



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE INGENIERÍA ELECTRÓNICA**

**DISEÑO E IMPLEMENTACIÓN DE UN BANCO DE PRUEBAS DE
LABORATORIO APLICADO A FM BROADCASTING UTILIZANDO
SOFTWARE OPENSOURCE GNURADIO, RASPBERRY PI Y RADIOS
DEFINIDAS POR SOFTWARE**

Trabajo de titulación previo a la obtención del
Título de Ingeniero Electrónico

**AUTORES: VANESSA STEPHANIA PAZMIÑO QUIGUANGO
LUIS ERICK ESPINOZA PIONCE
TUTOR: ING. HÓLGER SANTILLÁN MSC.**

Guayaquil – Ecuador

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Vanessa Stephania Pazmiño Quiguango con documento de identificación N° 0950611020 y Luis Erick Espinoza Pionce con documento de identificación N° 0952379907; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 10 de febrero del año 2022

Atentamente,



Vanessa Stephania Pazmiño Quiguango
0950611020



Luis Erick Espinoza Pionce
0952379907

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Vanessa Stephania Pazmiño Quiguango con documento de identificación N° 0950611020 y Luis Erick Espinoza Pionce con documento de identificación N° 0952379907, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del proyecto de investigación: “Diseño e implementación de un banco de pruebas de laboratorio aplicado a FM broadcasting utilizando software Opensource GNURadio, Raspberry PI y radios definidas por software”, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 10 de febrero del año 2022

Atentamente,

Vanessa Stephania Pazmiño Quiguango
0950611020

Luis Erick Espinoza Pionce
0952379907

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Holger Jorge Santillán Carranza con documento de identificación N° 0916303134, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO E IMPLEMENTACIÓN DE UN BANCO DE PRUEBAS DE LABORATORIO APLICADO A FM BROADCASTING UTILIZANDO SOFTWARE OPENSOURCE GNURADIO, RASPBERRY PI Y RADIOS DEFINIDAS POR SOFTWARE, realizado por Vanessa Stephania Pazmiño Quiguango con documento de identificación N° 0950611020 y por Luis Erick Espinoza Pionce con documento de identificación N° 0952379907, obteniendo como resultado final el trabajo de titulación bajo la opción tesis que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 10 de febrero del año 2022

Atentamente,



Firmado electrónicamente por:

**HOLGER JORGE
SANTILLAN
CARRANZA**

Ing. Holger Jorge Santillán Carranza, MSc
0916303134

Dedicatoria

El presente proyecto de titulación lo dedico principalmente a Dios, por ser el inspirador y darme la fuerza para continuar en este proceso de obtener uno de los anhelos más deseados por mí.

A mis padres, por su amor, trabajo y sacrificio en todos estos años, gracias a ellos he logrado llegar hasta aquí.

A mis amigos, hermanos, por estar siempre presentes acompañándome a lo largo de esta etapa de mi vida por aconsejarme y siempre mostrarme que debo estar optimista aún en dónde las situaciones te digan lo contrario. Gracias por ser especiales los quiero mucho.

A todas las personas que nos han apoyado y han hecho que el trabajo se realice con éxito en especial a aquellos que nos abrieron las puertas y compartieron sus conocimientos, a los maestros de la Universidad Politécnica Salesiana que supieron llenarnos de muchos conocimientos, los cuales serán muy importantes para nuestra vida personal y profesional.

Este esfuerzo va dedicado con amor para todos ustedes, gracias a todos por confiar en mis capacidades.

Vanessa Stephania Pazmiño Quiguango

Dedicatoria

Este proyecto de titulación va dedicado a mi familia, mis padres Paula Pionce, Luis Espinoza y mi hermana Erika Espinoza. Quienes se han caracterizado por ser personas trabajadoras y luchadoras, de las cuales eh aprendido los valores mas importantes en la vida.

Eh aquí mi principal motor para alcanzar cada una de mis metas y desafíos que se presentan cada día. Es por tal motivo que les agradezco y dedico un logro más de los muchos que tendremos en adelante como equipo y como familia.

Agradezco a Dios por darnos fuerzas cada día y salud para continuar y cumplir nuestros sueños.

Y que esto sirva de ejemplo para nuestras futuras generaciones, en especial a mis sobrinas Ammy y Paula que contarán siempre con todo nuestro apoyo, así como se lo hizo conmigo.

Por último, pero lo más importante, dedico esto a mi hijo Luis Espinoza para que en un futuro esté seguro de que tendrá muchas más oportunidades gracias a todo un equipo que trabajo sin descanso por el bienestar de los suyos.

Gracias por la confianza.

Luis Erick Espinoza Pionce

Resumen del proyecto

Año	Alumnos	Director de proyecto	Tema de proyecto de titulación
2022	Vanessa Stephania Pazmiño Quiguango Luis Erick Espinoza Pionce	Ing. Holger Santillán MSc.	Diseño e implementación de un banco de pruebas de laboratorio aplicado a FM broadcasting utilizando software Opensource GNURadio, Raspberry PI y radios definidas por software

Las tecnologías de las comunicaciones inalámbricas se han desarrollado en los últimos años, dando lugar a nuevos sistemas tecnológicos como LTE Advanced, 5G, WIFI6, IEEE 802.11ax, Zigbee y muchas más que son muy utilizadas en nuestro diario vivir para servicios de banda ancha, sin embargo, otras tecnologías inalámbricas como FM no han tenido desarrollos importantes en los últimos años ya que la tendencia actual es la transmisión de streaming de audio vía internet, estas nuevas tecnologías que serán aplicadas dependerán fundamentalmente del uso que hagamos de estas tecnologías.

Existen en el país muchas radios FM que transmiten por streaming vía internet, sin embargo, aplicando las dos tecnologías de radiodifusión y streaming pueden llegar no solo a las personas de una localidad sino también a miles de personas alrededor del mundo, ampliando así sus tecnologías y ampliando su utilidad de nuevas tecnologías, algunas de estas radios concesionadas por Arcotel transmiten al mundo mediante streaming vía internet. Es por esto por lo que en la actualidad en la tecnología FM específicamente en las empresas de radiodifusión aún sigue habiendo postulaciones de profesionales ingenieros en telecomunicaciones altamente capacitados en tecnologías de transmisión y recepción de señales en frecuencia modulada.

Los ingenieros en telecomunicaciones necesitan conocer cómo funcionan las tecnologías de radiodifusión FM y de streaming de audio sin embargo el laboratorio de telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil no cuenta con bancos de prácticas para la investigación sobre la radiodifusión FM, a pesar de contar con equipamiento como las

radios definidas por software y antenas de broadcasting FM.

Este proyecto de tesis trata sobre el diseño e implementación de un banco de pruebas de laboratorio aplicadas a la radiodifusión FM utilizando software Opensource GNURadio y radios definidas por software (SDR) para el uso en el laboratorio de telecomunicaciones en beneficio de los estudiantes de la carrera de ingeniería de telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil.

Abstract

Year	Students	Technical Project Manager	Item of project of titulation
2022	Vanessa Stephania Pazmiño Quiguango Luis Erick Espinoza Pionce	Ing. Hólger Santillán MSc.	Design and implementation of a laboratory test bench applied to FM broadcasting using open-source software GNURadio, Raspberry PI and software-defined radio

Wireless communications technologies have developed in recent years, giving rise to new technological systems such as LTE Advanced, 5G, WIFI6, IEEE 802.11ax, Zigbee and many more that are widely used in our daily lives for broadband services, however, other wireless technologies such as FM have not had important developments in recent years since the current trend is the transmission of audio streaming via the internet, these new technologies that will be applied will depend fundamentally on the use we make of these technologies.

There are many FM radios in the country that transmit by streaming via the Internet, however, applying the two technologies of broadcasting and streaming can reach not only the people of a locality but also thousands of people around the world, thus expanding their technologies and expanding their usefulness of new technologies, some of these radios concessioner by Arcotel transmit to the world through streaming via the Internet. That is why today FM technology and broadcasting companies still have applications from professional telecommunications engineers highly trained in fm frequency signal transmission and reception technologies.

Telecommunications engineers need to know how FM broadcasting and audio streaming technologies work; however, the telecommunications laboratory of the Salesian Polytechnic University in Guayaquil does not have practice banks for research on FM broadcasting, despite

having equipment such as software-defined radios and FM broadcasting antennas.

This thesis project deals with the design and implementation of a laboratory test bench applied to FM broadcasting using open source GNURadio software and radios defined by SDR software for use in the telecommunications laboratory for the benefit of students of the telecommunications engineering career of the Salesian Polytechnic University headquarters Guayaquil.

Índice general

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN.....	II
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN	IV
Dedicatoria	VI
Dedicatoria	VII
Resumen del proyecto.....	VIII
Abstract.....	X
Índice general.....	XII
Índice de figuras.....	XV
Índice de tablas	XVII
Introducción.....	XVIII
1. El problema	1
1.1. Descripción del problema	1
1.2. Antecedentes	2
1.3. Importancia y alcance	2
1.4. Delimitación	2
1.5. Beneficiarios de la propuesta	3
1.6. Justificación	3
1.7. Innovación e impacto del proyecto	3
1.8. Objetivos de la investigación	4
1.8.1. Objetivo general.....	4
1.8.2. Objetivos específicos	4
2. Fundamentos teóricos	5
2.1. Radiodifusión Sonora y Televisión abierta.....	5
2.1.1. Radiodifusión Sonora	5
2.1.2. Radiodifusión de Televisión.....	5
2.1.3. Servicios de Radiodifusión Sonora y de Televisión Abierta	5
2.2. Bandas de frecuencias.....	5
2.2.1. Frecuencias principales.....	5
2.2.2. Frecuencias auxiliares.....	5

2.3.	Canalización de la banda de FM	6
2.3.1.	Distribución de frecuencias	6
2.4.	Área de cobertura	6
2.4.1.	Área de cobertura principal	6
2.4.2.	Área de cobertura secundaria o de protección	6
2.5.	Características técnicas:	7
2.5.1.	Ancho de banda	7
2.5.2.	Frecuencias de banda base para audio	7
2.5.3.	Separación entre portadoras	7
2.5.4.	Porcentaje de modulación	7
2.5.5.	Potencia de operación	7
2.6.	Relaciones de protección señal deseada / señal no deseada	8
2.7.	RTL-SDR	8
2.8.	Bosquejo Básico del RTL-SDR	9
2.9.	Aplicación informática para el RTL-SDR	10
2.10.	Dispositivo USRP Ettus	11
2.10.1.	Opciones de SDR autónoma o conectadas al host	12
2.10.2.	Dispositivo Autónomo de Radio Definido por Software USRP	13
2.10.3.	Bundle de Dispositivo de Radio Definido por Software	14
2.11.	Amplificador Minicircuits 50 Watts	14
2.11.1.	Cómo funcionan los amplificadores FM	16
2.12.	GNU Radio	17
2.12.1.	Tipo variable	18
2.13.	Raspberry Pi	19
2.13.1.	Raspberry Pi OS	19
2.14.	Cómo instalar Raspberry Pi OS	20
2.14.1.	Qué es NOOBS	20
2.15.	Función de Raspberry Pi	21
2.16.	Antena Yagi FM	22
2.16.1.	Reflector	22
2.16.2.	Director	22
2.16.3.	Ganancia	23
2.16.4.	Pérdida de señal	24
2.16.5.	Relación Señal – Ruido	25

3.	Marco metodológico	26
3.1.	Título de la propuesta.....	26
3.2.	Justificación	26
3.3.	Metodología	26
3.4.	Descripción de la propuesta	26
3.4.1.	Práctica 1: Introducción a la programación por bloques en software Opensource GNURadio.....	30
3.4.2.	Práctica 2: Configuración e implementación de Transmisor FM con GNURadio, USRP, amplificador FM y antena FM.....	30
3.4.3.	Práctica 3: Configuración e implementación de Receptor FM con RTL-SDR, Raspberry PI	30
3.4.4.	Práctica 4: Configuración de analizador de espectros con RTL-SDR y Raspberry PI.....	31
3.4.5.	Práctica 5: Configuración de APP SDR touch y RTL-SDR para usarlo como analizador de espectros.	31
4.	Resultados	32
5.	Análisis de resultados.....	42
5.1.	Práctica #1: Introducción a la programación por bloques en software Opensource GNURadio.....	42
5.2.	Práctica #2: Configuración e implementación de Transmisor FM con GNURadio, USRP, amplificador FM y antena FM.....	43
5.3.	Práctica #3: Configuración e implementación de Receptor FM con RTL- SDR, Raspberry PI	43
5.4.	Práctica #4: Configuración de analizador de espectros con RTL-SDR y Raspberry PI.	44
5.5.	Práctica #5: Configuración de APP SDR touch y RTL-SDR para usarlo como analizador de espectros.	46
6.	Conclusiones.....	48
7.	Recomendaciones.....	49
	Bibliografía	50
	ANEXOS	53

Índice de figuras

Figura 2.1 RTL-SDR.....	8
Figura 2.2 Bosquejo del RTL-SDR	10
Figura 2.3 Modelo USRP.....	12
Figura 2.4 Dispositivo definido por Software USRP	13
Figura 2.5 Bundle de Dispositivo de Radio Definido por Software	14
Figura 2.6 Amplificador Minicircuits 50 Watts	14
Figura 2.7 Función de un amplificador FM.....	16
Figura 2.8 GNU Radio.....	17
Figura 2.9 Tarjeta Raspberry Pi.....	19
Figura 2.10 Raspberry Pi OS.....	20
Figura 2.11 Elementos de Raspberry Pi	22
Figura 2.12 Antena Yagi/FM.....	23
Figura 3.1 Banco de pruebas de TX y RX FM broadcasting	27
Figura 3.2 Diagramas de bloques del banco de pruebas.....	28
Figura 3.3 Banco de pruebas en laboratorio de telecomunicaciones	29
Figura 4.1 Conexión de Amplificador con USRP	32
Figura 4.2 Fuente de poder para amplificador RF	33
Figura 4.3 Antena de cono para FM Tx	33
Figura 4.4 Abrazadera de antena y pedestal	34
Figura 4.5 Antena con pedestal.....	35
Figura 4.6 Conexiones del banco de pruebas del lado del transmisor	36
Figura 4.7 Pruebas de conectividad del transmisor	36
Figura 4.8 Conectividad del transmisor del banco de pruebas.....	37
Figura 4.9 Raspberry PI y RTL-SDR	37
Figura 4.10 Conexión de antena a RTL-SDR	38
Figura 4.11 Inicio de OpenWebRX	38
Figura 4.12 OpenWebRX	39
Figura 4.13 Recepción en OpenWebRX.....	39
Figura 4.14 Pruebas de recepción con celular.....	40

Figura 4.15 Pruebas de recepción.....	40
Figura 4.16 Espectro de señal visto desde GNURadio.....	41
Figura 5.1 Banco de pruebas FM	42
Figura 5.2 Diagrama de bloques del transmisor FM	43
Figura 5.3 Conexión de RTL-SDR y Raspberry PI.....	44
Figura 5.4 OpenWebRX	45
Figura 5.5 Receptor FM.....	45
Figura 5.6 APP SDR touch.....	46
Figura 5.7 Conexión de Celular y RTL-SDR	47

Índice de tablas

Tabla 2.1 Relación de señal deseada.....	8
Tabla 2.2 Aspecto de los sistemas SDR.....	9
Tabla 2.3 Modelos y sus procesadores	13
Tabla 2.4 Parámetros del amplificador FM	15

Introducción

En el Ecuador el ente regulador de las telecomunicaciones Arcotel ha designado varias concesiones de frecuencias FM tanto para la empresa privada como pública, por lo tanto, estas empresas requieren de profesionales altamente capacitados en este campo, para el correcto funcionamiento del envío de señales de audio mediante la frecuencia modulada.

En este trabajo de investigación se dispone implementar un banco de pruebas de laboratorio aplicada al uso y práctica de la tecnología FM. Utilizando software Opensource como GNURadio, Raspberry PI, y radio definida por software.

Cabe destacar que muchas de estas radios concesionadas hacen su transmisión broadcast vía radio y también vía streaming gracias a los servicios web y el internet. Varias radios con licencia de Arcotel se transmiten al mundo a través de Internet. Esta es la razón por la que hoy en día las empresas de tecnología y radiodifusión de FM todavía necesitan ingenieros de telecomunicaciones profesionales que estén bien capacitados en la transmisión y recepción de señales de FM con conocimientos en radiofrecuencia y en streaming.

El diseño del sistema solo utiliza el software GNURadio de código abierto, radios definidos por software como el USRP Ettus, RTL-SDR y hardware de bajo costo como Raspberry PI. Estos hardware trabajarán con antena de radio FM para buscar desarrollar habilidades en los estudiantes de ingeniería en telecomunicaciones.

La principal motivación para este trabajo es que los ingenieros de telecomunicaciones deben comprender los principios de funcionamiento de la tecnología de transmisión de audio y radio FM, adicionando así un aporte en el laboratorio de telecomunicaciones de la Universidad Politécnica Salesiana en la sede de Guayaquil con un banco de prácticas para la investigación de la radiodifusión FM.

El libro se ordena de la siguiente manera: En el capítulo 1 explica el problema y sus objetivos. El capítulo 2 presentará las bases teóricas de FM, GNURadio, Raspberry PI y radio definida por software para la transmisión de FM.

El tercer capítulo trata sobre el marco metodológico, tipo de investigación, diseño de la investigación, título de la propuesta, motivo y descripción de la propuesta.

En el Capítulo 4, se indica los resultados de la investigación. El quinto capítulo es el análisis de resultados, conclusiones y recomendaciones.

Finalmente, en los anexos se desarrolla una práctica para la implementación de FM broadcasting software Opensource GNURadio, Raspberry PI y radios definidas por software.

1. El problema

1.1. Descripción del problema

La tecnología de transmisión de radio FM sigue siendo un método de comunicación ampliamente utilizado por los ecuatorianos. Por esta razón las empresas de tecnología y radiodifusión de FM de hoy en día todavía necesitan ingenieros de telecomunicaciones profesionales que estén bien capacitados en la transmisión y recepción de señales de FM.

En la actualidad, hay 774 estaciones de radio comerciales privadas de radio FM concesionadas en el Ecuador, 189 estaciones de radio FM de servicio público concesionadas y estaciones de radio FM concesionadas de servicio público comunitario según Arcotel (Según datos de la Agencia Ecuatoriana de Regulación y Control de las Telecomunicaciones), a febrero de 2021, un total de 94 estaciones de radio brindarán a todos los ecuatorianos 1057 estaciones de radio FM con licencia para transmisión de audio FM por radio en todo el país.

Actualmente, en el Laboratorio de Telecomunicaciones de la Universidad Politécnica Salesiana en la sede de Guayaquil carece de pruebas de laboratorio para la práctica de tecnologías de radiodifusión FM, sin embargo, se cuenta con equipos de radio definidos por software que actualmente pueden ser utilizados por estudiantes que cursan la carrera de ingeniería en telecomunicaciones, para desarrollar prácticas de la capa física de redes inalámbricas. Estas herramientas y equipos pueden ser utilizado para el aprendizaje de la tecnología de frecuencia modulada.

En este contexto, el trabajo de investigación de este proyecto incluye el desarrollo de un banco de pruebas de laboratorio para la investigación, análisis y prueba de tecnología de transmisión de FM y flujos de audio, utilizando software y hardware como el USRP, RTL-SDR y Raspberry PI, combinándolos con antenas de transmisión de FM buscan desarrollar habilidades y destrezas en los estudiantes de ingeniería de telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil.

1.2. Antecedentes

En los últimos años, se ha desarrollado la tecnología de comunicación inalámbrica y han surgido nuevos sistemas tecnológicos como LTE Advanced, 5G, WIFI6, IEEE 802.11ax y Zigbee, los cuales se utilizan ampliamente en los servicios de banda ancha en nuestra vida diaria. Sin embargo, otras tecnologías inalámbricas las transmisiones FM de audio aún se mantienen debido a su popularidad, cero costo para el usuario y fácil recepción de servicio.

En la actualidad, tener conocimientos avanzados en la tecnología FM ayuda a resolver problemas en la transmisión y recepción de señales de frecuencia modulada, siendo de vital importancia para las empresas de radiodifusión disponer de personal altamente calificado en conocimientos de radiofrecuencia FM y TV.

1.3. Importancia y alcance

Con la contribución de este proyecto de investigación, los estudiantes de la carrera de ingeniería en telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil podrán desarrollar prácticas relacionadas a la transmisión y recepción de señales broadcasting de FM. Los cuales a futuro podrán poner en práctica en sus puestos laborales dentro de una estación de radio o un canal de televisión.

1.4. Delimitación

Espacial: Debido a la emergencia sanitaria por motivo de la pandemia del COVID-19 y siguiendo las indicaciones de la dirección de carrera de Ingeniería Electrónica de la Universidad Politécnica Salesiana, sede Guayaquil, el diseño e implementación del banco de pruebas de laboratorio aplicado a la radiodifusión FM y streaming de audio se desarrollará en la vivienda de la autora de tesis Vanessa Pazmiño ubicada en el centro de la ciudad de Guayaquil.

Temporal: El tiempo estimado para realizar el diseño, implementación y pruebas de funcionamiento del proyecto es de 5 meses según el cronograma elaborado.

Académica: En este proyecto de investigación se tiene como resultado un banco de pruebas de radio FM que servirá como herramienta fundamental para el desarrollo de competencias y habilidades de los futuros ingenieros en telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil. A través de este proyecto se ponen en práctica los conocimientos adquiridos en las disciplinas de electrónica, comunicaciones digitales, redes inalámbricas y medios de transmisión.

1.5. Beneficiarios de la propuesta

La implementación de este proyecto fortalecerá los conocimientos adquiridos en los estudiantes de la carrera de ingeniería en telecomunicaciones e investigadores de las tecnologías de radiodifusión específicamente empleando equipos de radio definida por software.

Estas herramientas y equipos se pueden utilizar para practicar en detalle acerca de la tecnología FM, con el fin de ampliar sus conocimientos y mejorar el desarrollo de habilidades en los estudiantes de la carrera de ingeniería en telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil.

1.6. Justificación

Mejorar las habilidades prácticas y técnicas de los estudiantes de ingeniería en telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil para que puedan desempeñarse mejor en el futuro en empresas del ramo de radiodifusión o canales de televisión. El uso de radios definidas por software en el Laboratorio de Telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil también facilitará la investigación en el campo de la radiodifusión.

1.7. Innovación e impacto del proyecto

En este proyecto de investigación, se recomienda utilizar el software de código abierto GNURadio, Raspberry pi y radio definida por software para diseñar e implementar un banco

de pruebas de laboratorio para radiodifusión FM.

El proyecto también impulsa la investigación en el campo de la radiodifusión mediante el uso de equipos de radio definidos por software del laboratorio de telecomunicaciones de la Universidad Politécnica Salesiana de Guayaquil.

1.8. Objetivos de la investigación

1.8.1. Objetivo general

Diseñar e implementar un banco de pruebas de laboratorio aplicado a la radiodifusión FM utilizando software Opensource GNURadio, Raspberry PI y Radios Definidas por Software.

1.8.2. Objetivos específicos

- Diseñar un banco de prácticas aplicadas a la tecnología de modulación FM mediante radios definidas por software RTL-SDR y USRP B205.
- Configurar NI USRP para la transmisión de audio en FM utilizando software Opensource GNURadio y antena de transmisión en la banda FM.
- Implementar un banco de recepción FM con RTL-SDR, pantalla touch, Raspberry PI y batería de litio para la monitorización y análisis de señales FM.
- Desarrollar guía de pruebas para prácticas con amplificador FM para la transmisión de audio a larga distancia y monitorizar potencia mediante analizador de espectros con RTL-SDR.

2. Fundamentos teóricos

2.1. Radiodifusión Sonora y Televisión abierta

2.1.1. Radiodifusión Sonora

Es un servicio de radiocomunicación que permite la emisión de programas de audio a distancia, cuya misión es recibida directamente por el público [1].

2.1.2. Radiodifusión de Televisión

Es un servicio de radiocomunicación que permite la transmisión remota de programas de imagen y sonido, y su transmisión es recibida directamente por el público [1].

Ç

2.1.3. Servicios de Radiodifusión Sonora y de Televisión Abierta

En el plan nacional de frecuencias se asignan bandas de frecuencia para los servicios de radiodifusión sonora de onda media, onda corta y FM; radiodifusión de TV (analógica o digital) VHF o UHF [2].

2.2. Bandas de frecuencias

Como parte del servicio de radiodifusión sonora FM, se han establecido las siguientes bandas de frecuencia [1].

2.2.1. Frecuencias principales

La banda de frecuencia utilizada para el servicio de radiodifusión sonora FM está establecida por la banda de frecuencia 88-108MHz aprobada por el plan nacional de frecuencias [1].

2.2.2. Frecuencias auxiliares

Los utilizados para los radioenlaces auxiliares definidos en el apartado 2.5 y se otorgan en el

plan nacional de frecuencias [1].

Los enlaces auxiliares se pueden proporcionar a través de infraestructura propia, sin necesidad de prestar servicios a terceros a través de operadores de servicios de telecomunicaciones legalmente autorizados[1].

2.3. Canalización de la banda de FM

Se tendrá en cuenta el rango de 100 frecuencias con un intervalo de 200 KHz, las cuales se encuentran enumeradas del 1 al 100, y comienza con la primera frecuencia desde 88,1 MHz [1].

2.3.1. Distribución de frecuencias

La asignación de frecuencias se implementará en áreas de operación independientes, un conjunto de frecuencias correspondientes a las operaciones normales y locales de la central eléctrica, de manera que se minimicen las interferencias. Y canales adyacentes [2].

2.4. Área de cobertura

2.4.1. Área de cobertura principal

Correspondiente a la intensidad de campo de la ciudad o pueblo a ser servido, su potencia de campo es igual o mayor que la potencia de campo mínima a proteger [3].

2.4.2. Área de cobertura secundaria o de protección

Corresponde al área alrededor de la ciudad denotada como área de cobertura primaria, y su potencia de campo se encuentra entre los datos o valores correspondientes a los límites de las áreas de cobertura primaria y secundario [4].

2.5. Características técnicas:

Los parámetros técnicos de instalación y emisión de las estaciones de radiodifusión sonora FM deben cumplir con esta norma y cumplir con:

2.5.1. Ancho de banda

El ancho de banda estéreo es de 220 kHz y el ancho de banda mono es de 180 kHz, con una tolerancia de hasta el 5% [1].

2.5.2. Frecuencias de banda base para audio

La frecuencia es de 50 Hz a 15 kHz

2.5.3. Separación entre portadoras

Determinado por el grupo de frecuencias correspondiente a cada área de trabajo independiente.

2.5.4. Porcentaje de modulación

Los siguientes valores no deben excederse en picos frecuentes:

- Para sistemas mono o estéreo, solo 100%.
- Si utilizan subportadoras: 95%.
- Si utilizan dos o más subportadoras: 100%.

2.5.5. Potencia de operación

Es la fuerza de salida del dispositivo de transmisor en vatios (watts) que se provee al procedimiento radiante [2].

2.6. Relaciones de protección señal deseada / señal no deseada

Tabla 2.1 Relación de señal deseada

SEPARACIÓN DESEADA ENTRE PORTADORAS	SISTEMAS ESTEREOFONICO	SISTEMA MONOFONICO
0 KHz (co-canal)	37 dB	28 dB
200 kHz	7 dB	6 dB
400 kHz	-20 dB	-20 dB
600 kHz	-30 dB	-30 dB

Fuente: [3]

Como podemos apreciar en la tabla 2.1 la distancia mínima entre estaciones transmisoras se determinará de acuerdo con la relación de protección entre el mismo canal y los canales adyacentes en el área de cobertura autorizada [2].

2.7. RTL-SD

El precursor en la introducción del concepto de radio definida por software y Hardware (RTL-SDR) fue Antti Palosaari, quien colabora con LINUX a lo largo del siglo XXI, utilizando una plataforma de hardware única llamada radio para transmitir y recibir señales. Este concepto nació, dando paso al desarrollo de diversas aplicaciones para aplicaciones civiles y militares, porque las tarjetas de radio están integradas en varios dispositivos, como teléfonos móviles, computadoras, equipos de audio, etc.[4].



Figura 2.1 RTL-SDR

Fuente: [4].

Como se aprecia en la figura 2.1 el RTL-SDR se utiliza actualmente para la investigación y el desarrollo de aplicaciones, por lo que una organización ha desarrollado este concepto llamado WIRELESS INNOVATION FORUM. La organización define SDR como: "radio, en la que parte o todas las funciones de la capa física están definidas por software", la tecnología SDR trabaja en las capas inferiores del modelo OSI, especialmente en la capa física y la capa de enlace, porque controla algunas funciones que generalmente son procesadas por simulaciones electrónicas [4].

Tabla 2.2 Aspecto de los sistemas SDR

ASPECTO	FUNCIÓN
Hardware	Conjunto de equipos y componentes que forman un dispositivo de radio.
Software	Conjunto de servicios e interfaces a través de los cuales las aplicaciones de software interactúan con el hardware.
Aplicaciones y servicios	Defina capas de aplicaciones y servicios.
Usuario	Controla las aplicaciones y la transferencia de datos.

Fuente: [4]

Es trascendental referirse a los aspectos básicos que constituyen los fragmentos de un sistema SDR, cuáles son y qué ocupaciones desempeñan. Un sistema SDR está constituido por cuatro aspectos primordiales, los mencionados en la tabla 2.2 [4].

2.8. Bosquejo Básico del RTL-SDR

Desde su aparición, el SDR ha estado en su concepto y arquitectura. Sin embargo, su desarrollo se basa en un esquema básico, a saber, consta de tres partes: parte de RF, parte de FI y parte de banda de frecuencia base, como se observa en la figura 2.2 [4].

Diagrama de bloques simplificado de NooElec RTL-SDR

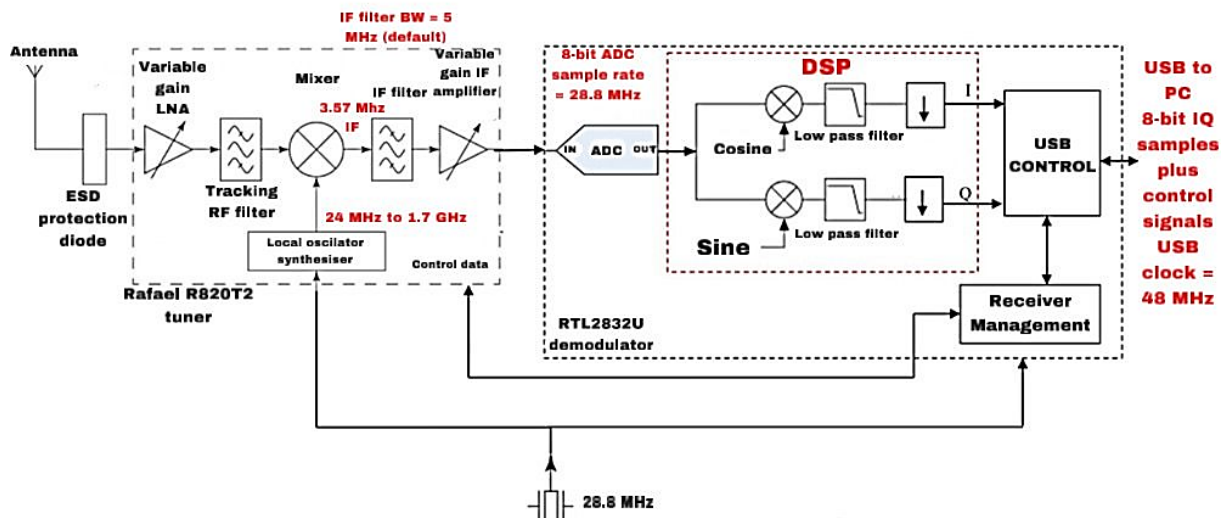


Figura 2.2 Bosquejo del RTL-SDR
Fuente: [5]

Es importante especificar los dos conceptos que forman parte de la arquitectura, por ejemplo: El Front End es la parte que contiene el concepto de radiofrecuencia, que se utiliza para la transmisión y recibir la señal; otro concepto es el back end, que es una función de procesamiento de señal, también incluye parte IF [4].

