

UNIVERSIDAD TÉCNICA DE AMBATO



FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

MAESTRÍA EN TELECOMUNICACIONES

Tema: SIMULACIÓN DE APLICACIONES CON SERIES TRIGONOMÉTRICAS DE FOURIER EN SISTEMAS DE COMUNICACIONES ELECTRÓNICAS

Trabajo de Titulación, previo a la obtención del Grado Académico de
Magister en Telecomunicaciones

Modalidad de Titulación “Proyecto de Desarrollo”

Autor: Ingeniero Alex Francisco Rodríguez Ortiz

Director: Ingeniero Carlos Diego Gordón Gallegos, PhD.

Ambato – Ecuador

2021

APROBACIÓN DEL TRABAJO DE TITULACIÓN

A la Unidad Académica de Titulación de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

El Tribunal receptor de la Defensa del Trabajo de Titulación, presidido por la Ingeniera Elsa Pilar Urrutia Urrutia Magíster, e integrado por los señores Ingeniero Fabián Rodrigo Salazar Escobar, Doctor e Ingeniero Geovanni Danilo Brito Moncayo Magíster, designados por la Unidad Académica de Titulación de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, para receptor el Trabajo de Titulación con el tema: “SIMULACIÓN DE APLICACIONES CON SERIES TRIGONOMÉTRICAS DE FOURIER EN SISTEMAS DE COMUNICACIONES ELECTRÓNICAS”, elaborado y presentado por el señor Ingeniero Alex Francisco Rodríguez Ortiz, para optar por el Grado Académico de Magister en Telecomunicaciones; una vez escuchada la defensa oral del Trabajo de Titulación el Tribunal aprueba y remite el trabajo para uso y custodia en las bibliotecas de la Universidad Técnica de Ambato.

Ing. Elsa Pilar Urrutia Urrutia Mg.
Presidente y Miembro del Tribunal de Defensa

Ing. Fabián Rodrigo Salazar Escobar, Dr.
Miembro del Tribunal de Defensa

Ing. Geovanni Danilo Brito Moncayo, Mg
Miembro del Tribunal de Defensa

AUTORÍA DEL TRABAJO DE INVESTIGACIÓN

La responsabilidad de las opiniones, comentarios y críticas emitidas en el Trabajo Titulación presentado con el tema: “SIMULACIÓN DE APLICACIONES CON SERIES TRIGONOMÉTRICAS DE FOURIER EN SISTEMAS DE COMUNICACIONES ELECTRÓNICAS”, le corresponde exclusivamente a: Ingeniero Alex Francisco Rodríguez Ortiz, Autor bajo la Dirección del Ingeniero Carlos Diego Gordón Gallegos PhD, director del Trabajo de Titulación; y el patrimonio intelectual a la Universidad Técnica de Ambato.

Ing. Alex Francisco Rodríguez Ortiz

AUTOR

Ing. Carlos Diego Gordón Gallegos, PhD.

DIRECTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que el Trabajo de Titulación, sirva como un documento disponible para su lectura, consulta y procesos de investigación, según las normas de la Institución.

Cedo los Derechos de mi Trabajo de Titulación, con fines de difusión pública, además apruebo la reproducción de este, dentro de las regulaciones de la Universidad Técnica de Ambato.

Ing. Alex Francisco Rodríguez Ortiz
c.c..1804790507

INDICE GENERAL

Contenido

PORTADA.....	i
APROBACIÓN DEL TRABAJO DE TITULACIÓN.....	ii
AUTORÍA DEL TRABAJO DE INVESTIGACIÓN.....	iii
DERECHOS DE AUTOR.....	iv
INDICE GENERAL.....	v
INDICE DE TABLAS.....	ix
INDICE DE FIGURAS.....	x
AGRADECIMIENTO.....	xiii
DEDICATORIA.....	xiv
RESUMEN EJECUTIVO.....	xv
EXECUTIVE SUMMARY.....	xvii
1 CAPÍTULO I.....	1
1.1 Introducción.....	1
1.2 Justificación.....	2
1.3 Objetivos.....	3
1.3.1 General.....	3
1.3.2 Específicos.....	3
2 CAPÍTULO II.....	4
ANTECEDENTES INVESTIGATIVOS.....	4
2.1 Estado del Arte.....	4
2.2 Marco Teórico.....	7
ANÁLISIS DE LAS TELECOMUNICACIONES.....	7
2.3 Estructura de un sistema básico de Telecomunicaciones.....	7
2.3.1 Fuente de Información.....	7
2.3.2 Transductor.....	8
2.3.3 Transmisor.....	8
2.3.4 Canal.....	8
2.3.5 Receptor.....	9
2.3.6 Ancho de banda y potencia de la señal.....	9
2.4 Modulación.....	10
2.4.1 ¿Por qué se modula una señal?.....	11
2.5 Modos de Transmisión.....	11
2.5.1 Simplex.....	11
2.5.2 Half Dúplex o Semi Dúplex.....	12
2.5.3 Full Dúplex.....	12
2.6 Demodulación.....	13
2.7 Problemas en la recepción de señales.....	14

