestra información acerca del modulador.

**Log:** muestra la información modulada si se activa.

Otro bloque importante el cual es importante mencionar el “file source”, en este se le puede cargar todo de tipo de archivos los cuales puede ser enviados, esta manera ya no se tiene la necesidad de transmitir bits aleatorios como información

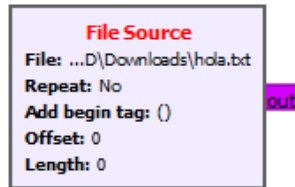


Figura 4.35 Bloque file source

En este bloque los únicos parámetros que vale la pena resaltar son:

Figura 4.36 Parámetros de file source

**File:** indicar donde se encuentra el archivo que se quiera enviar

**Output type:** este si se quiere hacer modulación digital hay que dejarlo como tipo byte.

**Repeat:** si quiere que el mensaje se envíe nuevamente cuando este termine

Para poder realizar el diagrama de bloques de transmisión y recepción de la información, se necesita un bloque para poder decir cuál es la cantidad de bits que se va enviar en cada paquete y cuando este termina, para esto se utiliza el bloque “Stream to tagged Stream”

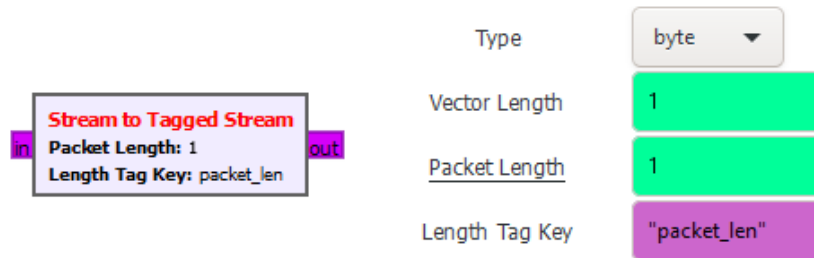


Figura 4.37 Bloque Stream to Tagged Stream

En la figura 189 se puede ver la configuración típica del bloque nombrado anteriormente, con todos los bloques importantes nombrados es momento de realizar el sistema de transmisión de un archivo de texto, esto se puede logara como se ve en el siguiente diagrama de bloques.

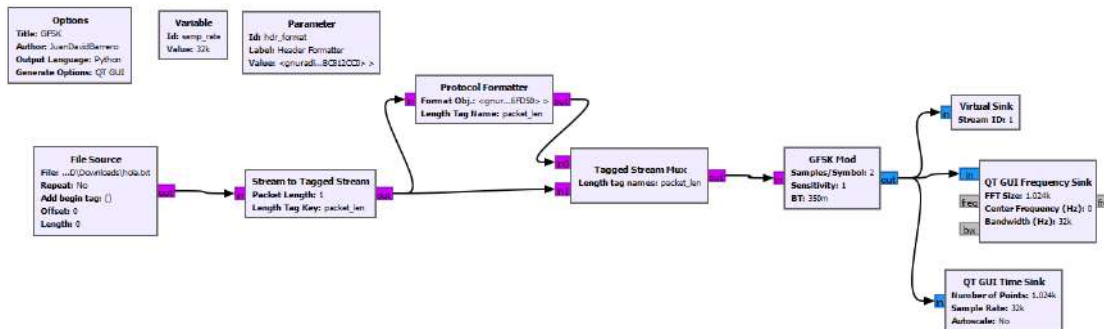


Figura 4.38 Transmisión de datos GFSK

Finalmente queda hacer la recepción de datos, para esto se utiliza el bloque de GFSK demo visto en la figura 186, además de un bloque el cual crear el archivo con la información recibida, este bloque se llama file sink.

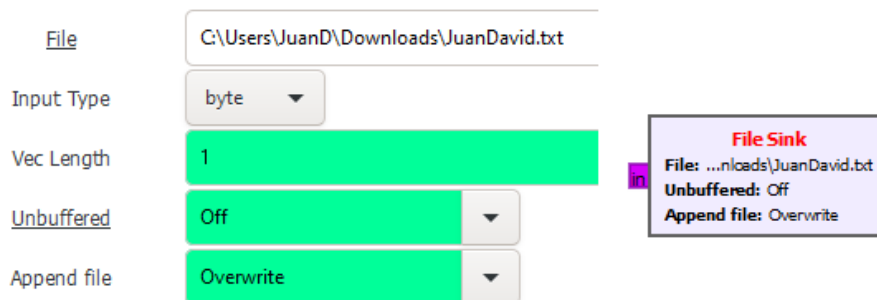


Figura 4.39 File sink

De este bloque es importante resaltar que se debe habilitar la opción de sobrescribir el archivo, y darte la dirección en donde se quiere guardar la información. Con esto presente a continuación se muestra el diagrama de bloques para la recepción de la información.

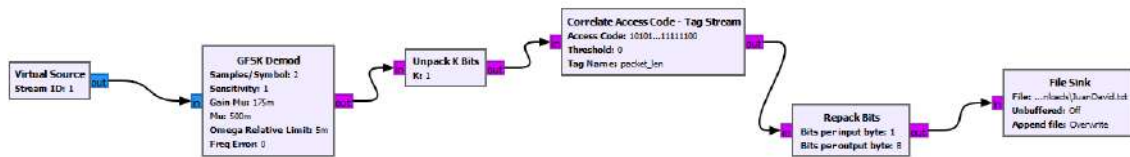


Figura 4.40 Receptor GFSK

Todo el diagrama de bloques se puede ver a continuación

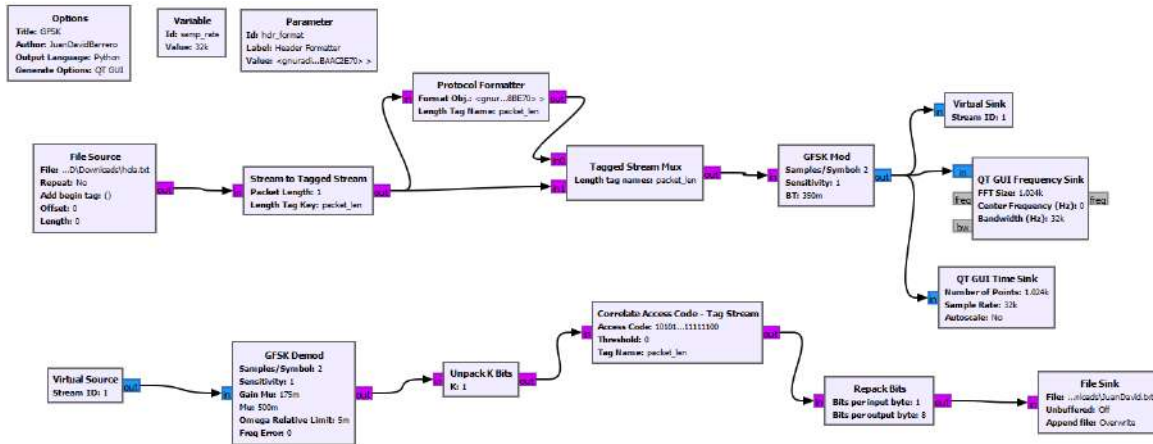


Figura 4.41 Transmisión y Recepción GFSK

## 4.2.7 Tx y Rx con GFSK utilizando RDS

Utilizando la misma estructura se puede construir el diagrama de bloques del transmisor utilizando el RDS, este solo requiere unas ligeras modificaciones para que funcione. Esto se puede ver en la siguiente figura.

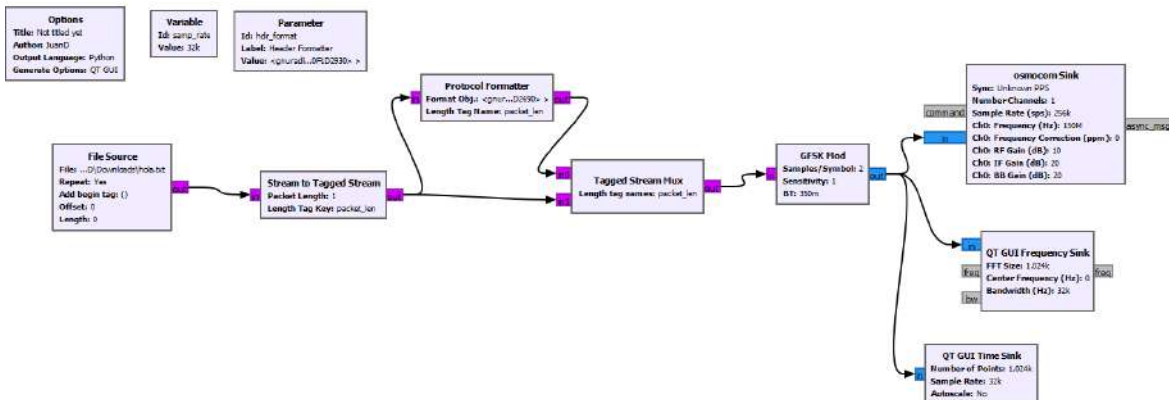


Figura 4.42 Transmisión RDS GFSK

El problema radica al momento de recibir la información, ya que no existe ningún tipo de sincronizador entre las señales envidadas esta no puede ser decodificada correctamente, estos problemas se resolverán en la siguiente sección.

Por ahora la primera solución que se puede dar para poder recibir información, es mostrada en el siguiente diagrama de bloques. Es importante decir que la información si está siendo transmitida pero como no hay ningún sistema de sincronización esta no puede ser reconstruida correctamente.

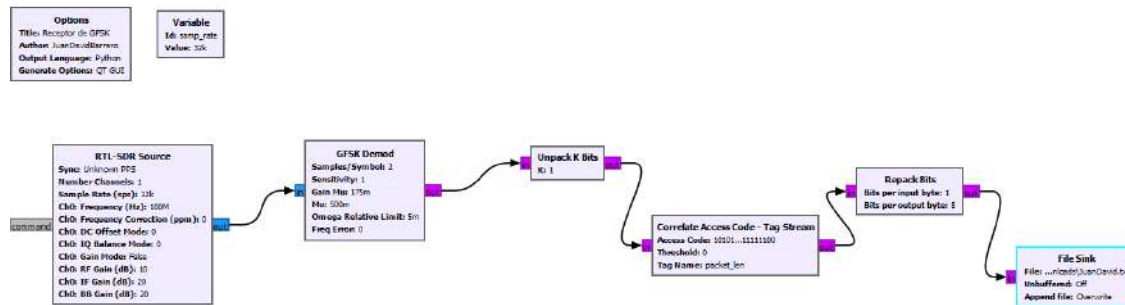


Figura 4.43 Receptor GFSK con RDS

## 4.3 modulación PSK

PSK (Phase-shift keying), es una modulación de fase donde la señal moduladora es digital. En la modulación PSK convencional el valor de la fase de la señal portadora varía dependiendo de valor de la señal de bits que aparezca en la entrada, esto da como resultado una señal modula con la siguiente forma.

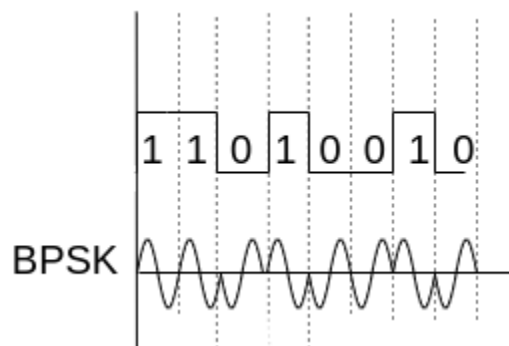


Figura 4.44 PSK demostración

Dependiendo de la velocidad que se necesiten transmitir los datos, se pueden generar variaciones de la modulación PSK. Esta puede agrupar varios bits en un solo paquete para que estos determinen el valor de la fase de salida. De este concepto nace la modulación, BPSK, QPSK, 8PSK y M-PSK.

Este tipo de modulaciones se pueden generar en GNU radio, pero antes de hacer esta parte es importante tener en mente otro concepto y el diagrama de constelaciones, este se puede definir como “es una representación gráfica (llamada I-Q) de los símbolos digitales recibidos en un periodo de tiempo” (Promax, 2017)

En la siguiente imagen se puede observar cómo se presenta un diagrama de constelación, dependiendo de la fase que se tenga la señal de salida.

Figura 11 Tomada de (Introducción a los sistemas de comunicación, 2000)

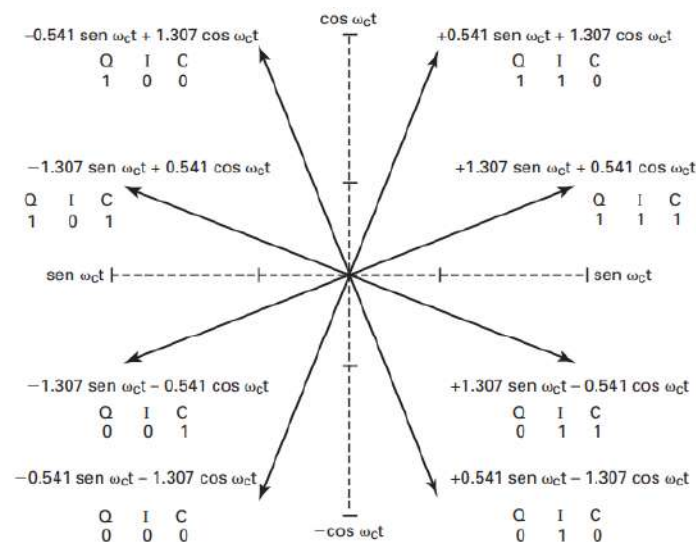


Figura 4.45 Diagrama de constelación modulación 16-PSK

### 4.3.1 2-PSK simulación con GNU radio

Antes de realizar la simulación de transmisión de una señal utilizando la modulación 2-PSK o BPSK, esta sección mostrara alguno de los bloques que son necesarios, primero es importante ver como poder generar la señal.

El primero es importante para ver es el diagrama de constelación como se puede ver en la siguiente figura.

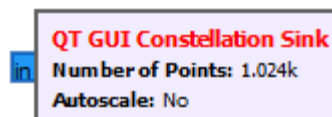


Figura 4.46 Diagrama de constelación

El diagrama de constelación esperado en un sistema BPSK se ve a continuación, Este se puede interpretar como dependiendo el valor de bit, la señal tiene una fase de 0 o de 180 grados.

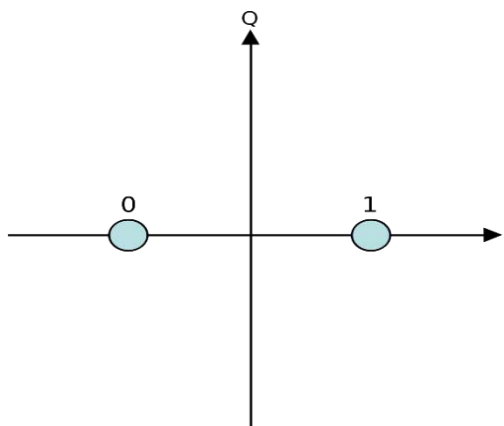


Figura 4.47 BPSK constelación

El siguiente diagrama de bloques muestra cómo construir una modulación BPSK, y para verificar que se está haciendo correcta se ha puesto el bloque de “Constellation sink”, de esta forma se podrá observar si la fase está cambiando.