2.9. Aplicación informática para el RTL-SDR

Se han desarrollado decenas de aplicaciones para el dongle RTL-SDR, que se clasifican como aplicaciones de uso general, lo que les permite ser utilizados como receptores de radio de banda ancha ordinarios. La diferencia entre ellos radica en sus respectivas funciones específicas, en el rendimiento real, en cómo funciona el sistema operativo, independientemente de si liberan el código fuente, si son gratuitos o de pago [4].

Entre muchas aplicaciones, SDRSharp se considera una de las aplicaciones más completas en esta categoría: SDR # (Windows / Linux con Mono). Se llama SDRSharp, y es el más

popular en la actualidad, ya que permite recibir y de modular varios tipos de señales, como NFM, WFM, AM, USB, LSB, DSB, CW. Es muy fácil de operar y se han agregado nuevas funciones (complementos), lo que la convierte en una excelente herramienta [4].

Permite la visualización simultánea de bandas de espectro de hasta 3,2 MHz, la grabación de toda la banda y el audio demodulador de las estaciones de radio seleccionadas en I / Q, la conexión remota a través de una red de datos, la exploración del espectro y la creación de datos de base de datos con las siguientes frecuencias [4].

Estaciones de radio de interés, modificar el filtro de paso de banda, etc. Su frecuencia de operación se puede ampliar agregando un convertidor ascendente y un convertidor descendente. Se recomienda utilizar un PC de cuatro núcleos y 4 GB de RAM para obtener resultados satisfactorios en todas sus características [4].

2.10. Dispositivo USRP Ettus

Es una versión mejorada de USRP porque contiene un FPGA más grande, y está diseñado para aplicaciones que requieren este tipo de comunicación. Con un rápido desarrollo, este equipo hace posible varias aplicaciones de comunicación por radio, como radiodifusión, televisión analógica y digital, redes teléfono celular, etc. [6].

Proporcione una arquitectura de RF definida por software para diseñar, crear prototipos e implementar rápidamente sistemas inalámbricos con funciones de procesamiento de señales personalizadas [6].

En la figura 2.3 el equipo de radio definido por software de USRP es un dispositivo de radiofrecuencia reconfigurable, que incluye una combinación de un procesador basado en host, FPGA y terminal frontal de radiofrecuencia [6].



Figura 2.3 Modelo USRP
Fuente: [7].

El equipo de radio definido por software de USRP incluye opciones que van desde opciones de bajo costo con características FPGA fijas hasta radios de alta gama con FPGA abiertos grandes y ancho de banda instantáneo. Estos dispositivos se pueden utilizar para aplicaciones tales como múltiples entradas múltiples salidas (MIMO) y pruebas de referencia LTE / Wifi, inteligencia de señales y sistemas de radar [6].

2.10.1. Opciones de SDR autónoma o conectadas al host

El USRP está diseñado como un dispositivo periférico de PC para conectar software al espectro electromagnético. Desde el primer USRP, las aplicaciones han seguido evolucionando y muchas aplicaciones requieren un procesador integrado [6].

Si su aplicación tiene un SDR que se distribuye físicamente desde un sistema de control centralizado o se implementa automáticamente, es posible que necesite esta configuración independiente [6].

Si los sistemas autónomos son el requisito clave, deberá decidir si se trata del sistema multiprocesador en chip Xilinx Zynq (MPSOC) o del sistema RF en chip (RFSOC). La Tabla 2.3 detalla diferentes modeladores y los procesadores de los cuales están compuestos [6].

Tabla 2.3 Modelos y sus procesadores

Modelo de radio	Procesador integrado
USRP N320, USRP N321, USRP N310	Xilinx Zynq 7100 MPSOC
USRP E31X	Xilinx Zynq 7020 MPSOC
USRP E320	Xilinx Zynq 7045 MPSOC
NI Ettus USRP X410	Xilinx Zynq UltraScale+ RFSoC ZU28DR
USRP 2974	Intel Core i7 6822EQ (2 GHz Quad Core)

Fuente: [7].

2.10.2. Dispositivo Autónomo de Radio Definido por Software USRP

La FPGA y el procesador interno se utilizan para proporcionar un control determinista del transceptor para construir rápidamente prototipos de sistemas de comunicación inalámbrica de alto rendimiento [8].

Como se puede apreciar en la Figura 2.4 se muestra el dispositivo autónomo de radio definido por software de la USRP que consta de un procesador interno, una FPGA y una interfaz de RF. Puede programar el dispositivo usando LabVIEW Communications Suite o un flujo de trabajo de software de código abierto, dependiendo del sistema operativo que elija [8].

Puede configurar el dispositivo con los sistemas operativos NI Linux Real-Time, Linux Fedora o Linux Ubuntu en tiempo real. Los dispositivos autónomos de radio definidos por software de USRP le permiten crear prototipos de diversas aplicaciones de investigación avanzada, como LTE o simulación de dispositivos autónomos 802.11; desarrollo de algoritmos de control de acceso a medios (MAC); entrada múltiple, salida múltiple (MIMO); redes heterogéneas; relé LTE; muestreo de RF [8].



Figura 2.4 Dispositivo definido por Software USRP
Fuente: [8].

2.10.3. Bundle de Dispositivo de Radio Definido por Software

El paquete de equipo de radio definido por software ayuda a los estudiantes a experimentar con señales del mundo real usando el software LabVIEW y el hardware USRP [9] como se muestra en la Figura 2.5.



Figura 2.5 Bundle de Dispositivo de Radio Definido por Software

Fuente: [9]

El paquete de radio definido por software es un paquete de hardware y software que incluye equipos de radio definidos por software de periféricos de radio definidos por software universal (USRP), soporte de software LabVIEW y cursos listos para usar. El paquete también incluye componentes de la estación de laboratorio, diseñados para ayudar a los estudiantes a adquirir experiencia práctica en enlaces de comunicación en tiempo real entre dispositivos de radio definidos por software de USRP [9].

2.11. Amplificador Minicircuits 50 Watts

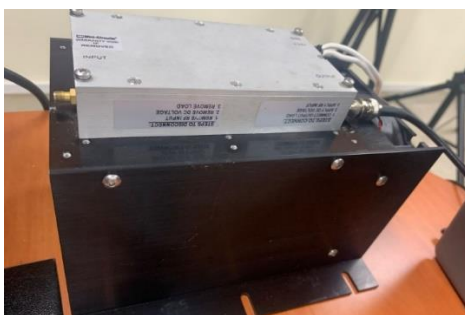


Figura 2.6 Amplificador Minicircuits 50 Watts

Fuente: [Los Autores]

En la figura 2.6 se observa el amplificador minicircuits de 50 Watts el cual tiene un rango de frecuencia de 50 a 500 Mhz adecuado para amplificar señales FM.

La impedancia de entrada del equipo es de 50 ohmios por tal motivo todo el sistema de antena, cables y conectores deben ser de esta impedancia. Amplificador de clase A, con conectores de entrada y salida del tipo sma. En la tabla 2.4 se detallan las características principales del amplificador FM.

Tabla 2.4 Parámetros del amplificador FM
Fuente: [15].

Parameter	Condition (MHz)	ZHL-50W- 52-S+			Units
		Min.	Typ.	Max.	
Frequency Range		50		500	MHz
Gain		47	52	—	dB
Gain Flatness		—	±1.2	±1.8	dB
Output Power at 1dB compression	200 - 300 50 - 500	+46.0 +44.0	+47.5 +46.5	— —	dBm
Saturated Output Power at 3dB compression	200 - 300 50 - 500	+47.0 +45.5	+49.0 +48.0	— —	dBm
Noise Figure		—	4.5	7.0	dB
Output third order intercept point		—	+59	—	dBm
Input VSWR		—	1.75	—	:1
Output VSWR		—	2.5	—	:1
DC Supply Voltage		—	24	25	V
Supply Current		—	—	9.3	A

2.11.1. Cómo funcionan los amplificadores FM

En la Figura 2.7 se muestra la función de un transmisor FM el cual utiliza la frecuencia FM disponible en la radio para reproducir música a través del altavoz y convierte la salida de audio del dispositivo en una señal de radio FM.

De esta manera, puede escuchar música de fuentes distintas de la radio del automóvil integrada, como CD, puertos USB e incluso cintas. Esto aprovecha la potencia y el volumen del sistema de altavoces y el control de radio, lo que facilita su manejo mientras conduce [10].

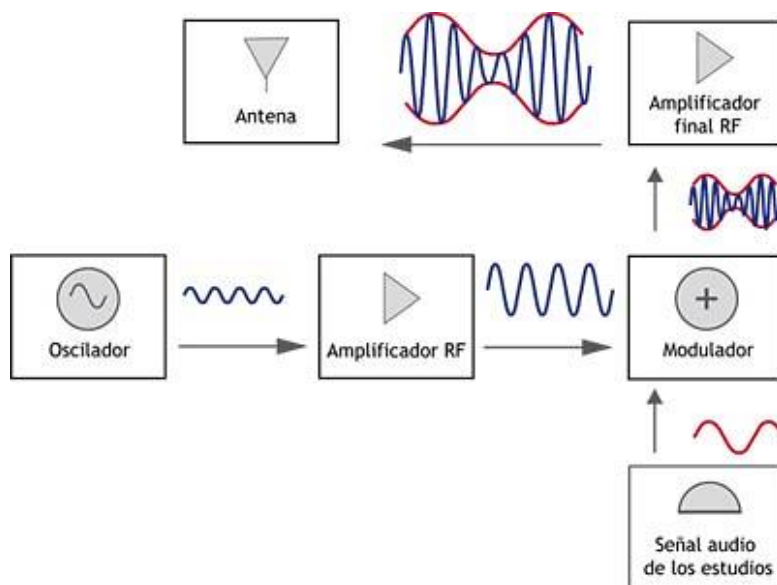


Figura 2.7 Función de un amplificador FM
Fuente: [11]

La primera pieza que encontramos es el oscilador, que como su nombre indica oscilará o vibrará para producir una señal constante de alta frecuencia, que usaremos como portadora.

Dado que la señal generada por el oscilador es muy pequeña, necesitamos usar un amplificador de radiofrecuencia (RF) para aumentarla. Ahora hemos preparado la portadora para recibir la señal modulada del estudio, es decir, la señal a modular. El modulador conecta el modulador y la portadora. Esto se puede hacer en frecuencia (FM) o amplitud (AM), dependiendo del transmisor.

Una característica importante de este dispositivo es que no requiere ningún tipo de conexión física con la radio y funciona vía Bluetooth, sin embargo, debe estar conectado a un encendedor de cigarrillos, el cual proporcionará energía [11].

2.12. GNU Radio

GNU Radio es un software de código abierto para aplicaciones SDR que proporciona una gran cantidad de procesamiento de señales, interfaces gráficas de usuario y bibliotecas de utilidades. Cada aplicación desarrollada se construye a partir de bloques conectados a través de puertos, y cada bloque realiza ciertos aspectos del procesamiento de señales [12], como se visualiza en la Figura 2.8.

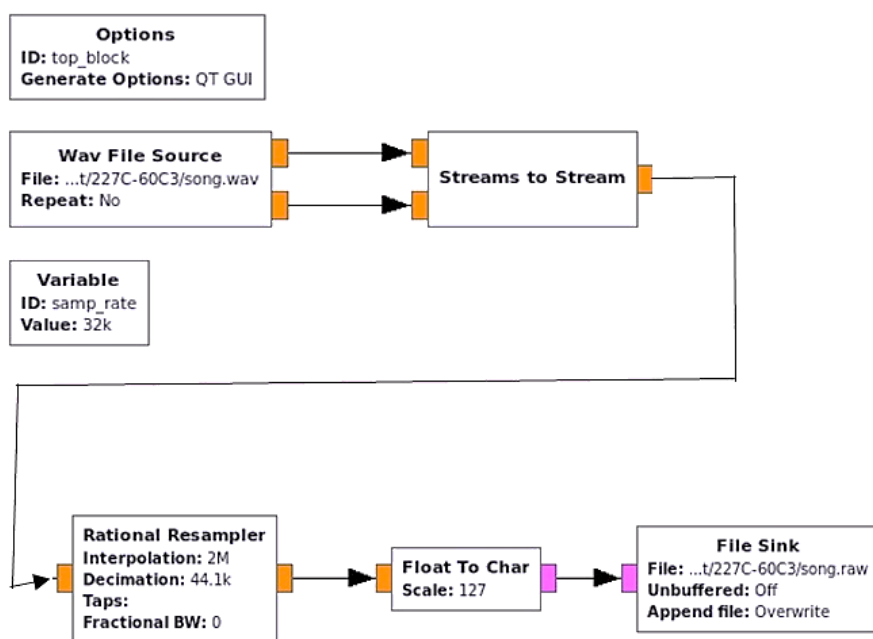


Figura 2.8 GNU Radio
Fuente: [13].

Área de trabajo: Es el lugar de trabajo donde se colocan los bloques que se utilizan para desarrollar aplicaciones (como fuentes, sumideros, moduladores, demoduladores, etc.).

Barra de herramientas: La barra de herramientas proporciona varias opciones para ser utilizadas, como editar, guardar, ejecutar, copiar, cortar y también mostrar la situación en la que el programa no se puede ejecutar debido a algunos errores en la compilación.

Bibliotecas: En la biblioteca, encontrará varias bibliotecas, como moduladores analógicos y digitales, tipos de datos, fuentes, sumideros, codificadores, entre ellos, todos estos bloques son propios de GNU Radio y pueden utilizarse en aplicaciones.

Consola: En esta parte de la pantalla se muestra la ejecución del proceso de envío a través de un mensaje, el cual también brinda información sobre cualquier error en la ejecución del programa.

Ventana de variables: Le permite ver las variables que se están utilizando y también le permite crear nuevas variables [12].

2.12.1. Tipo variable

Dependiendo de la aplicación a desarrollar en GNU Radio, se pueden manejar los siguientes tipos de variables:

Int: Datos de tipo entero que constan de 4 bytes.

Float: Datos de coma flotante que constan de 4 bytes.

Complejo: Datos de tipo complejo compuestos por una parte real y una parte imaginaria compuesta por 8 bytes.

Byte: Datos compuestos por 1 byte

Corto: Datos de tipo entero que constan de 2 bytes.

Para encontrar el tipo de una variable en GNU Radio, debe mirar la opción Ayuda y la barra

de menú en Tipos [12].

2.13. Raspberry Pi

Como muestra la Figura 2.9 la Tarjeta Raspberry Pi es una computadora compacta de bajo costo diseñada para ser desarrollada de manera que todos los usuarios puedan usar la informática. Otra característica de Raspberry Pi es que se usa ampliamente para el desarrollo de pequeños prototipos y la capacitación de equipos electrónicos y de computadoras escolares [14].



Figura 2.9 Tarjeta Raspberry Pi
Fuente: [15].

Todos los diseños de Raspberry Pi se basan en hardware gratuito y también se utilizan habitualmente sistemas operativos gratuitos basados en GNU / Linux.

Raspberry Pi OS (anteriormente Raspbian) fue desarrollado para esta computadora y es una versión personalizada de Debian. Además, se pueden instalar diferentes sistemas operativos, como veremos más adelante. La Fundación Raspberry Pi lanzó la primera Raspberry Pi en febrero de 2012 [15].

2.13.1. Raspberry Pi OS

Como cualquier ordenador del mercado, debemos instalar un sistema operativo para interactuar fácilmente con él. En comparación con las computadoras tradicionales, Raspberry

Pi tiene funciones limitadas y algunas funciones especiales. El sistema operativo Raspberry Pi, también conocido como Raspbian, ha sido "desarrollado" para este propósito.

El sistema operativo Raspberry Pi OS Figura 2.10 se basa en la distribución GNU / Linux, especialmente en Debian. Dado que el sistema operativo Debian es de código abierto, cualquiera puede personalizarlo e instalarlo sin pagar licencias ni regalías [15].

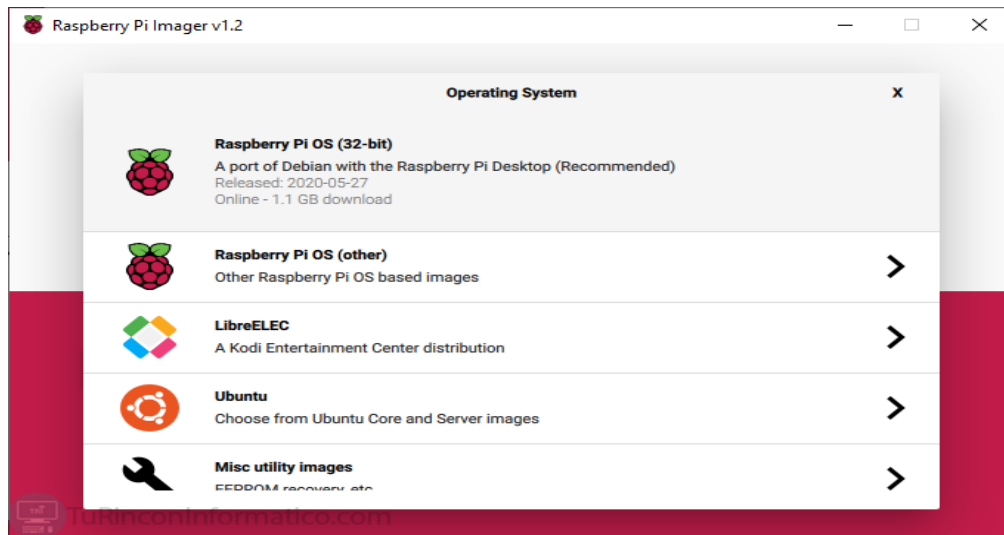


Figura 2.10 Raspberry Pi OS
Fuente: [16].

2.14. Cómo instalar Raspberry Pi OS

Si queremos instalar el sistema operativo oficial de este microordenador, primero debemos descargarlo de la web oficial de la Fundación Raspberry Pi. Esta página nos proporciona dos versiones diferentes de instalación del sistema operativo Raspberry Pi, a través de NOOBS o mediante imágenes, esta última es más adecuada para usuarios avanzados [17].

2.14.1. Qué es NOOBS

Es la versión del sistema operativo Raspberry Pi con asistente de instalación del sistema operativo. NOOBS es un acrónimo de software nuevo listo para usar. En este modo de instalación, tenemos la versión Pixel (con GUI) y la versión Lite [17].

La versión NOOBS del sistema operativo Raspberry Pi integra un método de configuración rápida del sistema operativo, instaló un navegador y un modo de recuperación de línea de comandos basado en la partición de rescate generada automáticamente. NOOBS Pixel pesa alrededor de 2 GB, mientras que NOBS Lite pesa solo 32 MB [15].

2.15. Función de Raspberry Pi

El funcionamiento es el mismo que el de una computadora de escritorio porque tiene los mismos componentes como se observa en la Figura 2.11 en un nivel básico. La placa Raspberry Pi se basa en la arquitectura ARM SoC con bajo consumo de energía y buen rendimiento. Vienen con un modelo de RAM, cuya capacidad varía de un modelo a otro [17].

Tienen múltiples salidas de video y se ha agregado un conector Jack de 4 polos en la versión más reciente para entrada de micrófono y salida de audio. Disponen de lector de tarjetas con sistema operativo y varios puertos USB. También cuenta con una gran cantidad de conectores GPIO y puede desarrollar una gran cantidad de proyectos [17].

Raspberry Pi 4 es la versión más moderna, con un conector Gigabit Ethernet y una tarjeta Wifi + Bluetooth integrada. Este modelo más moderno tiene un conector de alimentación tipo USB-C y dos conectores MIPI, uno para la pantalla y otro para la cámara [17].

Para que funcione, debemos alimentar la placa de circuito a través de un conector específico. Debemos cargar el sistema operativo que queramos en la tarjeta de memoria SD para poder disfrutar de todas sus funciones. Debemos conectar el teclado y el mouse, así como la pantalla de visualización de datos para interactuar con la placa [15].

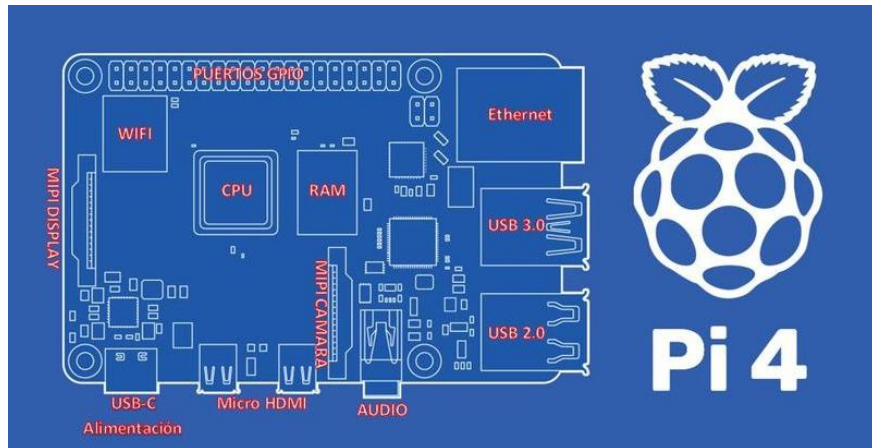


Figura 2.11 Elementos de Raspberry Pi
Fuente: [15].

2.16. Antena Yagi FM

Es una antena direccional, la presente invención mejora la antena tradicional, a través de una estructura simple dipolo, combinada con elementos parásitos denominados reflectores y directores, se obtiene una antena sencilla y de muy altas prestaciones [18] como se muestra en la Figura 2.12.

- El elemento de excitación. (Dipolos) Pueden estar activos o excitados, están conectados directamente a la línea de transmisión y reciben energía de la fuente.
- Elemento parasitario. No están conectados a la línea de transmisión y reciben energía mediante inductancia mutua. Estos elementos se dividen en reflectores y directores.

2.16.1. Reflector

El elemento parásito es más largo que el elemento de excitación. Reduce la fuerza de la señal en su dirección y aumenta la fuerza de la señal en la dirección del dipolo.

2.16.2. Director

El elemento parásito es más corto que su elemento de excitación. Aumenta (n) la intensidad

del campo a lo largo de su dirección y disminuye (n) la intensidad del campo hacia el reflector.

El elemento parásito puede ser un reflector o director que define la dirección máxima de radiación, mientras que el elemento activo es un dipolo de $\frac{1}{2}$ onda. El Yagi más simple consta de 3 elementos. Para las bandas de frecuencia VHF y UHF, este tipo de antenna utiliza 30 Hay un reflector y varias guías entre ellos [18].



Figura 2.12 Antena Yagi/FM
Fuente: [19]

Debido a su simplicidad y ancho de banda relativamente grande, la antenna Yagi-Uda se usa ampliamente en aplicaciones de transmisión de TV, estaciones de radioaficionados y en los últimos años se ha utilizado para otros tipos de servicios, como enlaces de radio puntuales. Listo para su uso en sistemas Wi-Fi, ya sea en la banda de frecuencia de 2,4 GHz o en la banda de frecuencia de 5,7 GHz, así como en otros sistemas, sus características de directividad hacen que tenga muy buen rendimiento [19].

2.16.3. Ganancia

En una antenna con 2 a 4 elementos, el intervalo aproximado es de $0,15$ a $0,2 \lambda$. En algunos casos, si el segundo director difiere del primero en $0,25 \lambda$, se puede obtener una ganancia mayor, incluso Mayor, el tercero y cuarto elementos son separados hasta el valor máximo de $0,4 \lambda$.

Otro factor importante es la ganancia, porque aumentará rápidamente cuando haya pocos elementos, y para más elementos, aumentará lentamente. Para el gráfico con 2 elementos (1

director, dipolo y un reflector), la ganancia es de 5 a 7 dB, 5 es aproximadamente 10 dB y 8 es aproximadamente 12 dB, por lo que la antena Yagi ya no está diseñada. De 12 a 15 elementos [18].

A medida que aumenta el número de directores, la reactancia de los directores aumenta y la amplitud de corriente entre ellos disminuye. Al mismo tiempo, la amplitud actual de los directores alejados del dipolo activo cae bruscamente.

Estas antenas direccionales de banda ancha están disponibles en dos versiones con 3 o 5 elementos. Están hechos de acero inoxidable y están diseñados para transmisores de potencia de salida media a alta. El diseño robusto de estas antenas las hace adecuadas para cualquier condición climática y vida útil.

Se utilizan materiales de alta calidad y se seleccionan en cada detalle: todos los aisladores están hechos de PTFE y los tornillos están hechos de acero inoxidable.

Las partes metálicas están conectadas a tierra. El modelo ADR3 también es adecuado para las bandas de frecuencia de 140-174 MHz y 174-215 MHz. Estas antenas se pueden desmontar en dos partes para reducir los costos de envío.

2.16.4. Pérdida de señal

Otro factor para considerar al seleccionar e instalar una antena es la pérdida de señal. La antena debe estar lo más alejada posible de objetos metálicos, chimeneas, paredes y ramas que absorben las ondas de radio para reducir la fuerza de la señal que llega a la antena. Una antena inestable o suelta provocará una atenuación de la señal.

Torres de comunicación, como transmisores y repetidores de teléfono, radio, televisión u otras torres de comunicación ubicadas entre la antena y el transmisor o repetidor deseado.

Las estructuras metálicas actuarán como "escudos" para resistir señales, como puentes o edificios que contienen un alto contenido de metal.

Si hay una gran resistencia en el circuito de la antena, también aumentará la pérdida de señal. Para reducir la resistencia, todas las uniones y conexiones deben soldarse cuidadosamente, y la antena y la línea de transmisión deben estar compuestas por dos conductores como máximo. como sea posible no combinar Al igual que la antena de mástil Como aislamiento, brinda protección a la señal para que no se filtre al suelo [18].

2.16.5. Relación Señal – Ruido

El ruido es importante al instalar una antena está compuesto por ondas de radio de múltiples frecuencias, que pueden ser generadas por equipos electrónicos o interferencias eléctricas naturales. Los dispositivos que a menudo generan ruido incluyen ascensores, refrigeradores, sistemas de encendido de automóviles, aspiradoras, lavadoras, ventiladores, computadoras, luces fluorescentes, televisores, hornos microondas, termostatos y otros aparatos eléctricos. y muchos más. Centrales eléctricas, azucareras u otras industrias que utilizan grandes equipos eléctricos, consumen mucha energía y generan ruido. Línea eléctrica [18]

3. Marco metodológico

3.1. Título de la propuesta

Diseño e implementación de un banco de pruebas de laboratorio aplicado a FM broadcasting utilizando software opens urce GNURadio, Raspberry pi y radios definidas por software.

3.2. Justificación

La necesidad de mejorar las destrezas técnicas y prácticas en el campo de la radiofrecuencia FM en los estudiantes en ingeniería en telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil para que en un futuro se desempeñen de una mejor manera en empresas del área de la radiodifusión o en canales de TV.

La investigación en el campo de la radiodifusión también se verá facilitada por el uso de equipos de radio definidos por software del Laboratorio de Telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil.

3.3. Metodología

La metodología que se utilizará en la investigación es la experimental e inductiva, ya que para este proyecto se realizará un prototipo de banco de pruebas de TX y RX FM broadcasting donde se utilizarán equipamiento de radios definidas por software para la transmisión y recepción de señales FM, de igual manera se realizará el análisis de la data obtenida de potencia y SNR en el analizador de espectros que se implementará mediante la Raspberry PI y el RTL-SDR.

3.4. Descripción de la propuesta

En este trabajo de tesis se propone un banco de pruebas de FM broadcasting para la

transmisión y recepción de señales FM mediante software Opensource como GNURadio, hardware como Raspberry PI, USRP B205 mini-i y RTL-SDR, el banco de pruebas utiliza el USRP B205 mini-i de Ettus para la transmisión de señales broadcasting FM a larga distancia lo cual se lo realiza mediante el software Opensource GNURadio y un amplificador de potencia de 50 Watts que junto con una antena tipo cono omnidireccional enviará la señal FM para ser utilizado en las diferentes prácticas de laboratorio propuestas en este trabajo de investigación.

Del lado del receptor se tendrá un equipamiento móvil con una Raspberry PI, Laptop y Celular, SDR de bajo costo RTL-SDR para la recepción y análisis de señales FM mediante la Raspberry PI.

La figura 3.1 detalla los elementos del banco de pruebas FM broadcasting.

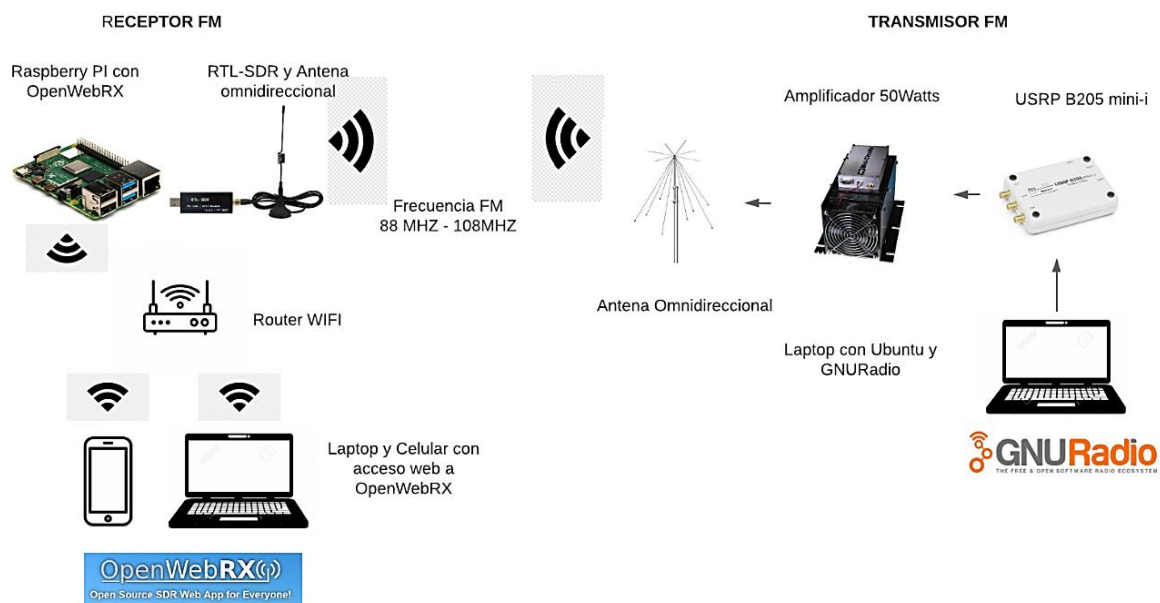


Figura 3.1 Banco de pruebas de TX y RX FM broadcasting
Fuente: [Los Autores]

A continuación, se describen los equipos y elementos utilizados en el banco de pruebas.

Hardware de recepción:

- Raspberry PI 3 model B V 1.2
- RLT-SDR Dongle
- Antena Omnidireccional para recepción FM
- Router WIFI
- Laptop
- Celular

Hardware de transmisión:

- Laptop con Ubuntu y GNURadio
- Amplificador FM 50 Watts
- USRP B205 mini-i
- Antena Omnidireccional de cono
- Laptop
- Cables RF con conectores SMA

A continuación, en la figura 3.2 se muestra un diagrama de bloques del hardware de transmisión y recepción del prototipo para el banco de pruebas.