ANALISIS DE SEÑALES	14
2.8 Procesamiento de señales.....	14
2.9 Serie de Fourier.....	15
2.9.1 Serie Trigonométrica.....	16
2.9.2 Series de Fourier de cosenos y senos	21
2.9.3 Fidelidad.....	22
2.10 Analizador de Espectro	22
2.10.1 Definición.....	22
2.11 Clasificación del Analizador de Espectro.....	22
2.11.1 Analizador de Espectro Analógico.....	22
2.11.2 Analizador de Espectro Digital.....	23
2.12 Funcionamiento	23
2.12.1 Frecuencia	23
2.12.2 Amplitud	24
2.12.3 Ventana de visualización horizontal SPAN	24
2.13 Aplicaciones	24
ANALISIS DE SOFTWARE LIBRE	25
2.14 Software Libre	25
2.14.1 Introducción	25
2.14.2 Características	25
2.14.3 Ventajas y Desventajas.....	25
2.14.4 Comparación de software libre con software propietario.....	27
2.15 Arquitectura de software.....	28
2.16 Lenguaje de programación Python.....	28
2.17 Computación simbólica.	29
2.18 Software utilizado para el desarrollo.....	30
2.18.1 Sympy.	30
2.18.2 Numpy.....	31
2.18.3 Python.	31
2.18.4 Spyder.	32
2.18.5 Utilidades	34
Utilidades	34
2.19 Radio Definido por Software SDR.....	34
2.19.1 GNU Radio.....	35
2.19.2 Requisitos de GNU Radio.....	35
2.19.3 Programación con GNU Radio	37
2.19.4 Uso de variables en GNU Radio	40
2.19.5 Visualización de Resultados	41
2.20 Conexión de GNU Radio y el LTR-SDR.....	41
2.21 Sistema Operativo Raspberry Pi.....	42
ANALISIS DE HARDWARE.....	43
2.22 Receptor - Decodificador.....	43
2.22.1 Definición.....	43
2.22.2 Características del receptor - decodificador.....	43

2.22.3	Historia de RDS	44
2.22.4	Radio definido por software (RDS)	45
2.22.5	Arquitectura del RDS	45
2.22.6	Componentes del receptor – decodificador.....	47
2.22.7	RTL-SDR	47
2.22.8	Comparativa de dispositivos SDR	47
2.22.9	Demodulador RTL2832U	48
2.22.10	Aplicaciones.....	48
2.22.11	Tuner R820T.....	49
2.22.12	Características	49
2.22.13	Aplicaciones.....	49
2.23	Raspberry Pi	50
2.23.1	Definición.....	50
2.23.2	Generalidades técnicas de un Raspberry Pi.....	50
2.23.3	Modelos	50
2.23.4	Aplicaciones.....	51
2.24	Introducción ADF4351	54
2.25	Sintetizador de Frecuencias	54
2.26	Características módulo generador de señales ADF4351	56
2.27	Comparativa tipos de generador de frecuencia:.....	57
3	CAPITULO III.....	59
	MARCO METODOLÓGICO.....	59
3.1	Ubicación.....	59
3.2	Equipos y materiales	59
3.3	Tipo de investigación.....	59
3.4	Hipótesis - pregunta científica – idea a defender	59
3.5	Población o muestra:.....	59
3.6	Recolección de información:.....	59
3.7	Variables respuesta o resultados esperados	60
4	CAPITULO IV	61
4.1	Sistema de Control.....	64
4.2	Presentación del Prototipo	65
4.2.1	Conexión Física del Prototipo.....	65
4.2.2	Implementación de prototipo en GNU RADIO	66
4.3	Hardware y software para simulación	68
4.3.1	Diseño y Construcción del Hardware	69
4.3.2	Acoplamiento al chasis	69
4.3.3	Pantalla.....	72
4.4	Desarrollo de aplicación Serie Trigonométrica de Fourier.....	84
4.4.1	Interfaz Gráfica.	84
4.4.2	Etapas del desarrollo	86
4.4.3	Entrada de datos.	86
4.4.4	Graficar la función de entrada.....	87

4.4.5	Obtener el período de la función de entrada.	87
4.4.6	Determinar la simetría de la función de entrada.	87
4.5	Cálculo de los coeficientes.	88
4.5.1	Cálculo del coeficiente a_0	88
4.5.2	Cálculo del coeficiente a_n	89
4.5.3	Cálculo del coeficiente b_n	89
4.6	Simulación con diferentes números de armónicas.	91
4.7	Generación de señales con distintas potencias	94
4.8	Aplicación práctica para detección de frecuencias específicas.	103
4.9	Aplicación prácticas para almacenamiento de energía	108
5	CAPÍTULO V	116
	CONCLUSIONES, RECOMENDACIONES	116
5.1	Conclusiones.....	116
5.2	Recomendaciones	117
	BIBLIOGRAFÍA	118
	ANEXOS	121