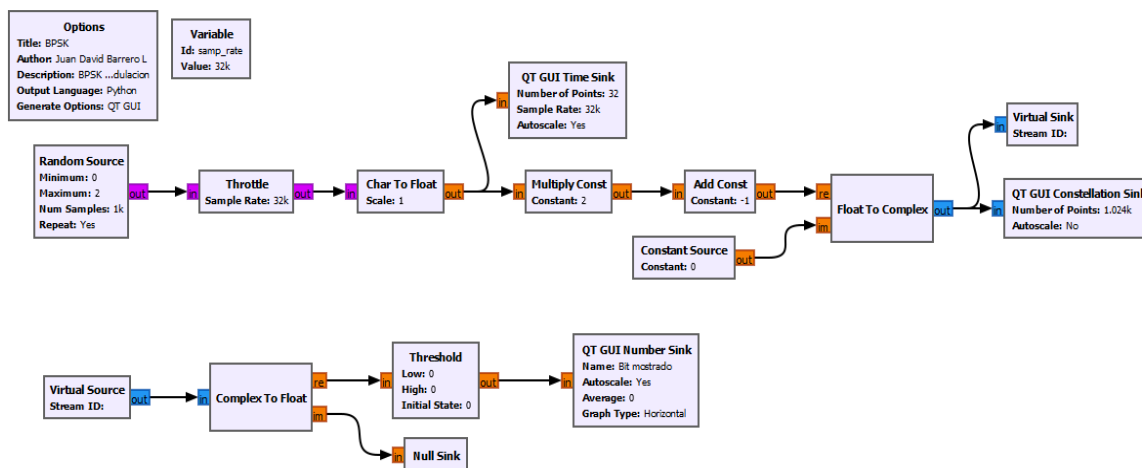


Figura 4.48 BPSK GNU radio

Al ejecutar el siguiente diagrama de bloques, se observa el siguiente resultado. Tal como se espera aparecen los valores esperado en el diagrama de constelación.

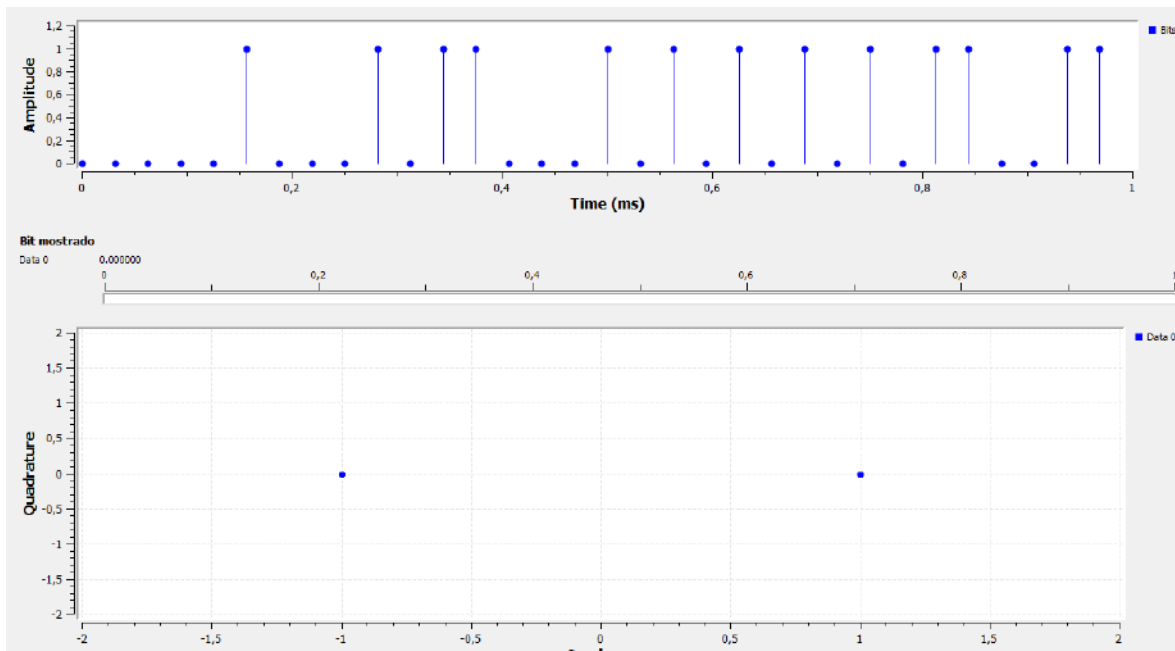


Figura 4.49 Constelación BPSK GNU radio

### 4.3.2 4-PSK simulación con GNU radio

Utilizando GNU radio se construirá, el sistema para genera una señal QPSK o 4-PSK, este tipo de modulación toma 2 bits de la señal de entrada de esta manera se puede obtener la siguiente tabla de la verdad para obtener las 4 fases distintas.

| Bit 0 | Bit 1 | Fase |
|-------|-------|------|
| 0     | 0     | 135  |
| 0     | 1     | 225  |
| 1     | 0     | 45   |
| 1     | 1     | 315  |

Tabla 8 Tabla de verdad QPSK

Con la tabla de verdad se pude generar el diagrama de constelaciones, este se puede ver a continuación.

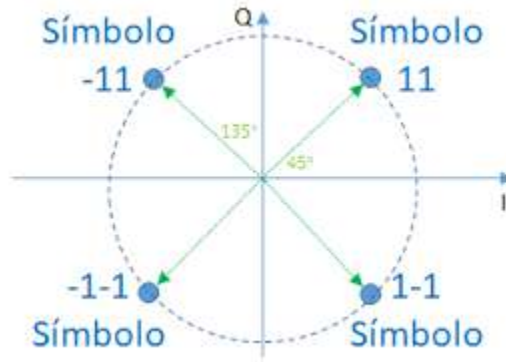


Figura 4.50 Constelación QPSK

Para realizar la modulación QPSK en GNU radio, se utiliza haciendo el siguiente diagrama de bloque, una adición importante que se le agrega es un bloque llamada "Noise Gaussian Source", este adiciona ruido a la señal, lo cual hace tener una sistema más real antes de pasar a la transmisión con los RDS.

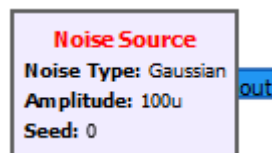


Figura 4.51 Noise Source

La única precaución importante a tener en cuenta al utilizar este bloque es que un valor grande en el parámetro de la amplitud puede resultar en la pérdida total de la información o modulación que se quiera hacer.

El siguiente diagrama de bloques muestra cómo se construye la modulación QPSK, y mostrar el diagrama de constelación esperada, además de mostrar como el ruido Gaussiano puede afectar la señal esperada.

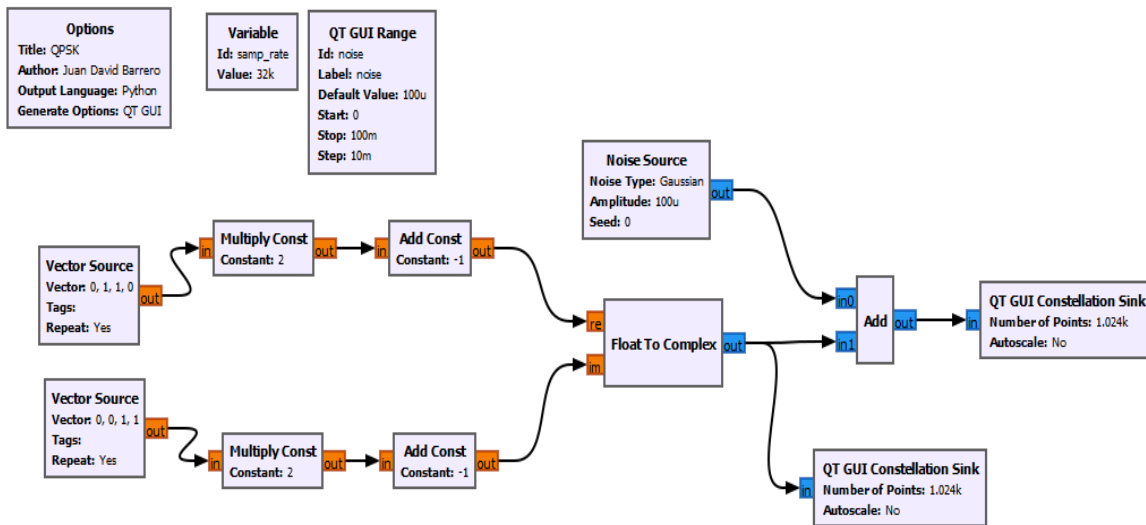


Figura 4.52 QPSK GNU radio

Al ejecutar el código se observa el diagrama de constelación esperado y visto en la figura 203, además cuenta con una barra deslizadora para ajustar el valor de amplitud de la señal de ruido, y observar el comportamiento que este tiene sobre los puntos de la constelación.

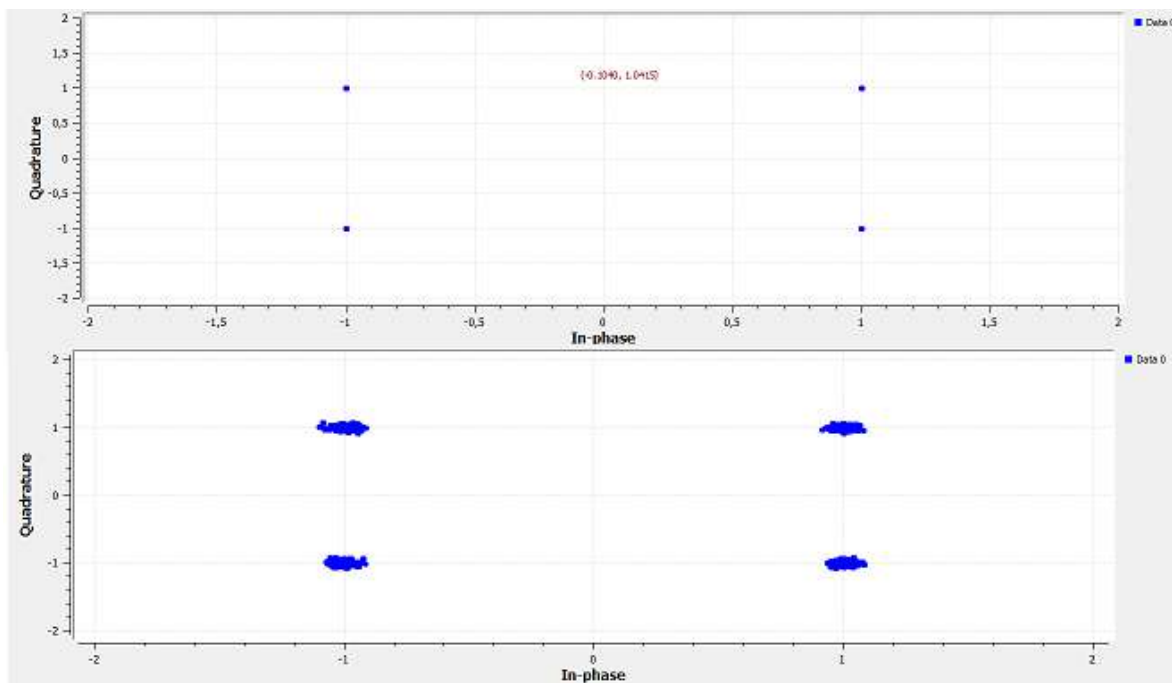


Figura 4.53 QPSK constelación

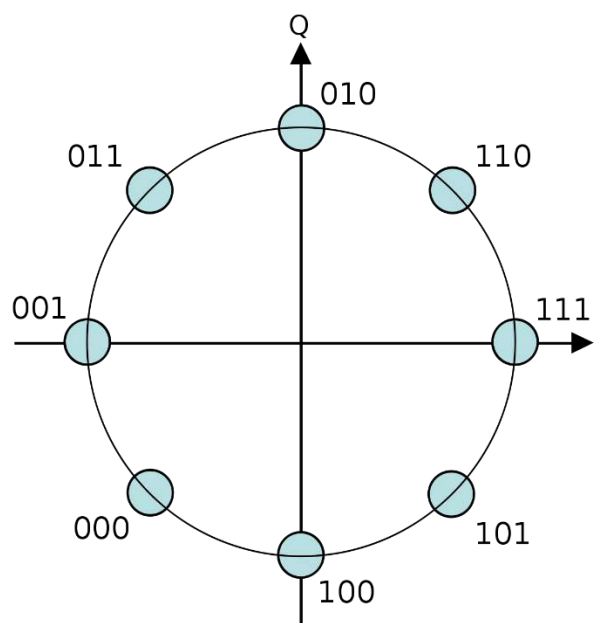
### 4.3.3 8-PSK GNU radio

En este tipo de modulación se tiene ocho fases distintas eso se puede logara agrupando en paquetes de 3 bits, en la siguiente tabla de verdad se puede ver el valor necesario para lograr separar correctamente los bits y realizar la modulación.

| Bit 0 | Bit 1 | Bit 2 | Fase   |
|-------|-------|-------|--------|
| 0     | 0     | 0     | -112.5 |
| 0     | 0     | 1     | -157.5 |
| 0     | 1     | 0     | -67.5  |
| 0     | 1     | 1     | -22.5  |
| 1     | 0     | 0     | 112.5  |
| 1     | 0     | 1     | 157.7  |
| 1     | 1     | 0     | 67.5   |
| 1     | 1     | 1     | 22.5   |

*Tabla 9 8PSK tabla de verdad*

Con la tabla de verdad se puede generar el diagrama de constelación de esta modulación, el cual debe tener la siguiente forma. Algo importante para tener en cuenta es que utiliza el método gray para seleccionar las fases, esto facilita a la hora de realizar la demodulación y recuperar la información



*Figura 4.54 Constelación 8PSK*

Para realizar esto se genera el código de la señal envolvente, esto se puede lograr con el bloque Python block, en el cual se puede escribir código en Python para procesar la señal que se desee, esto es muy conveniente para poder trabajar con la señal de la manera que se necesite.

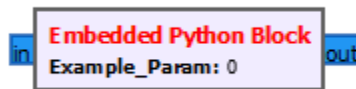


Figura 4.55 Python block GNU radio

Una vez en pantalla, se le hace doble clic y se debe escoger el editor de código que se quiera tal y como se ve en la figura.

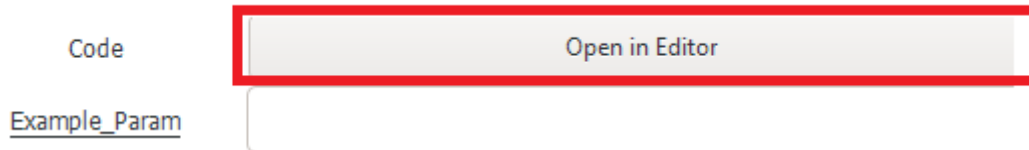


Figura 4.56 escoger el editor de código

Al abrir el editor escogido se mostrara la siguiente plantilla sobre este se debe escribir el nuevo código que se quiera generar

```

1  """
2  Embedded Python Blocks:
3
4  Each time this file is saved, GRC will instantiate the first class it finds
5  to get ports and parameters of your block. The arguments to __init__ will
6  be the parameters. All of them are required to have default values!
7  """
8
9  import numpy as np
10 from gnuradio import gr
11
12
13 class blk(gr.sync_block): # other base classes are basic_block, decim_block, interp_block
14     """Embedded Python Block example - a simple multiply const"""
15
16     def __init__(self, example_param=1.0): # only default arguments here
17         """arguments to this function show up as parameters in GRC"""
18         gr.sync_block.__init__(
19             self,
20             name='Embedded Python Block', # will show up in GRC
21             in_sig=[np.complex64],
22             out_sig=[np.complex64]
23         )
24         # if an attribute with the same name as a parameter is found,
25         # a callback is registered (properties work, too).
26         self.example_param = example_param
27
28     def work(self, input_items, output_items):
29         """example: multiply with constant"""
30         output_items[0][:] = input_items[0] * self.example_param
31         return len(output_items[0])
32

```

Figura 4.57 Plantilla código GNU radio

Para generar la función envolvente se usa el siguiente código.

```
class blk(gr.sync_block): # other base classes are basic_block, decim_block, interp_block
    """Embedded Python Block example - a simple multiply const"""

    def __init__(self, example_param=1.0): # only default arguments here
        """arguments to this function show up as parameters in GRC"""
        gr.sync_block.__init__(
            self,
            name='M-PSK', # will show up in GRC
            in_sig=[np.float32,np.float32],
            out_sig=[np.complex64]
        )
        # if an attribute with the same name as a parameter is found,
        # a callback is registered (properties work, too).
        self.example_param = example_param

    def work(self, input_items, output_items):
        Q = input_items[0]
        A = input_items[1]
        PSK = output_items[0]
        PSK[:]=A*np.exp(1*j*Q)
        return len(PSK)
```

Figura 4.58 código de Python

Una vez puesto se puede ver el nuevo bloque que se ha generado.