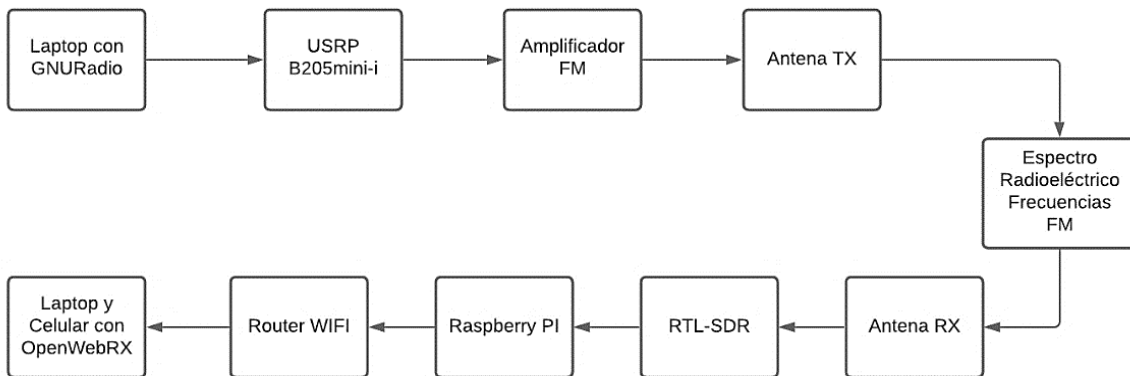


Figura 3.2 Diagramas de bloques del banco de pruebas
Fuente: [Los Autores]

En la figura 3.3. se observa el banco de pruebas FM broadcasting armado en el laboratorio de telecomunicaciones de la Universidad Politécnica Salesiana.

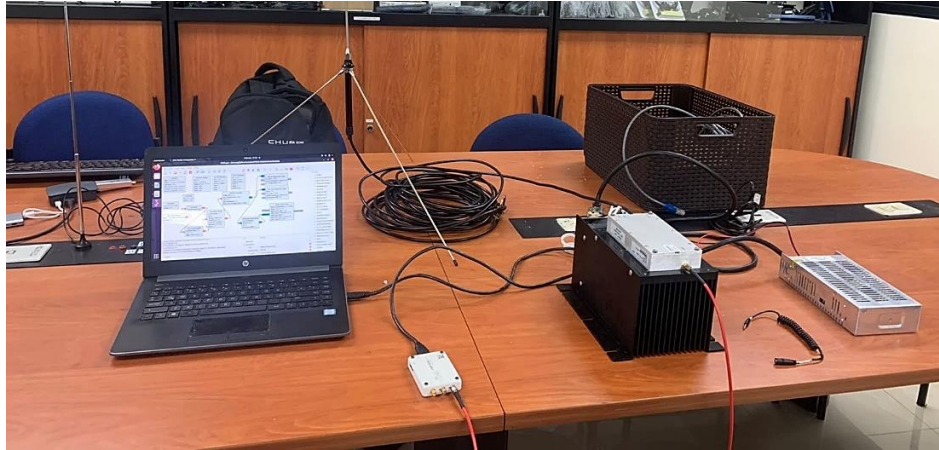


Figura 3.3 Banco de pruebas en laboratorio de telecomunicaciones
Fuente: [Los Autores]

Las prácticas que se proponen en este proyecto de investigación son las siguientes:

- 1) Introducción a la programación por bloques en software Opensource GNURadio.
- 2) Configuración e implementación de Transmisor FM con GNURadio, NI USRP, amplificador FM y antena FM.
- 3) Configuración e implementación de Receptor FM con RTL-SDR, Raspberry PI y pantalla touch.
- 4) Configuración de analizador de espectros con RTL-SDR y Raspberry PI.
- 5) Configuración de APP SDR touch y RTL-SDR para usarlo como analizador de espectros.

En las prácticas se cumplirán los siguientes objetivos por cada práctica, a continuación, el resumen:

3.4.1. Práctica 1: Introducción a la programación por bloques en software Opensource GNURadio.

OBJETIVO GENERAL:

Aprender el uso básico de la programación por bloques de GNURadio.

OBJETIVOS ESPECÍFICOS:

- Instalar GNURadio en Windows
- Instalar GNURadio en Ubuntu
- Probar y verificar la configuración de GNURadio con la programación por bloques.

3.4.2. Práctica 2: Configuración e implementación de Transmisor FM con GNURadio, USRP, amplificador FM y antena FM.

OBJETIVO GENERAL:

Configurar e implementar transmisor FM con GNURadio, NI USRP, amplificador FM y antena FM.

OBJETIVOS ESPECÍFICOS:

- Configurar transmisor FM con GNURadio
- Configurar USRP para transmisión de señales FM
- Conectar y configurar amplificador FM con antena FM.

3.4.3. Práctica 3: Configuración e implementación de Receptor FM con RTL-SDR, Raspberry PI

OBJETIVO GENERAL:

Configurar e implementar receptor FM con RTL-SDR, Raspberry PI y pantalla touch.

OBJETIVOS ESPECÍFICOS:

- Configurar receptor FM con RTL-SDR
- Configurar Raspberry PI y OpenWebRX
- Conectar Dispositivo Móvil para monitorización web

3.4.4. Práctica 4: Configuración de analizador de espectros con RTL-SDR y Raspberry PI.

OBJETIVO GENERAL:

Configurar e implementar analizador de espectros con RTL-SDR y Raspberry PI.

OBJETIVOS ESPECÍFICOS:

- Configurar analizador de espectros con RTL-SDR
- Configurar Raspberry PI
- Configurar OpenWebRX.
- Verificar la configuración de los dispositivos Raspberry PI, RTL-SDR y OpenWebRX.

3.4.5. Práctica 5: Configuración de APP SDR touch y RTL-SDR para usarlo como analizador de espectros.

OBJETIVO GENERAL:

Configurar e implementar APP SDR touch y RTL-SDR para usarlo como analizador de espectros.

OBJETIVOS ESPECÍFICOS:

- Configurar APP SDR.
- Configurar RTL-SDR como analizador de espectros.

- Aprender el uso de APP para SDR.

4. Resultados

En este capítulo se realizará la descripción de los resultados obtenidos con el banco de pruebas FM. Se describirá los elementos que se usan en el banco de pruebas, y se realizarán las pruebas que posteriormente se detallan en las prácticas ubicadas en anexos.

En la figura 4.1 se observa la conexión del amplificador RF de 50 Watts de la marca Mini-Circuits



Figura 4.1 Conexión de Amplificador con USRP
Fuente: [Los Autores]

En la figura 4.2 se observa la fuente de poder que alimenta al amplificador Mini-Circuits de 50 Watts.



Figura 4.2 Fuente de poder para amplificador RF
Fuente: [Los Autores]

La Figura 4.3 muestra la Antena de transmisor FM, 1/4 Wave Profesional GP FM Broadcast Antena para 87-108MHZ



Figura 4.3 Antena de cono para FM Tx
Fuente: [Los Autores]

Como se puede observar en la Figura 4.4 la antena es sostenido por un pedestal mediante abrazaderas.



Figura 4.4 Abrazadera de antena y pedestal
Fuente: [Los Autores]

La figura 4.5 muestra la antena, pedestal y cable coaxial que será conectado en la salida del amplificador.



Figura 4.5 Antena con pedestal
Fuente: [Los Autores]

La figura 4.6 muestra las conexiones de los equipos del lado del transmisor. Como se puede apreciar la salida del USRP se conecta mediante cable coaxial y conector SMA a la entrada del amplificador Mini-circuits. La salida del amplificador Mini-circuits se conecta a la antena omnidireccional.

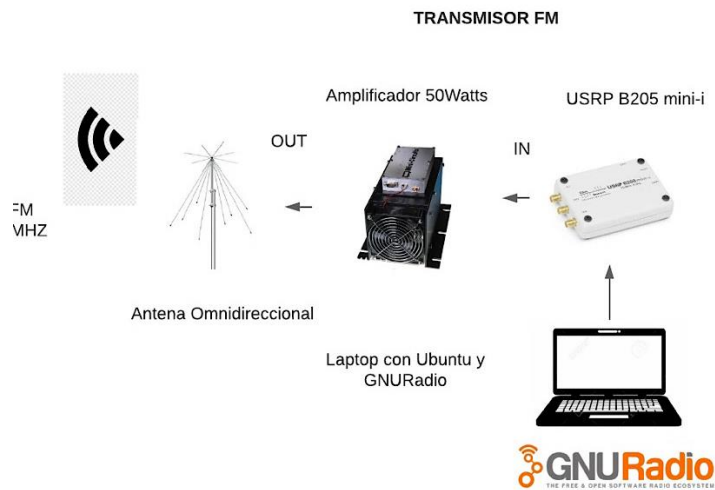


Figura 4.6 Conexiones del banco de pruebas del lado del transmisor
Fuente: [Los Autores]

Luego de las conexiones del banco de pruebas del lado de la transmisión se procede a realizar pruebas de conectividad con la Laptop Ubuntu, software GNURadio y el USRP. Ver figura 4.7 y 4.8.



Figura 4.7 Pruebas de conectividad del transmisor
Fuente: [Los Autores]

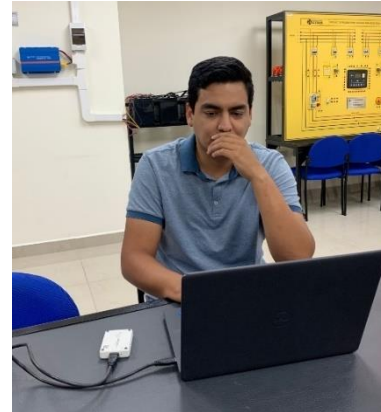


Figura 4.8 Conectividad del transmisor del banco de pruebas
Fuente: [Los Autores]

Del lado del receptor se encuentra el RTL-SDR y la Raspberry PI que tendrá cargado el software OpenWebRX en su microSD. Observe la conectividad del RTL-SDR y la raspberry Pi el cual es por USB. Ver figura 4.9



Figura 4.9 Raspberry PI y RTL-SDR
Fuente: [Los Autores]

En la figura 4.10 se observa la conectividad de la antena de recepción y el RTL-SDR, la conexión es mediante el puerto SMA.

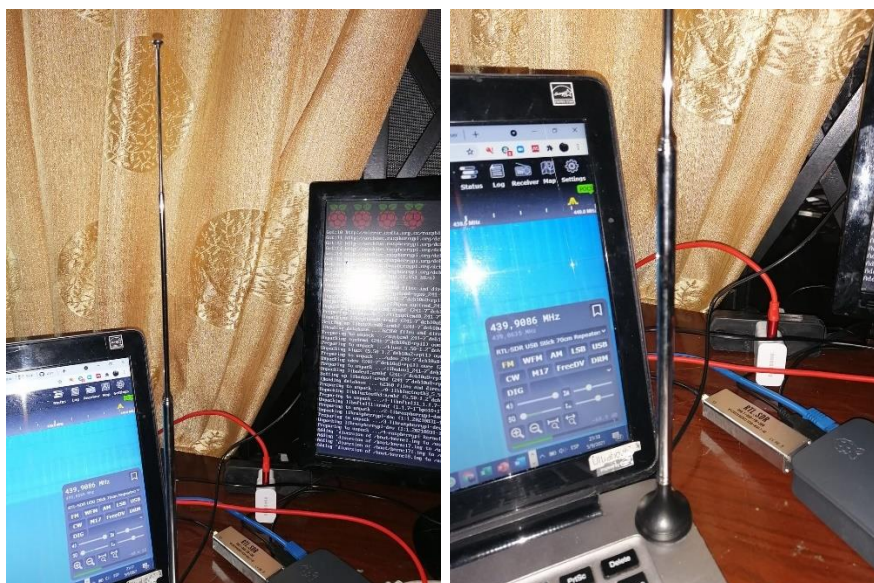


Figura 4.10 Conexión de antena a RTL-SDR
Fuente: [Los Autores]

Del lado del receptor se debe inicializar el software previamente instalado OpenWebRX. Ver figura 4.11 inicialización de OpenWebRX en Raspberry PI.

```

inot6 :::1 prefixlen 120 scopeid 0x10:Host>
loop txqueuelen 1000 (Local Loopback)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@openwebrx:~$ sudo apt-get update
sudo: update: command not found
pi@openwebrx:~$ sudo apt-get update
Get:1 http://archive.raspberrypi.org/debian buster InRelease [32.6 kB]
Hit:2 https://repo.openwebrx.de/debian buster InRelease
Get:3 http://archive.raspberrypi.org/debian buster/main armhf Packages [300 kB]
Get:4 http://rasbian.raspberrypi.org/rasbian buster InRelease [15.0 kB]
Get:5 http://rasbian.raspberrypi.org/rasbian buster/main armhf Packages [13.4 MB]
Fetched 13.4 MB in 23s (574 kB/s)
Reading package lists... Done
M: Repository 'http://rasbian.raspberrypi.org/rasbian buster InRelease' changed
pi@openwebrx:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  bluez libbluetooth3 libftnl11 libpan-systemd libraspberrypi-bin libraspberrypi-bootloader
  raspberrypi-kernel raspberrypi-sys-mods systemd system
17 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 124 MB of archives.
After this operation, 118 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.raspberrypi.org/debian buster/main armhf bluez armhf 5.50-0
Get:2 http://mirror.cedia.org.ec/rasbian/rasbian buster/main armhf systemd-sys
Get:3 http://mirror.cedia.org.ec/rasbian/rasbian buster/main armhf libpan-sys
Get:4 http://mirror.cedia.org.ec/rasbian/rasbian buster/main armhf libbluetooth3
Get:5 http://archive.raspberrypi.org/debian buster/main armhf libftnl11 armhf
Get:6 http://archive.raspberrypi.org/debian buster/main armhf libraspberrypi-do
Get:7 http://mirror.cedia.org.ec/rasbian/rasbian buster/main armhf systemd ar
Get:8 http://mirror.cedia.org.ec/rasbian/rasbian buster/main armhf udev armhf
Get:9 http://mirror.cedia.org.ec/rasbian/rasbian buster/main armhf libadcol
Get:10 http://mirror.cedia.org.ec/rasbian/rasbian buster/main armhf libraspberrypi-d
Get:11 http://archive.raspberrypi.org/debian buster/main armhf libraspberrypi-kern
Get:12 http://archive.raspberrypi.org/debian buster/main armhf raspberrypi-kern
132 raspberrypi-kernel 71.5 MB/79.6 MB 90%

```

Figura 4.11 Inicio de OpenWebRX
Fuente: [Los Autores]

En la figura 4.12 se observa la pantalla principal de OpenWebRX, el cual es un analizador de espectros para la banda de FM.

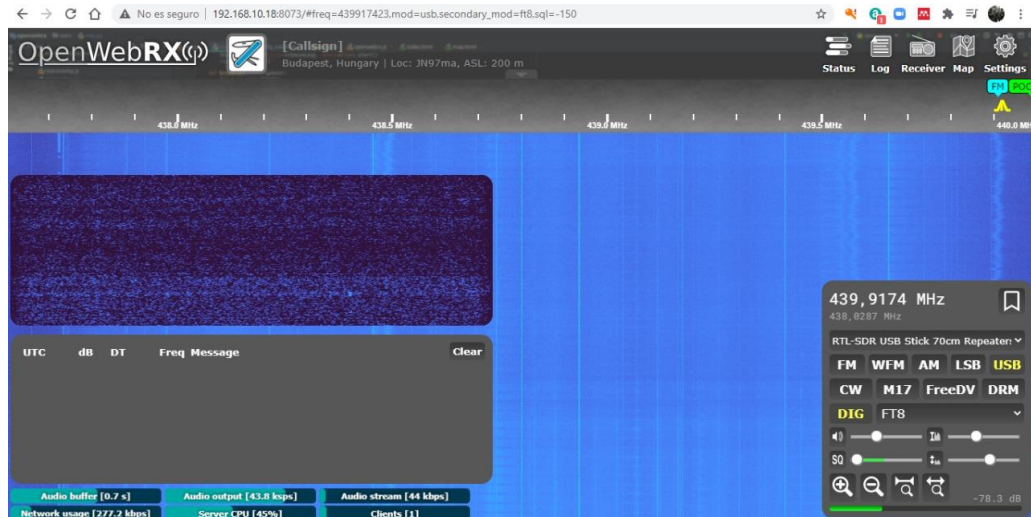


Figura 4.12 OpenWebRX
Fuente: [Los Autores]

En la figura 4.13 se observa los resultados de las señales obtenidas luego de ejecutar el transmisor y el receptor para las pruebas.

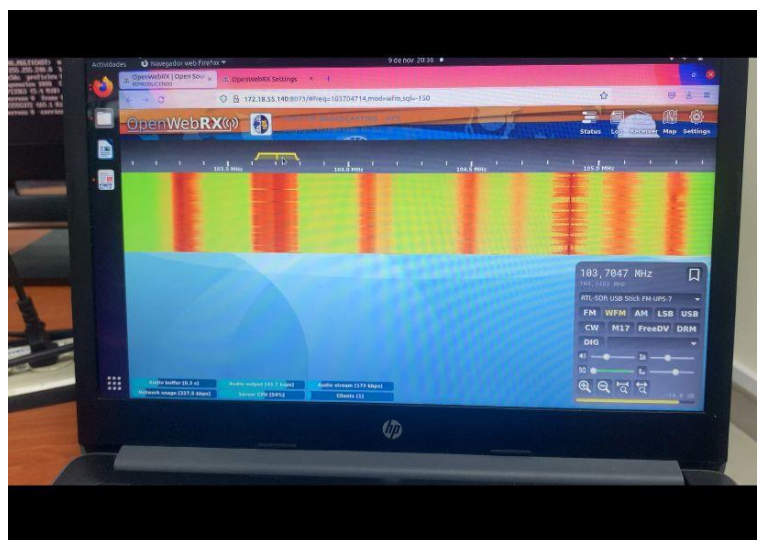


Figura 4.13 Recepción en OpenWebRX
Fuente: [Los Autores]

En la figura 4.14 y 4.15 se observan pruebas de recepción en la banda FM, se realiza sintonización de las radios con app FM del celular Android, previamente se conecta unos audífonos que harán de antena de recepción. Con estas pruebas se valida el audio que envía el transmisor FM.



Figura 4.14 Pruebas de recepción con celular
Fuente: [Los Autores]



Figura 4.15 Pruebas de recepción
Fuente: [Los Autores]

En la figura 4.16 se observa la Laptop Ubuntu que ejecuta el software GNURadio con la respectiva programación para el envío de audio en la banda de FM utilizando SDR y

amplificador RF de 50 Watts.

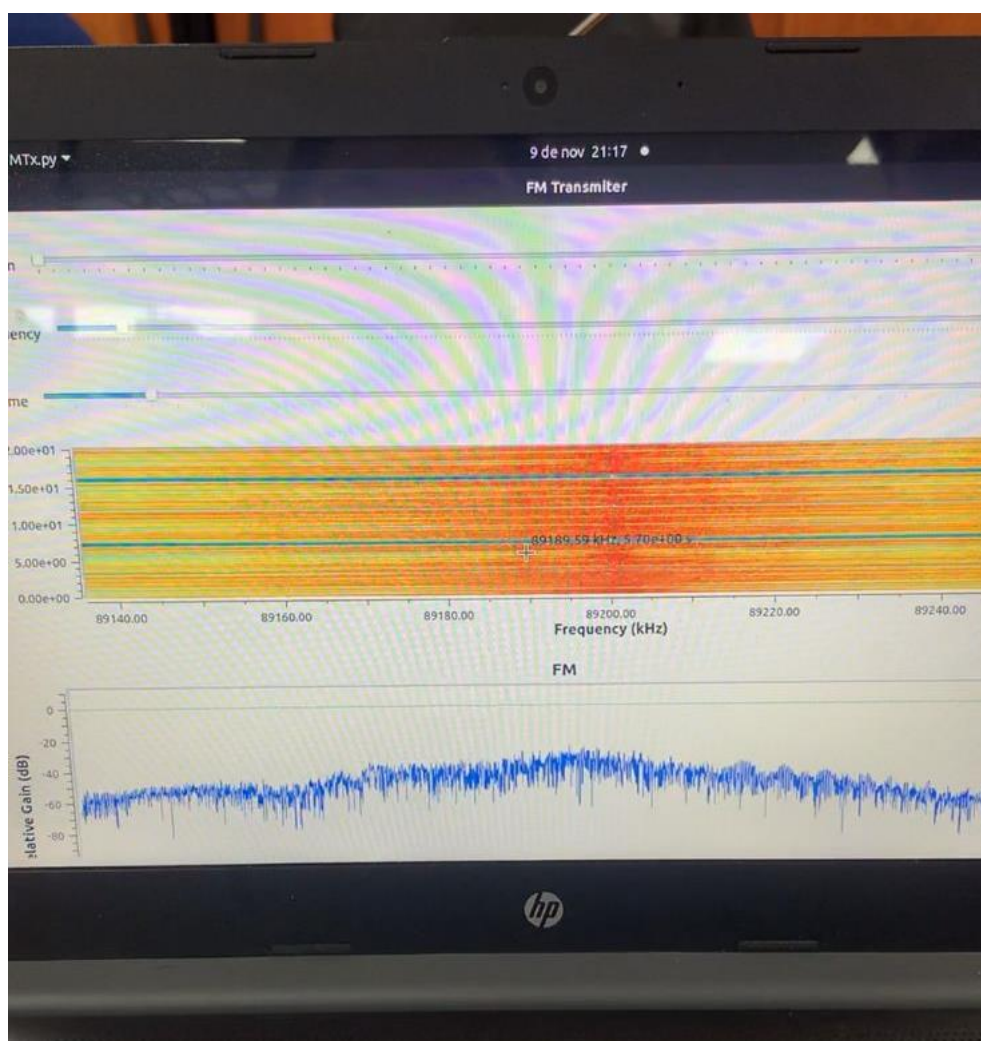


Figura 4.16 Espectro de señal visto desde GNURadio
Fuente: [Los Autores]

5. Análisis de resultados

A continuación, se muestra un resumen de los resultados obtenidos de cada procedimiento. Las prácticas detalladas se pueden encontrar en la sección de anexos de este libro de tesis.

5.1. Práctica #1: Introducción a la programación por bloques en software Opensource GNURadio.

En esta práctica se aprende el uso básico de la programación por bloques de GNURadio. Se logra realizar los objetivos de instalación de GNURadio en Ubuntu vía comandos y la configuración de programación por bloques en GNURadio. Se hace un análisis de los bloques de GNURadio para la programación del transmisor FM. La realización de la práctica se detalla en los anexos de este libro.

En la figura 5.1 se observa el banco de pruebas FM para el envío y la transmisión de las señales.



Figura 5.1 Banco de pruebas FM
Fuente: [Los Autores]

5.2. Práctica #2: Configuración e implementación de Transmisor FM con GNURadio, USRP, amplificador FM y antena FM.

En esta práctica se realiza la implementación del transmisor FM con GNURadio, USRP B205 mini-i, amplificador FM y antena FM. Se logra realizar los objetivos de configuración de transmisor FM mediante programación de bloques con GNURadio. Se realiza las conexiones del USRP con el amplificador FM para la transmisión y amplificación de las señales FM. La realización de la práctica se detalla en los anexos de este libro.

En la gráfica 5.2 se observa la programación por bloques realizada en GNURadio para el transmisor FM.

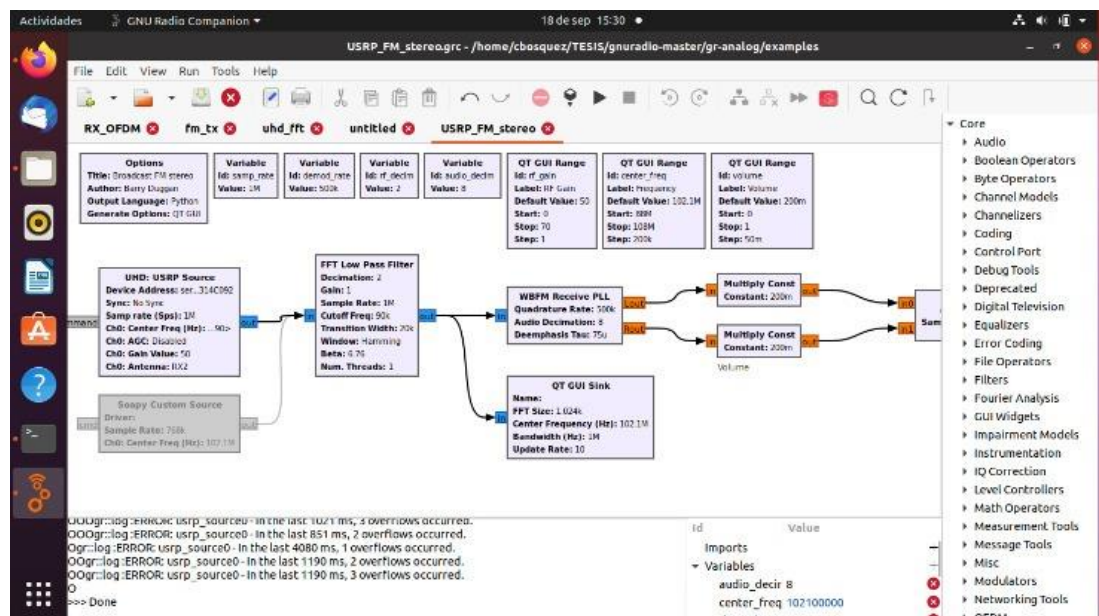


Figura 5.2 Diagrama de bloques del transmisor FM
Fuente: [Los Autores]

5.3. Práctica #3: Configuración e implementación de Receptor FM con RTL-SDR, Raspberry PI

En esta práctica se realiza la configuración e implementación del receptor FM con RTL-SDR y Raspberry PI, se logra realizar los objetivos de configurar el receptor FM configurado en la SD del Raspberry PI mediante programa OpenWebRX, la monitorización se la realiza mediante una Tablet ingresando vía web a la ip de gestión del OpenWebRX. La realización

de la práctica se detalla en los anexos de este libro.

En la figura 5.3 se observa la conexión del receptor FM con la Raspberry PI.



Figura 5.3 Conexión de RTL-SDR y Raspberry PI
Fuente: [Los Autores]

5.4. Práctica #4: Configuración de analizador de espectros con RTL-SDR y Raspberry PI.

En esta práctica se realiza la configuración e implementación de analizador de espectros con RTL-SDR y Raspberry PI. Se logran los objetivos como la configuración de analizador de espectros con RTL-SDR, el cual es configurado en la Raspberry PI. El analizador de espectros es visualizado mediante el programa OpenWebRX. La realización de la práctica se detalla en los anexos de este libro.

En la figura 5.4 se observa el analizador de espectros visto desde OpenWebRX.

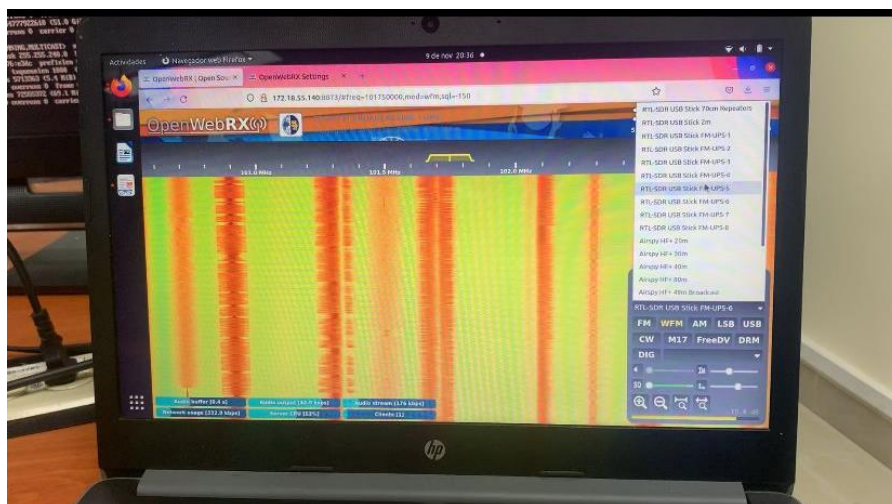


Figura 5.4 OpenWebRX
Fuente: [Los Autores]

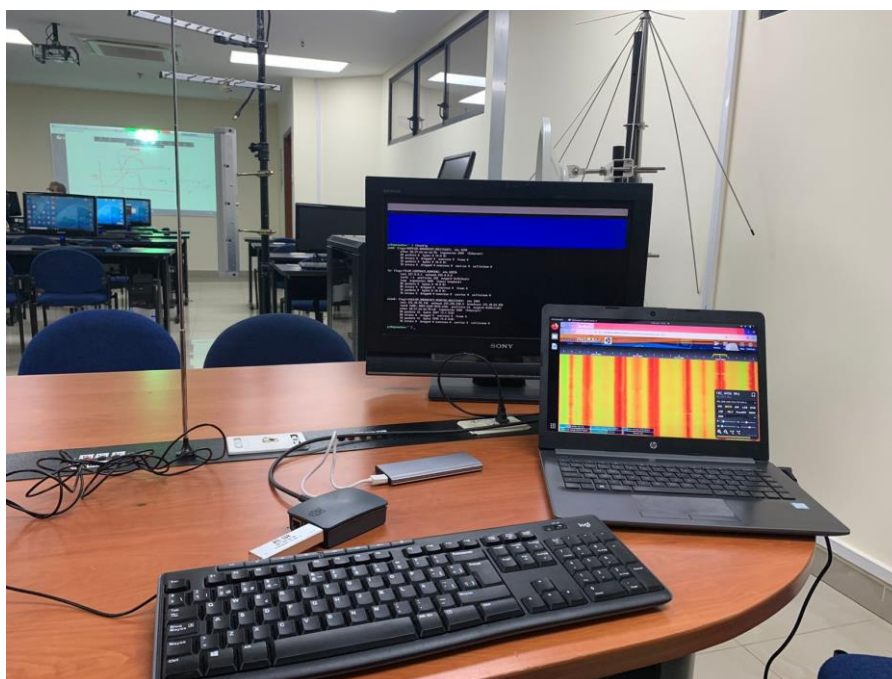


Figura 5.5 Receptor FM
Fuente: [Los Autores]

5.5. Práctica #5: Configuración de APP SDR touch y RTL-SDR para usarlo como analizador de espectros.

En la práctica 5 se realiza la configuración e implementación de APP SDR touch y RTL-SDR para usarlo como analizador de espectros. Se utiliza el RTL-SDR para monitorizar las señales RF el cual estará conectado al celular que funcionará con la APP SDR touch para ver las señales FM. La realización de la práctica se detalla en los anexos de este libro.

En la figura 5.6 se observa el espectro visto desde la APP SDR touch



Figura 5.6 APP SDR touch
Fuente: [Los Autores]

En la figura 5.7 se observa la conectividad entre el SDR y el celular, el cual es mediante adaptador USB.



Figura 5.7 Conexión de Celular y RTL-SDR
Fuente: [Los Autores]

6. Conclusiones

Este trabajo de investigación tiene las siguientes conclusiones basadas en sus objetivos:

- Se elaboró el diseño y se implementó un banco de pruebas de laboratorio aplicado a la radiodifusión FM utilizando software Opensource GNURadio, Raspberry PI y Radios Definidas por Software.
- Se realizó el diseño de un banco de prácticas aplicadas a la tecnología de modulación FM mediante radios definidas por software.
- Se ejecutó la configuración de NI USRP para la transmisión de audio en FM utilizando software Opensource GNURadio y antena de transmisión en la banda FM. Para el cumplimiento de este objetivo se utiliza el USRP B205 mini-i como transmisor de RF.
- Se implementó un banco de recepción FM con RTL-SDR, pantalla touch, Raspberry PI para la monitorización y análisis de señales FM. Para el cumplimiento de este objetivo se utiliza una Tablet o celular como monitor y una batería portable para la energización de la Raspberry Pi.
- Se realizó la instalación y pruebas con amplificador FM para la transmisión de audio a larga distancia y monitorizar potencia mediante analizador de espectros con RTL-SDR. Se utiliza amplificador de 50 Watts.