INDICE DE TABLAS

Tabla 2.1 Libertades del Software Libre	25
Tabla 2.2 Ventajas del Software Libre.	26
Tabla 2.3 Desventajas del Software Libre.	26
Tabla 2.4 Software libre vs. Software propietario.....	27
Tabla 2.5 Utilidades RTL-SDR.....	34
Tabla 2.6 Descripción de botones de la barra de herramientas.....	40
Tabla 2.7 Precios de dispositivos RDS.	47
Tabla 2.8 Características del chip R820T.	49
Tabla 2.9 Especificaciones Técnicas de los modelos de Raspberry PI.	51
Tabla 2.10 Especificaciones técnicas generador de señales ADF4351.....	57
Tabla 4.1 Características técnicas del prototipo.....	61
Tabla 4.2 Descripción de la arquitectura del prototipo.	63
Tabla 4.3 Resumen de mediciones.....	103
Tabla 4.4 Resumen de mediciones.....	108
Tabla 4.5 Resumen de mediciones Antena HRPSA	115

INDICE DE FIGURAS

Figura 2.1 Diagrama en bloques de un sistema básico de telecomunicaciones [14] ...	7
Figura 2.2 Representación del espectro en frecuencia de señal Periódica [16]	10
Figura 2.3 Representación del espectro en frecuencia de señal no Periódicas [16]...	10
Figura 2.4 Modelo de Tx SIMPLEX [18].....	12
Figura 2.5 Modelo de Tx HALF DUPLEX O SEMIDUPLEX [18]	12
Figura 2.6 Gráfico del Modelo de Tx FULL DUPLEX [18]	13
Figura 2.7 Ejemplo de demodulación y traslación de espectros [19]	13
Figura 2.8 Onda sinusoidal rectificada [20].....	15
Figura 2.9 Representación gráfica de una aplicación GNU Radio	37
Figura 2.10 Tipos de datos mostrados en el bloque Throttle	38
Figura 2.11 Tipos de datos completos en GNU Radio.....	39
Figura 2.12 Convertidores de tipos de datos disponibles.....	39
Figura 2.13 Bloque Variable	40
Figura 2.14 Cuadro de propiedades de un bloque dado.	41
Figura 2.15 Bloque recepción para conexión del LTR-SDR	42
Figura 2.16 Diagrama de bloques RDS[31]	46
Figura 2.17 Módulo RTL-SDR. A [32]	47
Figura 2.18 Módulo RTL-SDR. B [32].....	47
Figura 2.19 Generador de señales ADF4351. [34]	56
Figura 4.1 Arquitectura del analizador de espectro.	62
Figura 4.2 Arquitectura de un Raspberry Pi. [35].....	63
Figura 4.3 Espectro en forma de cascada.....	64
Figura 4.4 Componentes del analizador de espectro.....	66
Figura 4.5 Bloque Throttle - programación prototipo.....	66
Figura 4.6 Opciones enable, disable y bypass en un diagrama de GNU Radio.	67
Figura 4.7 Conexión Raspberry y RTL-SDR.....	68
Figura 4.8 Simulación GNU Radio.....	68
Figura 4.9 Parte interna del prototipo.	70
Figura 4.10 Vista frontal del prototipo.....	70
Figura 4.11 Vista posterior del prototipo.	71
Figura 4.12 Diferentes vistas del prototipo.	71
Figura 4.13 Parte frontal de la pantalla.	72

Figura 4.14 Diagrama de bloques prototipo.....	73
Figura 4.15 Actualización de paquetes.	74
Figura 4.16 Instalación de gr-osmosdr.....	74
Figura 4.17 Numero de paquetes a instalación gr-osmosdr.	75
Figura 4.18 Verificación de la instalación del software RTL-SDR.	75
Figura 4.19 Bloque Throttle.....	77
Figura 4.20 Bloque QT GUI range.	77
Figura 4.21 Variables de control deslizante.	78
Figura 4.22 Unión entre la entra y salida de los bloques.	78
Figura 4.23 Variable Potencia.....	79
Figura 4.24 Variable Frecuencia	79
Figura 4.25 Variable Ancho_Banda.....	80
Figura 4.26 Ejecución diagrama de flujo.	81
Figura 4.27 Interfaz gráfica variables de control deslizante.	81
Figura 4.28 Auto Scale.....	82
Figura 4.29 permeámetros de frecuencia y potencia.....	83
Figura 4.30 Diagrama de Cascada (Waterfall).....	83
Figura 4.31 Pantalla de la app para una función.	84
Figura 4.32 Etapas en la resolución del problema:	86
Figura 4.33 Programación para el cálculo del coeficiente a_0	88
Figura 4.34 Programación para el cálculo del coeficiente a_n	89
Figura 4.35 Programación para el cálculo del coeficiente b_n	91
Figura 4.36 Pantalla inicial ejecución del programa “Series de Fourier”.	91
Figura 4.37 Función diente de sierra con 10 armónicos.	92
Figura 4.38 Gráfica de la función diente de sierra con 10 armónico.	92
Figura 4.39 Función diente de sierra con 30 armónicos.	92
Figura 4.40 Gráfica de la función diente de sierra con 30 armónicos.....	93
Figura 4.41 Función impulso con 10 armónicos.....	93
Figura 4.42 Gráfica impulso con 10 armónico.....	93
Figura 4.43 Función impulso con 30 armónicos.....	93
Figura 4.44 Gráfica impulso con 30 armónicos.	94
Figura 4.45 Conexión ADF configuración de frecuencia a 88 MHz.	95
Figura 4.46 Configuración de parámetros-Frecuencia central a 88 MHz.....	96