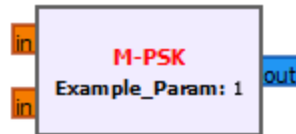


Figura 4.59 MPSK

El cual tiene dos entradas para la señal de bits y la otra para un constante que puede ser colocada en 1. Con esto ya se tiene lo necesario para montar el diagrama de bloques de la modulación 8PSK,

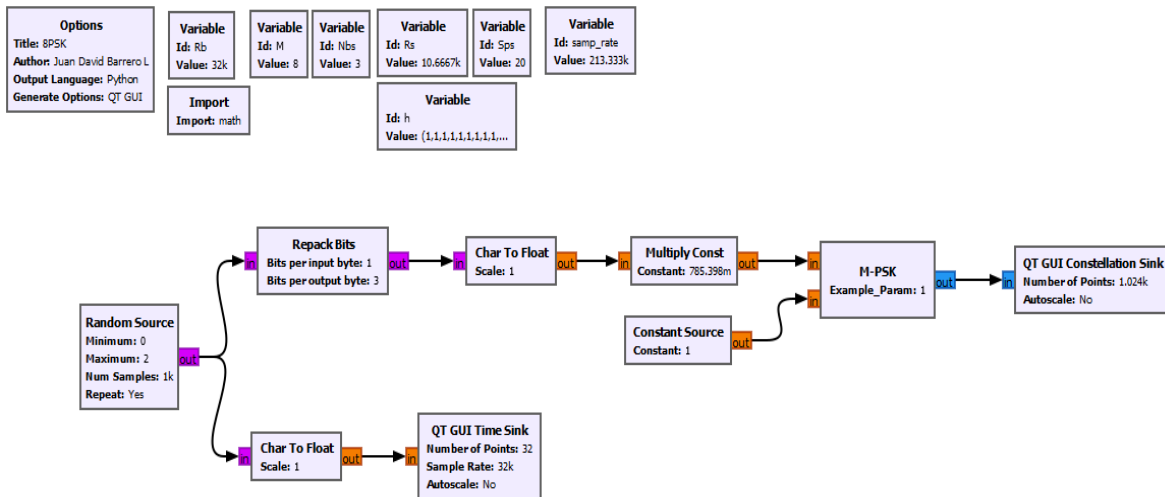


Figura 4.60 8PSK

Al ejecutar el código se puede observar el diagrama de constelación esperado para una modulación 8PSK.

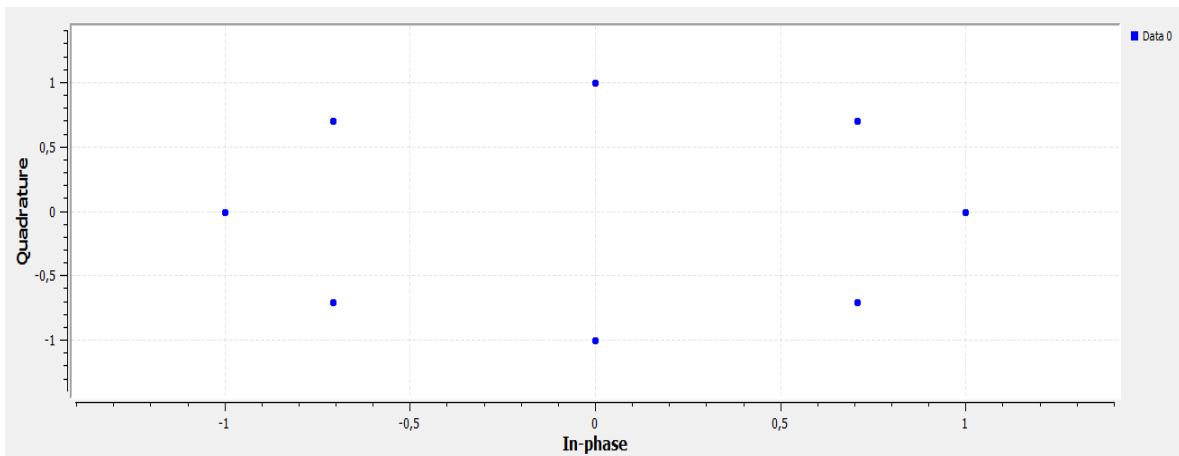


Figura 4.61 Constelación 8PSK

#### 4.3.4 Problemas con la comunicación digital

En telecomunicación, la interferencia entre símbolos (ISI) es una forma de distorsión de una señal en la cual un símbolo interfiere con símbolos posteriores. Es un fenómeno no deseado ya que los símbolos anteriores tienen un efecto similar al del ruido, lo que hace que la comunicación sea menos fiable.

Para poder combatir el problema de ISI se puede utilizar los siguiente métodos:

- Utilizar la conformación de pulso es el proceso de cambiar la forma de onda de los pulsos transmitidos. El método más efectivo para realizar esto es utilizando la transformación de la señal utilizando el coseno alzado.

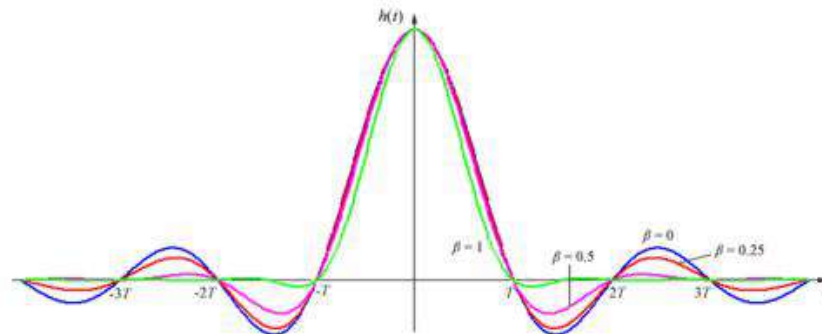


Figura 4.62 Coseno alzado

- Utilizar un ecualizador al momento de recibir la señal.

Ambos métodos pueden ser implementados en GNU radio con la intención de aprender cómo evitar la interferencia de símbolos al transmitir una señal digital, para poder ver esto se puede armar el siguiente diagrama de bloques.

Ya habiendo visto cómo generar las modulaciones con GNU radio, es momento de mostrar otro bloque que está en las librerías que permite hacer todo tipo de señal PSK en un solo bloque.

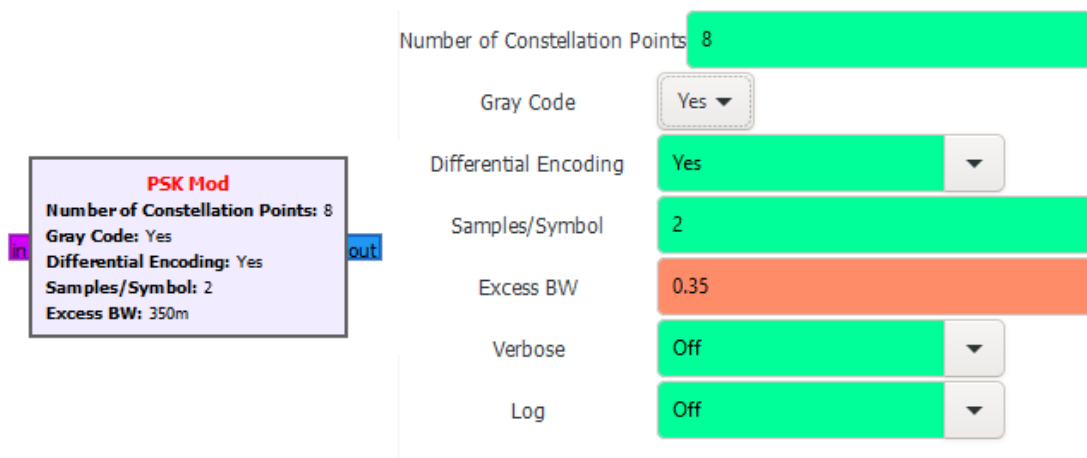


Figura 4.63 PSK mod

Para configurar este bloque hay que modificar los siguientes parámetros.

**Number of constellation points:** En ese se puede poner el tipo de modulación que se quiera 2, 4, 8 o 16

**Gray code:** Este es el sistema de numeración que organiza los símbolos, es recomendable dejar este activo

**Samples/symbol:** es el número de símbolos que se va tener en cada paquete

**Excess BW:** Es el ancho de banda permitido para señal, este se puede hacer tan grande como sea requerido.

Para poder simular la interferencia de símbolos se usa el bloque “Channel model”, el cual simula todos los cambios que pueden ocurrir al transmitir una señal, tal y como se puede ver en la siguiente figura.

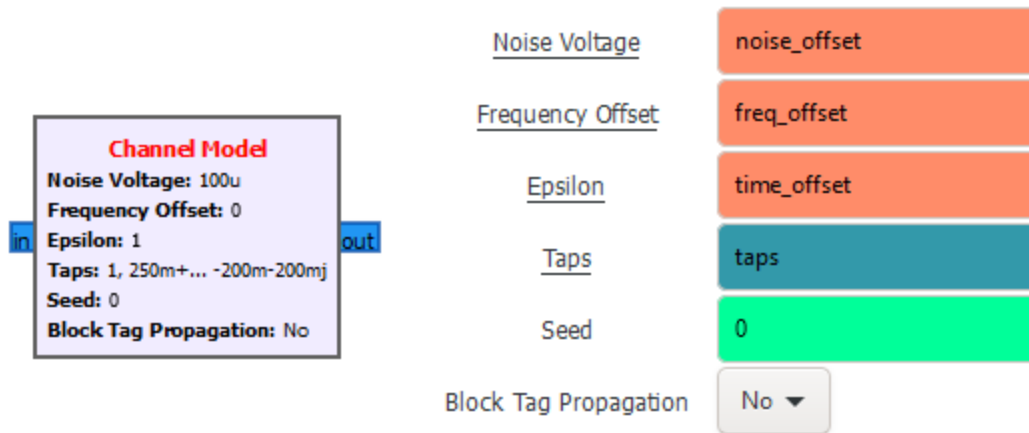


Figura 4.64 Channel model GNU radio

Con lo visto se puede ver el efecto que tiene diferentes factores como el ruido, la desviación de frecuencia, la interferencia de símbolos y la desviación del tiempo. Para esto se arma el siguiente diagrama de bloques.

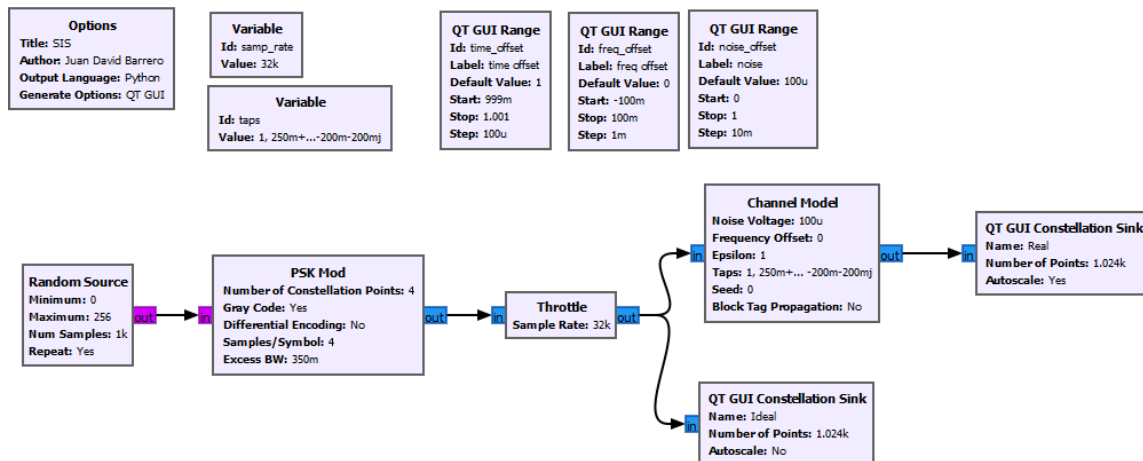


Figura 4.65 SIS GNU radio

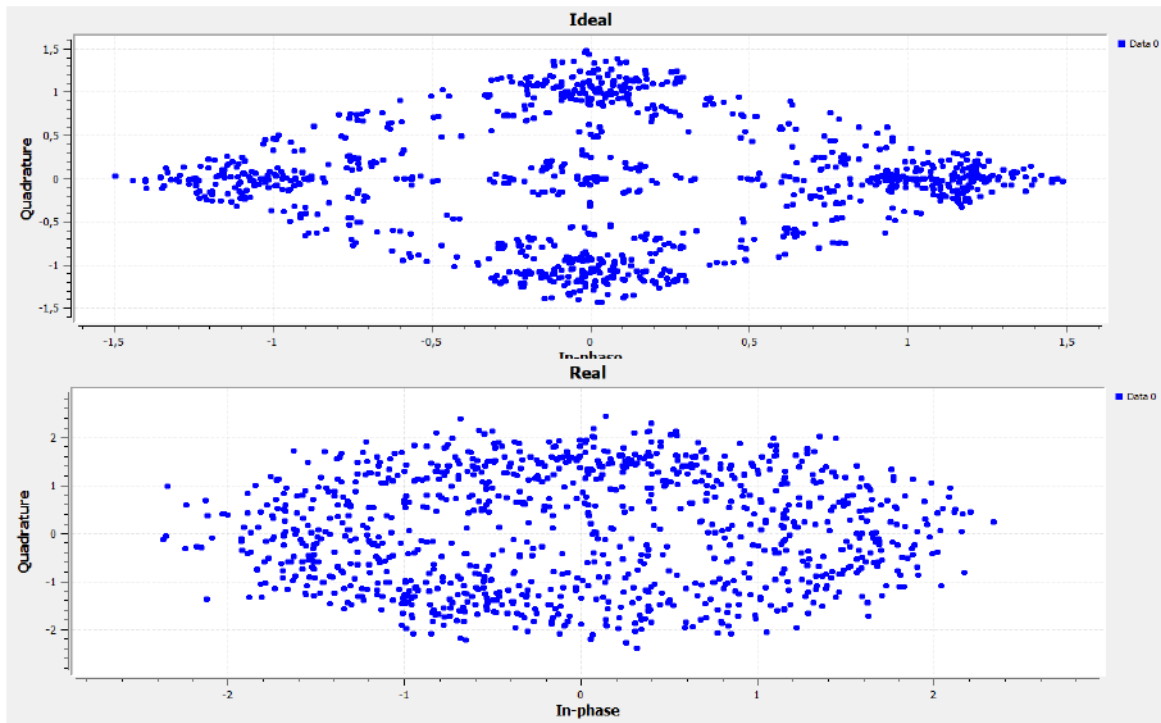


Figura 4.66 Señal distorsionada por la desviación de frecuencia

Para poder recuperar la señal se puede agregar el siguiente diagrama de bloque el cual contiene un coseno alzado, interno el cual es justo lo que se necesitaba para realizar la conformación de pulso.