7. Recomendaciones

Las recomendaciones de este proyecto de investigación son las siguientes:

- Para este banco de pruebas se recomienda usar antenas y amplificador de mayor ganancia para poder realizar prácticas de FM a más distancia, en este proyecto usamos una antena de 2.14 dBi de ganancia.
- Se recomienda utilizar software libre para la monitorización de señales FM, sin embargo, se debe considerar que la universidad dispone de licencias de National Instruments para poder utilizar Labview con los USRP sin ningún inconveniente.
- Las pruebas de las señales en FM que se realicen deben ser cortas en tiempo y en laboratorio, ya que no se puede utilizar por mucho tiempo las frecuencias licenciadas por temas legales y de concesión de frecuencias.
- Se recomienda utilizar Raspberry PI 4 para un mejor performance en procesamiento y memoria ya que la versión 3 tiene limitada memoria y procesador.
- Para las pruebas de envío de audio se debe considerar que sea audio de tipo WAV para que sea reconocido por GNURadio.
- Las configuraciones de GNURadio pueden ser modificadas para experimentar en el banco de pruebas el diseño y programación propuesta por los estudiantes.


Bibliografía

- [1]. Agencia de Regulación y Control de las Telecomunicaciones ARCOTEL. “Promovemos el desarrollo armónico del sector de las telecomunicaciones, radio, televisión y las TIC, mediante la administración y regulación eficiente del espectro radioeléctrico y los servicios.” <http://www.arcotel.gob.ec/arcotel-efectuo-mediaciones-a-la%E2%80%A6/> (accessed Jun. 14, 2020).
- [2]. Agencia de Regulación y Control de las Telecomunicaciones ARCOTEL. “Radiodifusión sonora y Televisión abierta.” <http://www.arcotel.gob.ec/radiodifusion-sonora-y-television-abierta2/> (accessed Apr. 5, 2021).
- [3]. Agencia de Regulación y Control de las Telecomunicaciones ARCOTEL. “Modificaciones técnicas”. <https://www.arcotel.gob.ec/modificaciones-tecnicas-radiodifusionsonora/> (accessed Jan. 21, 2021).
- [4]. EcuRed. “RTLSDR Dongle”. https://www.ecured.cu/RTLSDR_Dongle_ (accessed Sept. 1, 2022).
- [5]. Programador Clic. “Diagrama esquemático de RTL-SDR.” <https://programmerclick.com/article/9651382065/> (accessed Aug. 15, 2022).
- [6]. Ettus. “B200/B210/B200mini/B205mini Getting Started Guides.” https://kb.ettus.com/B200/B210/B200mini/B205mini_Getting_Started_Guides (accessed Jan. 11, 2021).
- [7]. NI National Instruments. “Which USRP Is Right for You?” <https://www.ni.com/es-cr/shop/wireless-design-test/what-is-a-usrp-software-defined-radio/which-usrp-is-right-for-you.html> (accessed Dec. 20, 2021).

- [8]. NI National Instruments. *Product flyer usrp software defined radios*. Accessed: Feb. 21, 2022. [Online]. Available: <https://www.ni.com/pdf/product-flyers/usrp-software-defined-radio.pdf>
- [9]. NI National Instruments. “Bundle de dispositivo de radio definido por software.” <https://www.ni.com/es-cr/shop/hardware/products/software-defined-radio-device-bundles.html> (accessed March 12, 2022).
- [10]. E. Rodríguez. “Cómo escuchar música de tu móvil a través de la radio del coche: guía de compra de transmisores FM Bluetooth.” Xataka. <https://www.xataka.com/seleccion/como-escuchar-musica-tu-movil-traves-radio-coche-guia-compra-transmisores-fm-bluetooth> (accessed Dec. 12, 2021)
- [11]. S. García. *Manual para radialistas analfatécnicos*. Accessed: Aug. 10, 2022. [Online]. Available: <https://www.analfatecnicos.net/ManualRadialistasAnalfatecnicos.pdf>
- [12]. Gnuradio. “Tutorials.” <https://wiki.gnuradio.org/index.php/Tutorials> (accessed Jan. 21, 2022)
- [13]. J. Ninoc. “Convert a wav file to raw using gnu radio companion (GRC) (radio).” Jesusninoc. <https://www.jesusninoc.com/04/18/convertir-un-fichero-wav-en-raw-utilizando-gnu-radio-companion-grc/> (accessed Feb. 12, 2022)
- [14]. Raspberry Pi. “Raspberry Pi 4.” <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/> (accessed Feb. 21, 2020).
- [15]. R. Solé. “Raspberry Pi: Crea proyectos DIY por muy poco dinero.” Profesional Review. <https://www.profesionalreview.com/2021/07/18/que-es-raspberry-pi/> (accessed Apr. 17, 2021).

- [16]. Tu rincón informático. “Instalar y Configurar Raspberry Pi Os (Raspbian).”
<https://turinconinformatico.com/raspberrypi/instalar-raspbian/> (accessed Sept. 4, 2021).
- [17]. E. Rodríguez. “De cero a maker: todo lo necesario para empezar con Raspberry Pi.” Xataka.
<https://www.xataka.com/makers/cero-maker-todo-necesario-para-empezar-raspberry-pi>
(accessed July 25, 2022)
- [18]. Ecured. “Antena Yagi Uda.” https://www.ecured.cu/Antena_Yagi_Uda (accessed Feb. 12, 2022)
- [19]. L. Martínez and K. Currea, “Sistema de comunicaciones para un cubesat de análisis de imágenes climatológicas.” B.A., thesis, Dept. Electr., Telecom. Univ. Catol. Colomb. Bogotá, Colombia, 2020. [Online]. Available:
<https://repository.ucatolica.edu.co/bitstream/10983/24505/1/Trabajo%20de%20Grado.pdf>
- [20]. Gnuradio. “Tutorials.” https://wiki.gnuradio.org/index.php/Tutorials_ (accessed Dec. 30, 2020).

ANEXOS
ANEXO 1: GUIAS DE PRÁCTICAS

		GUÍA DE PRÁCTICA DE LABORATORIO	
CARRERA: INGENIERÍA ELECTRÓNICA		ASIGNATURA:	
NRO. PRÁCTICA:	1	TÍTULO PRÁCTICA: Introducción a la programación por bloques en software Opensource GNURadio.	
<p>OBJETIVO GENERAL:</p> <p>Aprender el uso básico de la programación por bloques de GNURadio.</p> <p>OBJETIVOS ESPECÍFICOS:</p> <ul style="list-style-type: none"> Instalar GNURadio en Windows Instalar GNURadio en Ubuntu Probar la configuración de GNURadio con la programación por bloques. 			
INSTRUCCIONES		<ol style="list-style-type: none"> 1. Los estudiantes deben leer previamente el manual de práctica para el desarrollo. 2. Los estudiantes deben utilizar los equipos del banco de pruebas de radiodifusión FM de una manera responsable y calificada para evitar daños en los equipos. 3. Los estudiantes deben trabajar en grupo para el desarrollo de la práctica. 4. Se debe dejar en orden el sitio de práctica luego del desarrollo de esta. 	

MARCO TEÓRICO:

GNURadio:

GNU Radio es un kit de herramientas de desarrollo de software de código abierto y gratuito que proporciona bloques de procesamiento de señales para implementar radios de software. Se puede utilizar con hardware de RF externo de bajo costo y fácilmente disponible para crear radios definidas por software, o sin hardware en un entorno similar a una simulación. Se usa ampliamente en entornos de aficionados, académicos y comerciales para respaldar tanto la investigación de comunicaciones inalámbricas como los sistemas de radio del mundo real [20].

GNU Radio es un marco que permite a los usuarios diseñar, simular e implementar sistemas de radio del mundo real de alta capacidad. Es un marco altamente modular, orientado a un "diagrama de flujo" que viene con una biblioteca completa de bloques de procesamiento que se pueden combinar fácilmente para crear aplicaciones complejas de procesamiento de señales [20].

GNU Radio se ha utilizado para una gran variedad de aplicaciones de radio del mundo real, que incluyen procesamiento de audio, comunicaciones móviles, rastreo de satélites, sistemas de radar, redes GSM, Digital Radio Mondiale y mucho más, todo en software de computadora [20].

Por sí solo, no es una solución para hablar con ningún hardware específico. Tampoco proporciona aplicaciones listas para usar para estándares de comunicaciones de radio específicos (por ejemplo, 802.11, ZigBee, LTE, etc.), pero se puede (y se ha utilizado) para desarrollar implementaciones de básicamente cualquier banda limitada. estándar de comunicación [20].

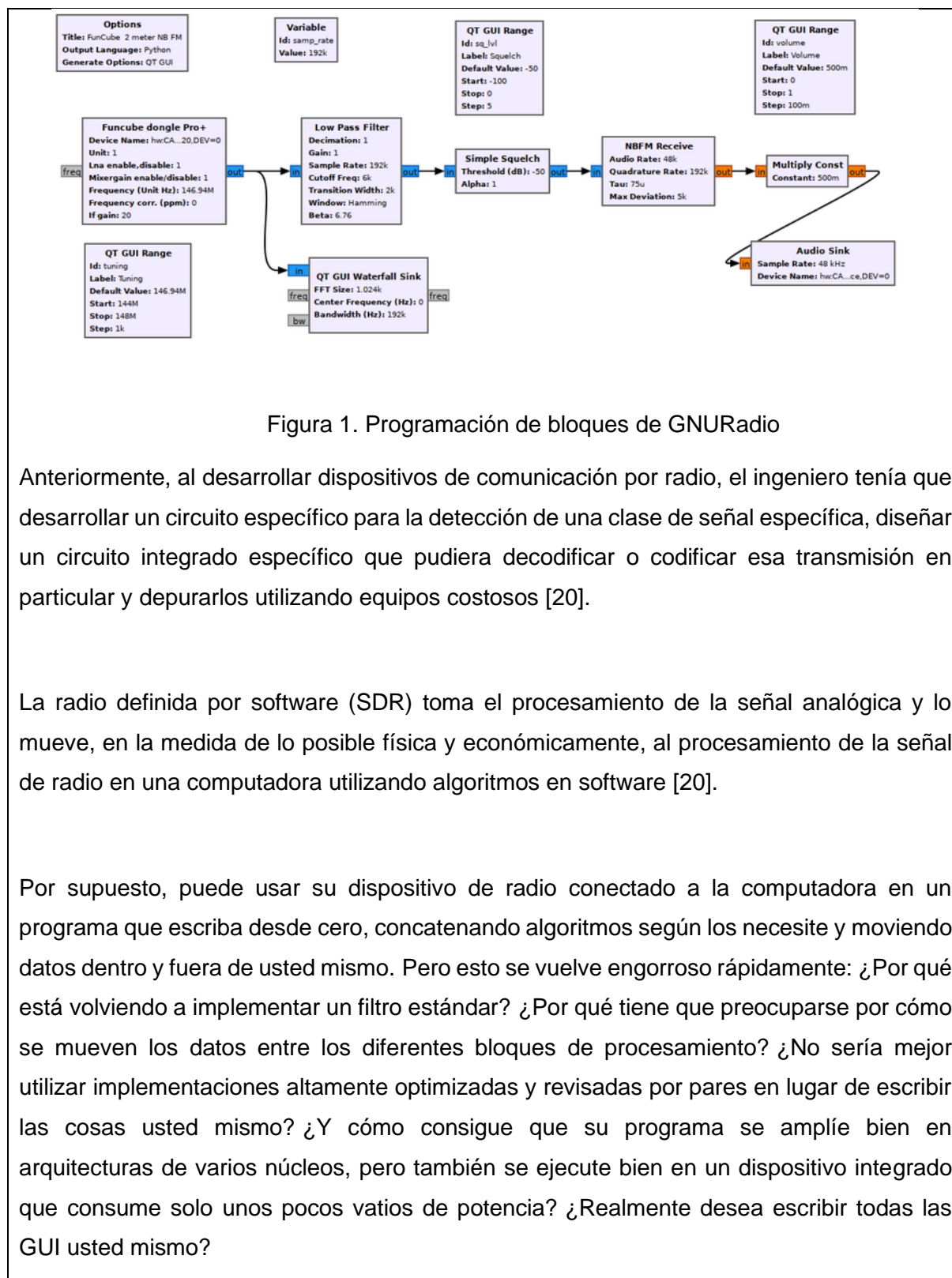


Figura 1. Programación de bloques de GNURadio

Anteriormente, al desarrollar dispositivos de comunicación por radio, el ingeniero tenía que desarrollar un circuito específico para la detección de una clase de señal específica, diseñar un circuito integrado específico que pudiera decodificar o codificar esa transmisión en particular y depurarlos utilizando equipos costosos [20].

La radio definida por software (SDR) toma el procesamiento de la señal analógica y lo mueve, en la medida de lo posible física y económicamente, al procesamiento de la señal de radio en una computadora utilizando algoritmos en software [20].

Por supuesto, puede usar su dispositivo de radio conectado a la computadora en un programa que escriba desde cero, concatenando algoritmos según los necesite y moviendo datos dentro y fuera de usted mismo. Pero esto se vuelve engorroso rápidamente: ¿Por qué está volviendo a implementar un filtro estándar? ¿Por qué tiene que preocuparse por cómo se mueven los datos entre los diferentes bloques de procesamiento? ¿No sería mejor utilizar implementaciones altamente optimizadas y revisadas por pares en lugar de escribir las cosas usted mismo? ¿Y cómo consigue que su programa se amplíe bien en arquitecturas de varios núcleos, pero también se ejecute bien en un dispositivo integrado que consume solo unos pocos vatios de potencia? ¿Realmente desea escribir todas las GUI usted mismo?

Ingrese GNU Radio: un marco dedicado a escribir aplicaciones de procesamiento de señales para computadoras comerciales. GNU Radio envuelve la funcionalidad en bloques reutilizables fáciles de usar, ofrece una escalabilidad excelente, proporciona una biblioteca extensa de algoritmos estándar y está altamente optimizado para una gran variedad de plataformas comunes. También viene con una gran cantidad de ejemplos para comenzar [20].

Un enfoque modular basado en diagrama de flujo para el procesamiento de señales digitales

Para procesar señales digitales, es sencillo pensar en las etapas de procesamiento individuales (filtrado, corrección, análisis, detección ...) como bloques de procesamiento, que se pueden conectar mediante simples flechas indicadoras de flujo [20]:

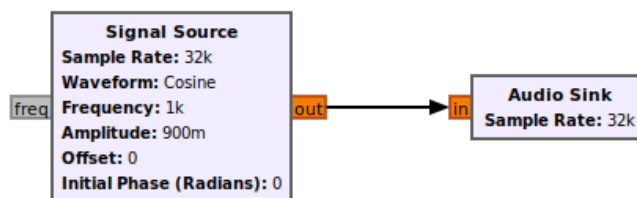


Figura 2. Signal Source

Al crear una aplicación de procesamiento de señales, se creará un gráfico completo de bloques. Dicho gráfico se denomina diagrama de flujo en GNU Radio [20].

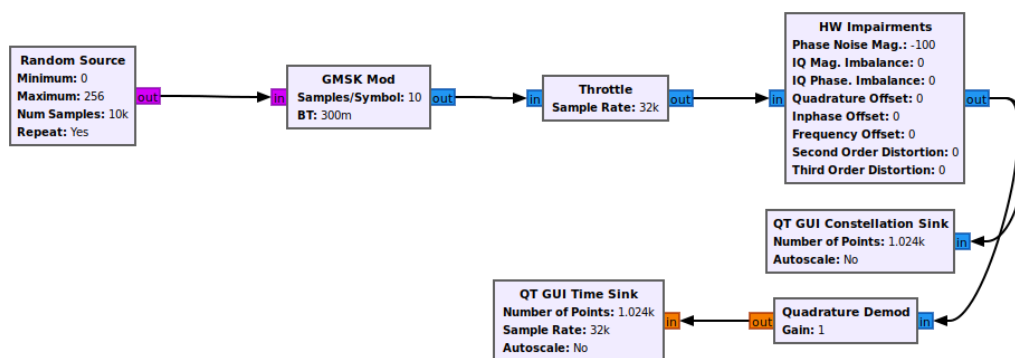


Figura 3. Flujo de GNURadio

GNU Radio es un marco para desarrollar estos bloques de procesamiento y crear diagramas de flujo, que comprenden aplicaciones de procesamiento de radio [20].

Como usuario de GNU Radio, puede combinar bloques existentes en un diagrama de flujo de alto nivel que hace algo tan complejo como recibir señales moduladas digitalmente y GNU Radio moverá automáticamente los datos de la señal entre estos y provocará el procesamiento de los datos cuando esté listo para su procesamiento [20].

GNU Radio viene con un gran conjunto de bloques existentes. Se puede encontrar un índice de todos ellos en Block Docs. Solo para darle un pequeño extracto de lo que está disponible en una instalación estándar, aquí están algunas de las categorías de bloques más populares y algunos de sus miembros [20]:

- Generadores de forma de onda
 - Fuente constante
 - Fuente de ruido
 - Fuente de señal (por ejemplo, seno, cuadrado, diente de sierra)
- Moduladores
 - AM Demod
 - Modulación de fase continua
 - Modificación / demodificación de PSK
 - Modificación / demodificación de GFSK
 - Modificación / demodificación de GMSK
 - QAM Mod / Demod
 - Recibir WBFM
 - Recibir NBFM
- Instrumentación (es decir, GUI)
 - Fregadero de la constelación
 - Fregadero de frecuencia

- Fregadero de histograma
- Fregadero de número
- Disipador de ráster de tiempo
- Disipador de tiempo
- Fregadero de cascada

- Operadores matemáticos
 - Abdominales
 - Agregar
 - Complejo conjugado
 - Dividir
 - Integrar
 - Log10
 - Multiplicar
 - RMS
 - Sustraer

- Modelos de canal
 - Modelo de canal
 - Modelo que se desvanece
 - Modelo de canal dinámico
 - Modelo de desvanecimiento selectivo de frecuencia

- Filtros
 - Filtro de paso / rechazo de banda
 - Filtro de paso bajo / alto
 - Filtro IIR
 - Banco de filtros genérico
 - Hilbert
 - Filtro FIR diezmador
 - Filtro de coseno elevado de raíz

- Filtro FFT
- Análisis de Fourier
 - FFT
 - Log Power FFT
 - Goertzel (remuestreadores)
 - Remuestreador fraccional
 - Remuestreador arbitrario polifásico
 - Remuestreador racional (sincronizadores)
 - Recuperación de reloj MM
 - Correlación y sincronización
 - Bucle de Costas
 - Borde de banda FLL
 - PLL Freq Det
 - Correlacionador de PN
 - Sincronización de reloj polifásico

Con estos bloques, se pueden realizar muchas tareas estándar, como normalizar señales, sincronización, mediciones y visualización, simplemente conectando el bloque apropiado a su diagrama de flujo de procesamiento de señales [20].

Además, puede escribir sus propios bloques, que combinen los bloques existentes con algo de inteligencia para proporcionar una nueva funcionalidad junto con algo de lógica, o puede desarrollar su propio bloque que opere con los datos de entrada y los datos de salida [20].

Por lo tanto, GNU Radio es principalmente un marco para el desarrollo de bloques de procesamiento de señales y su interacción. Viene con una extensa biblioteca estándar de bloques, y hay muchos sistemas disponibles sobre los que un desarrollador puede construir. Sin embargo, GNU Radio en sí no es un software que esté listo para hacer algo específico; es trabajo del usuario construir algo útil a partir de él, aunque ya viene con

muchos ejemplos útiles de trabajo. Piense en ello como un conjunto de bloques de construcción [20].

Conociendo el GRC

Hemos visto en Una breve introducción a GNU Radio, SDR y DSP que GNU Radio es una colección de herramientas que se pueden usar para desarrollar sistemas de radio en software en lugar de hacerlo completamente en hardware. En este tutorial, comenzamos de manera simple y exploramos cómo usar GNU Radio Companion (GRC), la herramienta gráfica de GNU Radio, para crear diferentes tonos. Debemos tener en cuenta que GRC fue creado para simplificar el uso de GNU Radio al permitirnos crear archivos de Python gráficamente en lugar de crearlos solo en código [20].

Lo primero que hay que cubrir es la interfaz. Hay cinco partes: biblioteca, barra de herramientas, terminal, espacio de trabajo y variables [20].

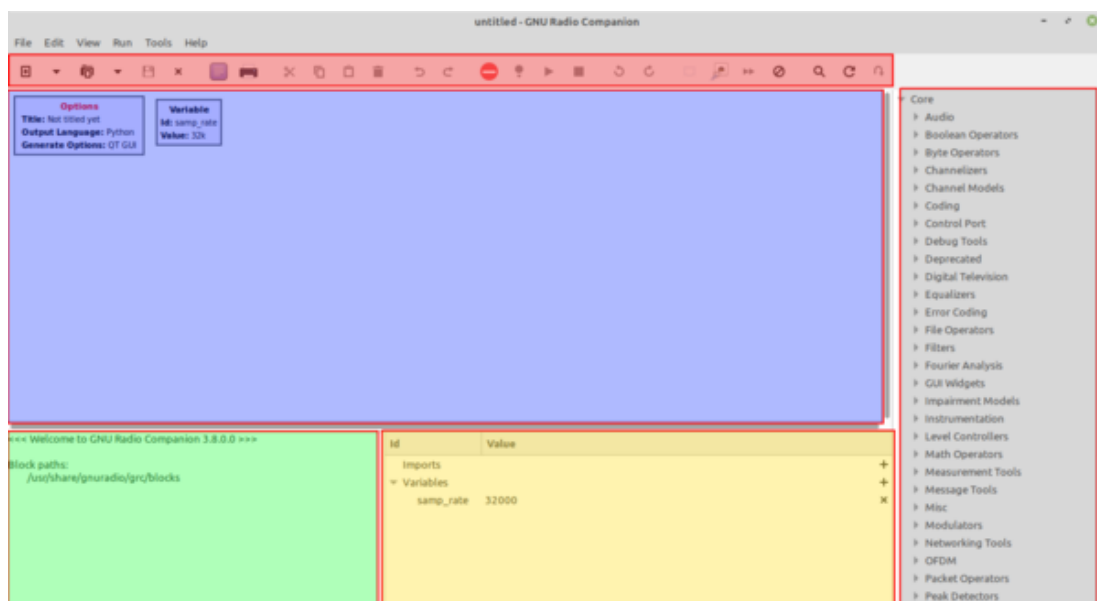


Figura 4. Panel principal GNURadio

El tutorial está destinado a ser práctico, así que tómese unos descansos de la lectura aquí y allá para experimentar. Reiteramos que estos tutoriales son simplemente guías y que la

mejor manera de aprender algo es probarlo: hacer una pregunta, pensar en una posible solución y probarlo. ¡Comencemos por poner en marcha el GRC! Esto generalmente se hace abriendo una ventana de terminal (ctrl + alt + t en Ubuntu) y escribiendo [20]:

```
$ gnuradio-companion
```

Si no puede abrir GRC, debe volver a InstallingGR y solucionar los problemas de instalación.

Buscando bloques

La Biblioteca contiene los diferentes bloques instalados en las rutas de los bloques GRC. Aquí encontramos bloques que están preinstalados en GNU Radio y bloques que están instalados en el sistema. Al principio parece abrumador buscar bloques. Por ejemplo, si queremos generar una forma de onda, ¿en qué categoría debemos buscar? Vemos que hay una categoría de Generadores de forma de onda, está bien, no está mal. Pero, ¿qué pasaría si quisiéramos encontrar alguna forma de mostrar nuestros datos, pero aún no estamos seguros de cuál es la mejor manera de mostrarlos? Sabemos por una breve introducción a GNU Radio, SDR y DSP que esto se conoce como fregadero; sin embargo, mirando a través de la lista no hay categoría de Fregaderos. La solución es utilizar la función de búsqueda haciendo clic en el icono de la lupa o escribiendo Ctrl + f y luego comenzar a escribir una palabra clave para identificar el bloque. Vemos aparecer un cuadro blanco en la parte superior de la biblioteca con un cursor. Si escribimos "hundir", podemos encontrar todos los bloques que contienen las palabras "hundir" y las categorías donde encontraremos cada bloque [20].

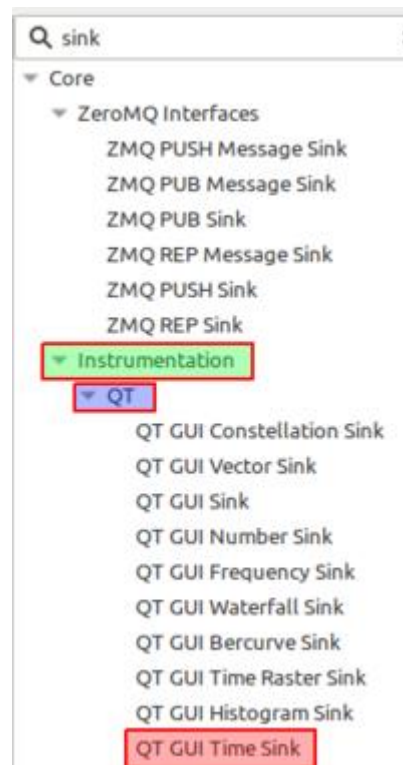


Figura 5. Biblioteca de GNURadio

Por ahora, agreguemos el bloque llamado QT GUI Time Sink haciendo clic en su nombre y arrastrándolo al área de trabajo o haciendo doble clic en su nombre para que se coloque automáticamente en el área de trabajo [20].

Modificar las propiedades del bloque

El espacio de trabajo (área principal de la pantalla) contiene todos nuestros bloques que componen nuestro diagrama de flujo, y dentro de cada bloque podemos ver todos los diferentes parámetros del bloque. Sin embargo, existe un bloque especial con el que cada nuevo diagrama de flujo comienza y debe tener, llamado Bloque de Opciones. Hagamos doble clic en el bloque de opciones para examinar sus propiedades. Vemos una ventana como la siguiente [20]:

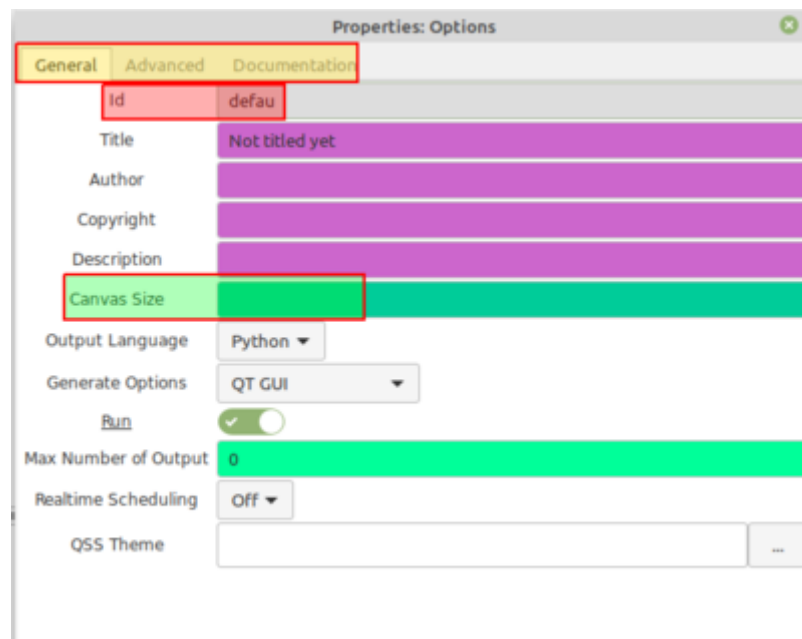


Figura 6. Propiedades de bloque

Estas propiedades de bloque se pueden cambiar de las predeterminadas para realizar diferentes tareas. Eliminemos parte del nombre actual y observemos que la identificación se vuelve azul. Este color significa que la información se ha editado, pero no se ha guardado. Si volvemos a las propiedades del Bloque de opciones, podemos ver que hay diferentes pestañas [20].

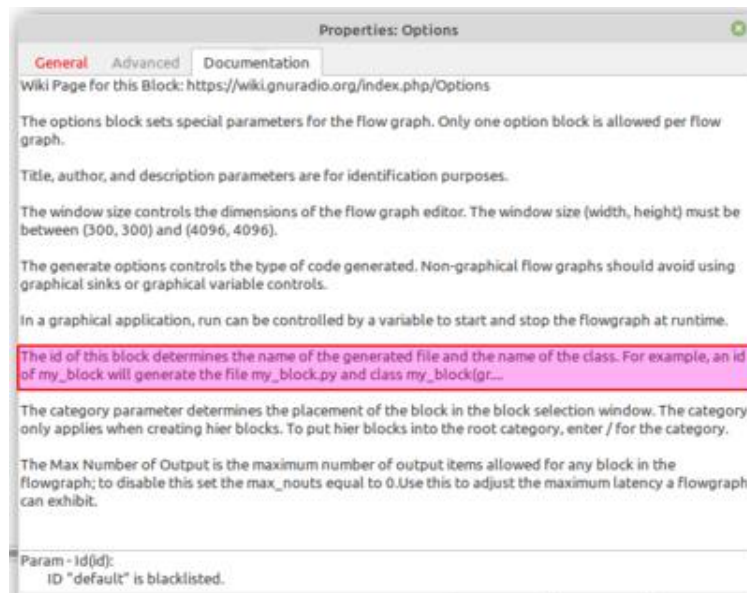
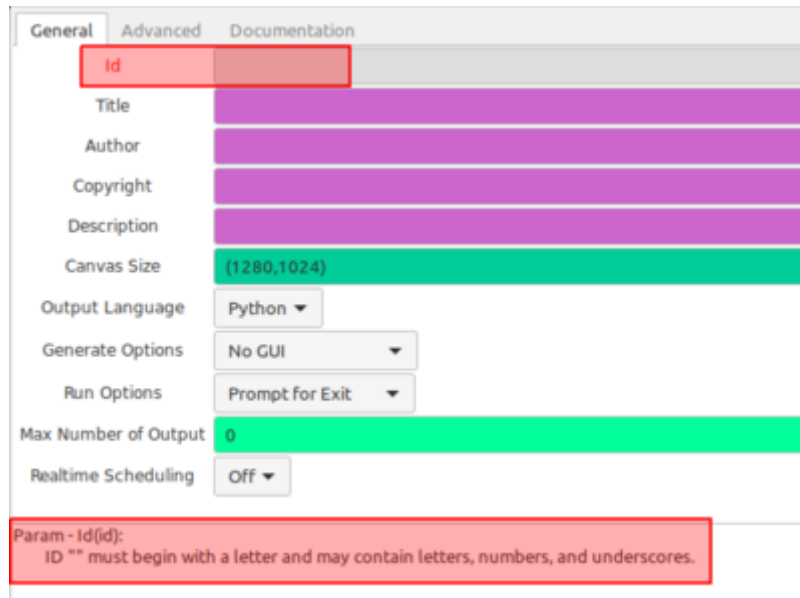


Figura 7. Opciones del bloque.

Si leemos un par de líneas, podemos ver que el ID de propiedad se usa realmente para el nombre del archivo de Python que genera el diagrama de flujo [20].



General	Advanced	Documentation
Id		
Title		
Author		
Copyright		
Description		
Canvas Size	(1280,1024)	
Output Language	Python	
Generate Options	No GUI	
Run Options	Prompt for Exit	
Max Number of Output	0	
Realtime Scheduling	Off	

Param - Id(id):
ID "" must begin with a letter and may contain letters, numbers, and underscores.

Figura 8. Configuraciones de bloque

Así que ahora eliminemos toda la cadena de identificación. Observe ahora que en la parte inferior aparece un mensaje de error. También observe que el ID del parámetro ahora es rojo para mostrarnos exactamente dónde ocurrió el error [20].

Para mantener las cosas organizadas, cambiemos el ID a "tutorial_two_1". Asegurémonos también de que la propiedad Generar opciones esté establecida en "QT GUI" ya que estamos usando un receptor gráfico. El campo ID nos permite administrar más fácilmente nuestro espacio de archivos. Si bien guardamos el diagrama de flujo de GRC como <nombre de archivo>. Grc, la generación y ejecución de este diagrama de flujo produce otra salida. GRC es una interfaz gráfica que se encuentra sobre el entorno de programación normal de GNU Radio que se encuentra en Python. GRC traduce el diagrama de flujo que creamos en el lienzo de la GUI aquí en un script de Python, por lo que cuando ejecutamos un diagrama de flujo, realmente estamos ejecutando un programa de Python. El ID se usa para nombrar ese archivo Python, guardado en el mismo directorio que el archivo. Grc. Por

defecto, el ID es el predeterminado y, por lo tanto, crea un archivo llamado default.py. Cambiar la ID nos permite cambiar el nombre del archivo guardado para una mejor administración de archivos. En GNURadio 3.8 obtendrá un error si no cambia la identificación predeterminada, por lo que debe cambiar esta identificación para ejecutar el diagrama de flujo [20].