Figura 4.47 Espectrograma con frecuencia resonante a 88 MHz.....	96
Figura 4.48 Acercamiento (Zoom) frecuencia resonante a 88 MHz.....	97
Figura 4.49 Configuración de prototipo – Frecuencia generada a 500MHz.....	97
Figura 4.50 Configuración de parámetros-Frecuencia central a 500 MHz.....	98
Figura 4.51 Espectrograma con frecuencia resonante a 500 MHz.....	98
Figura 4.52 Conexión y configuración de prototipo – Frecuencia generada a 1000MHz.....	99
Figura 4.53 Configuración de parámetros-Frecuencia central a 1000 MHz.....	99
Figura 4.54 Espectrograma con frecuencia resonante a 1000 MHz.....	100
Figura 4.55 Configuración de prototipo – Frecuencia generada a 1750MHz.....	101
Figura 4.56 Configuración de parámetros-Frecuencia central a 1750 MHz.....	102
Figura 4.57 Espectrograma con frecuencia resonante a 1750 MHz.....	102
Figura 4.58 Rango de frecuencias específicas de 88 MHz a 90 MHz.	104
Figura 4.59 Prototipo más antena omnidireccional -captación de radiofrecuencia.	104
Figura 4.60 Configuración de parámetros radiofrecuencia.....	105
Figura 4.61 Frecuencia resonante en cascada Waterfall.	105
Figura 4.62 Frecuencia resonante detectada a 88.1 MHz.	106
Figura 4.63 Frecuencia resonante detectada a 88.5 MHz.	106
Figura 4.64 Frecuencia resonante detectada a 88.9 MHz.	107
Figura 4.65 Frecuencia resonante detectada a 88.3 MHz.	107
Figura 4.66 Frecuencia resonante detectada a 89.7 MHz	108
Figura 4.67 Conexión de prototipo a antena HRPSA.	109
Figura 4.68 Configuración de parámetros antena HRPSA	110
Figura 4.69 Configuración generador de frecuencia ADF.....	110
Figura 4.70 Resonancia de antena HRPSA a 914 MHz.....	111
Figura 4.71 Gráfica en cascada Waterfall en resonancia 914.5 MHz.....	111
Figura 4.72 Resonancia de antena HRPSA a 914.5 MHz.....	112
Figura 4.73 Gráfica en cascada Waterfall en resonancia 915 MHz.....	112
Figura 4.74 Resonancia de antena HRPSA a 915 MHz.....	113
Figura 4.75 Gráfica en cascada Waterfall en resonancia 913.5 MHz.....	113
Figura 4.76 Resonancia de antena HRPSA a 913.5 MHz.....	114
Figura 4.77 Gráfica en cascada Waterfall en resonancia 913 MHz.....	114
Figura 4.78 Resonancia de antena HRPSA a 913 MHz.....	115

AGRADECIMIENTO

A la Universidad Técnica de Ambato, por acogerme en sus aulas y laboratorios para seguirme preparando académicamente. A la Dirección de Sistemas Informáticos y Redes (DISIR) de la Universidad Técnica de Ambato por permitirme desarrollar el presente trabajo de investigación la cual está vinculada al proyecto de Investigación: Sistema de Captación de Energía Electromagnética para Abastecimiento de Energía en Terminales de Internet de las Cosas (IoT) en entornos de Quinta Generación (5G) coordinado y dirigido principalmente por PhD. Carlos Diego Gordón Gallegos.

A la Ing. Mg. Pilar Urrutia, por toda la ayuda brindada para el presente trabajo investigativo. Al personal docente y administrativo de la Maestría en Telecomunicaciones de la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato.

DEDICATORIA

El presente trabajo investigativo dedico a las personas que más amo y representan en mi vida lo más valioso que tengo.

A mi padre Julio Rodríguez, quien con su ejemplo y constancia supo forjarme y hacer de mí un hombre de bien.

A mi madre Esthela Ortiz, por su apoyo incondicional que me ha permitido ser un excelente profesional.

UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL
MAESTRÍA EN TELECOMUNICACIONES

TEMA:

“SIMULACIÓN DE APLICACIONES CON SERIES TRIGONOMÉTRICAS DE FOURIER EN SISTEMAS DE COMUNICACIONES ELECTRÓNICAS”

AUTOR: Ing. Alex Francisco Rodríguez Ortiz

DIRECTOR: Ing. Carlos Diego Gordón Gallegos, PhD.

LÍNEA DE INVESTIGACIÓN: Comunicaciones digitales

FECHA: 23 de julio del 2021

RESUMEN EJECUTIVO

En el primer escenario el objetivo fue desarrollar una aplicación que graficara Series Trigonómicas de Fourier de funciones continuas y representar funciones graficas con alto grado de fidelidad sin la necesidad de Internet para ejecutar los procesos necesarios.