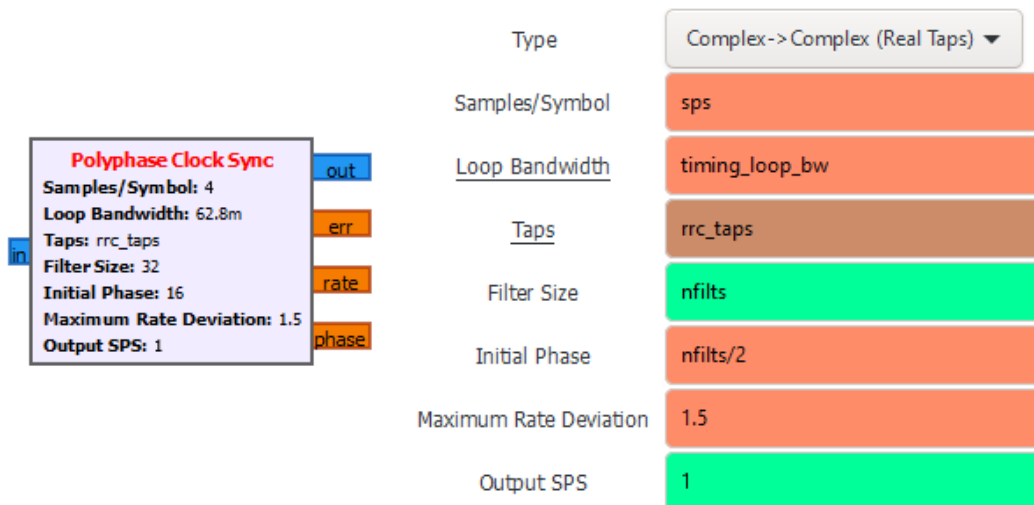


Figura 4.67 polyphase clock

**Samples/symbols:** Es el valor que se ajustó en la modulación PSK

**Loop bandwidth:** es valor es normalmente fijo y se puede colocar el siguiente valor  $2\pi/100$

**Taps:** Aca se coloca el valor coseno alzado para el cual se quiere recuperar la señal,

El siguiente diagrama de bloques muestra cómo unir todos los bloques para poder codificar la señal, y recuperarla a pesar de interferencia de símbolos

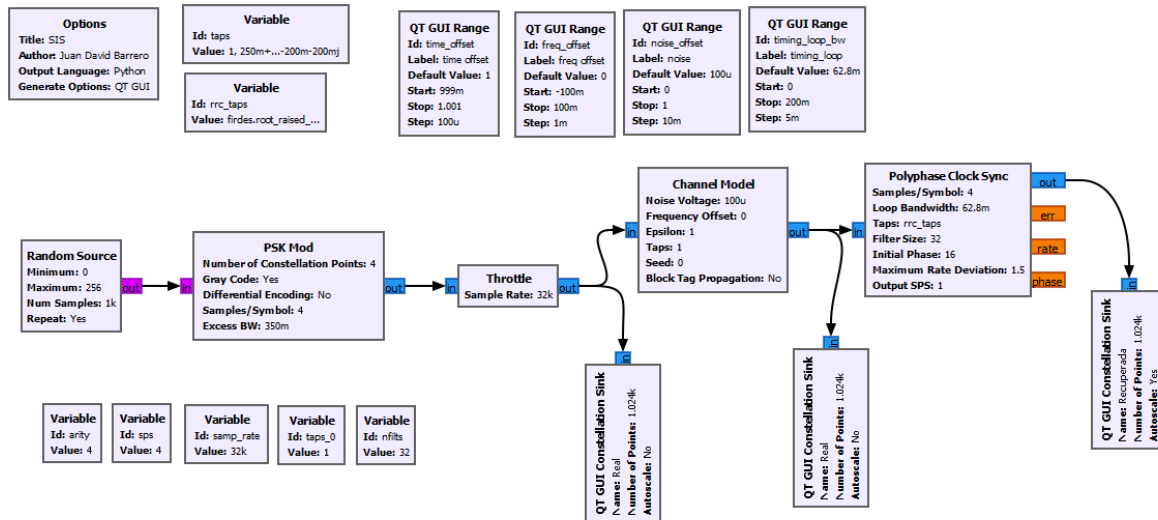


Figura 4.68 Recuperando la señal

Al ejecutar el código se obtiene el siguiente resultado

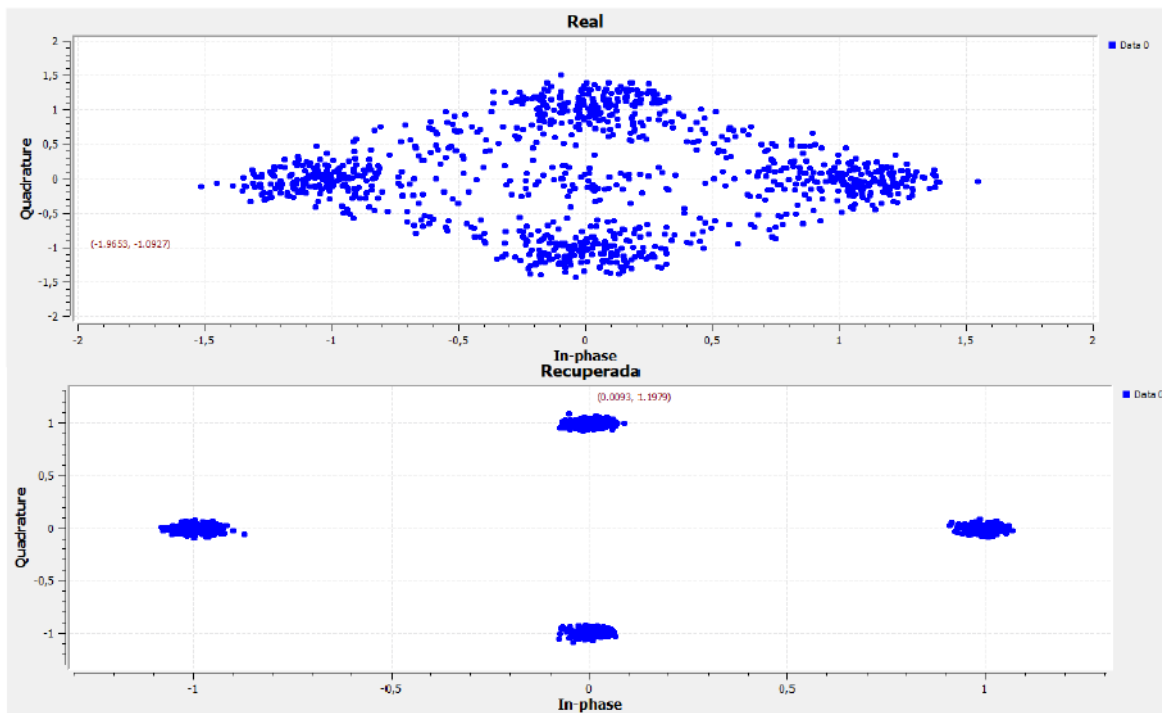


Figura 4.69 Señal recuperada

La segunda forma de evitar la interferencia de símbolos es utilizando un ecualizador para esto se utiliza el siguiente bloque

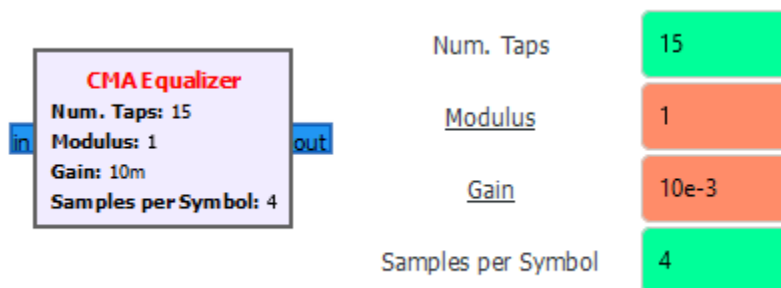


Figura 4.70 Ecualizador

Este ecualizador tiene un coseno alzado por esta razón no es necesario colocar uno a la salida, los parámetros que es importante cambiar es la ganancia de este, si es muy grande la señal no se puede recuperar, el siguiente en la muestras por símbolo las cuales tienen que coincidir con la puestas en el modulador PSK del inicio.

En el siguiente diagrama de bloque se puede observar utilizar todo en conjunto para estudiar cómo evitar la interferencia de símbolos.

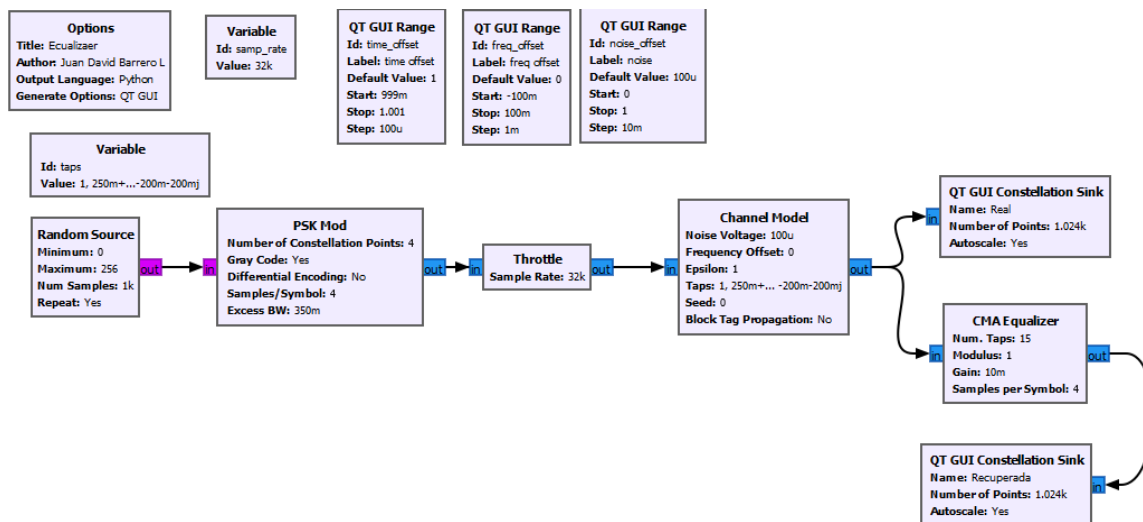


Figura 4.71 Ecualizador para evitar SIS

Al ejecutar el código se observa cómo funciona.

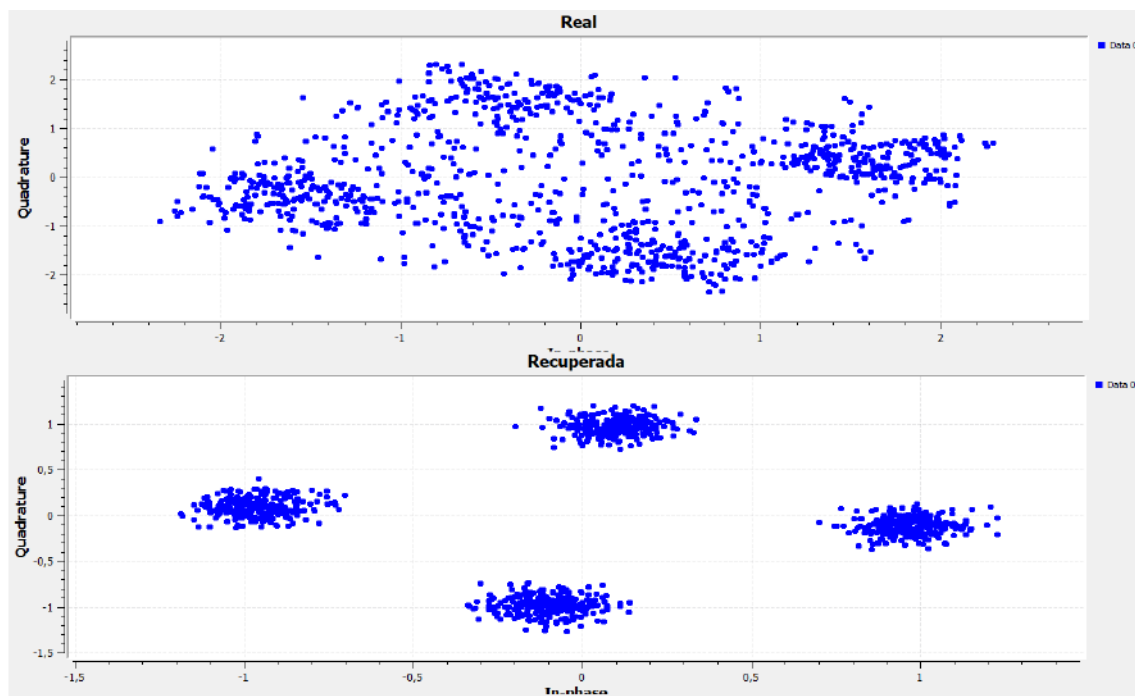


Figura 4.72 Evitando SIS con ecualizador

Aun se puede ver que la señal no es recupera correctamente para esto se puede usar ambos métodos para de esta manera dar una solución definía al problema de interferencia por símbolos.

Para hacer esto se realizó el siguiente diagrama de bloques, el cual cuenta con todo lo necesario para recuperar le señal sin importar que tipo de modulación PSK se escoja, es esta sección se utilizó 8PSK

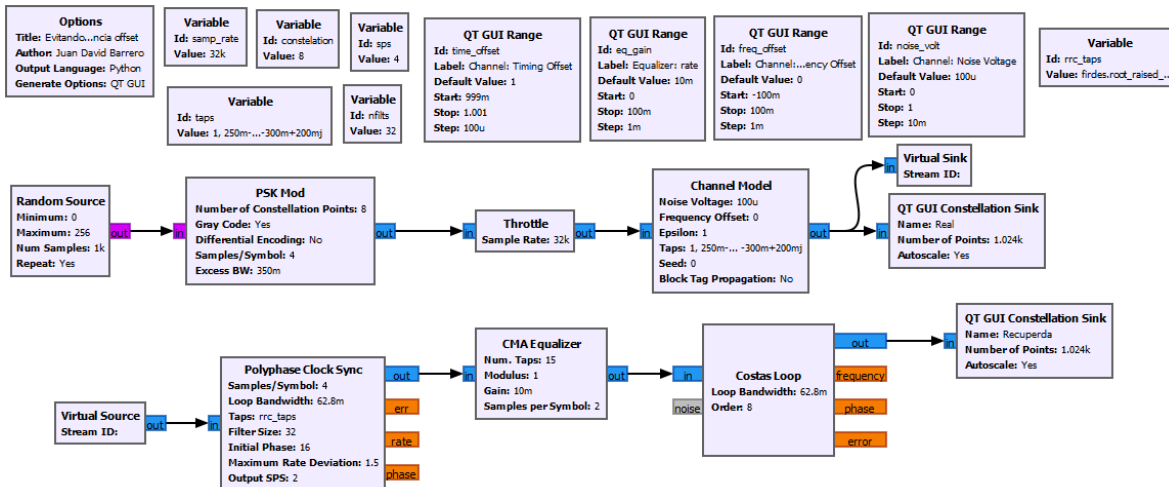


Figura 4.73 SIS

Al ejecutar el código se observa que el resultado es el esperado

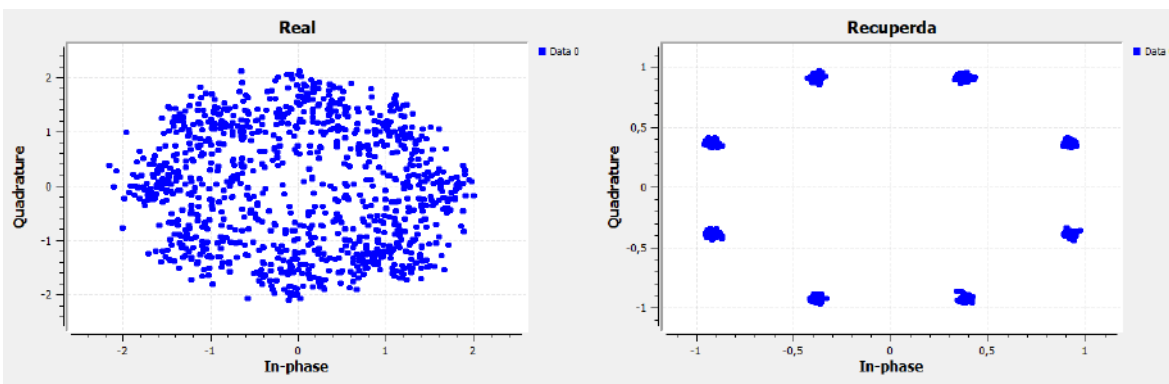


Figura 4.74 Eliminado problema de frecuencia offset

### 4.3.5 Transmisión y recepción 8-PSK GNU radio

Con los sistemas visto anteriormente es ya se puede hacer la transición poniendo a prueba la teoría, pero antes es necesario ver cómo se puede hacer la demodulación de una señal 8-PSK (la misma teoría se puede aplicar para cualquier variación).