Otro resultado de esta conexión GRC-Python es que GRC es en realidad todo Python. De hecho, todos los cuadros de entrada en las propiedades del bloque o las variables que usamos se interpretan como Python. Eso significa que podemos establecer propiedades usando llamadas de Python, como llamar a un numpy u otras funciones de GNU Radio. Un uso común de esto es llamar a la herramienta de diseño de filtros filter. Firdes de GNU Radio para construir nuestros grifos de filtro [20].

Otra clave para notar son los diferentes colores presentes en los campos en los que podemos ingresar información. En realidad, estos corresponden a diferentes tipos de datos que cubriremos más adelante en este tutorial.

Controles variables

Ciertos parámetros de un bloque se pueden modificar durante el tiempo de ejecución. Se indican en la sección Parámetros, después del nombre del parámetro.

Como se muestra en esta captura de pantalla, los parámetros 'Taps' y 'Center Frequency' están subrayados, lo que indica que se pueden cambiar durante el tiempo de ejecución [20].

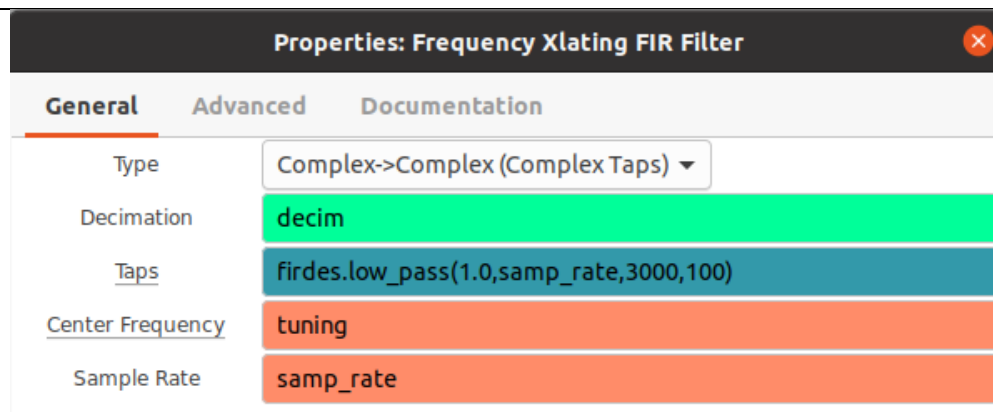


Figura 9. Controles de variables

Diagrama de flujo

Ahora que tenemos una mejor comprensión de cómo encontrar bloques, cómo agregarlos al espacio de trabajo y cómo editar las propiedades del bloque, procedamos a construir el siguiente diagrama de flujo de una fuente de señal que se envía a un Throttle Block y luego a nuestro Time Sink haciendo clic en los puertos / pestañas de color del tipo de datos uno tras otro para hacer las conexiones [20]:

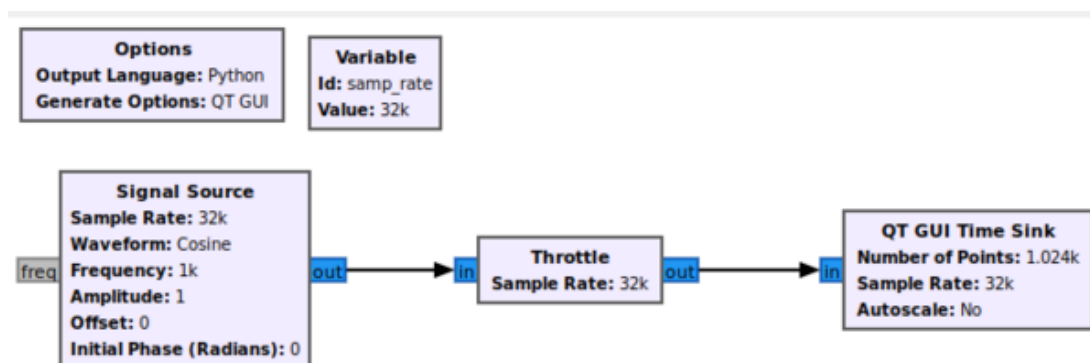


Figura 10. Diagrama de flujo

Con un diagrama de flujo utilizable, ahora podemos analizar la barra de herramientas [20].

Una nota sobre el bloque del acelerador: lo que hace exactamente se explica con mayor detalle más adelante en este tutorial. Por ahora, será suficiente saber que este bloque

acelera el gráfico de flujo para asegurarse de que no consume el 100% de los ciclos de su CPU y hace que su computadora no responda [20].

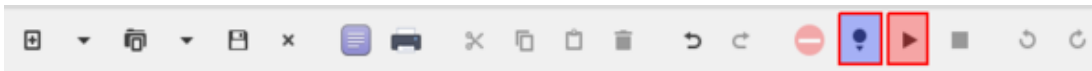


Figura 11. Panel de GNURadio

Esta sección de la interfaz contiene comandos presentes en la mayoría del software, como nuevo, abrir, guardar, copiar y pegar. Comencemos guardando nuestro trabajo hasta ahora y titulando nuestro gráfico de flujo tutorial_two. Herramientas importantes aquí son Generar flowgraph, Ejecutar flowgraph, y Kill flowgraph todos accesibles a través F5, F6, F7 y respectivamente. Hay una buena referencia disponible en Ayuda -> Tipos que muestra el mapeo de colores de los tipos que veremos más adelante [20].

Examinando la salida

Vamos a seguir adelante y Ejecutar el flowgraph para ver nuestra onda sinusoidal.

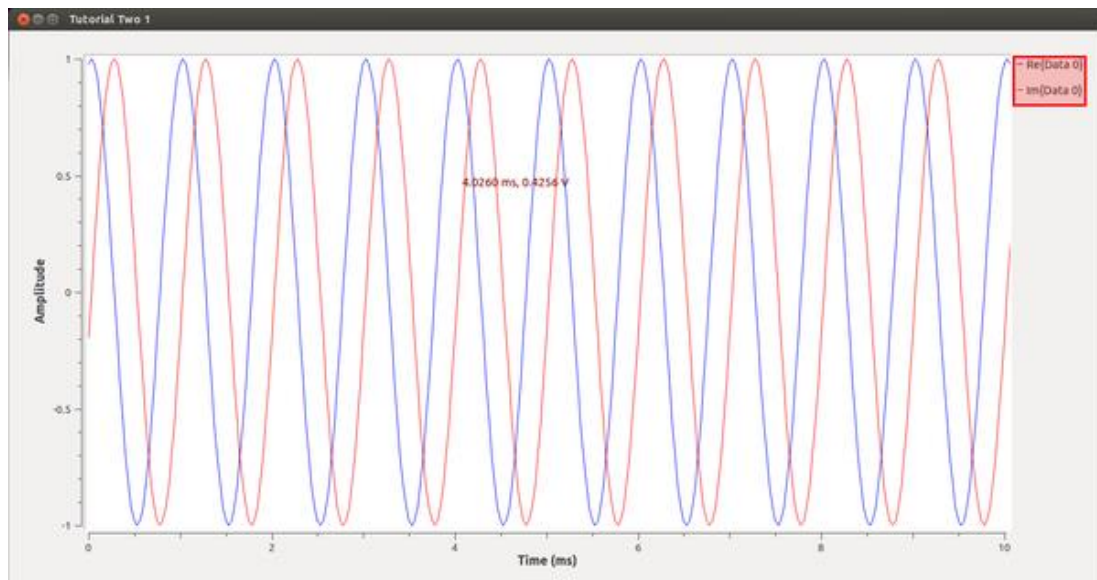


Figura 12. Diagrama de señal

Deberíamos obtener lo anterior, que es una senoide compleja de la forma $e^{j\omega t}$. Mantengamos las cosas simples evitando señales complejas por ahora. Primero queremos matar el diagrama de flujo con el botón Kill flowgraph o simplemente cerrando la GUI de Time Sink . Ahora es un buen momento para repasar los tipos de datos en GNU Radio abriendo la ventana Ayuda -> Tipos [20].

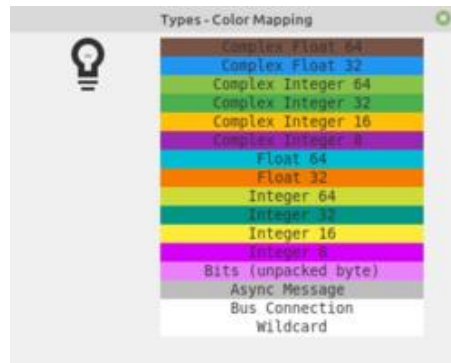


Figura 13. Color Mapping

Podemos ver los tipos de datos comunes que se ven en muchos lenguajes de programación. Podemos ver que nuestros bloques (puertos azules) son actualmente del tipo Complex Float 32, lo que significa que contienen tanto una parte real como una imaginaria, cada una de las cuales es del tipo Float 32. Podemos razonar que cuando Time Sink toma un tipo de datos complejo, genera tanto la parte real como la imaginaria en canales separados. Así que ahora cambiemos la Fuente de señal a un flotante yendo a sus propiedades de bloque y cambiando el parámetro Tipo de salida. Vemos que su puerto se vuelve naranja, ahora hay una flecha roja apuntando al Throttle Block, y en la Toolbar hay un botón rojo “-“ (rojo) que dice “Ver errores del gráfico de flujo”. Sigamos adelante y hagamos clic en eso [20].

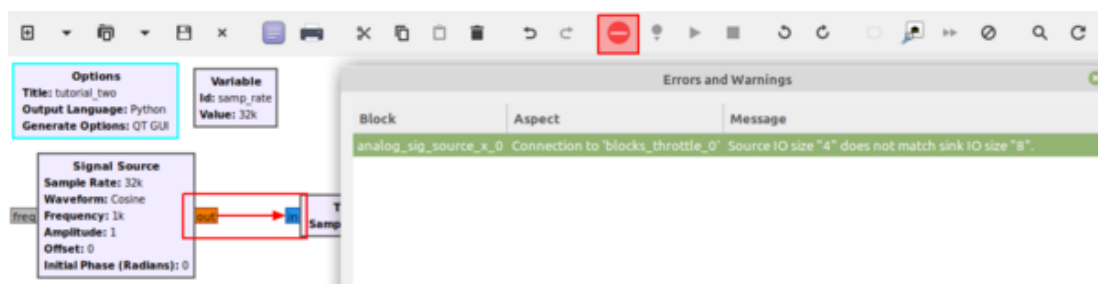


Figura 14. Flujo

Vemos que, en la conexión especificada, hay una discrepancia de tamaño. Esto se debe a que el tamaño de nuestro tipo de datos no coincide. GNU Radio no nos permitirá encadenar bloques de diferentes tamaños de datos, así que cambiemos el tipo de datos de todos nuestros bloques posteriores. Ahora podemos generar y ejecutar como antes. Ahora vemos nuestra onda sinusoidal en un canal [20].

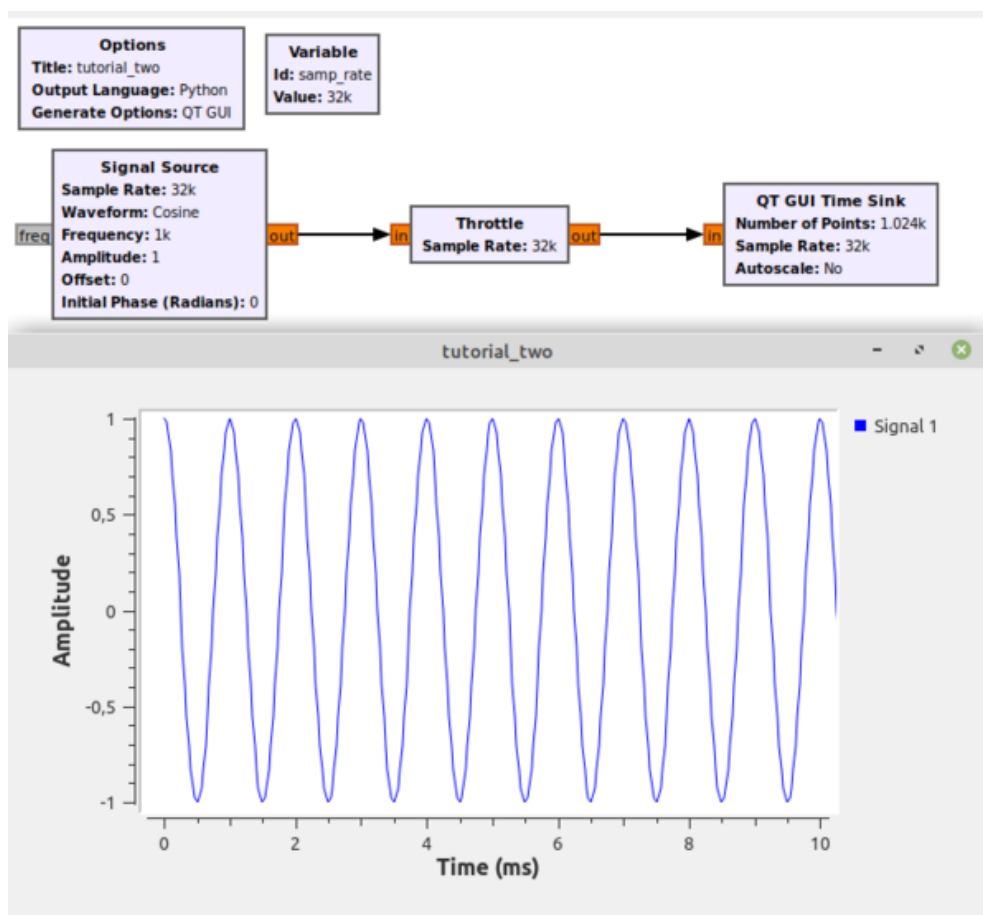


Figura 15. Resultado del diagrama de flujo

Instalación de GNU Radio

La mayoría de las distribuciones contienen un paquete con nombre `GNURadio` similar en sus repositorios estándar. Para la mayoría de los casos de uso, es suficiente instalar este paquete y comenzar a desarrollar [20].

El desarrollo de GNU Radio puede ser acelerado y los binarios proporcionados por su distribución pueden estar desactualizados. ¡Compruebe si la versión que está instalando está actualizada! A veces, las versiones antiguas no se actualizan en los sistemas de empaque. Si encuentra un error en una versión anterior de GNU Radio, verifique si el error aún existe en la versión más reciente de GNU Radio antes de presentar un nuevo problema [20].

Si la versión enviada en su distribución está desactualizada, comuníquese con el mantenedor correspondiente para actualizarla en el sistema de empaque.

Aquí hay ejemplos de cómo instalar GNU Radio en varias distribuciones de Linux [20].

Tabla 1: Comandos para instalar GNU Radio

Distribución	Comando
<u>Debian/Ubuntu and derivates</u>	# apt install gnuradio
<u>Fedora</u>	# dnf install gnuradio
RHEL/CentOS	# yum install gnuradio
<u>Archlinux</u>	# pacman -S gnuradio
<u>Gentoo Linux</u>	# emerge net-wireless/gnuradio

Instalación de Ubuntu PPA

Para Ubuntu, las últimas compilaciones (tanto publicadas como extraídas de la rama maestra) se mantienen como PPA en Launchpad . Asegúrese de desinstalar primero cualquier versión de gnuradio previamente instalada [20].

Para acceder a lo último desde la rama maestra, agregue el ppa gnuradio / gnuradio-master (eliminando otros ppas gnuradio si ya están configurados)

```
$ sudo add-apt-repository ppa: gnuradio / gnuradio-master
```

Para acceder a la versión publicada actual (3.9), agregue el ppa gnuradio / gnuradio-releases (eliminando otros ppas gnuradio si ya están configurados)

```
$ sudo add-apt-repository ppa: gnuradio / gnuradio-releases
```

Para acceder a la versión 3.8, agregue el ppa gnuradio / gnuradio-releases-3.8 (eliminando otros ppas gnuradio si ya están configurados)

```
$ sudo add-apt-repository ppa: gnuradio / gnuradio-releases-3.8
```

Para acceder a la versión 3.7 lanzada (heredada), agregue el ppa gnuradio / gnuradio-releases-3.7 (eliminando otros ppas gnuradio si ya están configurados)

```
$ sudo add-apt-repository ppa: gnuradio / gnuradio-releases-3.7
```

Luego, actualice las fuentes de apt e instale gnuradio

```
$ sudo apt-get update
```

```
$ sudo apt install gnuradio
```

Atención: Do no intente instalar nuevos paquetes como `gr-osmosdr` a través de la gestión de paquetes de Ubuntu (es decir, el uso de "apt"). Ubuntu intentará instalar una versión potencialmente incompatible y su sistema estará en un estado indefinido [20].

ACTIVIDADES POR DESARROLLAR:

- 1) En Laptop o PC instalar Ubuntu última versión.
- 2) Instalar GNURadio última versión.
- 3) Abrir GNURadio, explorar y validar lo visto en el marco teórico.
- 4) Realizar diagrama de bloques básico de transmisión de señales con GNURadio.
- 5) Realizar diagrama de bloques básico de recepción de señales con GNURadio.

RESULTADO(S) OBTENIDO(S):

El estudiante debe colocar los resultados obtenidos en la práctica con imágenes y descripciones de cada imagen.

CONCLUSIONES:

El estudiante debe colocar las conclusiones de las prácticas de acuerdo con los objetivos planteados.

CARRERA: INGENIERÍA ELECTRÓNICA

ASIGNATURA:

NRO. PRÁCTICA:

2

TÍTULO PRÁCTICA: Configuración e implementación de Transmisor FM con GNURadio, USRP, amplificador FM y antena FM.

OBJETIVO GENERAL:

Configurar e implementar transmisor FM con GNURadio, NI USRP, amplificador FM y antena FM.

OBJETIVOS ESPECÍFICOS:

- Configurar transmisor FM con GNURadio
- Configurar USRP para transmisión de señales FM
- Conectar y configurar amplificador FM con antena FM.

INSTRUCCIONES

1. Los estudiantes deben leer previamente el manual de práctica para el desarrollo.

2. Los estudiantes deben utilizar los equipos del banco de pruebas de radiodifusión FM de una manera responsable y calificada para evitar daños en los equipos.

3. Los estudiantes deben trabajar en grupo para el desarrollo de la práctica.

4. Se debe dejar en orden el sitio de práctica luego del desarrollo de esta.

ACTIVIDADES POR DESARROLLAR:

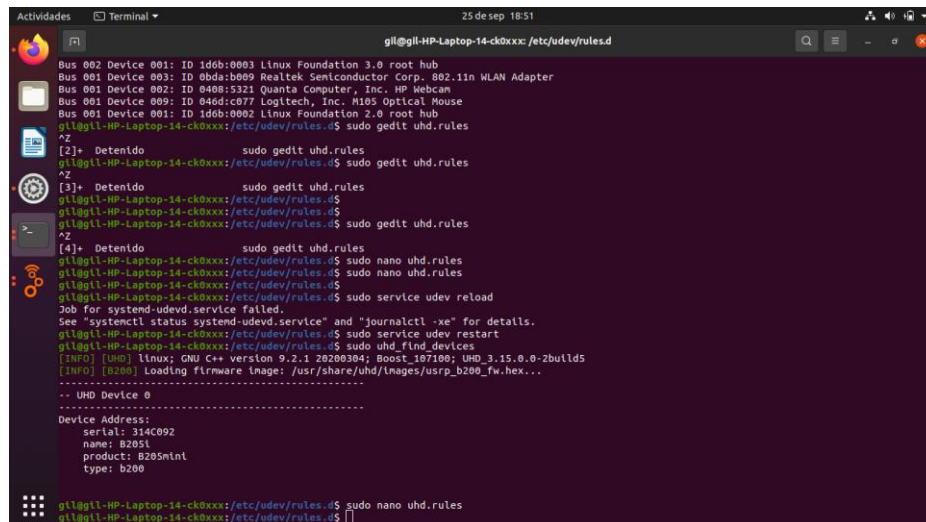
- En esta práctica se explica en detalle la configuración de la programación del transmisor FM configurado en GNURadio.
- Para la práctica 2 se necesita una Laptop o PC con Ubuntu 20.
- Se realiza la instalación de GNURadio en Ubuntu con los siguientes comandos ejecutados desde el terminal.

```
sudo add-apt-repository ppa:gnuradio/gnuradio-releases
```

```
sudo apt-get update
```

```
sudo apt-get install gnuradio
```

- Se debe validar que el USRP B205 mini-i sea reconocido en la Laptop con Ubuntu, para esto se debe realizar las siguientes configuraciones de permisos en el puerto USB.



```
gll@gll-HP-Laptop-14-ck8xxx: /etc/udev/rules.d
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 0bda:b009 Realtek Semiconductor Corp. 802.11n WLAN Adapter
Bus 001 Device 002: ID 0408:3221 Quanta Computer, Inc. HP Webcam
Bus 001 Device 009: ID 046d:c077 Logitech, Inc. M105 Optical Mouse
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
gll@gll-HP-Laptop-14-ck8xxx:/etc/udev/rules.d$ sudo gedit uhd.rules
^Z
[2]+ Detenido          sudo gedit uhd.rules
gll@gll-HP-Laptop-14-ck8xxx:/etc/udev/rules.d$ sudo gedit uhd.rules
^Z
[3]+ Detenido          sudo gedit uhd.rules
gll@gll-HP-Laptop-14-ck8xxx:/etc/udev/rules.d$ sudo gedit uhd.rules
gll@gll-HP-Laptop-14-ck8xxx:/etc/udev/rules.d$ sudo gedit uhd.rules
^Z
[4]+ Detenido          sudo gedit uhd.rules
gll@gll-HP-Laptop-14-ck8xxx:/etc/udev/rules.d$ sudo nano uhd.rules
gll@gll-HP-Laptop-14-ck8xxx:/etc/udev/rules.d$ sudo nano uhd.rules
gll@gll-HP-Laptop-14-ck8xxx:/etc/udev/rules.d$ sudo nano uhd.rules
gll@gll-HP-Laptop-14-ck8xxx:/etc/udev/rules.d$ sudo service udev reload
Job for systemd-udev.service failed.
See "systemctl status systemd-udev.service" and "journalctl -xe" for details.
gll@gll-HP-Laptop-14-ck8xxx:/etc/udev/rules.d$ sudo service udev restart
gll@gll-HP-Laptop-14-ck8xxx:/etc/udev/rules.d$ sudo uhd_find_devices
[INFO] [UHD] linux; GNU C++ version 9.2.1 20200304; Boost 107100; UHD 3.15.0.0-2build1
[INFO] [B200] Loading firmware image: /usr/share/uhd/firmware/usrp_b200_fw.hex...
-----
-- UHD Device 0
-----
Device Address:
  serial: 314C092
  name: B205i
  product: B205mini
  type: b200
gll@gll-HP-Laptop-14-ck8xxx:/etc/udev/rules.d$ sudo nano uhd.rules
gll@gll-HP-Laptop-14-ck8xxx:/etc/udev/rules.d$
```

Figura 1. Comandos uhd

```

gll@gll-HP-Laptop-14-ck0xxx: /usr/lib/uhd/uttl$ sudo uhd_images_downloader.py
sudo: uhd_images_downloader.py: orden no encontrada
gll@gll-HP-Laptop-14-ck0xxx: /usr/lib/uhd/uttl$ sudo python3 uhd_images_downloader.py
[INFO] Images destination: /usr/share/uhd/images
[INFO] No inventory file found at /usr/share/uhd/images/inventory.json. Creating an empty one.
19442 KB / 19442 KB (100%) x3xx_x310_fpga_default-gfde2a94e.zip
18697 KB / 18697 KB (100%) x3xx_x300_fpga_default-gfde2a94e.zip
01534 KB / 01534 KB (100%) e3xx_e310_sgl_fpga_default-gfde2a94e.zip
01522 KB / 01522 KB (100%) e3xx_e310_sgl_fpga_default-gfde2a94e.zip
09070 KB / 09070 KB (100%) e3xx_e320_fpga_default-gfde2a94e.zip
23071 KB / 23071 KB (100%) n3xx_n310_fpga_default-gfde2a94e.zip
16072 KB / 16072 KB (100%) n3xx_n300_fpga_default-gfde2a94e.zip
24996 KB / 24996 KB (100%) n3xx_n320_fpga_default-gfde2a94e.zip
00479 KB / 00479 KB (100%) b2xx_b200_fpga_default-gfde2a94e.zip
00464 KB / 00464 KB (100%) b2xx_b200mini_fpga_default-gfde2a94e.zip
00879 KB / 00879 KB (100%) b2xx_b210_fpga_default-gfde2a94e.zip
00523 KB / 00523 KB (100%) b2xx_b205mini_fpga_default-gfde2a94e.zip
00162 KB / 00162 KB (100%) b2xx_common_fw_default-g2bdad498.zip
00007 KB / 00007 KB (100%) usrp2_usrp2_fw_default-gsbea23d.zip
00450 KB / 00450 KB (100%) usrp2_usrp2_fpga_default-gsbea23d.zip
02415 KB / 02415 KB (100%) usrp2_n200_fpga_default-gsbea23d.zip
00009 KB / 00009 KB (100%) usrp2_n200_fw_default-gsbea23d.zip
02757 KB / 02757 KB (100%) usrp2_n210_fpga_default-gsbea23d.zip
00009 KB / 00009 KB (100%) usrp2_n210_fw_default-gsbea23d.zip
02076 KB / 02076 KB (100%) n230_n230_fpga_default-gfde2a94e.zip
00319 KB / 00319 KB (100%) usrp1_usrp1_fpga_default-gsbea23d.zip
00522 KB / 00522 KB (100%) usrp1_b100_fpga_default-gsbea23d.zip
00006 KB / 00006 KB (100%) usrp1_b100_fw_default-gsbea23d.zip
00017 KB / 00017 KB (100%) octoclock_octoclock_fw_default-g14000041.zip
04839 KB / 04839 KB (100%) usb_common_wlndrv_default-g14000041.zip
[INFO] Images download complete.
gll@gll-HP-Laptop-14-ck0xxx: /usr/lib/uhd/uttl$ sudo uhd_find_devices
[INFO] [UHD] linux; GNU C++ version 9.2.1 20200304; Boost 107100; UHD 3.15.0.0-2builds
[INFO] [B200] Loading firmware image: /usr/share/uhd/images/usrp_b200_fw.hex...
-- UHD Device 0
Device Address:
.....
serial: 314C092
name: B2051
product: B205mini
type: b200
gll@gll-HP-Laptop-14-ck0xxx: /usr/lib/uhd/uttl$ sudo uhd_find_devices
[INFO] [UHD] linux; GNU C++ version 9.2.1 20200304; Boost 107100; UHD 3.15.0.0-2builds
[INFO] [B200] Loading firmware image: /usr/share/uhd/images/usrp_b200_fw.hex...
-- UHD Device 0
Device Address:
.....
serial: 314C092
name: B2051
product: B205mini
type: b200
gll@gll-HP-Laptop-14-ck0xxx: /usr/lib/uhd/uttl$ cd /etc/udev/rules.d/
gll@gll-HP-Laptop-14-ck0xxx: /etc/udev/rules.d$ sudo touch uhd.rules
gll@gll-HP-Laptop-14-ck0xxx: /etc/udev/rules.d$ sudo gedit uhd.rules
^Z
[1]+ Detenido          sudo gedit uhd.rules
gll@gll-HP-Laptop-14-ck0xxx: /etc/udev/rules.d$ sudo lsusb -v
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB                  3.00
  bDeviceClass             9 Hub
  bDeviceSubClass          0

```

Figura 2. Instalación uhd_images_downloader.py

```

gll@gll-HP-Laptop-14-ck0xxx: /usr/lib/uhd/uttl$ cd /etc/udev/rules.d/
gll@gll-HP-Laptop-14-ck0xxx: /etc/udev/rules.d$ sudo touch uhd.rules
gll@gll-HP-Laptop-14-ck0xxx: /etc/udev/rules.d$ sudo gedit uhd.rules
^Z
[1]+ Detenido          sudo gedit uhd.rules
gll@gll-HP-Laptop-14-ck0xxx: /etc/udev/rules.d$ sudo lsusb -v
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB                  3.00
  bDeviceClass             9 Hub
  bDeviceSubClass          0

```

Figura 3. Configuración de reglas de USB.

```

Actividades Terminal 25 de sep 18:43
gll@gll-HP-Laptop-14-c0xxx: /etc/udev/rules.d

No power switching (usb 1.0)
Per-port overcurrent protection
TT think time 8 FS bits
bWron2PerGood 10 * 2 milli seconds
bHubConcurrent 0 milli Ampere
DeviceRenovable 0x20 0x01
PortPwrCtrlMask 0xff 0xff
Hub Port Status:
Port 1: 0000.0100 power
Port 2: 0000.0100 power
Port 3: 0000.0100 power
Port 4: 0000.0100 power
Port 5: 0000.0100 power
Port 6: 0000.0503 highspeed power enable connect
Port 7: 0000.0100 power
Port 8: 0000.0103 power enable connect
Port 9: 0000.0100 power
Port 10: 0000.0100 power
Port 11: 0000.0100 power
Port 12: 0000.0100 power
can't get device qualifier: Resource temporarily unavailable
can't get debug descriptor: Resource temporarily unavailable
Device Status: 0x0001
Self Powered
gll@gll-HP-Laptop-14-c0xxx: /etc/udev/rules.d$ sudo lsusb
Bus 002 Device 006: ID 2500:0022 Ettus Research LLC Unknown
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 0bda:b009 Realtek Semiconductor Corp. 802.11n WLAN Adapter
Bus 001 Device 002: ID 0408:5321 Quanta Computer, Inc. HP Webcam
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
gll@gll-HP-Laptop-14-c0xxx: /etc/udev/rules.d$ sudo lsusb
Bus 002 Device 006: ID 2500:0022 Ettus Research LLC Unknown
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 0bda:b009 Realtek Semiconductor Corp. 802.11n WLAN Adapter
Bus 001 Device 002: ID 0408:5321 Quanta Computer, Inc. HP Webcam
Bus 001 Device 009: ID 046d:c077 Logitech, Inc. M105 Optical Mouse
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
gll@gll-HP-Laptop-14-c0xxx: /etc/udev/rules.d$

```

Figura 4. Configuración lsusb

- Se realiza la siguiente configuración en GNURadio para el transmisor TX

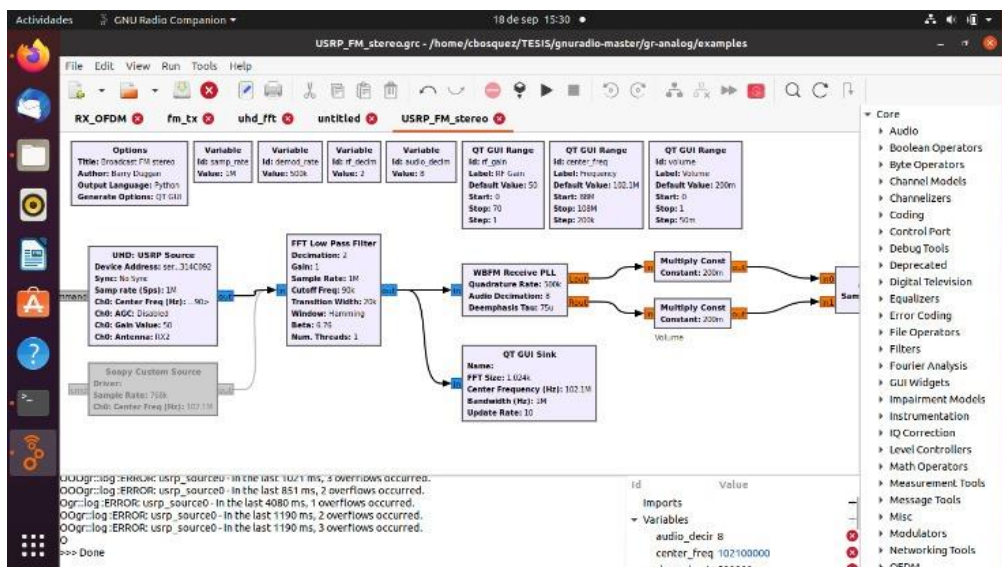


Figura 5. Diagrama de bloques del TX FM

Descripción de pantalla principal de GNURadio

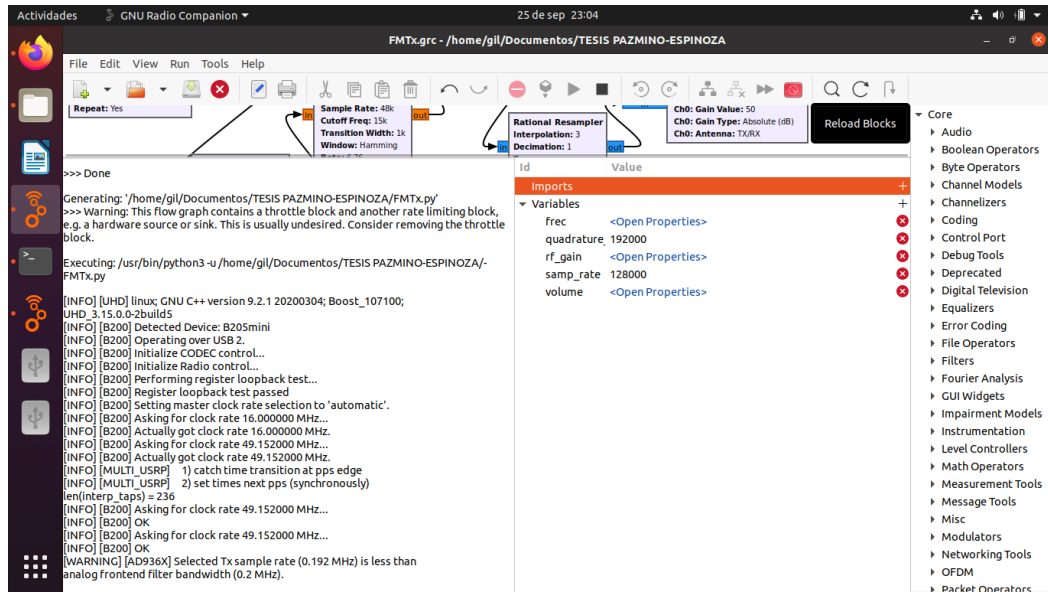


Figura 6. Paneles de visualización de procesos de GNURadio

A continuación, se detalla la programación de cada uno de los bloques.