Esta aplicación se desarrolló e implementó en Python con las librerías SymPy y numpy para utilizar cálculo informático. Los resultados mostrados por la aplicación están basados en la fidelidad con el desarrollo teórico.

En el segundo escenario actual, ha habido un inmenso avance en el campo de las comunicaciones inalámbricas en este mundo de la ingeniería moderna. La tecnología actual de radio definida por software (SDR) es una tecnología emergente indiscutible y presenta nuevos desafíos para los ingenieros de comunicaciones. El avance del sistema SDR ha logrado un progreso significativo en los últimos años, lo que lo convierte en un sustituto de las arquitecturas de radio de hardware tradicionales donde los procedimientos matemáticos son obligatorios para decodificar y procesar señales de radio utilizando circuitos analógicos. Recientemente, las computadoras han resultado ser lo suficientemente poderosas para hacer los cálculos matemáticos requeridos usando software. Por lo tanto, el objetivo de este proyecto de investigación es demostrar que un analizador de espectro basado en RTL-SDR se puede utilizar de manera competente como una alternativa al analizador de espectro de hardware

existentes. Este enfoque reducirá la complejidad del sistema de hardware analógico con la mayor manejabilidad de las técnicas de filtrado y demodulación basadas en software. Como los dispositivos RTL-SDR son relativamente económicos (aproximadamente \$30) y de tamaño pequeño, este sistema también ofrece rentabilidad con la provisión de portabilidad. Se realizó un estudio experimental con las condiciones adecuadas para examinar la viabilidad y eficiencia del sistema propuesto.

Básicamente este proyecto de investigación presenta la implementación, puesta a prueba de un aplicativo para la simulación de Series Trigonómicas de Fourier y una radio definida por software de bajo costo basados en la computadora Raspberry Pi4, el receptor de radio RTL-SDR y el software GNU Radio. Se abordan aspectos clave relacionados con la instalación y configuración de la radio GNU y los controladores RTL-SDR en el sistema operativo Raspberry Pi OS. El prototipo resultante puede demodular señales de AM y FM en el rango de frecuencia de 30MHz a 1.7GHz y también puede cumplir con la función de Spectrum Analyzer en aplicaciones exigentes de sensibilidad moderada.

Descriptor: Series Trigonómicas de Fourier, Python, SymPy, numpy, comunicaciones inalámbricas, radio definida por software (SDR), RTL-SDR, analizador de espectro, simulación, Raspberry Pi4, GNU Radio, sistema operativo.

UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL
MAESTRÍA EN TELECOMUNICACIONES

THEME:

“SIMULATION OF APPLICATIONS WITH FOURIER TRIGONOMETRIC
SERIES IN ELECTRONIC COMMUNICATIONS SYSTEMS”

AUTHOR: Ing. Alex Francisco Rodríguez Ortiz

DIRECTED BY: Ing. Carlos Diego Gordón Gallegos, PhD.

LINE OF RESEARCH: Digital communications

DATE: July 23th, 2021

EXECUTIVE SUMMARY

In the first scenario, the objective was to develop an application that would graph Fourier Trigonometric Series of continuous functions and represent graphical functions with a high degree of fidelity without the need for the Internet to execute the necessary processes.

This application was developed and implemented in Python with the SymPy and numpy libraries to use computer calculation. The results displayed by the application are based on fidelity with the theoretical development.

In the second current scenario, there has been an immense advance in the field of wireless communications in this world of modern engineering. Today's software-defined radio (SDR) technology is an undisputed emerging technology and presents new challenges for communications engineers. The advancement of the SDR system has made significant progress in recent years, making it a substitute for traditional hardware radio architectures where mathematical procedures are required to decode and process radio signals using analog circuitry. Recently, computers have turned out to be powerful enough to do the required mathematical calculations using software. Therefore, the aim of this research project is to demonstrate that an RTL-SDR-based spectrum analyzer can be used competently as an alternative to the existing hardware spectrum analyzer. This approach will reduce the complexity of the analog hardware system with the greater manageability of software-based filtering and demodulation techniques. As RTL-SDR devices are relatively inexpensive (approximately \$ 30) and

small in size, this system also offers cost-effectiveness with the provision of portability. An experimental study was carried out with the appropriate conditions to examine the viability and efficiency of the proposed system.

Basically this research project presents the implementation, testing of an application for the simulation of Fourier Trigonometric Series and a low-cost software-defined radio based on the Raspberry Pi4 computer, the RTL-SDR radio receiver and the GNU software. Radio. Key issues related to installing and configuring the GNU radio and RTL-SDR drivers on the Raspberry Pi OS are covered. The resulting prototype can demodulate AM and FM signals in the 30MHz to 1.7GHz frequency range and can also fulfill the role of Spectrum Analyzer in demanding applications of moderate sensitivity.

Keywords: Trigonometric Series of Fourier, Python, SymPy, numpy, wireless communications, Software Defined Radio (SDR), RTL-SDR, spectrum analyzer, simulation, Raspberry Pi4, GNU Radio, operating system.