Lo primero es determinar la el valor de las variables Q e I para poder hacer una modulación 8-PSK, para esto se tiene la siguientes ecuaciones, para poder determinar los ángulos

$$\text{Angulo} = \frac{360}{2^M}$$

Ecuación 32 Angulo

Donde M es el número de bits en el paquete (en el caso de 8PSK M = 3)

Este es el valor del ángulo que hay entre las fases, ahora es momento de determinar la magnitud y el ángulo que tiene cada fase con respecto al eje para así poder dibujar el diagrama de fase y el de constelación.

$$2 = (Q^2 + I^2)^2$$

$$\phi = \tan^{-1}\left(\frac{I}{Q}\right)$$

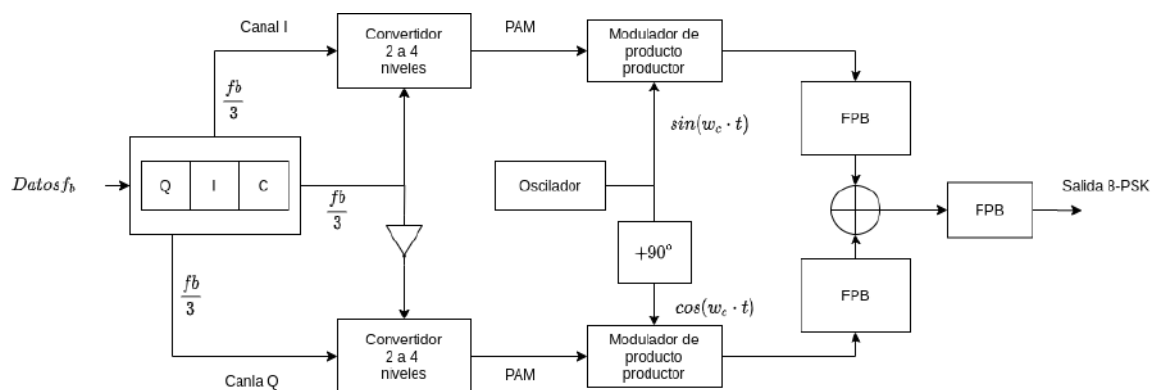
*Ecuación 33 parámetro de modulación*

Es importante recalcar que I o Q determinan la polaridad de la señal analógica mientras que C determinar el valor de la tensión de la salida, es decir que C es quien dice si la salida vale 0.541[v] o 1.307[v] como se ve en la siguiente tabla.

| Q | I | C | Angulo  |
|---|---|---|---------|
| 0 | 0 | 0 | -112,49 |
| 0 | 0 | 1 | -157,57 |
| 0 | 1 | 0 | -67,51  |
| 0 | 1 | 1 | -22,49  |
| 1 | 0 | 0 | 112,19  |
| 1 | 0 | 1 | 157,57  |
| 1 | 1 | 0 | 67,51   |
| 1 | 1 | 1 | 22,49   |

*Tabla 10 tabla de verdad 8-PSK*

Este resultado se puede expresar mejor con el siguiente diagrama de bloques



*Figura 4.75 Bloque decodificador PSK*

Por suerte en GNU radio este implementado bloque que hace la decodificación de la modulación PSK

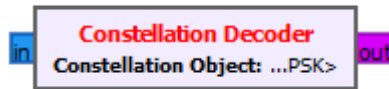


Figura 4.76 De modulador PSK

El único parámetro que se le debe colocar es el tipo modulación que se está usando ya se BPSK QPSK 8-PSK. Finalmente solo queda mostrar como quedo el diagrama de bloques del sistema transmisor para una modulación 8-PSK.

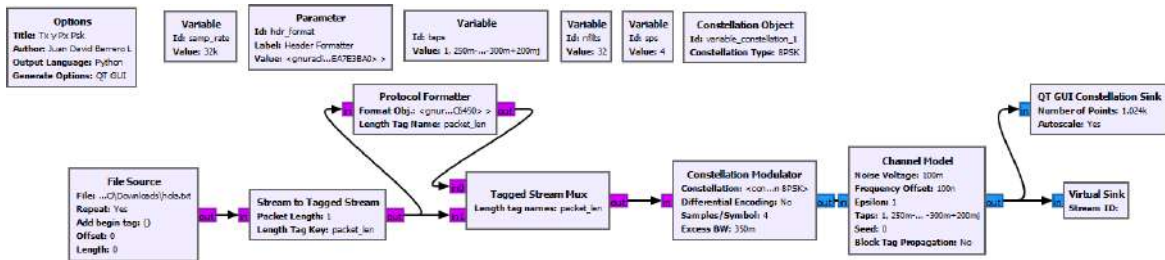


Figura 4.77 Transmisión 8-PSK con GNU radio

Y para el sistema receptor

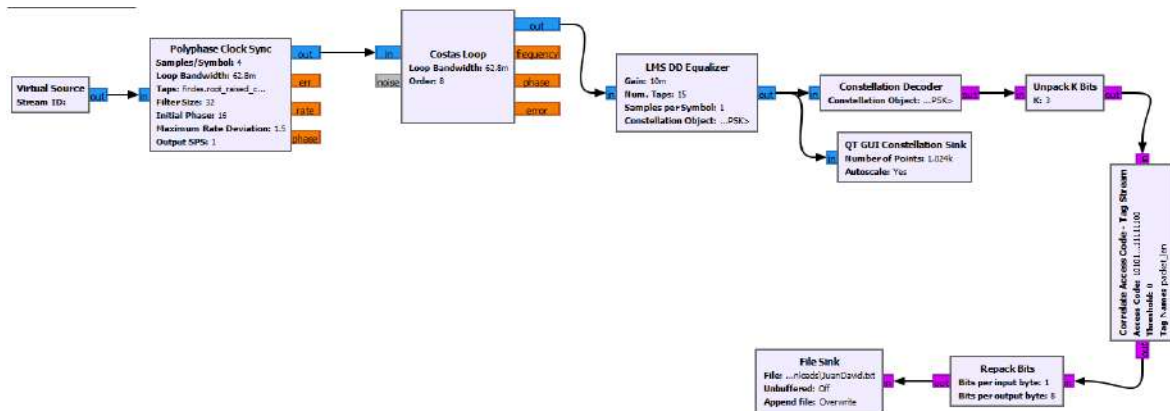
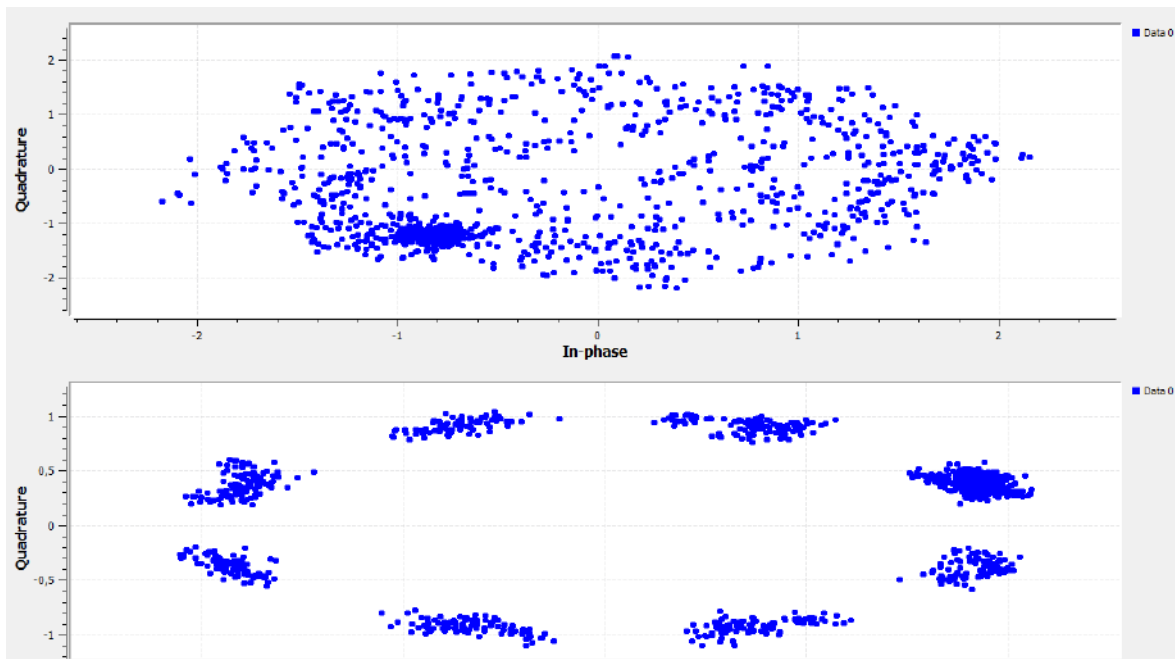


Figura 4.78 Receptor modulación 8-PSK

Al ejecutar el código se ve como la señal poder ser reconstruida correctamente, y la información llega sin ningún problema



### 4.3.6 Rx y Tx PSK con RDS

Haciendo los siguientes cambios en el diagrama de bloques mostrado anteriormente, se puede realizar la transmisión utilizando un RDS. Hay que tener cuenta que entre más muestre se agreguen en el Osmocom sink más ancho de banda la señal ocupara.

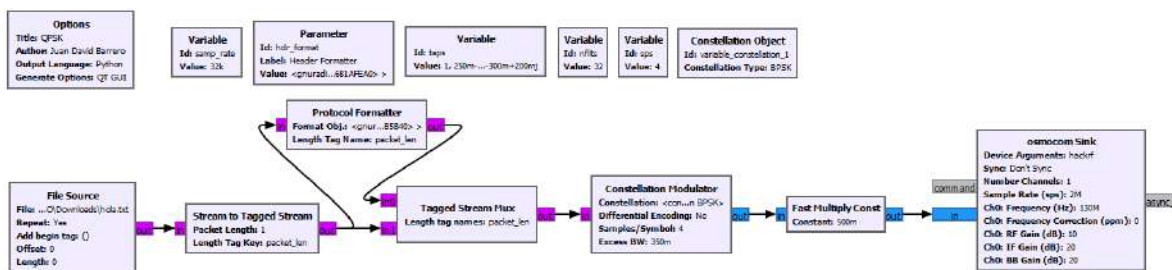


Figura 4.80 Transmisor PSK RDS

Este está hecho de tal forma que se puede escoger el tipo de modulación que se quiere, para realizar este cambio solo hay que escoger en este bloque, tal y como se muestra en la siguiente imagen.



## 5. Televisión digital con RDS

La televisión digital es una nueva “tecnología” que se tomaba poco a poco el mundo ya que tiene ciertas ventajas frente a la analógica, en el mundo digital ahora la imagen y el sonido se transmite poco a poco en paquetes de información discreta. Y un equipo programado decodifica esta información y la separa para posteriormente implementarla en la pantalla y los bafles de un televisor.

En la actualidad ya varias tecnologías que permiten acceder a este servicio como lo son

- Ondas terrestres (TDT)
- Cable
- Satélite
- ADSL
- Dispositivos móviles

Entre las ventajas se puede decir que tiene más canales, la calidad de la imagen y el sonido es superior y ya que la señal ahora es totalmente digital se puede proporcionar muchos más servicios agregando solo un poco más de programación, Esta sección se mostrará como se puede implementar un sistema de transmisión y recepción de televisión digital.

Es importante aclarar que para esta práctica es necesario el uso de **dos HackRf one** ya que el ancho de banda estándar para realizar un enlace de televisión digital este entre 6 MHz y 8 MHz, el RTL\_SDR solo puede recibir un ancho de banda máximo de hasta 3 MHz.

### 5.1 Transmisión de televisión digital ATSC con SDR

Aclarado que se debe utilizar dos hackRF para realizar la transmisión y recepción de la imagen y el audio es momento de ver como se puede lograr esto utilizando el software de GNU radio, por suerte este ya cuenta con los bloques necesarios para que hacen todo el trabajo pesado.

Así que en esta sección se mostrarán los pasos que se debe seguir para poder transmitir un video y su audio, lo primero es tener un video en el formato correcto, este puede ser de cualquier resolución, normalmente los computadores están en formato mp4, mov, wmv entre otros, pero ninguno de estos es apto para ser transmitido así que de cualquier formato que se tenga se debe pasar a un **.ts**.

Para lograr esto se recomienda ir a la siguiente página [link](#), donde se puede realizar la conversión, solo se tiene que seleccionar cual es el formato de entrada del video y

la pagina hacer el resto del trabajo, en la siguiente figura se muestra como se ve la página al momento de realizar este trabajo.



Figura 5.1 Convertidor de formato de video

Una vez con el formato de video correcto solo queda poner seleccionarlo en el bloque file source que tiene GNU radio para poder llamarlo y así comenzar con todo el proceso de separación de audio e imagen para ser transmitido, tal y como se puede ver en la siguiente figura.

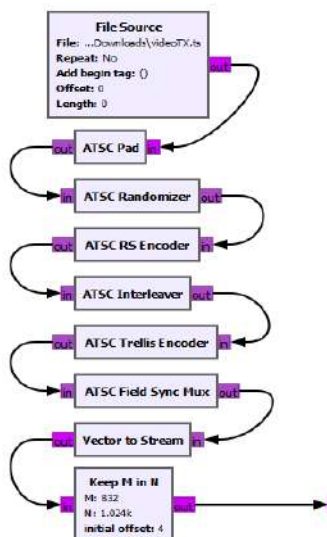


Figura 5.2 Codificación de video tv

Ninguno de estos bloques recibe algún parámetro todos ya están hechos de tal manera que es conectarlos en ese orden par que todas funciones de manera

correcta, para que finalmente se conecte a un modulador que pueda transmitir la señal correctamente. En la siguiente figura se muestra cual debe ser usado y los parámetros que se debe poner para realizar la transmisión de televisión digital.

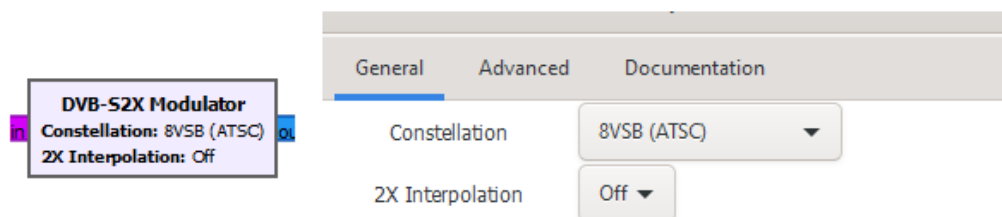


Figura 5.3 Modulador ATSC

Ya con todo configura de manera correcta, solo queda conectar el bloque de osmocom sink, para que el HackRf pueda empezar la transmitir la información al aire, hasta que llegue a un receptor ese se verá en la siguiente sección por ahora al ejecutar el código se debe ver la señal donde viaja la información, además, y en esta se puede el ancho de banda que ocupa.