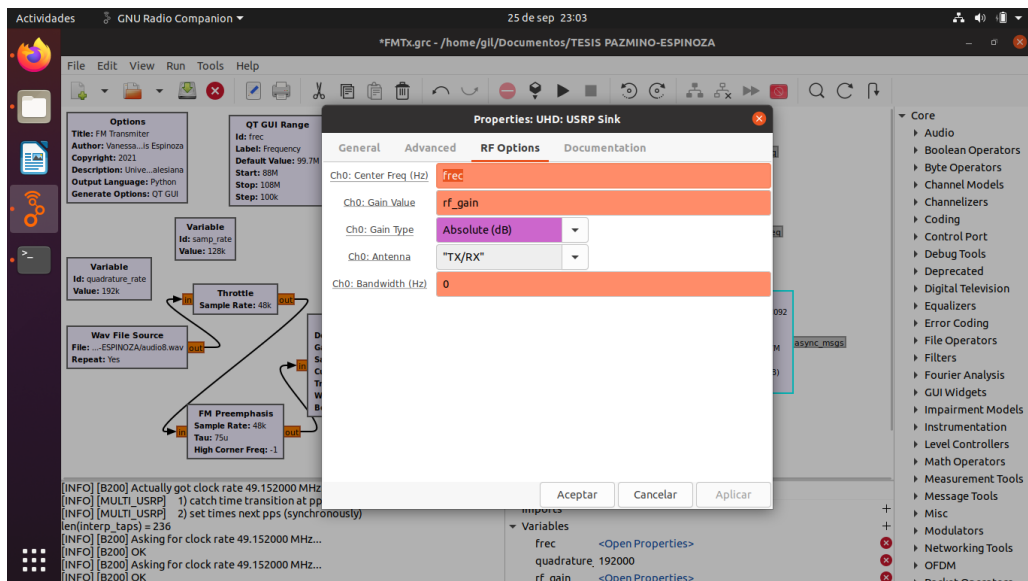


Figura 7. Configuración USRP Sink

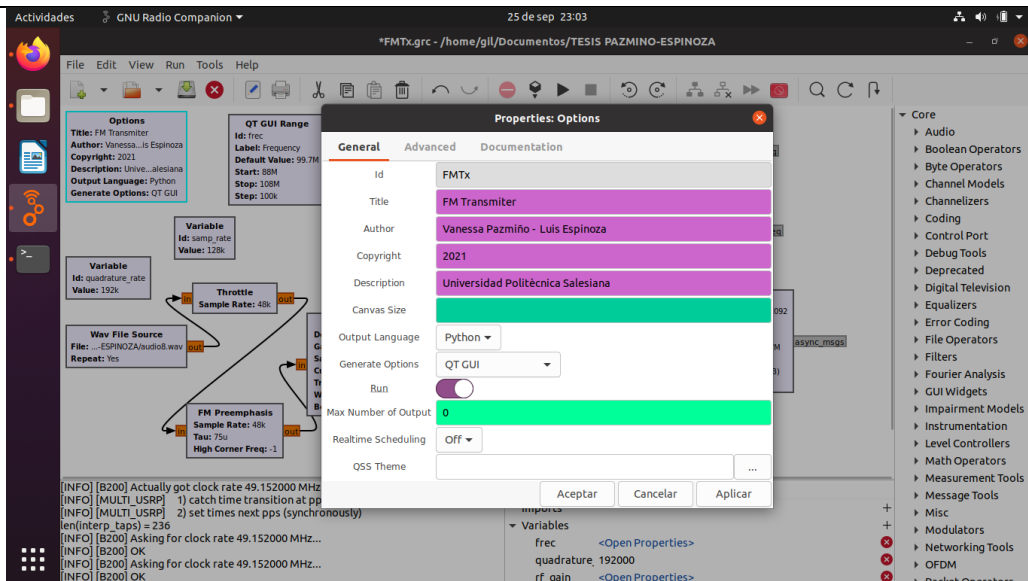


Figura 8. Opciones de USRP Sink

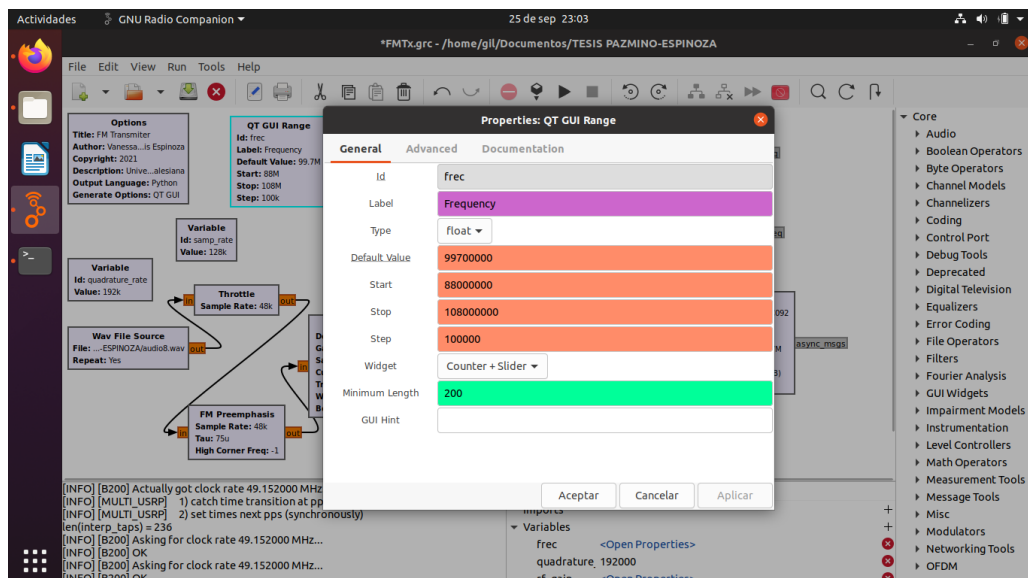


Figura 9. QT GUI Range

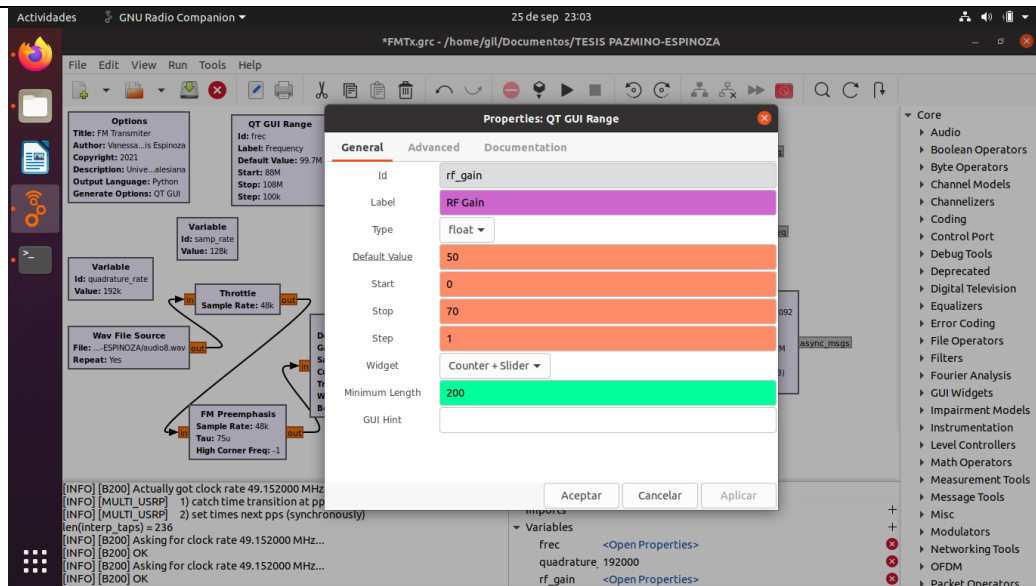


Figura 10. RF Gain

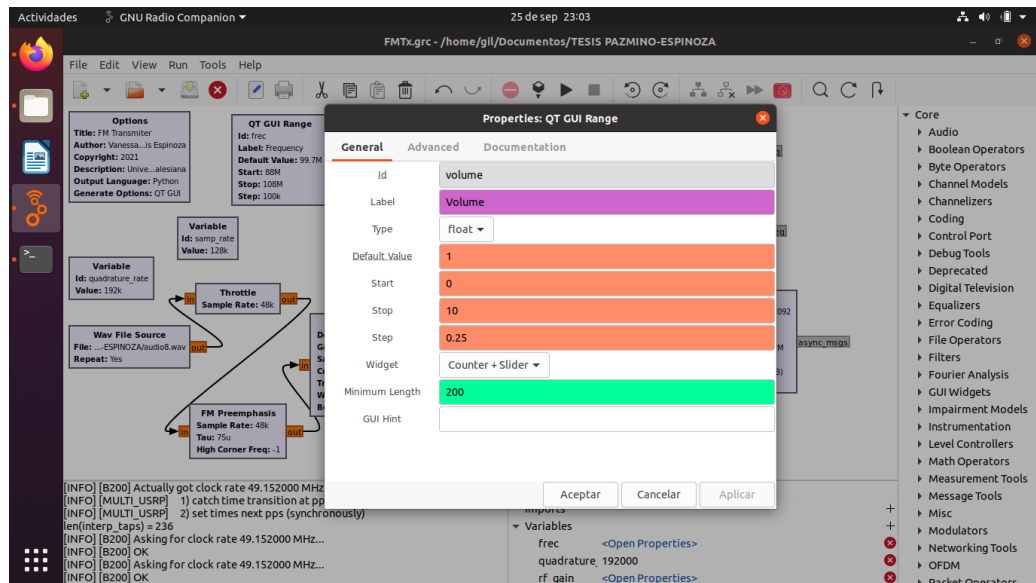


Figura 11. QT GUI Range

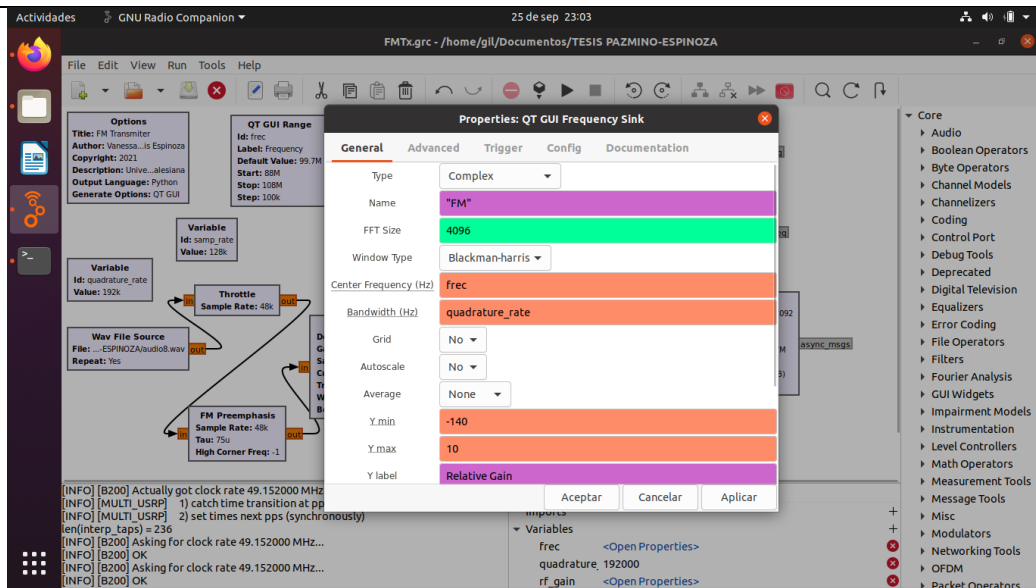


Figura 12. QT GUI Frecuencia

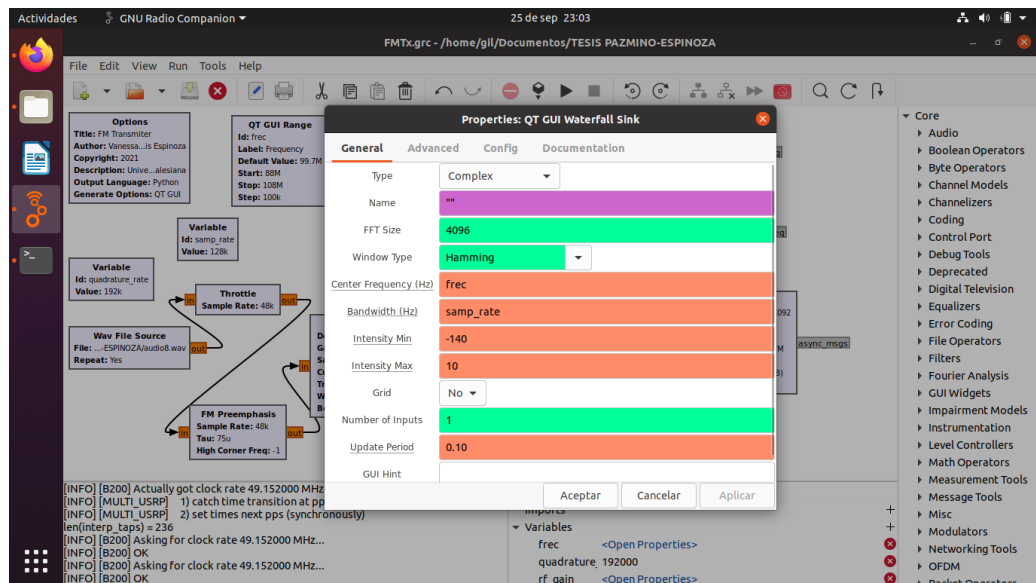


Figura 13. QT Waterfall Sink

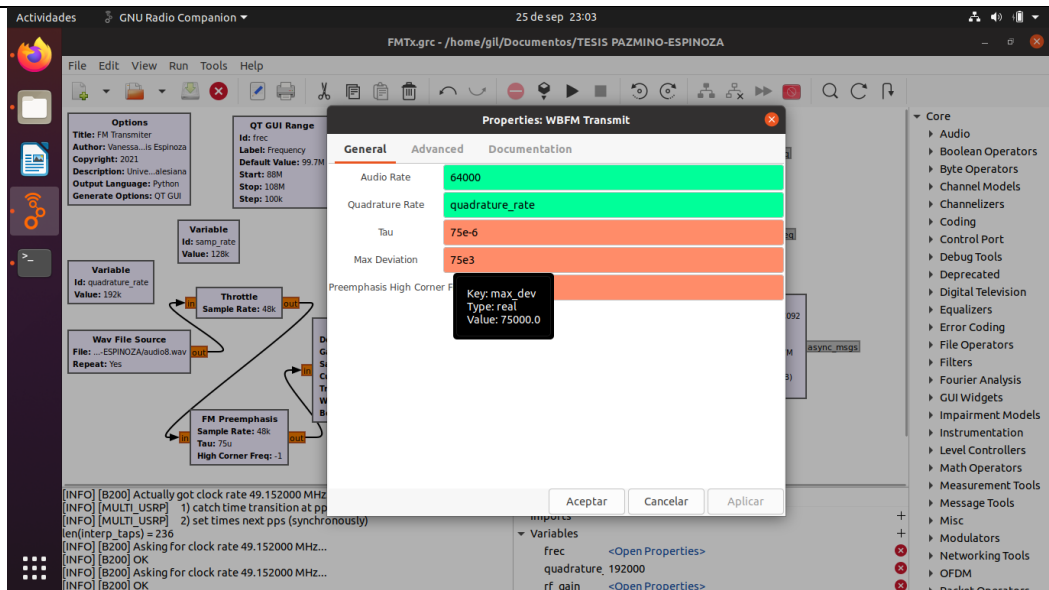


Figura 14. Transmisor WBFM

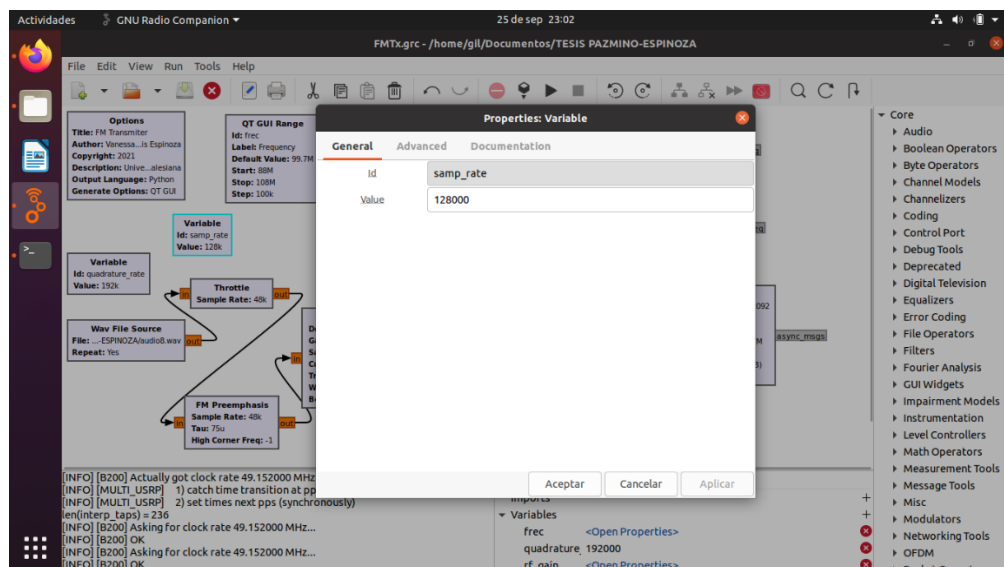


Figura 15. Variable samp_rate

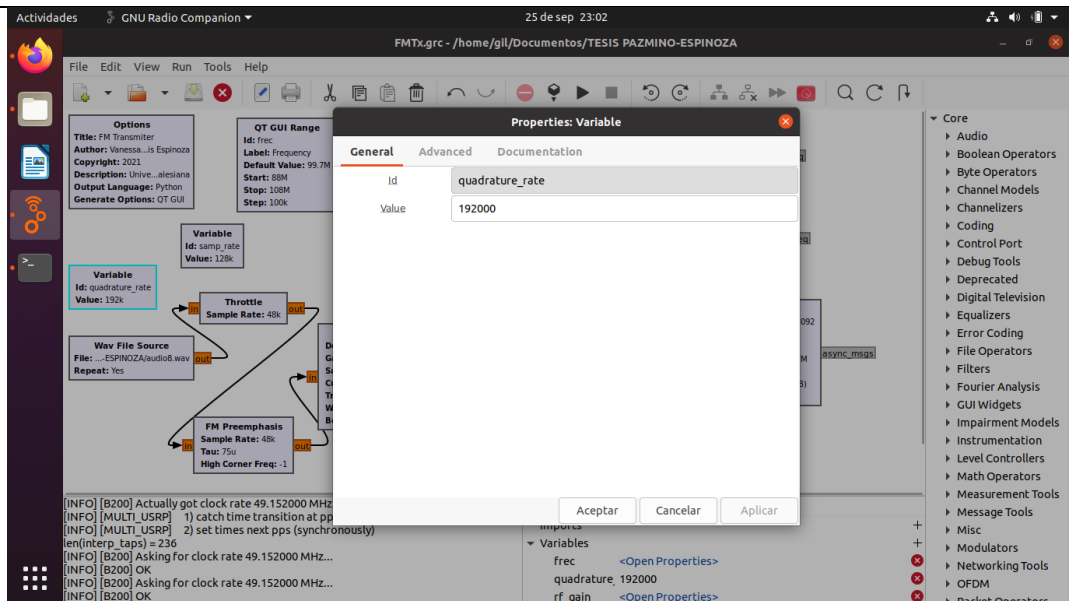


Figura 16. Variable quadrature_rate

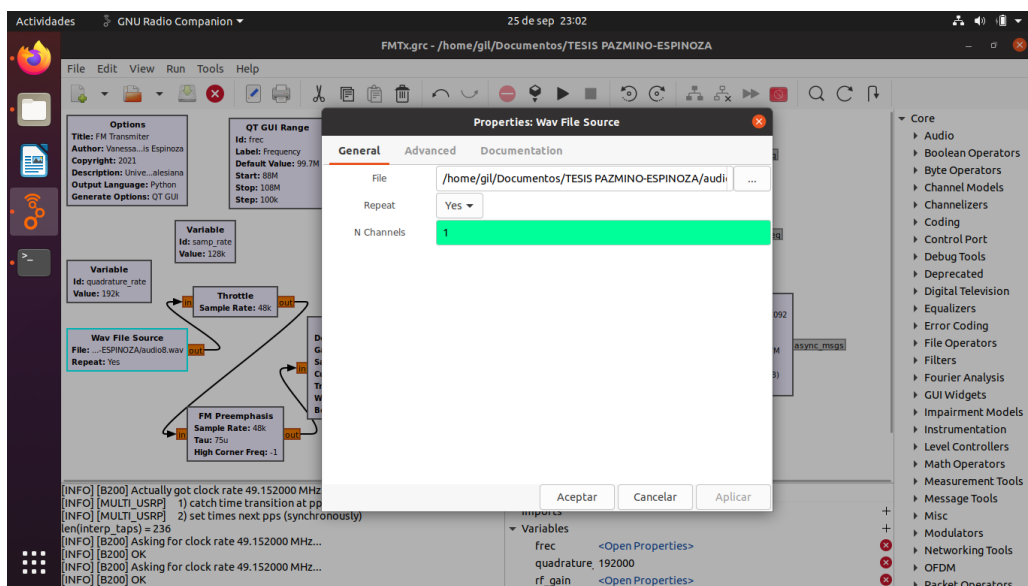


Figura 17. Wav file source

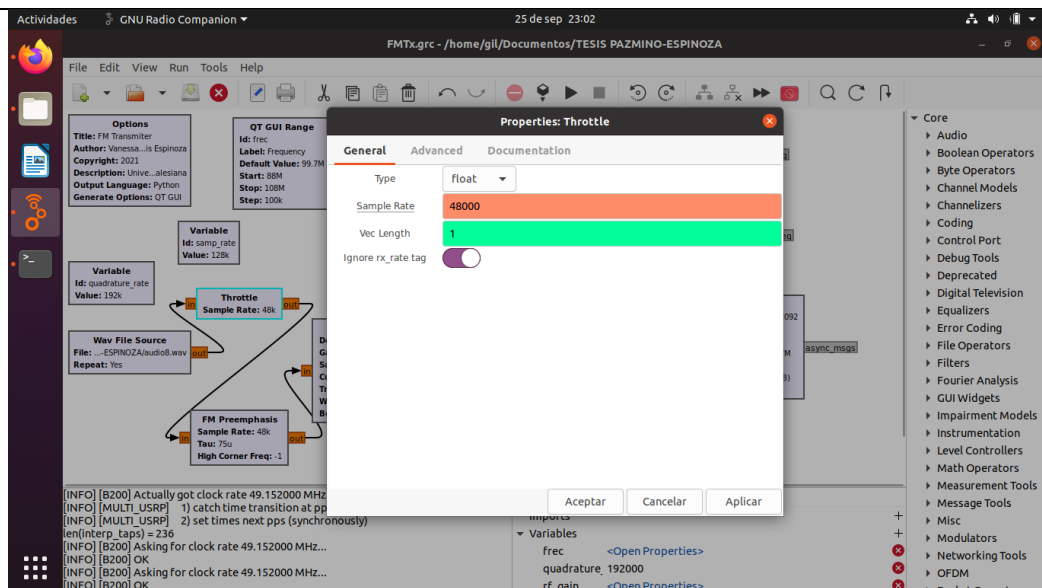


Figura 18. Throttle

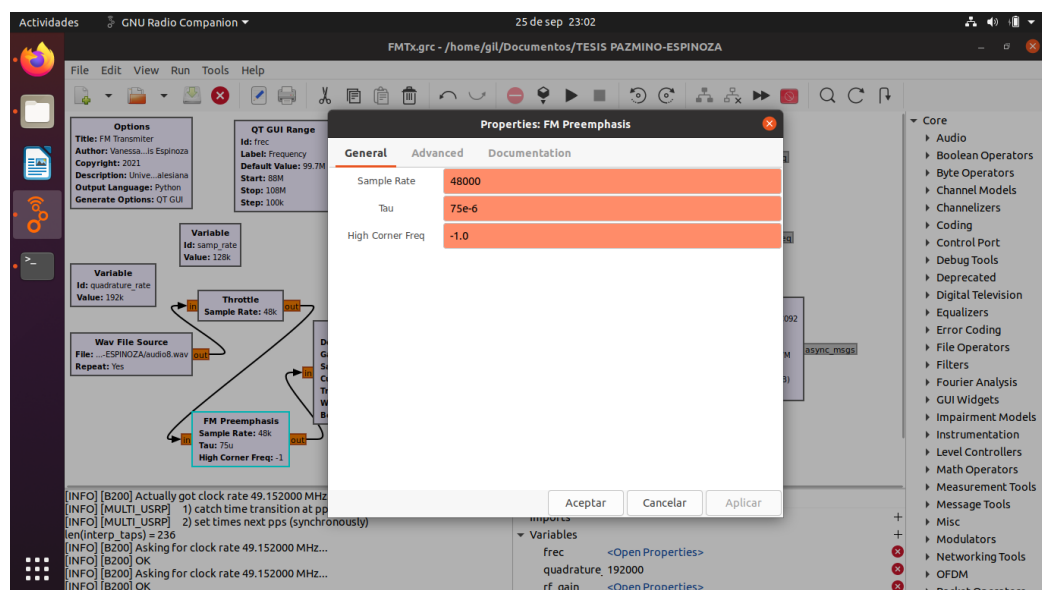


Figura 19. FM Preemphasis

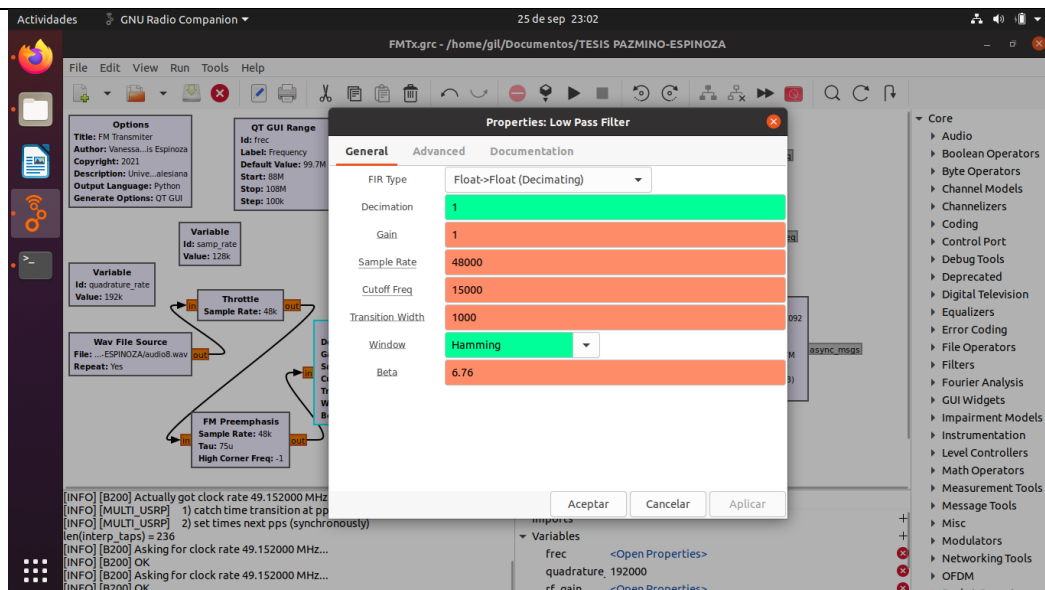


Figura 20. Low Pass filter

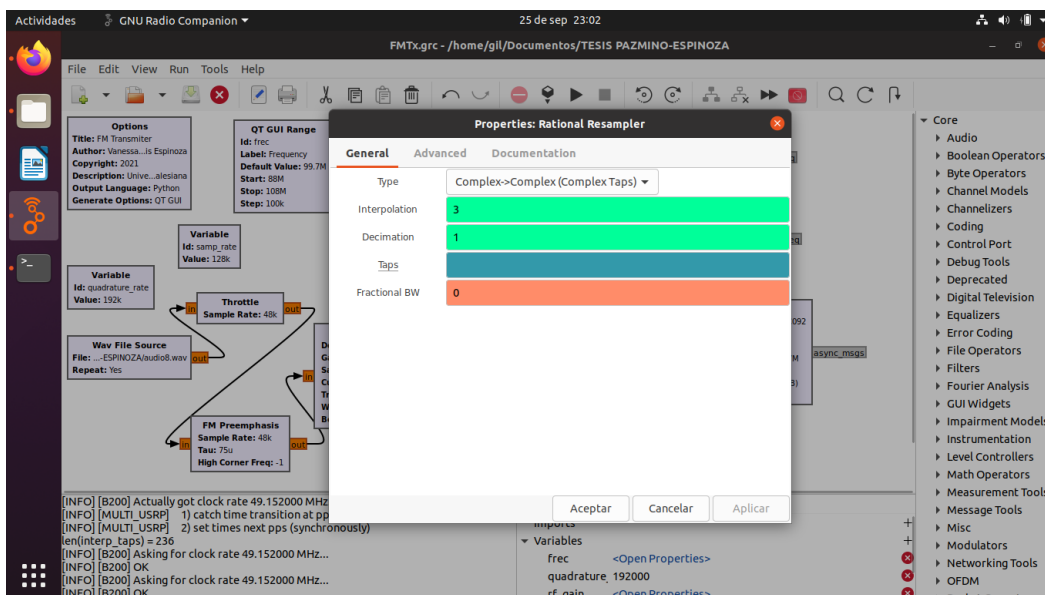


Figura 21. Rational resampler

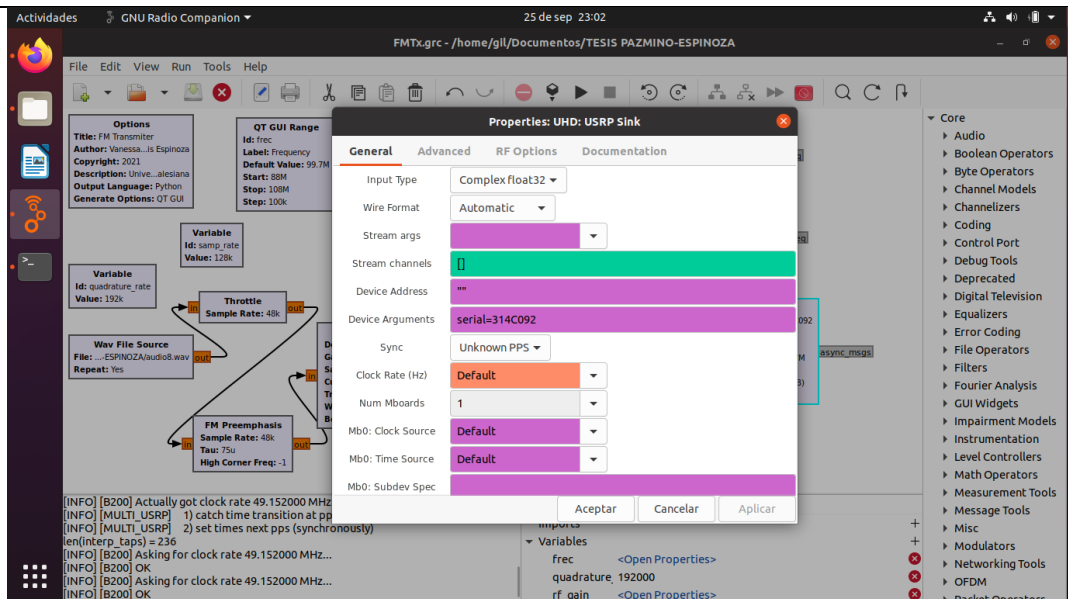


Figura 22. UHD Sink

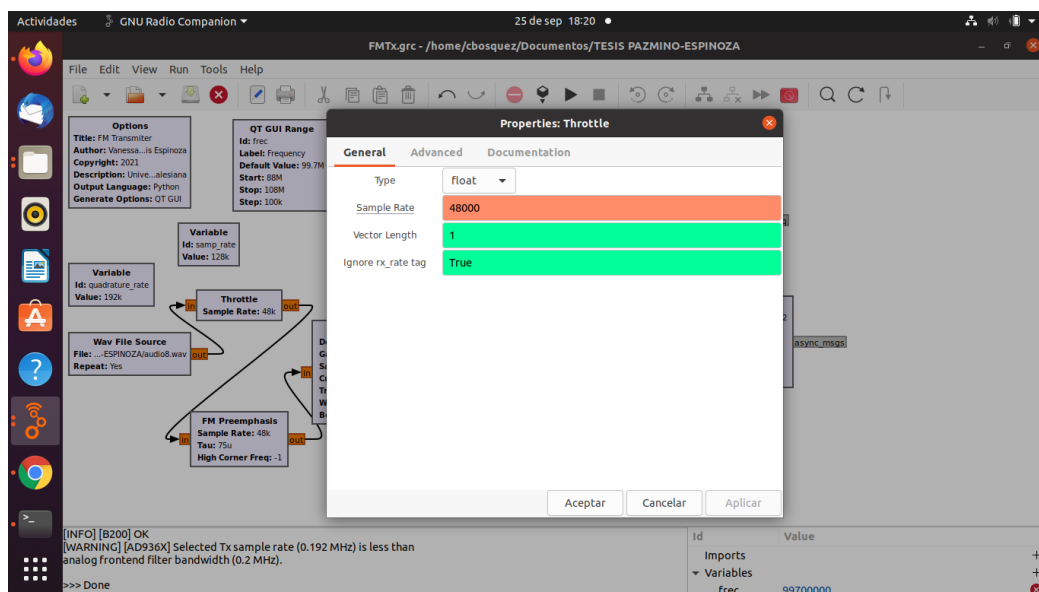


Figura 23. Throttle

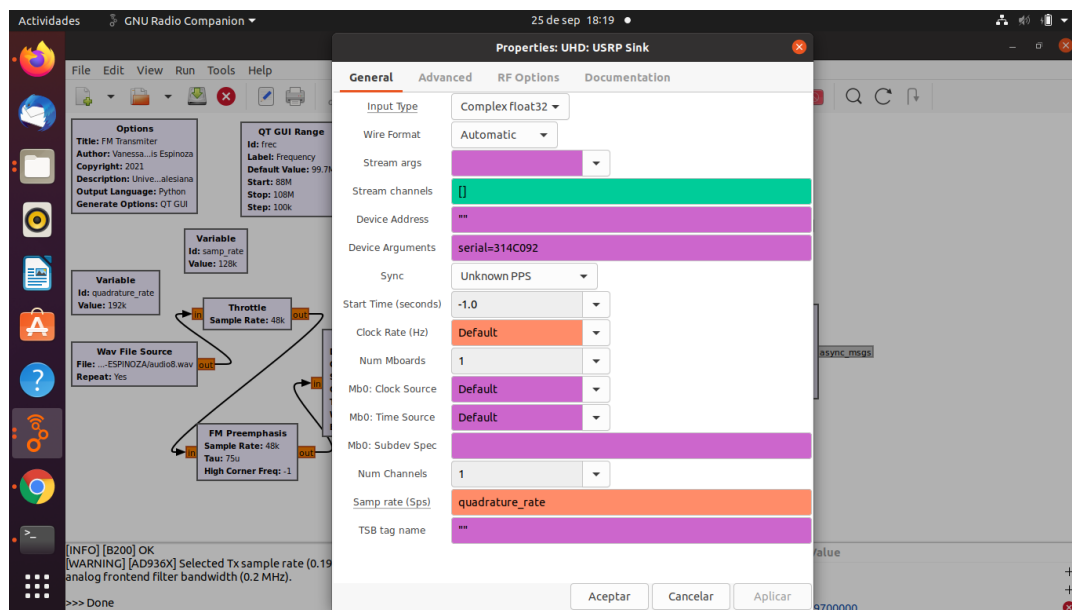


Figura 24. UHD Sink general

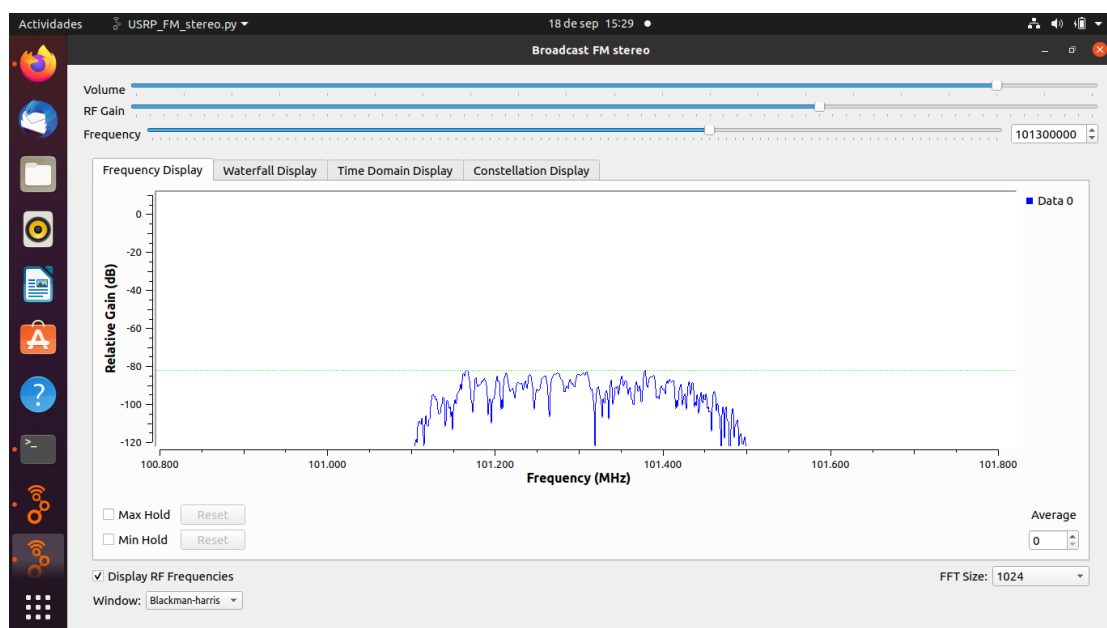


Figura 25. Analizador de espectros

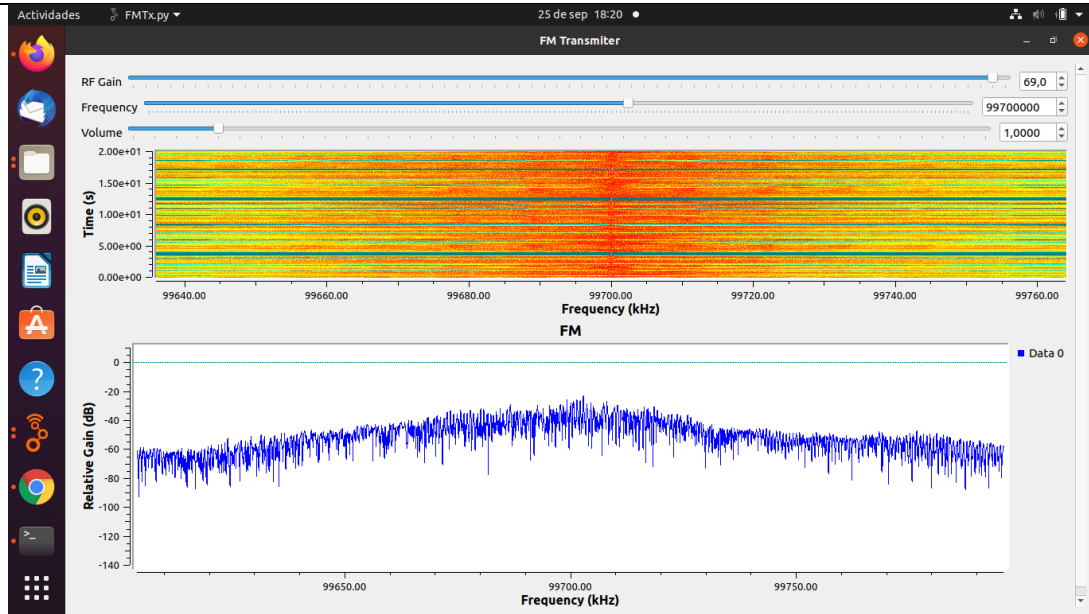


Figura 26. Waterfall

Las conexiones físicas del banco de pruebas se observan en la figura 27 y 28.

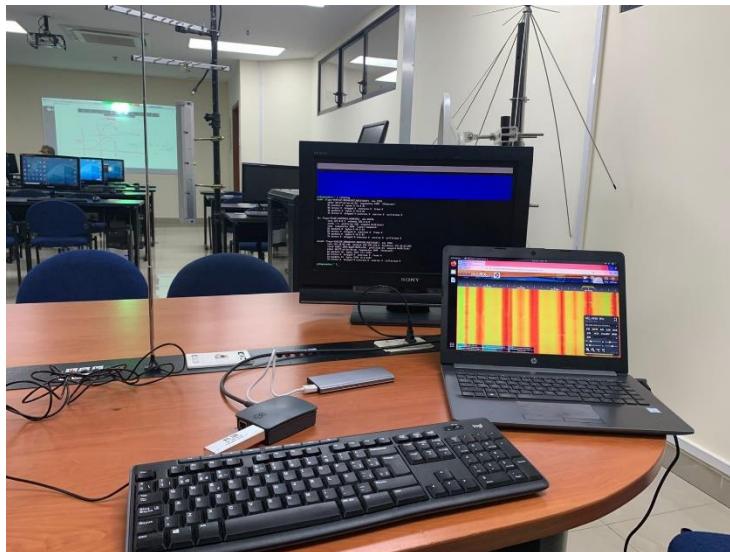


Figura 27. Conexiones del banco de pruebas



Figura 28. Conexiones del banco de pruebas



Figura 29. Pruebas del banco FM

RESULTADO(S) OBTENIDO(S):
El estudiante debe colocar los resultados obtenidos en la práctica con imágenes y descripciones de cada imagen
CONCLUSIONES:
El estudiante debe colocar las conclusiones de las prácticas de acuerdo con los objetivos planteados.

CARRERA: INGENIERÍA ELECTRÓNICA

ASIGNATURA:

NRO. PRÁCTICA:

3

TÍTULO PRÁCTICA: Configuración e implementación de Receptor FM con RTL-SDR, Raspberry PI

OBJETIVO GENERAL:

Configurar e implementar receptor FM con RTL-SDR, Raspberry PI.

OBJETIVOS ESPECÍFICOS:

- Configurar receptor FM con RT-SDR
- Configurar Raspberry PI y OpenWebRX
- Conectar Laptop o celular para monitorización web

INSTRUCCIONES

1. Los estudiantes deben leer previamente el manual de práctica para el desarrollo.

2. Los estudiantes deben utilizar los equipos del banco de pruebas de radiodifusión FM de una manera responsable y calificada para evitar daños en los equipos.

3. Los estudiantes deben trabajar en grupo para el desarrollo de la práctica.

4. Se debe dejar en orden el sitio de práctica luego del desarrollo de esta.

ACTIVIDADES POR DESARROLLAR:

- 1) Realizar la instalación de OpenWebRX en la microSD de la Raspberry PI.

OpenWebRX

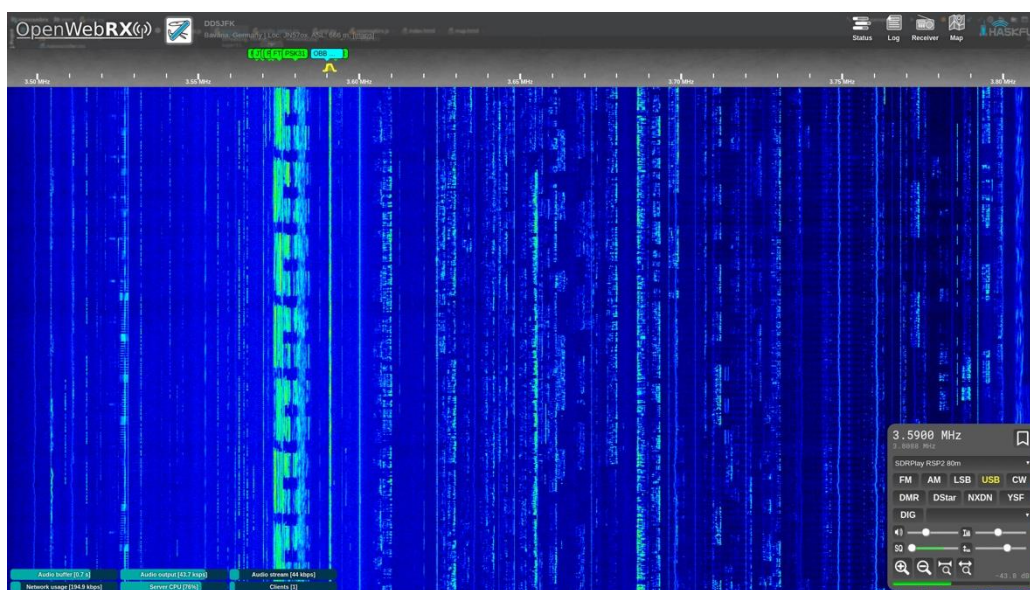


Figura 1. OpenWebRX

OpenWebRX es un receptor SDR multiusuario que se puede operar desde cualquier navegador web sin la necesidad de ningún software de cliente adicional. Es la solución ideal para proporcionar acceso al espectro de HF en su ubicación de elección a una amplia audiencia. Todo lo que necesita es una computadora, un dispositivo SDR y acceso a la red.

OpenWebRX tiene como objetivo admitir tantas modulaciones y codificaciones como sea posible, sin dejar de centrarse en una interfaz fácil de usar, de modo que incluso los usuarios sin experiencia puedan explorar el espectro de HF sin la necesidad de comprar equipos de radio costosos.

Antes de comenzar con la instalación real, eche un vistazo rápido a la página de hardware compatible para asegurarse de que está listo para comenzar.

Hay algunas formas básicas de configurar un receptor OpenWebRX entre las que puede elegir. Las instrucciones son diferentes para cada método, así que siga su elección. Es probable que elegir pasos de otras instrucciones no funcione ni mejore sus resultados.

Cuando haya terminado con la instalación básica, no dude en continuar con la guía de configuración .

Imágenes de la tarjeta SD de Raspberry Pi

La forma más conveniente es configurar un sistema basado en Raspberry Pi, todo lo que necesita es una tarjeta SD de repuesto.

Algunas notas sobre las imágenes (todas ellas también se aplican al sistema operativo Raspberry Pi):

- El usuario y la contraseña predeterminados son usuario: pi, contraseña: frambuesa
- La distribución del teclado predeterminada es "gb", por lo que las teclas pueden estar en una ubicación inusual (dependiendo de lo que esté acostumbrado).
- El servidor ssh está deshabilitado de forma predeterminada. Si desea habilitarlo, coloque un archivo con el nombre ssh(no se necesita contenido) en el volumen de "arranque" después de flashear la imagen.
- Puede utilizar la herramienta oficial "raspi-config" para personalizar su sistema.
- Wi-Fi está inicialmente desactivado hasta que ajuste su ubicación Wi-Fi en raspi-config virtud de Localisation Options > Change WLAN Country

Una vez que haya arrancado una Raspberry con la tarjeta SD, aparecerá en su red con el nombre de host "openwebrx", que debería hacer que esté disponible como `https://openwebrx/` en la mayoría de las redes. Esto puede variar según su configuración específica.

Para la voz digital, el requisito mínimo en este momento parece ser una Raspberry Pi 3B+. Me gustaría trabajar en la optimización de esto para especificaciones más bajas, pero en este punto no estoy seguro de cuánto se puede hacer.

Instalación del paquete