CAPÍTULO I

EL PROBLEMA DE INVESTIGACIÓN

1.1 Introducción

La necesidad de las personas por obtener una forma estable de comunicación es la razón por la cual se han constatado grandes descubrimientos a través de la historia, demostrando que las capacidades humanas van a paso de gigante, de esta manera los análisis respectivos de este tema a constituido las bases respectivas de grandes descubrimientos, que como ya se han mencionado se han venido refutando en el tiempo, evolucionando y desarrollando nuevas teorías e hipótesis que constituyen la construcción de un conocimiento basado en la demostración de la verdad, aunque es un poco contradictorio, es muy coherente, debido a que no debemos caer en la verdades absolutas; por ello, la importancia de estudiar este tema es apreciar y comprobar las respectivas temas considerados como verdad, llevando al análisis práctico de modelos matemáticos y simulados, refutando las teorías impuestas, constituyendo un continuo aprendizaje y sobre todo heredando un conocimientos para las futuras generaciones.

Una serie de Fourier, “en ciencias como en la física, se utilizan básicamente para analizar funciones que son periódicas, analizamos su correspondiente Serie de Fourier, que no es más que una descomposición de la función original en una suma infinita de funciones elementales en senos y cosenos que tienen frecuencias múltiplos de la señal inicial. En ingeniería se usan en óptica, acústica, procesamiento de señales (audio, video o simplemente imágenes), estudio de vibraciones y perturbaciones de sistemas, etc...” [1]

El analizador espectro es un instrumento de medición multipropósito para medir señales en el dominio de la frecuencia. Un analizador que se puede usar para medir señales en un transmisor que requiere parámetros de medición como frecuencia, potencia, ganancia y ruido. Los analizadores espectro también se utilizan para investigar la distribución de energía a lo largo de la frecuencia espectral de señales eléctricas conocidas, de esta investigación se obtuvo información muy valiosa sobre el ancho del campo de frecuencia (ancho de banda), el efecto de varios tipos de modulación, la generación de falsos señales y así sucesivamente todos sus beneficios en la planificación y prueba de frecuencias de RF (radiofrecuencia)[2]. El analizador

de espectro tiene una función especial para medir una serie de señales en un espectro de frecuencia limitado. Considerando que GNU Radio es un software que se puede utilizar de forma gratuita y se utiliza para crear una Radio definida por software (SDR), sin necesidad de hardware [3]. GNU Radio utiliza el lenguaje de programación C++ o Python en su implementación. La ventaja de GNU Radio es el software de código abierto y no cuesta mucho en términos de sus operaciones. Los investigadores pretenden medir la potencia de una frecuencia utilizando GNU Radio y luego compararla con un analizador de espectro [4].

Cada individuo o institución que usa una onda de frecuencia debe usar un analizador de espectrometría el cual costará más que comprar un analizador de espectrometría. Este prototipo quiere simular un analizador de espectro utilizando SDR y GNU Radio para reducir los precios en el uso de analizadores de espectro. La simulación llevada a cabo mide la relación potencia & señal / ruido en las frecuencias, 30MHz a 1.7GHz, la diferencia de unidades de potencia entre un analizador de espectrometría dados en (dBm) y GNU Radio (dB) requiere que esta simulación convierta unidades en las mismas unidades en este caso la simulación se realiza en unidades dB, para cambiar las unidades dB en GNU Radio se usa el principio de $30 \text{ dBm} = 1000 \text{ mW} = 1 \text{ W}$ [5]. GNU Radio no solo es útil en la investigación, sino también en el mundo de la educación, de acuerdo con la investigación [6] se ha sugerido como un método alternativo para desarrollar prácticas de la Teoría de la Comunicación. Muchas universidades aplican el aprendizaje utilizando un entorno basado en ecuaciones como Matlab, o solo se enfocan en las capacidades de procesamiento de señales que ofrece DSP, pero la incorporación de sistemas de software, hardware en GNU Radio y SDR-RTL permite que el aprendizaje sea más efectivo. SDR es la mejor solución para la tecnología de próxima generación, SDR es una de las soluciones para que los investigadores prueben técnicas en tiempo real, SDR demuestra avances tecnológicos especialmente en telecomunicaciones y electrónica [7].

1.2 Justificación

El presente estudio se origina debido al interés de mejorar el conocimiento práctico y teórico en la facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato, tanto en la simulación de aplicaciones con series trigonométricas de Fourier para el área de comunicaciones a través de un equipo portátil basado en software libre.

Existe factibilidad para realizar el presente trabajo investigativo, pues se cuenta con el apoyo necesario de FISEI de la Universidad Técnica de Ambato, los conocimientos suficientes del investigador, bibliografía especializada, recursos tecnológicos y económicos necesarios.

La investigación tuvo una utilidad teórica y práctica, pues se contribuyó con la teoría a mejorar el conocimiento científico con las temáticas relacionadas al problema de investigación, y en la práctica se lo demostró con la presentación de una propuesta de solución al problema investigado.

La originalidad de este proyecto de investigación se basó en la aplicación de estudios actualizados, libros de consulta, apegados en software libre, además de un estudio minucioso de las necesidades prácticas que tiene la FISEI y su aportación a dicha necesidad.