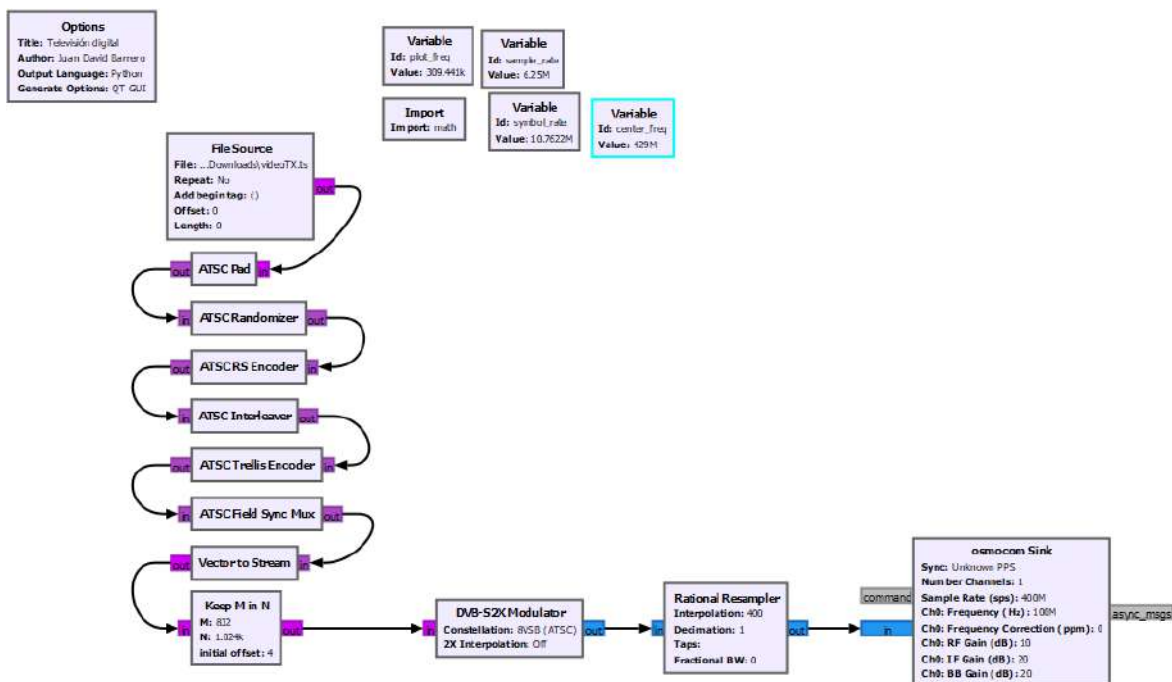
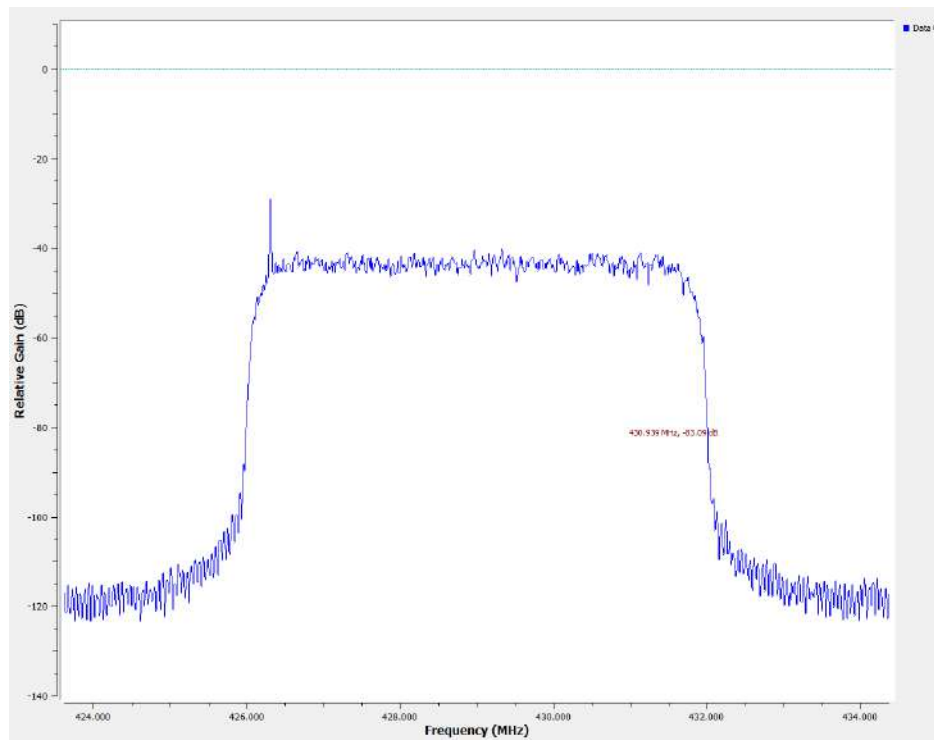


Figura 5.4 Transmisión de televisión digital

En la siguiente figura se puede apreciar cómo es la señal de salida el modulador



*Figura 5.5 Señal ATSC transmitida*

## 5.2 Recepción de televisión digital ATSC con SDR

Finalmente, solo queda hacer el diagrama de bloques del sistema receptor, para esto GNU radio ya ofrece los bloques necesarios que hacer toda la decodificación por nosotros, sin embargo, hay que realizar un filtrado antes para poder separar las posibles interferencias de símbolos y ruido que se producen al momento de realizar la transmisión.

Como se mencionó anteriormente es necesario este proceso también debe ser hecho por un HackRf ya que este tiene la capacidad para poder recibir un ancho de banda de hasta 9 MHz, si se realiza con un RTL\_SDR se corre el peligro de dañarlo y en el mejor de los casos que la recepción falle.

En la siguiente figura se muestra el diagrama de bloques del sistema receptor.

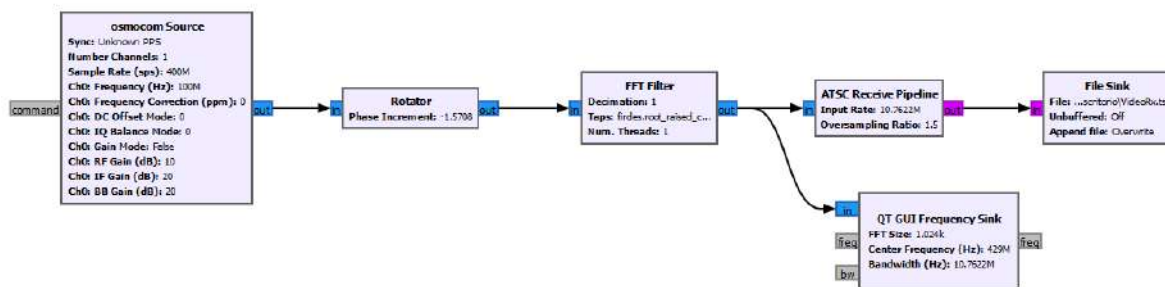


Figura 5.6 Receptor televisión digital

Si todo sale correctamente con el bloque file sink se debe guardar el video en formato ts en el lugar donde se colocó, y este puede ser visto por cualquier reproductor multimedia que soporte este formato.

Un reproductor de video que acepta este formato es SMPlayer este se descarga de desde manera sencilla desde su página oficial, una vez instalado se ve de la siguiente manera.

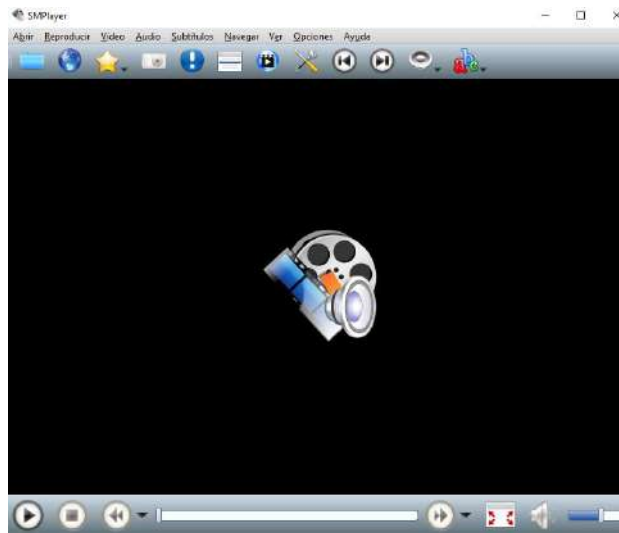


Figura 5.7 SM player

Y se puede abrir el archivo .ts para poder corroborar que el video fue transmitido exitosamente.

## 6. Conclusiones

Al realizar las guías se presenta una oportunidad de que quienes estudian las modulaciones apliquen la teoría con el software.

La versatilidad que ofrece el software de GNU radio para investigar las modulaciones digitales y analógicas, lo hace una herramienta ideal a la hora de querer aprender sobre el mundo de las telecomunicaciones.

La gran cantidad de bibliotecas que ofrece GNU radio brinda la posibilidad de simular modulaciones de alto nivel como por ejemplo 64QAM

Los componentes como los son el HackRf One y el RTL SDR dongle son más económicos comparados con módulos didácticos que ofrece el mercado.

Al terminar de realizar las guías se adquiere destreza en el desarrollo de software, que hace al ingeniero electrónico avance en temas que involucran programación y programación orientada a objetos.

## 7. Referencias

- Andreotti, J. I. (06 de 2015). */ingenieroandreotti.blogspot.com*. Obtenido de */ingenieroandreotti.blogspot.com*:  
<https://ingenieroandreotti.blogspot.com/2015/06/modulacion-modulation-2da-parte.html?m=1>
- Ascencio, O. L. (2010). Oscilador controlado por voltaje con compesación de proceso, voltaje y temperatura . En O. L. Ascencio, *Oscilador controlado por voltaje con compesación de proceso, voltaje y temperatura* (págs. 15-17). INAE.
- Gareth Montenegro Chaidez, V. I. (2016). *Radio Definido Por Software futuro de las comunicaciones*.
- Introducción a los sistemas de comunicación. (2000). En F. G. Stremler, *Introducción a los sistemas de comunicación* (pág. 219). Mexico: Addison-wesley iberoamericano.
- Nave, M. O. (2010). *hyperphysics*. Obtenido de hyperphysics:  
<http://hyperphysics.phy-astr.gsu.edu/hbasees/Audio/radio.html>
- Rodrigurz, J. E. (s.f.). *Telecomundo*. Obtenido de Telecomundo:  
<https://telecomundo.wordpress.com/modulaciones-digitales/>
- Significados. (11 de 06 de 2015). Obtenido de Significados:  
<https://www.significados.com/am-y-fm/>

## 8. Anexos

### 8.1 Links a los códigos y resultados de las practicas

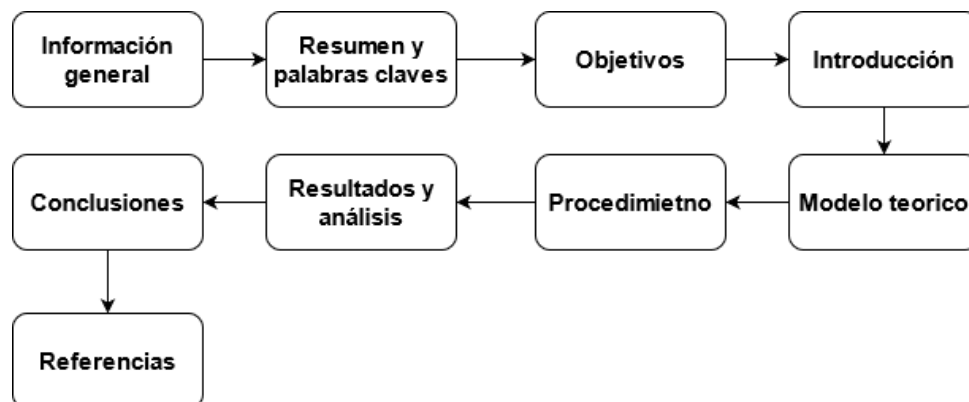
En los siguientes enlaces de puede encontrar información pertinente sobre este trabajo de grado

- Enlace al repositorio donde se encuentra todos los códigos y archivos multimedia ( <https://github.com/JuanDavidBarrero/GNURadio> )
- Enlace al video de la practica de modulación AM ( <https://www.youtube.com/watch?v=IU0ij0F7RS0> )
- Enlace al video de la practica de modulación FM ( <https://www.youtube.com/watch?v=ndbm73-T27Y> )
- Enlace al video de la practica de modulación digital ( [https://www.youtube.com/watch?v=tDHXOxo\\_kQs](https://www.youtube.com/watch?v=tDHXOxo_kQs) )
- Enlace al video de televisión digital ( <https://www.youtube.com/watch?v=0TrnDbnrfT0> )

### 8.2 Modelo para realizar guías de laboratorio

Este proyecto se hace con el fin de realizar cinco prácticas para manejar y utilizar los sistemas de radiocomunicaciones conocidos como RDS (Radio Definido por Software) para el laboratorio comunicaciones de la Universidad Pontificia Bolivariana de Bucaramanga.

Para poder realizar los laboratorios es necesario tener en cuenta la siguiente figura donde se muestran los pasos a seguir para poder redactar de forma correcta un laboratorio que se pueda implementar.



#### Actividad 1: Pasos para realizar una guía de laboratorio

## **Información general**

Acá debe ir el nombre y el número de laboratorio además se debe poner los miembros del grupo o el nombre del estudiante que lo vaya a realizar, también debe ir el nombre del profesor, la materia de comunicaciones y finalmente la fecha de realización y la fecha de entrega del informe,

## **Resumen y palabras clave**

En esta parte debe ir un resumen de lo que se quiere realizar en la práctica esto se hace con la finalidad de dar una idea al estudiante de que se trata el laboratorio.

## **Objetivos**

Acá se presenta lo que se quiere lograr en la práctica además también se pretender verificar las hipótesis que se realizaron y verificar las ecuaciones.

## **Introducción**

Se plasma la relevancia del laboratorio y las aplicaciones que se pueden realizar en la vida cotidiana con estos nuevos conocimientos.

## **Modelo teórico**

Se expone de manera corta y concisa las leyes que se van a comprobar las cuales rigen el comportamiento de los experimentos que se van a realizar en el laboratorio.

## **Procedimiento**

En esta parte se explican los procedimientos que se debe realizar a la hora de hacer las prácticas de laboratorio.

## **Resultado y análisis**

En esta parte se evidencian todos los datos tomados de las prácticas, una práctica siempre tiene gráficas o tablas que ayuden a soportar los datos que fueron tomados y las posteriores conclusiones.

## **Conclusiones**

Esta es la última parte de un laboratorio, son la respuesta que se obtuvieron en la práctica estas deben ser coherentes y responder los objetivos propuestos.

## **Referencias**

Se muestra evidencia de los autores que investigaron para poder realizar la práctica.

Para poder realizar y alcanzar cada uno de los objetivos propuestos se sugiere utilizar el siguiente modelo de actividades para que no haya ningún incidente a la hora de realizar cada una de las prácticas.

## **Actividad 1: Identificar los materiales para desarrollar las prácticas**

Para poder comenzar a hacer la prueba es necesario identificar las herramientas de software y hardware que se piensa aplicar, para esto se propone la siguiente tabla donde se puede ver lo materiales necesarios para iniciar.

| Software                   | Hardware                       |
|----------------------------|--------------------------------|
| GNU radio Companion        | RTL DVB T FM DAB               |
| Python                     | HackRf                         |
| SDR Sharp                  | Cable USB tipo c o micro USB   |
| VirtualBox máquina virtual | Computador con Windows o Linux |

*Tabla 11 Materiales*

La ventaja del software de GNU radio Companion es que es de código abierto, gratis y sirve para todos los sistemas operativos, ya que este se encuentra basado en los lenguajes de programación C y Python. Este software permite programar con diagrama de bloques y trae consigo varias funciones que pueden ser aprovechadas por los SDR.

Otra de las herramientas que se piensa utilizar es el SDR Sharp el cual ya cuenta con una interfaz gráfica que permite analizar de manera más rápida y didáctica las señales que hay en el espectro electromagnético.