Estamos proporcionando repositorios de paquetes para las distribuciones de Debian y Ubuntu Linux que contienen paquetes para OpenWebRX y la mayoría de sus dependencias. También hay un repositorio experimental disponible que le brinda acceso a las últimas actualizaciones de desarrollo.

Limitaciones:

- Limitado a Debian Buster o Ubuntu Hirsute Hippo en x86_64, armv7l y aarch64 (Sí, funcionará en Raspberry Pi OS)
- No todos los controladores SDR están disponibles empaquetados en un repositorio, por lo que la compatibilidad con dispositivos es limitada (consulte esta tabla para obtener más detalles). Los controladores y módulos aún se pueden instalar manualmente, por supuesto.
- Lo mismo se aplica al software, sobre todo WSJT-X, que en este momento solo está disponible en la versión 2.0 en el repositorio estable de Debian. Eso significa que FT4 (que se agregó en 2.1) no funcionará hasta que lo reemplace con una versión más nueva de la página de inicio de WSJT-X.

Resultados de la configuración de OpenWebRX.

- Una vez realizado la configuración de OpenWebRX en la microSD de la Raspberry se ingresa por la ip y el puerto designado.
- 2) Abrir desde la laptop vía HTTP con la ip del Raspberry y el puerto de OpenWebRT.
Ver figura 2

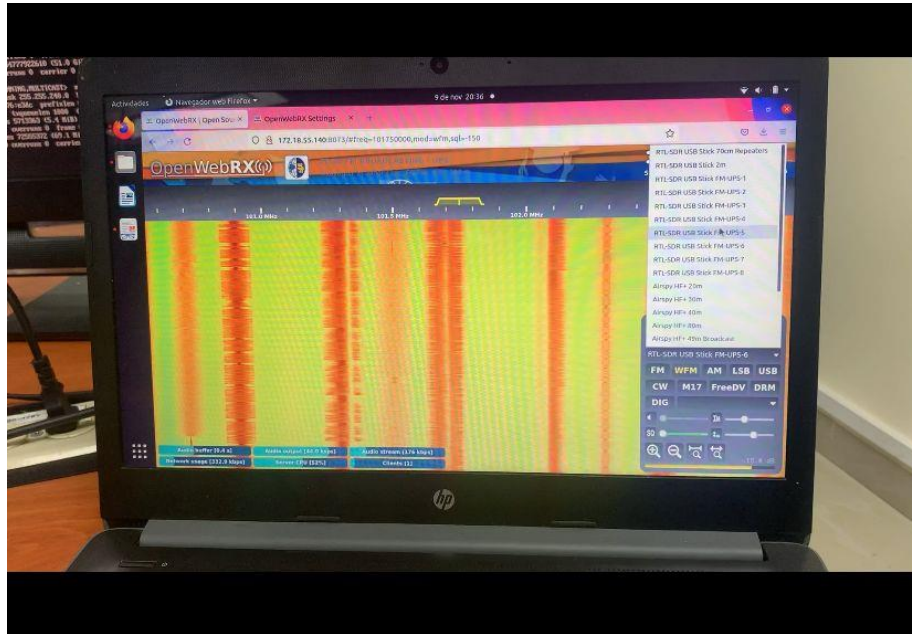


Figura 2. OpenWebRx visto vía web



Figura 3. Conexión de RTL-SDR y Raspberry PI

RESULTADO(S) OBTENIDO(S):

El estudiante debe colocar los resultados obtenidos en la práctica con imágenes y descripciones de cada imagen.

CONCLUSIONES:

El estudiante debe colocar las conclusiones de las prácticas de acuerdo con los objetivos planteados.

CARRERA: INGENIERÍA ELECTRÓNICA

ASIGNATURA:

NRO. PRÁCTICA:

4

TÍTULO PRÁCTICA: Configuración de analizador de espectros con RTL-SDR y Raspberry PI.

OBJETIVO GENERAL:

Configurar e implementar analizador de espectros con RTL-SDR y Raspberry PI.

OBJETIVOS ESPECÍFICOS:

- Configurar analizador de espectros con RTL-SDR
- Configurar Raspberry PI
- Configurar OpenWebRX.

INSTRUCCIONES

1. Los estudiantes deben leer previamente el manual de práctica para el desarrollo.

2. Los estudiantes deben utilizar los equipos del banco de pruebas de radiodifusión FM de una manera responsable y calificada para evitar daños en los equipos.

3. Los estudiantes deben trabajar en grupo para el desarrollo de la práctica.

4. Se debe dejar en orden el sitio de práctica luego del desarrollo de esta.

ACTIVIDADES POR DESARROLLAR:

Para esta práctica se debe realizar las configuraciones necesarias en el OpenWebRX para poder visualizar las frecuencias requeridas.

- 1) Configurar el banco de frecuencias con ancho de banda de 10 MHz.
- 2) Guardar las configuraciones de las frecuencias en OpenWebRX.
- 3) Realizar el análisis de las señales obtenidas con el analizador de espectros.

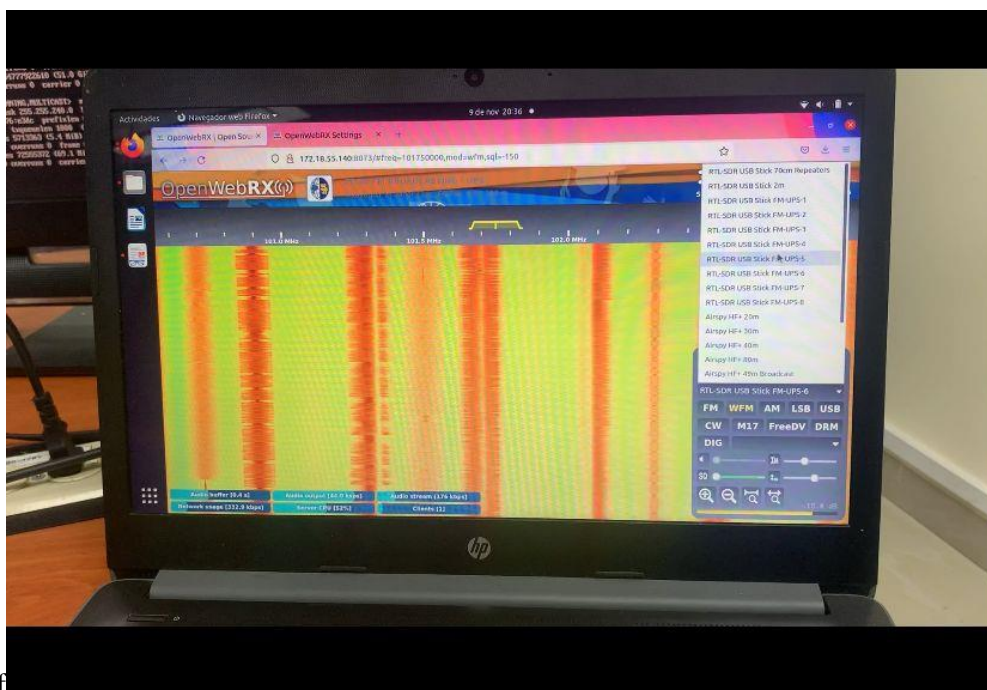


Figura 1. Banco de frecuencias en OpenWebRX.

RESULTADO(S) OBTENIDO(S):

El estudiante debe colocar los resultados obtenidos en la práctica con imágenes y

descripciones de cada imagen

CONCLUSIONES:

El estudiante debe colocar las conclusiones de las prácticas de acuerdo con los objetivos planteados.

CARRERA: INGENIERÍA ELECTRÓNICA

ASIGNATURA:

NRO. PRÁCTICA:

5

TÍTULO PRÁCTICA: Configuración de APP SDR touch y RTL-SDR para usarlo como analizador de espectros.

OBJETIVO GENERAL:

Configurar e implementar APP SDR touch y RTL-SDR para usarlo como analizador de espectros.

OBJETIVOS ESPECÍFICOS:

- Configurar APP SDR.
- Configurar RTL-SDR como analizador de espectros.
- Aprender el uso de APP para SDR.

INSTRUCCIONES

1. Los estudiantes deben leer previamente el manual de práctica para el desarrollo.
2. Los estudiantes deben utilizar los equipos del banco de pruebas de radiodifusión FM de una manera responsable y calificada para evitar daños en los equipos.
3. Los estudiantes deben trabajar en grupo para el desarrollo de la práctica.
4. Se debe dejar en orden el sitio de práctica luego del desarrollo de esta.

ACTIVIDADES POR DESARROLLAR:



Figura 1. Conexión del dispositivo APP SDR touch y RTL-SDR

Como podemos observar en la figura 1, el procedimiento es conectar el RTL-SDR con el dispositivo que se encuentra internamente conectado en el prototipo de la imagen.

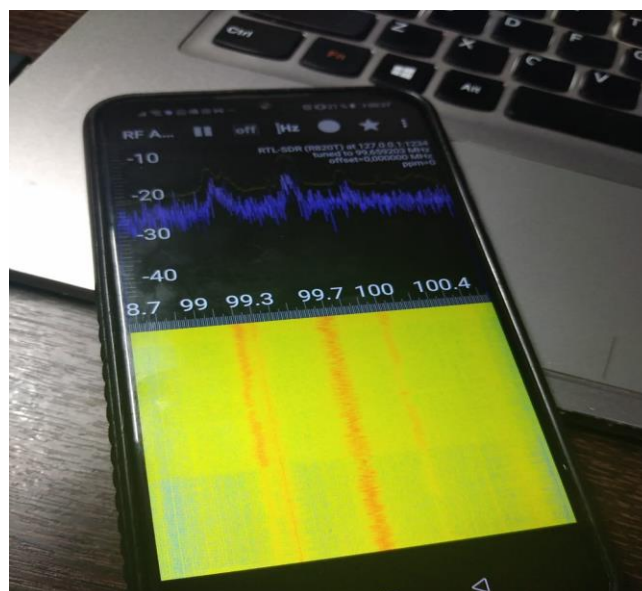


Figura 2. Permeámetros de frecuencias y potencia en la APP SDR touch

Iniciamos la aplicación y ajustamos a una frecuencia para poder examinar las señales que se van a manejar, esta aplicación trabaja dentro del rango de frecuencia que trabaja el RTL-SDR, como observamos en la figura 2.



Figura 3. Conexión del RTL-SDR con la aplicación en frecuencia

Se observa en la figura 3, las frecuencias que se van a utilizar con el dispositivo conectado con el RTL-SDR.



Figura 4. RTL-SDR conectado al dispositivo

La figura 4 muestra la conexión entre los dispositivos, esta conexión garantizara el acceso de una manera eficiente.



Figura 5. Cubierta plástica

Como vemos en la figura 5 se utiliza una cubierta plástica para colocar en el un conector USB.



Figura 6. Conector USB

Como vemos en la figura 6 el conector USB.



Figura 7. Antena de RTL-SDR

Como vemos en la figura 7 la antena de RTL-SDR es una antena que nos permitirá recibir las señales.

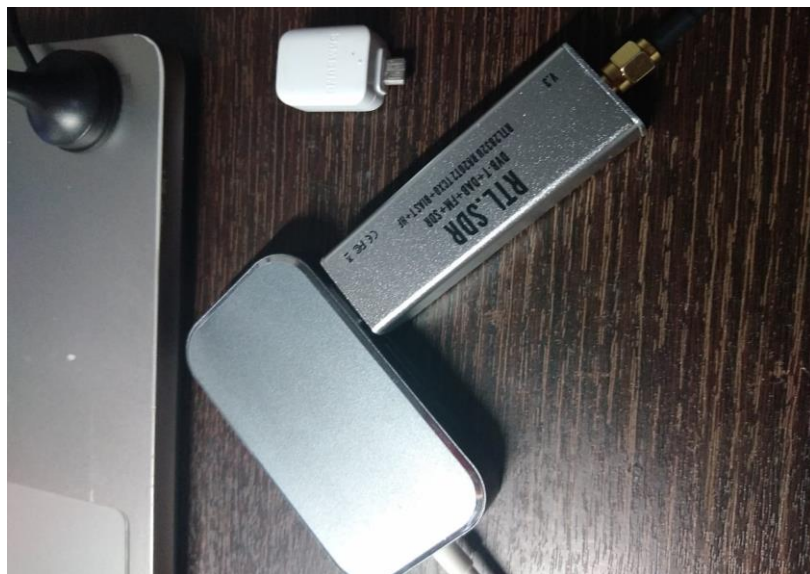


Figura 8. RTL-SDR conectado a un adaptador

Como vemos en la figura 8 el RTL-SDR está conectado a un adaptador que nos permitirá conectarlo a nuestro dispositivo móvil.

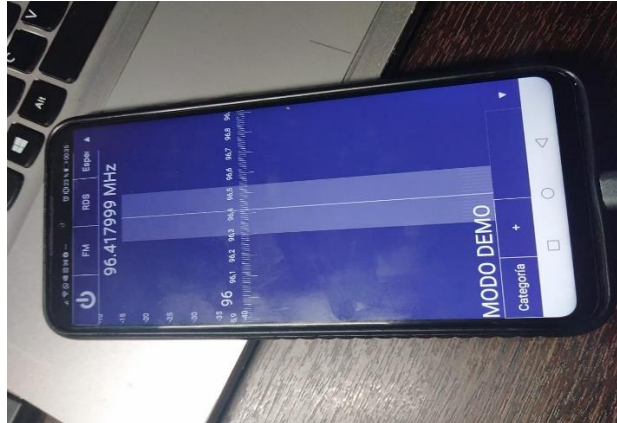


Figura 9. Pantalla de la APP SDR touch

En la figura 9 observamos la pantalla de la aplicación APP SDR touch donde podremos ingresar las frecuencias que vallamos a utilizar al momento de la configuración del dispositivo RTL-SDR.