Los beneficiarios de este proyecto son la FISEI de la Universidad Técnica de Ambato, en lo relacionado a un equipo práctico, e indirectamente a docentes y estudiantes.

1.3 Objetivos

1.3.1 General

Simular aplicaciones con Series Trigonométricas de Fourier en Sistemas de Comunicaciones Electrónicas.

1.3.2 Específicos

- Investigar el estado del arte y la literatura de las series trigonométricas de Fourier para identificar sus características principales.
- Elaborar el modelo del sistema integrado de la aplicación para serie trigonométricas de Fourier.
- Desarrollar e implementar el modelo del sistema integrado de la aplicación con series trigonométricas de Fourier para el área de comunicaciones con software libre.
- Realizar pruebas y validación de resultados del sistema integrado de la aplicación con series trigonométricas de Fourier.

CAPÍTULO II

ANTECEDENTES INVESTIGATIVOS

2.1 Estado del Arte

XXVI Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas (2018), de la Escuela Politécnica de Ingeniería de Gijón, indica que muchos fenómenos naturales tienen un carácter periódico y oscilante. Las series de Fourier y, en general, las series trigonométricas constituyen una excelente herramienta para el estudio de dichos procesos y tienen numerosas aplicaciones en la ingeniería: análisis de vibraciones, acústica, óptica, electrónica, economía, medicina, procesamiento de imágenes y señales, compresión de datos, etc. Se trata de descomponer una señal periódica en términos de señales periódicas básicas cuyas frecuencias son múltiplos de la señal original. En este trabajo se presenta un laboratorio virtual desarrollado como una interfaz gráfica de usuario de Matlab y una propuesta metodológica para el estudio de las propiedades de este tipo de series, tanto desde un punto de vista analítico como gráfico, con el objetivo de que los alumnos puedan afianzar sus conocimientos sobre el tema. [8]

Enrique Arévalo Zamudio et al (2018), presenta la Obtención y Validación de la Serie de Fourier para el Análisis del Contenido Armónico en un Control por Fase en la cual expone el análisis del contenido armónico de un circuito convertidor de CA/CA basado en la técnica de Control por Fase. En primer lugar, se presenta el cálculo de los coeficientes de la Serie de Fourier correspondientes a la forma de onda de salida del convertidor. La serie calculada es validada mediante programación de la Serie de Fourier con MATLAB para diversos ángulos de disparo en los tiristores. Finalmente, se presentan los resultados obtenidos en forma experimental, empleando un medidor de armónicas comercial. El comportamiento observado en la simulación, así como en el circuito real, muestra que los resultados obtenidos analíticamente cumplen con la representación de la forma de onda del convertidor con un margen de error aceptable. El ejercicio realizado tiene una finalidad didáctica y es evaluar la posibilidad de incorporar el tema en alguna asignatura relacionada con Electrónica de Potencia en las carreras de ingeniería que se imparten en la UAMAzcapotzalco. [9]

Rosa María Farfán Márquez et al (2019), en su artículo realizado a cerca de la “Situación de Aprendizaje para la Serie Trigonométrica de Fourier desde la Teoría Socio epistemológica” manifiesta que el objetivo de este es mostrar una propuesta didáctica que propicie la significación de la Serie Trigonométrica de Fourier a través de una situación de aprendizaje, cuyo fundamento se basa en la Teoría Socio epistemológica de la Matemática Educativa, en investigaciones donde se ha problematizado este saber. La Serie Trigonométrica de Fourier es un tema complejo para su aprendizaje en el nivel superior, donde por lo general se mecaniza el proceso sin comprender del todo su funcionamiento y características. Se quiere comprobar que con actividades que apoyen la relación entre lo algebraico y lo geométrico, haciendo uso de GeoGebra – software de geometría dinámica – como variable de control, se puede significar a la serie y su convergencia mediante un contexto físico-geométrico.[10]

José Silva, et al (2020), en sus estudios realizados y presentados en la Universidad de La Salle de la facultad de Ingeniería Eléctrica manifiesta que la necesidad de las personas por obtener una forma estable de comunicación es la razón por la cual se han constatado grandes descubrimientos a través de la historia, demostrando que las capacidades humanas van a paso de gigante, de esta manera los análisis respectivos de este tema (SERIES DE FOURIER DE SEÑALES COMPLEJAS) a constituido las bases respectivas de grandes descubrimientos, que como ya se han mencionado se han venido refutando en el tiempo, evolucionando y desarrollando nuevas teorías e hipótesis que constituyen la construcción de un conocimiento basado en la demostración de la verdad, aunque es un poco contradictorio, es muy coherente, debido a que no debemos caer en la verdades absolutas; por ello, la importancia de estudiar este tema es apreciar y comprobar las respectivas temas considerados como verdad, llevando al análisis práctico de modelos matemáticos y simulados, refutando las teorías impuestas, constituyendo un continuo aprendizaje y sobre todo heredando un conocimientos para las futuras generaciones. [11]