Por el lado del hardware es necesario contar con SDR para poder programarlos, en la tabla 1 se nombran dos de los más conocidos en el mercado por el bajo precio y por la gran comunidad que tiene detrás, la cual está siempre activa y dando respuesta a las dudas que se pueden presentar a la hora de utilizar estos dispositivos.

También es necesario contar con un computador que tenga los sistemas operativos indicados ya que estas aplicaciones solo se encuentran para instalarse de forma sencilla en estos entornos, de no poder contar con uno de estos se propone utilizar una máquina virtual para poder realizar las instalaciones ahí y trabajar desde allí las prácticas de laboratorio que se van a proponer.

## **Actividad 2: Realizar el esquema de los diagramas de bloques**

En este punto y con la lista materiales ya definidos se comenzará a realizar los diagramas de bloques de los cuales saldrá el código que permitirá programar los SDR, pero como este es un entorno de desarrollo nuevo para los estudiantes, es necesario primero realizar un practica cero en la cual es muestre como poder utilizar las diferentes herramientas que ofrece GNU radio Companion.

Gracias al software de GNU se pueden hacer los códigos en forma de diagrama de bloques, lo cual permite una facilidad a la hora de entender cómo se forma las distintas modulaciones que se verán en los laboratorios, de la misma forma este software permite crear bloques totalmente desde cero para cumplir con las necesidades requeridas para los diseños que se piensan montar.

Además, este software cuenta con una interfaz gráfica que permite ir analizando a cada paso como la señal portadora y la señal moduladora van actuando a medida que entran y salen los diagramas de bloques. Lo cual permite un mayor entendimiento de cómo se realiza todo el proceso de comunicación.

### **Actividad 3: Contratar el funcionamiento de los dispositivos**

Una vez que la simulación esta lista y el diagrama de bloques este completado se pasara a implementar el código realizado en los SDR, pero para esta parte hay que tener en cuenta que no todos estos dispositivos cuentan con la posibilidad de transmitir la información, para este caso se utiliza el HackRf mencionado en la tabla 1 él cuenta con esta habilidad.

Para poder recibir y decodificar la información transmitida se puede utilizar cualquiera de los dispositivos mencionados anteriormente, lo importante en esta parte es haber diseñado los protocolos que se usaran para demodular la información que fue enviada.

Una vez que los códigos estén subidos a los respectivos SDR se puede observar como el mensaje enviado por el HackRf es recibido y decodificado. Este mensaje puede ser texto, voz o video, siempre teniendo en cuenta la modulación escogida para la práctica de laboratorio que se esté realizando.

### **Actividad 4: Diseñar laboratorios**

Con las herramientas nombradas anteriormente, solo queda por hacer los informes de las prácticas de laboratorio, para esto se tiene en cuenta que cada una de estas debe tener objetivos fijos que ayude al estudiante a consolidar el tema que está viendo con lo que va a implementar físicamente.

Se van a realizar cuatro practicas más una adicional (que va a ser sobre cómo utilizar el software la cual será la practica número cero), cada una de estas va a tener su propia guía de laboratorio e informe. Para las otras cuatro practicas se van a tocar distintos métodos de modulación con distintos tipos de mensajes que se pueden transmitir, para se tenga un mayor conocimiento de cómo utilizar el hardware propuesto.

Todos los informes contarán con los pasos vistos en la figura 13 además él es necesario que para el planteamiento del problema se tiene que resolver utilizando algún modelo de modulación para poder observar si el estudiante tiene los conceptos claros vistos en la clase teórica, además al final se pedirá sacar conclusiones del trabajo que se realizó.

### **8.3 Guías de laboratorio**



Universidad Pontificia Bolivariana. Seccional Bucaramanga  
Ingeniería eléctrica y electrónica  
Laboratorio de comunicaciones

## Guía de laboratorio I

### Adaptación al entorno de GNU Radio Companion para el uso de SDR

|                            |                                                                 |
|----------------------------|-----------------------------------------------------------------|
| ASIGNATURA                 | LABORATORIO DE COMUNICACIONES                                   |
| CÓDIGO                     |                                                                 |
| GRUPO DE INVESTIGADORES I  |                                                                 |
| GRUPO DE INVESTIGADORES II |                                                                 |
| TÍTULO DEL LABORATORIO     | ADAPTACIÓN AL ENTORNO DE GNU RADIO COMPANION PARA EL USO DE SDR |

#### Información General

GNU radio es un software de código abierto, el cual cuenta con varias librerías optimizadas para trabajar con vectores y matrices en tiempo real. Este se encuentra disponible para descargar en sistemas operativos como Windows o Linux.

En esta guía familiariza al estudiante con el software GNU en el tema de telecomunicaciones.

En el trabajo de grado *Desarrollo implementación de un laboratorio de Radio Definido por Software RDS- para el laboratorio de telecomunicaciones de la UPB.* Juan David Barrero Lizcano. Lea y resuma los temas:

1. Introducción al Radio Definido por Software
2. Hardware necesario
3. Primeros pasos en GNU radio.

#### Palabras clave

Diagramas de bloque, ruido, filtros pasa altos, pasa bajos, pasa banda, vectores, operaciones aritméticas.

### Objetivo general

Conocer el entorno de trabajo GNU radio y los distintos tipos de bloques.

### Objetivos Específicos

Instalar GNU radio

Comprender la información vista en el analizador de espectro

Analizar el comportamiento de una señal en el dominio del tiempo

Filtrar los armónicos de una señal

Ubicar y trabajar los bloques de programación de GNU radio de manera rápida

### Introducción

Para observar el avance del estudiante en el uso de los bloques básicos de GNU radio, se plantea un problema el cual debe ser resuelto utilizando las herramientas vistas.

### Modelo teórico

**Teorema de Nyquist:** Es la frecuencia mínima con la que se debe muestrear una señal para ser recuperada.

**Filtros:** Operación matemática que se realiza sobre una señal con el fin de resaltar o atenuar algunas de sus características.

**POO:** Programación Orientada Objetos. Paradigma que usa GNU radio para programar los bloques de manera eficiente y ordenada.

## Procedimiento

### Parte 1

- Realizar un diagrama de bloques utilizando el software de GNU radio que haga la suma de tres señales senoidales a las cuales se les pueda variar la amplitud utilizando un indicador (*slider*). (rango de las señales 1000[Hz] - 10000[Hz])
- Mostrar el comportamiento de la señal resultante en el osciloscopio integrado en GNU radio. Observe lo que ocurre cuando aumenta la amplitud.
- Mostrar el comportamiento de la señal resultante en el dominio de la frecuencia ¿Qué puede observar?
- Calcule la potencia de cada una de las señales y contraste el valor obtenido con el visto en el analizador de espectro.
- Utilizando los filtros vistos, recuperar la señal de más baja frecuencia y muestre su comportamiento en un nuevo osciloscopio (Nombrar el nuevo osciloscopio)
- Utilizando filtros, recuperar la señal de más alta frecuencia y muestre el comportamiento de la señal en el dominio de la frecuencia.

### Parte 2

- Construya un diagrama de bloques que permita alternar entre dos audios grabados con anterioridad en su equipo, utilizando el bloque *selector*
- Filtre para el primer audio los sonidos que tengan una frecuencia menor a los 5000[Hz]
  1. ¿Qué sonidos o instrumentos puede escuchar ahora? Vea el comportamiento en el analizado de espectro
  2. Guarde la señal resultante en nuevo archivo
- Filtre para el segundo audio los sonidos que tengan una frecuencia mayor a los 8000[Hz]
  1. ¿Qué sonidos o instrumentos puede escuchar ahora? Vea el comportamiento en el analizado de espectro
  2. Guarda la señal resultante en nuevo archivo

### Aspectos evaluables

Realizar un informe que resuma la parte uno y la parte dos propuestas

- Evidencia (Imágenes o audios) de los diagramas de bloques pedido en la sección de arriba
- Orden y presentación de los diagramas de bloques
- Análisis de los datos, y las gráficas vistas
- Conclusiones

### Bibliografía

Desarrollo implementación de un laboratorio de Radio Definido por Software RDS- para el laboratorio de telecomunicaciones de la UPB. Trabajo de grado. Barrero Lizcano Juan David. 2021



## Guía de laboratorio II

### TRANSMISIÓN Y RECEPCIÓN DE SEÑALES ANALÓGICAS UTILIZANDO SDR MODULANDO LA AMPLITUD

|                            |                                                                                    |
|----------------------------|------------------------------------------------------------------------------------|
| ASIGNATURA                 | LABORATORIO DE COMUNICACIONES                                                      |
| CÓDIGO                     |                                                                                    |
| GRUPO DE INVESTIGADORES I  |                                                                                    |
| GRUPO DE INVESTIGADORES II |                                                                                    |
| TÍTULO DEL LABORATORIO     | TRANSMISIÓN Y RECEPCIÓN DE SEÑALES ANALÓGICAS UTILIZANDO SDR MODULANDO LA AMPLITUD |

#### Información General

En el mundo de las telecomunicaciones las modulaciones son necesarias para poder transmitir a larga distancia todo tipo de información.

La modulación por amplitud se logra al variar la amplitud de una señal sinusoidal con frecuencia y fase fija (señal moduladora) en proporción a una señal dada con una frecuencia superior (señal portadora)

En el trabajo de grado *Desarrollo implementación de un laboratorio de Radio Definido por Software RDS- para el laboratorio de telecomunicaciones de la UPB. Juan David Barrero Lizcano*. Lea y resuma los temas:

1. Modulación AM utilizando RDS
2. Modulación por amplitud portadora suprimida (DSB-SC)
3. Modulación por amplitud portadora suprimida (DSB-LC)
4. Modulación por cuadratura QAM

#### Palabras clave

Mezcladores, Modulación, Amplitud, RDS, Portadora, Moduladora, Ancho de banda, Índice de modulación, SDR#, Demodulación

### Objetivo general

Realizar la transmisión y recepción de una señal de audio entre dos RDS utilizando modulación AM

### Objetivos Específicos

Instalar los drivers para programar los RDS

Instalar el software de SDR# para detectar señales a diferentes frecuencias

Identificar el ancho de banda de ocupa una señal AM

Observar como el índice de modulación afecta a una señal AM

Recuperar la señal de información enviada por el RDS transmisor

Comprender el beneficio de trabajar con señales ortogonales para la transmisión de múltiples mensajes

### Introducción

Con la intención de realizar una comunicación entre dos RDS conectados en diferentes equipos, el estudiante debe tener presente como es el comportamiento de este tipo de señales para poder transmitirla y recibirla sin perjudicar la información del mensaje que se esté enviando, además, de estar en capacidad de poder ajustarse a cambiar los parámetros del sistema para adaptarse a los requisitos pedidos.

### Modelo teórico

**Ancho de banda:** Para señales analógicas, el ancho de banda es la longitud, medida en hercios (Hz), de la extensión de frecuencias en la que se concentra la mayor potencia de la señal.

**Índice de modulación:** O porcentaje de modulación se define como un número escalar que indicia cuando se manipula la amplitud de la portadora

**RDS:** Radio Definido por Software es un sistema de radiocomunicaciones donde varios de los componentes anteriormente utilizados en hardware ahora son implementados en software.

**Mezclador:** En GNU radio se puede definir como un bloque que multiplican las señales de entrada, entregando a la salida una combinación lineal de todas las entradas.

**Ortogonalidad:** Es un adjetivo que se emplea para nombrar a aquello que se encuentra en un ángulo de 90°.

## Procedimiento

### Parte 1

- Realizar un diagrama de bloques utilizando el software de GNU radio el cual realice una modulación DSB-SC con un tono como moduladora al cual se le puede variar la frecuencia.
- A la salida del sistema colocar un analizador de espectro y comience a variar la frecuencia 200[Hz]
  1. Observe que pasa con el ancho de banda
  2. ¿Cuál es la relación del ancho de banda con la frecuencia de la señal moduladora?
- A la salida del sistema colocar un osciloscopio
  1. ¿Cuál es el índice de modulación de este tipo de señal?
  2. ¿Cambia el índice de modulación al variar la frecuencia de la portadora?
- Recupere el tono modulado y observe el comportamiento en el osciloscopio

### Parte 2

- Con el software de Audacity grabar un audio de al menos 15 [s], este se guardo en formato WAV y muestreado a una frecuencia de 48[KHz]
- Utilizando GNU radio realizar un diagrama de bloques el genera una modulación DSB-LC, la cual tendrá como moduladora el audio grabado anteriormente.
  1. A la señal modulada se le debe poner una ganancia que puede ser variada con un indicador (*slider*)
- Limitar el ancho de banda del audio utilizando un filtro este no debe pasar de los 5000[Hz]
- Realizar la transmisión de la señal modulada utilizando el RDS HackRf-ONE a una frecuencia de 100[MHz]
- Colocar un analizador de espectro y osciloscopio a la salida del bloque transmisor.
  1. ¿Cuál es el ancho de banda de la señal transmitida?
  2. ¿Cuál es la potencia de la señal enviada?
  3. ¿Qué sucede al variar la amplitud de la señal moduladora?
- Con el Dongle RTL y GNU radio capturar y demodular la señal transmitida
- A la salida de la del receptor colocar el bloque *audio sink* para escuchar la señal recuperar
- Colocar el analizador de espectro a la salida del receptor para verificar el comportamiento de la señal recibida.

### Parte 3

- Utilizando el RDS HackRf-ONE y GNU radio realizar un diagrama de bloques que permita transmitir dos señales diferentes de audio al mismo tiempo
- Las señales deben ser enviada a una frecuencia de 200[MHz] y el ancho de banda de la señal modulada no debe ser de más de 20[KHz].
- Con el software SDR# sintonizar y captar los mensajes enviados
  - ¿Qué potencia tiene la señal recuperada?
  - ¿Cuánto ancho de banda ocupada cada señal?
  - Al separar los RDS 1[m] cuando disminuye la potencia en el receptor

### Aspectos evaluables

Realizar un informe que resuma la parte uno y la parte dos propuestas

- Evidencia (Imágenes o audios) de los diagramas de bloques pedido en la sección de arriba
- Orden y presentación de los diagramas de bloques
- Análisis de los datos, y las gráficas vistas
- Conclusiones

### Bibliografía

Desarrollo implementación de un laboratorio de Radio Definido por Software RDS- para el laboratorio de telecomunicaciones de la UPB. Trabajo de grado. Barrero Lizcano Juan David. 2021



### Guía de laboratorio III

## TRANSMISIÓN Y RECEPCIÓN DE SEÑALES ANALÓGICAS UTILIZANDO SDR MODULACIÓN DE FRECUENCIA

|                            |                                                                                         |
|----------------------------|-----------------------------------------------------------------------------------------|
| ASIGNATURA                 | LABORATORIO DE COMUNICACIONES                                                           |
| CÓDIGO                     |                                                                                         |
| GRUPO DE INVESTIGADORES I  |                                                                                         |
| GRUPO DE INVESTIGADORES II |                                                                                         |
| TÍTULO DEL LABORATORIO     | MODULACIÓN DE FRECUENCIA<br>APLICADA A SEÑALES ANALÓGICAS Y<br>DIGITALES UTILIZANDO SDR |

#### Información General

En el mundo de las telecomunicaciones las modulaciones son necesarias para poder transmitir a larga distancia todo tipo de información.

La modulación de frecuencia es una técnica de modulación utilizada ampliamente en el mundo, por sus dos grandes ventajas frente a la modulación de amplitud, la primera se puede transmitir sonido de mejor calidad debido al menor ancho de banda que necesita, y la segunda es su inmunidad al ruido debido a que esta solo varía la frecuencia mientras que la amplitud nunca cambia, por lo que el ruido nunca crece de manera desproporcional.