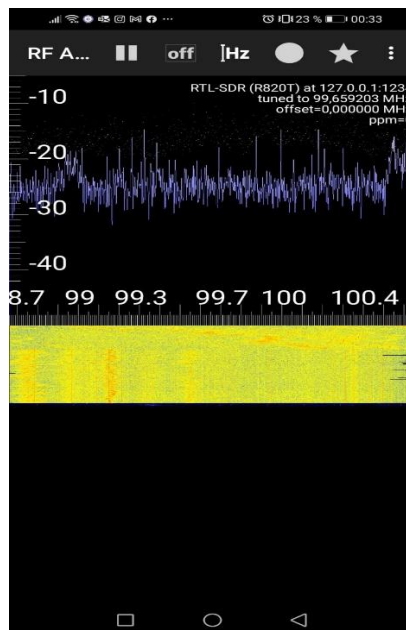


Figura 10. Grafica espectral, RTL-SDR

Observamos en la figura 10 la señal recibida por el RTL-SDR analizada por la APP SDR touch.



Figura 11. Instalar SDR

Como podemos observar en la figura 11, instalamos la aplicación SDR driver para poder conectarlo al programa.

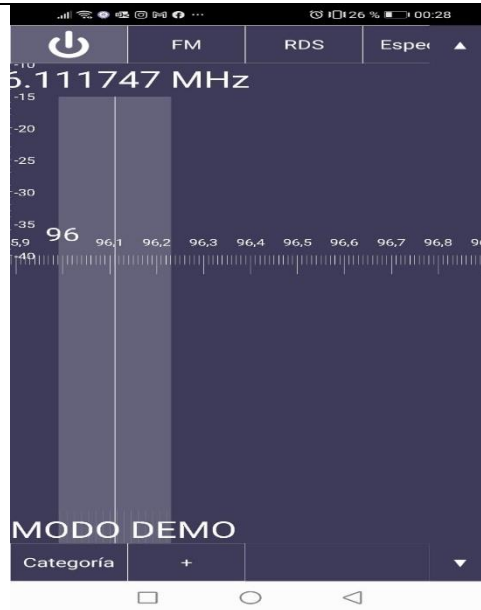


Figura 12. Iniciamos SDR driver

Como observamos en la figura 12 Iniciamos la aplicación SDR driver que nos permitirá conectarnos al dispositivo RTL-SDR.



Figura 13. Opciones del SDR driver

El SDR driver nos permite elegir el tipo de señal que nos sea más eficiente al utilizarlo, como vemos en la imagen 13.

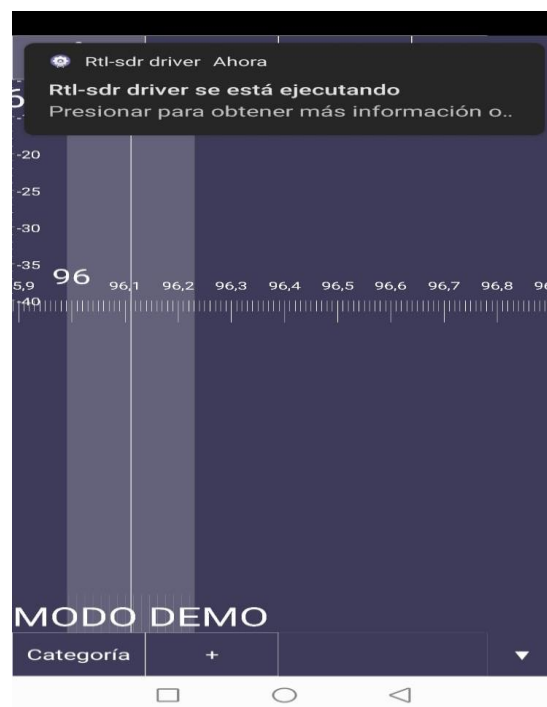


Figura 14. Ejecución del RTL-SDR

Visualizamos en la figura 14 la ejecución del RTL-SDR a través de la aplicación SDR driver.



Figura 15. Instalación del SDR touch

En la figura 15 instalamos la aplicación SDR touch para poder escanear y analizar señales de radios.

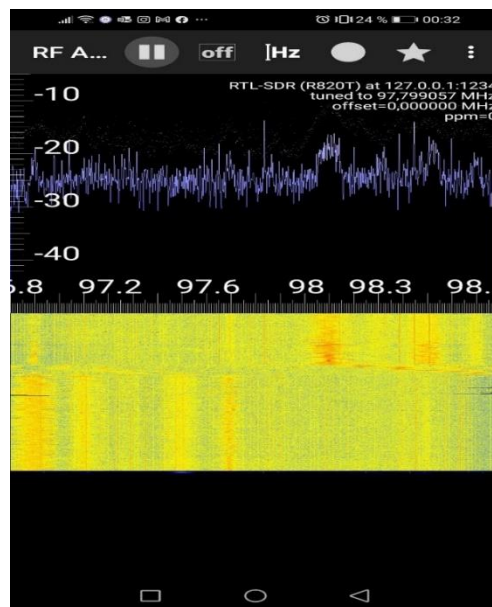


Figura 16. Conexión entre el RTL-SDR y el SDR touch

Como vemos en la figura 16 vemos la recepción de señales a través de los dispositivos conectados.

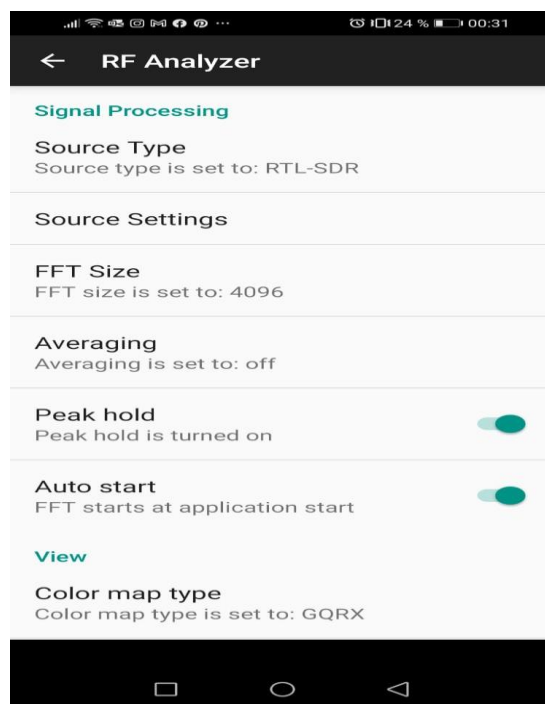


Figura 17. RF analizador

Como vemos en la figura 17 podemos seleccionar las opciones para el procesamiento de la señal.

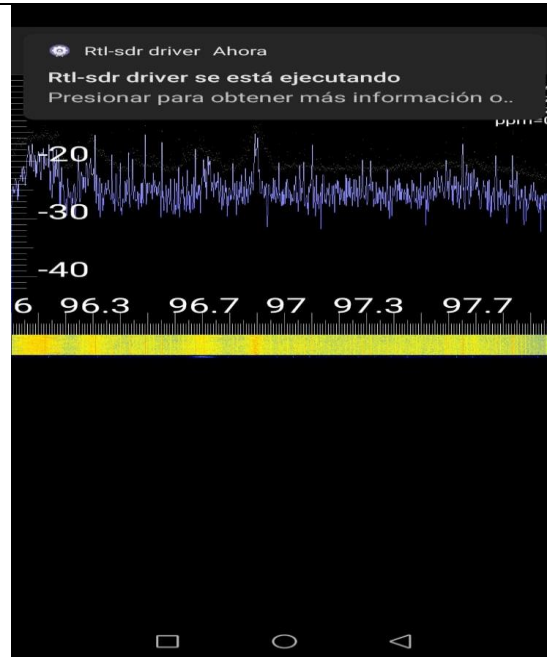


Figura 18. Grafica del espectral

En la figura 18 vemos la señal del espectro receptada por la aplicación SDR touch.

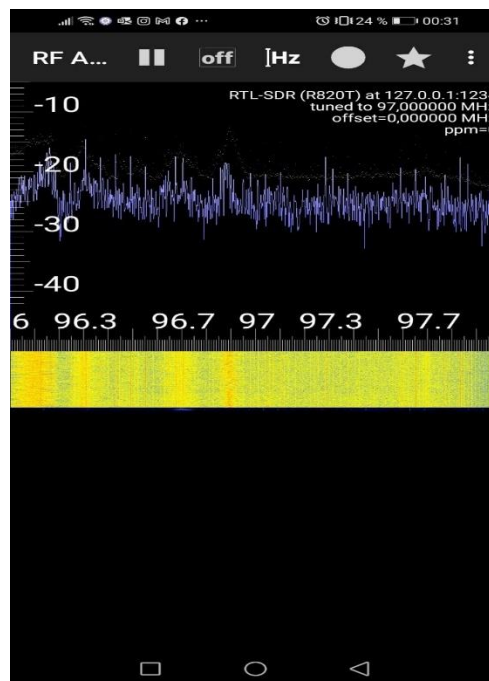


Figura 19. Señal receptada por la aplicación

En la figura 19 vemos la señal que capto el SDR touch en versión de espectro y la cascada de datos recibida.

RESULTADO(S) OBTENIDO(S):

El estudiante debe colocar los resultados obtenidos en la práctica con imágenes y descripciones de cada imagen

CONCLUSIONES:

El estudiante debe colocar las conclusiones de las prácticas de acuerdo con los objetivos planteados.

ANEXO 2: ESPECIFICACIONES DE EQUIPOS

DATASHEET DE EQUIPOS

Amplificador FM Minicircuits ZHL-50W-52-S+

Coaxial High Power Amplifier

50Ω 50W 50 to 500 MHz

ZHL-50W-52-S+

Features

- High power, 50 Watt
- Excellent IP3, +59 dBm typ.
- Class A amplifier, usable up to 75W.
- No damage with an open or short output load under full CW output power¹
- Shuts off when base plate temperature exceeds +100°C
- Internal power regulator (current remains constant over 22 to 28V)
- Over voltage protection, shut off above 29V
- Protected by US Patent 7,348,854



Model No.	ZHL-50W-52-S+
Case Style	BT1165
Connectors	IN-SMA, OUT-SMA

+RoHS Compliant

The +Suffix identifies RoHS Compliance. See our web site for RoHS Compliance methodologies and qualifications

Applications

- VHF/UHF transmitters
- Defense
- Amateur radio FM, TV
- Laboratory use

Electrical Specifications at 25°C

Parameter	Condition (MHz)	ZHL-50W-52-S+			Units
		Min.	Typ.	Max.	
Frequency Range		50		500	MHz
Gain		47	52	—	dB
Gain Flatness		—	±1.2	±1.8	dB
Output Power at 1dB compression	200 - 300 50 - 500	+46.0 +44.0	+47.5 +46.5	—	dBm
Saturated Output Power at 3dB compression	200 - 300 50 - 500	+47.0 +45.5	+49.0 +48.0	—	dBm
Noise Figure		—	4.5	7.0	dB
Output third order intercept point		—	+59	—	dBm
Input VSWR		—	1.75	—	:1
Output VSWR		—	2.5	—	:1
DC Supply Voltage		—	24	25	V
Supply Current		—	—	9.3	A

1. At constant open or short load 24V nominal supply voltage

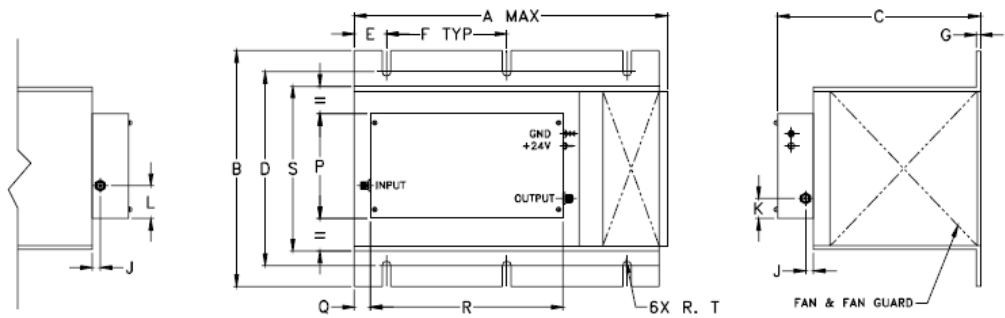
Maximum Ratings

Parameter	Ratings
Operating Temperature	-20°C to 85°C
Storage Temperature	-55°C to 100°C
Input RF Power ² (no damage)	+3dBm

2. At nominal output load, 24V nominal supply voltage.

Permanent damage may occur if any of these limits are exceeded.

Outline Drawing for models with heatsink



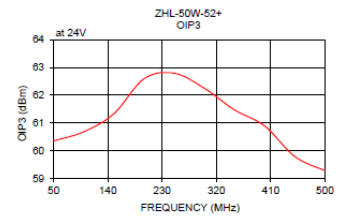
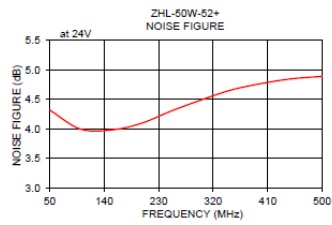
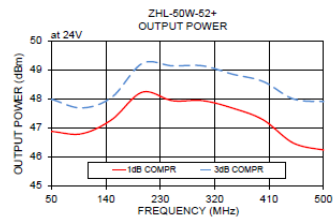
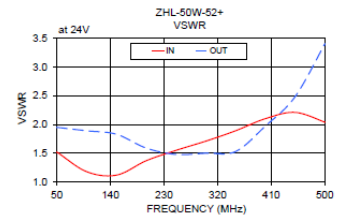
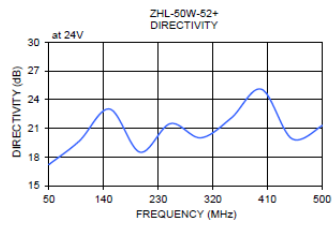
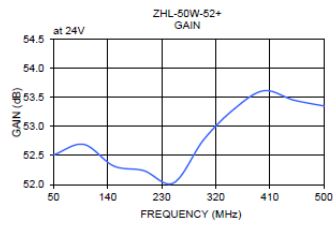
Outline Dimensions (inch/mm)

A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R	S	T	wt
9.85	7.3	6.3	6.00	1.00	3.75	.13	—	.25	.63	1.03	—	—	3.25	.5	6.00	5.1	.135	grams
250.19	185.42	160.02	152.40	25.40	95.25	3.30	—	6.35	16.00	26.16	—	—	82.55	12.70	152.40	129.54	3.43	4185

Typical Performance Data/Curves

ZHL-50W-52-S+

FREQUENCY (MHz)	GAIN (dB)	DIRECTIVITY (dB)	VSWR (:1)		POUT at 1 dB COMPR. (dBm)	POUT at 3 dB COMPR. (dBm)	NOISE FIGURE (dB)	OIP3 (dBm)
	24V	24V	IN	OUT	24V	24V	24V	24V
50	52.51	17.22	1.52	1.95	46.87	47.98	4.32	60.35
100	52.69	19.68	1.18	1.89	46.79	47.68	4.00	60.68
150	52.32	23.02	1.12	1.83	47.28	48.05	3.98	61.33
200	52.24	18.52	1.37	1.59	48.22	49.21	4.09	62.59
250	52.03	21.50	1.55	1.48	47.93	49.13	4.30	62.79
300	52.77	20.02	1.71	1.50	47.93	49.13	4.49	62.25
350	53.29	22.07	1.89	1.53	47.67	48.84	4.66	61.47
400	53.61	25.09	2.10	1.97	47.27	48.59	4.77	60.89
450	53.45	19.93	2.21	2.48	46.47	47.99	4.85	59.79
500	53.35	21.29	2.04	3.40	46.23	47.90	4.89	59.28



High Power Amplifier

ZHL-50W-52+

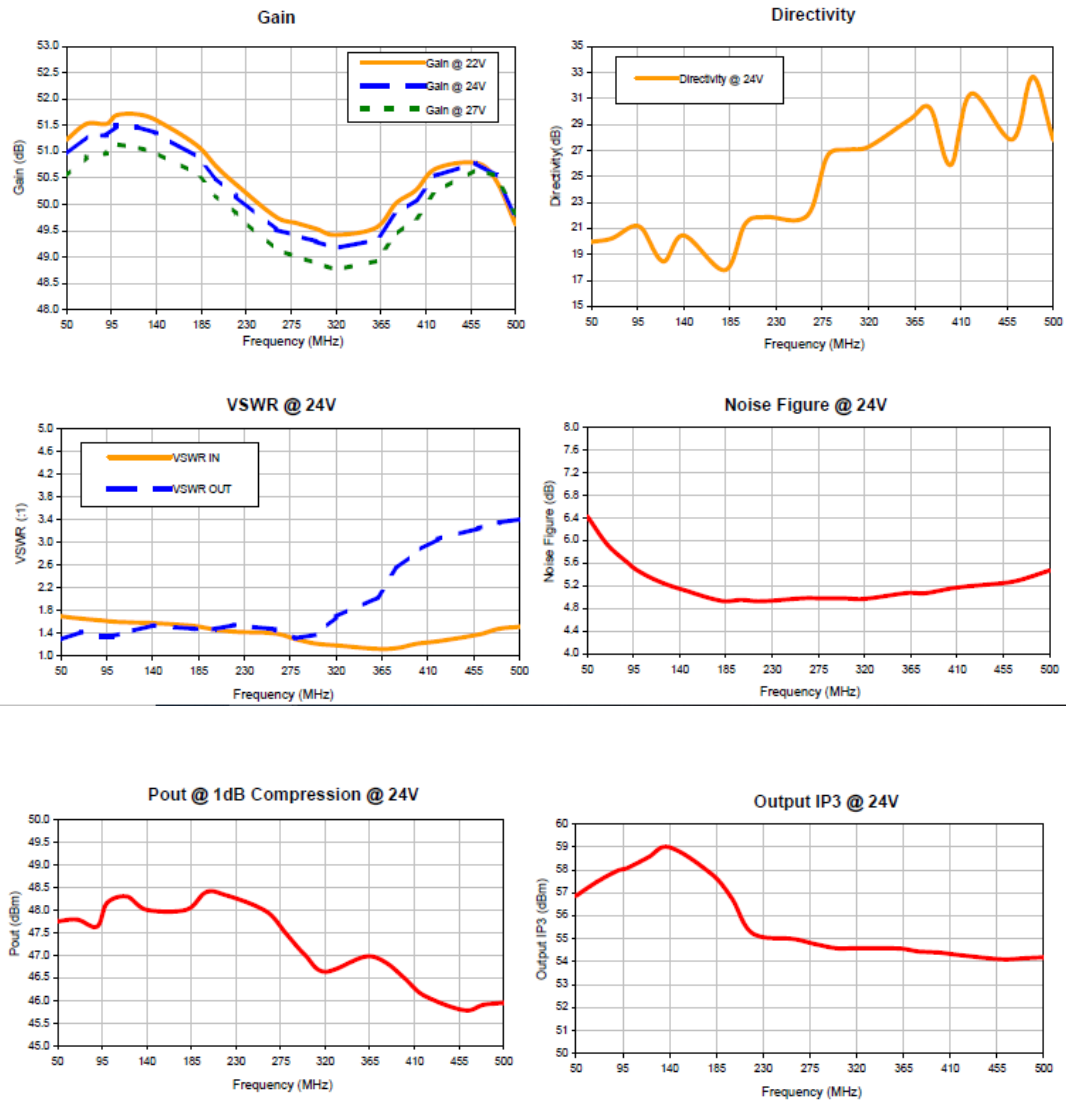
Typical Performance Data

Frequency (MHz)	Gain			Directivity (dB) 24V	VSWR In (:1) 24V	VSWR Out (:1) 24V	Noise Figure (dB) 24V	Pout at 1dB Compression (dBm) 24V	Output IP3 (dBm) 24V
	22V	24V	27V						
50.00	51.23	50.95	50.53	19.96	1.70	1.29	6.43	47.75	56.86
70.00	51.53	51.29	50.90	20.24	1.66	1.43	5.92	47.79	57.47
90.00	51.53	51.31	50.97	21.16	1.63	1.35	5.60	47.64	57.96
100.00	51.69	51.49	51.14	20.95	1.61	1.35	5.46	48.16	58.09
120.00	51.71	51.48	51.09	18.47	1.59	1.44	5.28	48.30	58.55
140.00	51.60	51.35	50.98	20.47	1.58	1.54	5.15	48.01	58.98
180.00	51.13	50.91	50.58	17.81	1.53	1.48	4.94	48.01	57.84
200.00	50.72	50.50	50.17	21.37	1.47	1.47	4.95	48.40	56.76
220.00	50.38	50.16	49.82	21.89	1.43	1.55	4.93	48.33	55.23
260.00	49.76	49.52	49.14	21.96	1.40	1.47	4.98	47.98	54.98
280.00	49.65	49.41	49.01	26.60	1.30	1.31	4.98	47.50	54.77
300.00	49.54	49.30	48.89	27.07	1.22	1.38	4.98	47.00	54.59
320.00	49.42	49.17	48.76	27.29	1.19	1.69	4.97	46.64	54.59
360.00	49.56	49.33	48.93	29.37	1.13	2.03	5.07	46.97	54.58
380.00	50.02	49.81	49.42	30.28	1.14	2.55	5.07	46.86	54.44
400.00	50.27	50.09	49.74	25.88	1.22	2.85	5.14	46.50	54.39
420.00	50.68	50.54	50.24	31.38	1.26	3.06	5.19	46.11	54.28
460.00	50.79	50.79	50.66	27.85	1.38	3.25	5.26	45.79	54.10
480.00	50.47	50.55	50.54	32.67	1.48	3.35	5.35	45.91	54.15
500.00	49.62	49.77	49.84	27.81	1.52	3.41	5.47	45.95	54.19

High Power Amplifier

ZHL-50W-52+

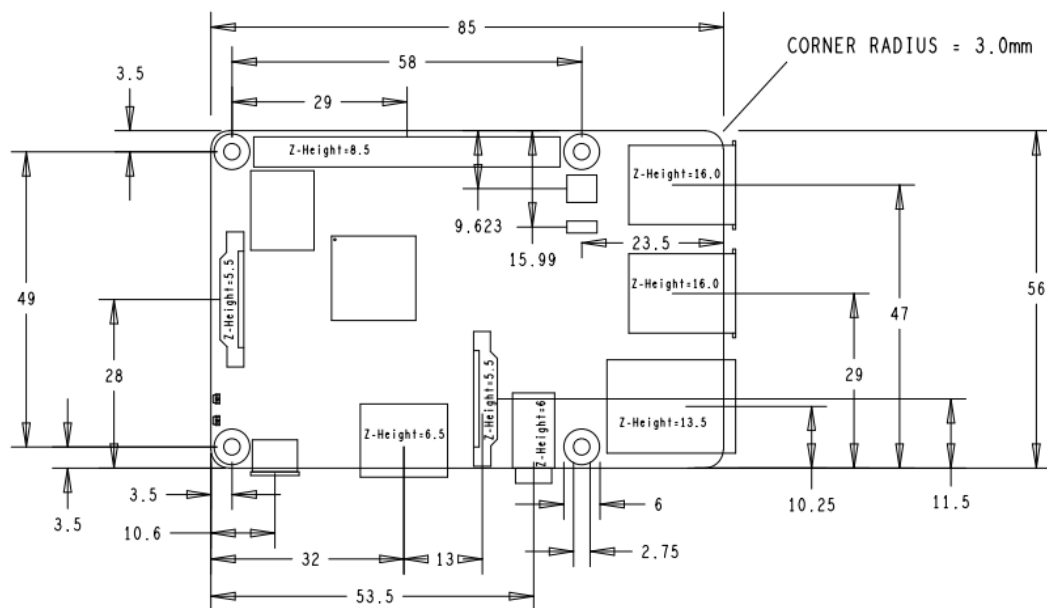
Typical Performance Curves



Raspberry Pi 3 Model B

Processor:	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory:	1GB LPDDR2 SDRAM
Connectivity:	<ul style="list-style-type: none">■ 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE■ Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps)■ 4 × USB 2.0 ports
Access:	Extended 40-pin GPIO header
Video & sound:	<ul style="list-style-type: none">■ 1 × full size HDMI■ MIPI DSI display port■ MIPI CSI camera port■ 4 pole stereo output and composite video port
Multimedia:	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card support:	Micro SD format for loading operating system and data storage
Input power:	<ul style="list-style-type: none">■ 5V/2.5A DC via micro USB connector■ 5V DC via GPIO header■ Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment:	Operating temperature, 0–50 °C
Compliance:	For a full list of local and regional product approvals, please visit www.raspberrypi.org/products/raspberry-pi-3-model-b+
Production lifetime:	The Raspberry Pi 3 Model B+ will remain in production until at least January 2023.

Physical specifications



RTL-SDR



The RTL-SDR Blog V3 is an improved RTL-SDR dongle. RTL-SDR dongles were originally designed for DVB-T HDTV reception, but they were found by hardware hackers to be useful as a general purpose SDR. The standard dongles are okay for DVB-T reception, but are just barely suitable for SDR users/experimenters. The RTL-SDR Blog V3 was redesigned with SDR user needs in mind, instead of DVB-T HDTV users who typically have more relaxed requirements.

Purchase at: www.rtl-sdr.com/store

Basic Information

- **Bandwidth:** Up to 2.4 MHz stable.
- **ADC:** RTL2832U 8-bits
- **Frequency Range:** 500 kHz – 1766 MHz (500 kHz – 24 MHz in direct sampling mode)
- **Typical Input Impedance:** 50 Ohms
- **Typical Current Draw:** 270 – 280 mA

Required Computing Hardware

Same requirements as a regular RTL-SDR. Compatible with Windows XP and above (SDR# requires Win 7 or newer), Linux, MacOS and Android. A dual core machine is recommended.

Single board PCs like the Raspberry Pi, Odroid, C.H.I.P are also supported with most command line apps.

USRP B205 mini-i

USRP™ B200mini Series

Product Overview

The USRP B200mini Series delivers a 1x1 software defined radio/cognitive radio in the size of a business card. With a wide frequency range from 70 MHz to 6 GHz and a user-programmable Xilinx Spartan-6 FPGA, this flexible and compact platform is ideal for both hobbyist and OEM applications. The RF front end uses the Analog Devices AD9364 RFIC transceiver with 56 MHz of instantaneous bandwidth. The board is bus-powered by a high-speed USB 3.0 connection for streaming data to the host computer. The USRP B200mini Series also includes connectors for GPIO, JTAG, and synchronization with a 10 MHz clock reference or PPS time reference input signal. There are three configurations in this product family with options for a larger or industrial-grade FPGA. The USRP Hardware Driver™ (UHD) software API supports all USRP products and enables users to efficiently develop applications then seamlessly transition designs between platforms as requirements expand.

Applications

Hobbyists and New Users

The powerful UHD software API reduces the learning curve and provides a quick start experience for new users and long-time hobbyists interested in AM/FM applications, cellular communication, and algorithm exploration.

Wireless Signal Discovery and Analysis

The content-rich GNU Radio community provides a wide range of tools and algorithms that enable discovery and analysis of air interface protocols.

OEM and Integration

The compact form factor and cost-effective design of the B200mini Series make it ideal for integration into larger systems for prototyping and deployment. The 10 MHz ref/PPS and GPIO features provide seamless synchronization and



Features

Device Variants

- [B200mini \(LX75 C-Grade FPGA\)](#)
- [B200mini-i \(LX75 I-Grade FPGA\)](#)
- [B205mini-i \(LX150 I-Grade FPGA\)](#)

RF Capabilities

- 1 TX, 1 RX
- 70 MHz to 6 GHz frequency range
- Up to 56 MHz bandwidth

Software

- UHD version 3.9.2 or later¹
- GNU Radio
- C/C++
- Python

High-Speed Interface and Power

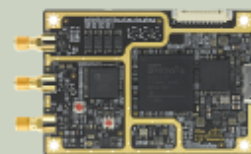
- USB 3.0
- USB powered

Synchronization

- 10 MHz clock reference or PPS time reference

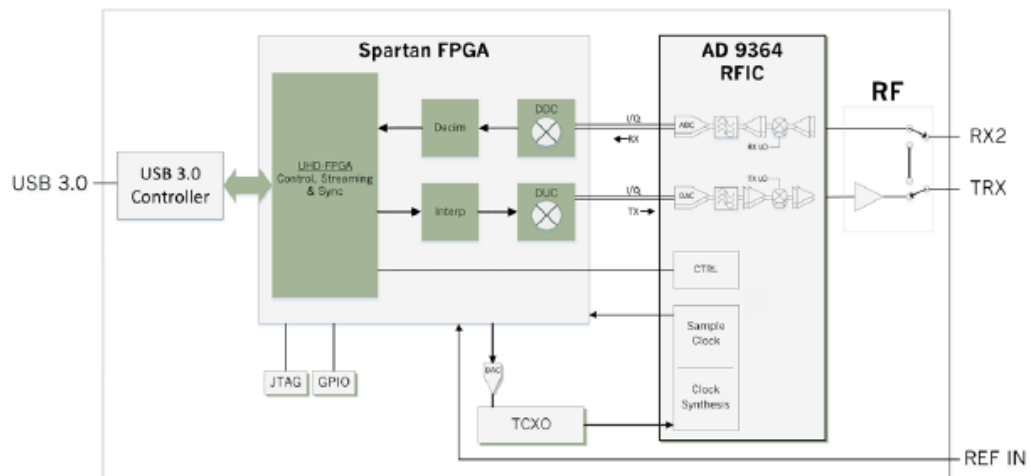
Peripherals

- GPIO
- JTAG



Specification	Typical	Unit
RF Performance³		
IIP3 (at typical NF)	-20	dBm
Power Output	> 10	dBm
Receive Noise Figure	< 8	dB
Conversion Performance and Clocks³		
ADC Sample Rate (Max.)	61.44	MS/s
ADC Resolution	12	bits
DAC Sample Rate (Max.)	61.44	MS/s
DAC Resolution	12	bits
Host Sample Rate (16b)	61.44	MS/s
Frequency Accuracy	±2.0	ppm

Specification	Typical	Unit
Power		
USB Power	5	V
Physical		
Dimensions	83.3 x 50.8 x 8.4	mm
Weight	24.0	g
Operating Temperature Range		
B200mini	0 – 40	°C
B200mini-i	0 – 45	°C
B205mini-i	0 – 45	°C



Antena Tx

Antena de transmisor FM, 1/4 Wave Profesional GP FM Broadcast Antena para 87-108MHZ
Transmisor FM Cable 15M TNC para 0.5W, 1W, 5W, 7W, 10W, 15W, 20W, 30W, 50w, 80w,
100W FM Transmisor





- Diseñado para transmisor FM: la mejor antena de transmisión GP FM de 1/4 de onda que puede ajustar fácilmente la frecuencia de precisión para fijar su transmisor de transmisión FM.
- Fácil de montar: desembalar y configurar sin esfuerzo en 5 minutos.
- Resistente y duradera: hecha de aleación de aluminio resistente al óxido y al agua, esta antena es tu mejor opción para tu transmisor FM.
- Potencia: 1 W-100 W (5 W, 7 W, 15 W, 25 W, 30 W, 50 W, 100 W). Frecuencia: 84,5 – 89,5 MHz (A+B1) 89,5 – 94,5 MHz (A+B2) 94,5 – 99,5 MHz (A+B3) 9,5 – – – – 104,5 Mhz (A+). B4) 104.5-109.5 (A+B5) --{Frecuencia central: 87MHz / 92MHz / 97MHz / 102MHz / 107MHz}
- Utilizado para campus, comunidad, minería, rural e iglesia, cine y otros lugares, es una antena al aire libre ideal para la transmisión FM.