En el trabajo de grado *Desarrollo implementación de un laboratorio de Radio Definido por Software RDS- para el laboratorio de telecomunicaciones de la UPB*. Juan David Barrero Lizcano. Lea y resuma los temas:

1. Modulación FM utilizando RDS
2. NBFM simulación utilizando RDS
3. Transmisión y receptor NBFM con RDS
4. Modulación de frecuencia banda ancha (WBFM)
5. Transmisión y recepción WBFM con RDS

### Palabras clave

VOC, Modulación, Frecuencia, RDS, Portadora, Moduladora, Ancho de banda, Desviación de frecuencia, Coeficiente de sensibilidad, Coeficientes de Bessel

### Objetivo general

Realizar la transmisión y recepción de una señal de audio entre dos RDS utilizando modulación de frecuencia (FM)

### Objetivos Específicos

Utilizar el software de SDR# para detectar señales a diferentes frecuencias

Identificar el ancho de banda de ocupa una señal FM

Identificar el tipo de modulación de frecuencia que se está transmitiendo

Recuperar la señal de información enviada por el RDS transmisor

### Introducción

Con la intención de realizar una comunicación entre dos RDS conectados en diferentes equipos, el estudiante debe tener presente como es el comportamiento de este tipo de señales para poder transmitirla y recibirla sin perjudicar la información del mensaje que se esté enviando, además, de estar en capacidad de poder ajustarse a cambiar los parámetros del sistema para adaptarse a los requisitos pedidos.

### Modelo teórico

**Ancho de banda:** Para señales analógicas, el ancho de banda es la longitud, medida en hercios (Hz), de la extensión de frecuencias en la que se concentra la mayor potencia de la señal.

**RDS:** Radio Definido por Software es un sistema de radiocomunicaciones donde varios de los componentes anteriormente utilizados en hardware ahora son implementados en software.

**El coeficiente de sensibilidad:** el cual es el número de veces que es necesario amplificar la señal moduladora para lograr el índice de modulación deseado.

**Desviación de frecuencia:** El cual es el cambio en la frecuencia que sucede que ocurre en la señal portadora cuando hay un cambio en la amplitud de la señal moduladora.

**Índice de modulación:** Es la relación entre la desviación de frecuencia y la frecuencia de modulación

### Parte 1

- Realizar un diagrama de bloques utilizando el software de GNU radio el cual realice muestre el funcionamiento de un VCO ante una señal senoidal
- A la salida del sistema colocar un analizador de espectro y comience a variar la amplitud de la señal de entrada al VCO
  1. Observe que pasa con el ancho de banda
  2. ¿Cuál es la relación del ancho de banda con la amplitud de la señal moduladora?
- A la salida del sistema colocar un osciloscopio
  1. ¿Qué pasa en la salida a medida que aumenta la amplitud de la señal de entrada?
  2. Cambia la señal senoidal por una señal cuadrada, ¿Qué ocurre ahora con la señal de salida?

### Parte 2

- Con el software Audacity grabar un audio de al menos 15[s], este de ser guardado en formato WAV y muestreado a una frecuencia de 44100[Hz]
- Utilizando el bloque de GNU radio NBFM realice un sistema de transmisión que pueda enviar por un canal un tono de la frecuencia que desee y el otro el audio guardado anteriormente
  1. EL bloque NBFM debe estar configurado con las características de una emisora comercial colombiana
- Colocar un analizador de espectro a la salida del bloque de transmisión
  1. ¿Cuál es el índice de modulación de la señal?
  2. ¿Cuál es el ancho de banda de la señal modulada?
- Realizar la transmisión utilizando el SDR HackRFOne a una frecuencia de 450[MHz]
- Con el SDR DongleRTL diseñar el diagrama de bloque de un receptor de la señal transmitida anteriormente
- A la salida del sistema receptor colocar el bloque *audio sink* para poder escuchar la señal transmitida
- Colocar el analizador de espectro a la salida del receptor, para observar los niveles de ruido que ocurren la transmitir la señal

### Parte 3

- Utilizando el RDS HackRfOne y GNU radio realizar un diagrama de bloques que permita transmitir la señal grabada en la parte uno de este laboratorio, con el protocolo WBFM
- La señal debe ser enviada en medio de alguna de las bandas comerciales de FM que se encuentran en la región de Bucaramanga y con un ancho de banda no mayor a 20[KHz]
- Utilizando SDR# y el DongleRTL SDR sintonizar y demodular la señal transmitida
  1. ¿Qué potencia tiene la señal?
  2. ¿Cómo es el ancho de banda comparado con las emisoras comerciales?

3. Aléjese poco a poco del transmisor mientras aun observa la señal transmitida ¿Qué ocurre cuando se aleja 1m, 2m, 3m?
- Utilizando el radio de su celular sintonice la frecuencia a la cual está transmitiendo
1. ¿Qué ocurre a medida que está alejando?

#### Aspectos evaluables

Realizar un informe que resuma la parte uno y la parte dos propuestas

- Evidencia (Imágenes o audios) de los diagramas de bloques pedido en la sección de arriba
- Orden y presentación de los diagramas de bloques
- Análisis de los datos, y las gráficas vistas
- Conclusiones

#### Bibliografía

Desarrollo implementación de un laboratorio de Radio Definido por Software RDS- para el laboratorio de telecomunicaciones de la UPB. Trabajo de grado. Barrero Lizcano Juan David. 2021



## Guía de laboratorio IV

### TRANSMISIÓN Y RECEPCIÓN DE SEÑALES UTILIZANDO SDR

### MODULACIONES DIGITALES

|                            |                                                                                     |
|----------------------------|-------------------------------------------------------------------------------------|
| ASIGNATURA                 | LABORATORIO DE COMUNICACIONES                                                       |
| CÓDIGO                     |                                                                                     |
| GRUPO DE INVESTIGADORES I  |                                                                                     |
| GRUPO DE INVESTIGADORES II |                                                                                     |
| TÍTULO DEL LABORATORIO     | ESTUDIO DE MODULACIÓN M-ARIAS DIGITALES PARA LA TRANSMISIÓN DE DATOS UTILIZANDO SDR |

#### Información General

El siglo XXI avanza cada vez a la tecnología digital por sus ventajas ante la analógica, como su facilidad de implementar en cualquier dispositivo tanto para transmisión y recepción, además la facilidad que se tiene para montar toda la modulación hasta ahora vistas en unas cuantas líneas de código.

Sin mencionar que tienen una gran inmunidad al ruido, pero para que esta sea una forma viable de mandar información por el aire es importante entender su funcionamiento y los problemas que se puede tener al querer transmitirla, y así poder evitarlos y realizar una radio conexión exitosa.

Estas modulaciones se dividen en 3 grandes grupos y estos a su vez tiene características únicas que los hacen muy atractivas a ciertas trasmisiones ya sea por su alta velocidad o por la fidelidad con la que pueden enviar una señal.

En el trabajo de grado *Desarrollo implementación de un laboratorio de Radio Definido por Software RDS- para el laboratorio de telecomunicaciones de la UPB. Juan David Barrero Lizcano*. Lea y resuma los temas:

1. Modulaciones digitales utilizando RDS
2. Modulación ASK
3. Modulación FSK
4. Modulación PSK

### Palabras clave

ASK, modulación, bits, bytes, baudios, FSK, GFSK, CFSK, VCO, portadora, Ancho de banda, PSK, 4-PSK, 8PSK, programación orientada a objetos

### Objetivo general

Realizar la transmisión y recepción de una señal de un archivo de texto (.txt) y una imagen en formato .jpg entre dos RDS utilizando la siguiente modulación digitales ASK GFSK y 4PSK

### Objetivos Específicos

Transmitir una imagen utilizando modulación GFSK entre 2 SDR

Determinar el ancho de banda de una modulación GFSK

Identificar el ancho de banda de ocupa una OOK utilizando SDR#

Transmitir un archivo de texto utilizando modulación 4-PSK

### Introducción

Con la intención de realizar una comunicación entre dos RDS conectados en diferentes equipos, el estudiante debe tener presente como es el comportamiento de este tipo de señales para poder transmitirla y recibirla sin perjudicar la información del mensaje que se esté enviando, además, de estar en capacidad de poder ajustarse a cambiar los parámetros del sistema para adaptarse a los requisitos pedidos.

### Modelo teórico

**Bit:** Es un número binario que puede valer 0 o 1 y es la unidad básica, que se utiliza en las comunicaciones digitales y en la computación. Este representa el paso de corriente o no en los transistores de los que esta compuestos los computadores.

**Byte:** Un conjunto de 8 bit forman un byte, y este el menor número de que puede tener un elemento de una memoria.

**ASK:** Amplitud Shift Keying es la implementación digital de la modulación AM, en esta se utiliza como portadora una señal digital, esta puede ser una señal bit generada por el computador

**FSK:**(Frequency-shift keying), es una modulación de frecuencia donde la señal moduladora es digital, Al igual que en ASK existen varios tipos de configuraciones para este proceso

**PSK:** (Phase-shift keying), es una modulación de fase donde la señal moduladora es digital. En la modulación PSK convencional el valor de la fase de la señal portadora varía dependiendo de valor de la señal de bits que aparezca en la entrada

### Parte 1

- Descargue una imagen de formato (jpg o png) de un tamaño menor a 1063 x 1535 pixeles
- Utilizando el bloque *GFSK* de GNU radio ajústelo para que tenga una desviación de frecuencia comercial.
- Utilizando el bloque *File Source* ajústelo de tal manera que la información no se repita una vez terminado de recorrer cada píxel de la imagen.
- Diseñe el transmisor de la imagen utilizando GFSK
  1. Determine utilizando GNU radio el ancho de banda
  2. ¿Qué sucede si aumenta el número de bits de la paquetación antes de realizar la modulación?
- Diseñe un sistema receptor que sea capaz de recibir la imagen enviada en el ítem anterior
  1. ¿Qué pasa si al separar los bits el número del receptor y transmisor son coincide?

### Parte 2

- Realice el diagrama de bloques de un sistema de modulación ASK utilizando GNU radio
- Generar una señal de bits aleatorios la cual será la señal moduladora
- Utilizando el HackRf one transmitir la señal a una frecuencia menor de 433[MHz]
- Con el SDR dongle y utilizando SDR# captar la señal enviada a la frecuencia que se haya escogido
- Determinar el ancho de banda de la señal
  1. Compare la señal vista de SDR# con un analizador de frecuencia colocado a la salida de la del sistema de modulación
  2. Aumente el número de bits ¿Qué le sucede al ancho de banda?
  3. Empaquete más bits antes de ser enviados ¿Qué pasa con el ancho de banda?

### Parte 3

- Utilizando los bloques vistos en GNU radio implemente una modulación digital 4-PSK
- Se debe transmitir un archivo de texto en formato .txt este puede contener cualquier cantidad de caracteres.
- Determine la fase de la señal en cada una de las posibles combinaciones de los bits
- A la salida coloque un analizador de espectro para poder determinar el ancho de banda de la señal
- Colocar a la salida un *Constellation Sink* para poder corroborar que sea un sistema 4-PSK
- Realice un sistema que sea capaz de decodificar la señal nuevamente a su estado original y que se guarde en el escritorio con el nombre "Señal\_recuperada.txt"
- Observar con el bloque *Constellation Sink* la salida del bloque *rtl\_source*
  1. ¿Qué ocurre mientras no se puede decodificar la señal?
  2. ¿Qué ocurre con la información mientras no se decodifica la señal?

### Aspectos evaluables

Realizar un informe que resuma la parte uno y la parte dos propuestas

- Evidencia (Imágenes o audios) de los diagramas de bloques pedido en la sección de arriba
- Orden y presentación de los diagramas de bloques
- Análisis de los datos, y las gráficas vistas
- Conclusiones

### Bibliografía

Desarrollo implementación de un laboratorio de Radio Definido por Software RDS- para el laboratorio de telecomunicaciones de la UPB. Trabajo de grado. Barrero Lizcano Juan David. 2021



## Guía de laboratorio V

### TRANSMISIÓN Y RECEPCIÓN DE TELEVISIÓN DIGITAL ATSC

|                            |                                                  |
|----------------------------|--------------------------------------------------|
| ASIGNATURA                 | LABORATORIO DE COMUNICACIONES                    |
| CÓDIGO                     |                                                  |
| GRUPO DE INVESTIGADORES I  |                                                  |
| GRUPO DE INVESTIGADORES II |                                                  |
| TÍTULO DEL LABORATORIO     | APROXIMACIÓN A TELEVISIÓN DIGITAL UTILIZANDO SDR |

#### Información General

La televisión digital llegó para quedarse en este mundo se tienen más ventajas y más servicios de los que alguna vez pudo ofrecer la televisión analógica, debido a esto es que se usa ampliamente a ahora por todo el mundo. Esto debido a que cada vez más empresas están atraídas al mundo digital tanto en audio como para imágenes.

Además de contar con la ventaja de poder transmitir más canales, y se tienen un control mayor sobre que frecuencia se quiere transmitir.

En el trabajo de grado *Desarrollo implementación de un laboratorio de Radio Definido por Software RDS- para el laboratorio de telecomunicaciones de la UPB. Juan David Barrero Lizcano*. Lea y resuma los temas:

1. Transmisión de televisión digital
2. Recepción de televisión digital

#### Palabras clave

GNU radio, televisión digital, ATSC, bit, bytes, comunicación digital

### Objetivo general

Realizar la transmisión y recepción de un video en formato .ts de un computador a otro utilizando dos HackRf One

### Objetivos Específicos

Transmitir un video en formato .ts (audio e imagen)

Recibir un video en forma .ts (audio e imagen)

### Introducción

La televisión digital es una nueva “tecnología” que se tomado poco a poco el mundo ya que tiene ciertas ventajas frente a la analógica, en el mundo digital ahora la imagen y el sonido se transmite poco a poco en paquetes de información discreta. Y un equipo programado decodifica esta información y la separa para posteriormente implementarla en la pantalla y los bafles de un televisor.

### Modelo teórico

**TDT:** Televisión digital terrestre, también llamada en algunos países de América televisión digital abierta. Esta es muy común verla en los hogares, con la llegada de este sistema de transmisión de televisión es que la tv analógica poco a poco se está dejando de lado.

**ATSC:** Advanced Television Systems Committee, en español Comité de Sistemas de Televisión Avanzada, es el grupo encargado del desarrollo de los estándares de la televisión digital

**ADSL:** Asymmetric Digital Subscriber List que traduce Línea de Abonado Digital Asimétrica) es un tipo de tecnología de transmisión de datos digitales y acceso a Internet.

**Satelital:** La televisión por satélite es un método de transmisión televisiva consistente en retransmitir desde un satélite de comunicaciones una señal de televisión emitida desde un punto de la Tierra

### Parte 1

- Grabe un video desde su celular de un tema que encuentre interesante, este video debe ser de menos de dos minutos
- Exporte el video del celular a su computador y guárdelo en una carpeta que puede ser encontrada fácilmente
- Guarde el video bajo el nombre videoTx.XX
- Cambie el formato de video a .Ts
- Utilizando el software de GNU radio realice un sistema que pueda transmitir este video
- Desde otro computador realice un sistema que pueda recibir la señal de televisión enviada desde el HackRf
- Coloque un analizador de espectro a la salida del filtro de la señal decodificada
  1. ¿Cuál es el ancho de banda de la señal recibida?
- Contraste el ancho de banda captado con GNU radio ahora con el software SDR#
  2. ¿Qué puede concluir al ver ambas señales?

### Aspectos evaluables

Realizar un informe que resuma la parte uno y la parte dos propuestas

- Evidencia (Imágenes o audios) de los diagramas de bloques pedido en la sección de arriba
- Orden y presentación de los diagramas de bloques
- Análisis de los datos, y las gráficas vistas
- Conclusiones

### Bibliografía

Desarrollo implementación de un laboratorio de Radio Definido por Software RDS- para el laboratorio de telecomunicaciones de la UPB. Trabajo de grado. Barrero Lizcano Juan David. 2021