Jesús Francisco Duarte Martínez (2020) en su investigación realizada para el Instituto Tecnológico de Chihuahua II, manifiesta que la aplicación de la serie de Fourier en la solución al aérea de comunicaciones es usada en muchas ramas de la ingeniería, además de ser una herramienta sumamente útil en la teoría matemática abstracta. Áreas de aplicación incluyen análisis vibratorio, acústica, óptica, procesamiento de

imágenes, señales y compresión de datos. En ingeniería, para el caso de los sistemas de telecomunicaciones, y a través del uso de los componentes espectrales de frecuencia de una señal dada, se puede optimizar el diseño de un sistema para la señal portadora del mismo. Además cabe destacar su importancia en los análisis del comportamiento armónico de una señal, en el reforzamiento de señales y la generación de formas de onda de corriente o tensión eléctrica por medio de la superposición de sinusoides generadas por osciladores electrónicos de amplitud variable cuyas frecuencias ya están determinadas.[12]

Jordi Mas i Hernández (2020) en su artículo define que el software libre tuvo sus inicios en pequeños grupos de entusiastas y activistas que querían cambiar la industria del software. Después de muchos años esta comunidad ha crecido rápidamente e incorporado no solamente voluntarios en todo el mundo, sino atraído la atención y la colaboración de centenares de empresas claves. El movimiento del software libre se ha desarrollado orgánicamente y cualquier instalación de éste está conformada de distintos componentes provistos por distintos participantes. Inicialmente el software libre fue popular en los servidores y con el paso del tiempo y el trabajo de miles de voluntarios las piezas faltantes se completaron. Linux que antes estaba limitado a ser un sistema que no era visible a los usuarios finales --- Google y Amazon son ejemplos de sistemas construidos sobre Linux--- ahora es un sistema que es usado por miles de usuarios en todo el mundo en sus computadoras personales, teléfonos o organizadores personales. Aunque prácticamente usar el software libre es muy similar a usar cualquier otro software, el software libre es un legado que le pertenece a toda la humanidad y como movimiento es interesante entender como un grupo de programadores, voluntarios y activistas ha creado esta plataforma. El software libre es propiedad de todos: cada persona en el mundo tiene derecho a usar el software, modificarlo y copiarlo de la misma manera que los autores de este mismo. Es un legado de la humanidad que no tiene propietario, de la misma manera que las leyes básicas de la física o las matemáticas. No existe un monopolio y no es necesario pagar peaje por su uso.

Un factor interesante en los últimos años ha sido la adopción del software libre en los gobiernos. Algunos lo han adoptado para ahorrar dinero, otros lo han hecho por cuestiones de seguridad, otros para ayudar a la creación de industrias locales y otros porque el software libre les pertenece.[13]

2.2 Marco Teórico

ANALISIS DE LAS TELECOMUNICACIONES

2.3 Estructura de un sistema básico de Telecomunicaciones

Los sistemas básicos de telecomunicaciones se dividen en 7 puntos:[14]

- Fuente de información
- Transductor de entrada
- Transmisor
- Canal
- Receptor
- Transductor de salida
- Destino

Dicho sistema se puede ver detallado mediante un diagrama de bloques como se observa en la Figura 2.1

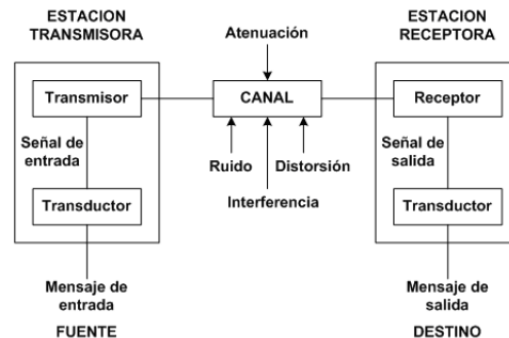


Figura 2.1 Diagrama en bloques de un sistema básico de telecomunicaciones [14]

2.3.1 Fuente de Información

Puede entenderse como fuente de información, cualquier elemento que pueda generar una secuencia de mensajes con suficiente claridad. Hay muchos tipos de fuentes, como máquinas y humanos. Cada persona o máquina desarrolla un algoritmo que produce una secuencia de mensajes según un protocolo de comunicación que conoce. La cadena de mensajes generada se envía a otros elementos que pueden manejar y comprender el mismo protocolo de comunicación. Al traducir este concepto a un entorno real, hemos encontrado una manera de vincular una secuencia de mensajes que se originan en una fuente a una secuencia de pulsos eléctricos con una amplitud diferente a la amplitud de la información transmitida. La cadena de mensajes enviados

por una fuente de información puede tener características digitales o analógicas que cambian con el tiempo. Independientemente de las características y protocolos que utilice la fuente de información para enviar una serie de mensajes, el objetivo siempre es intentar entregar al destinatario una copia casi exacta del mensaje original.[15]

2.3.2 Transductor

Se trata de asociar una serie de mensajes a una magnitud eléctrica. Este usa un transductor para convertir el mensaje en una "señal eléctrica" debido a que el mensaje generado por la fuente de información no contiene ninguna propiedad eléctrica. Con un transductor, el mensaje se convierte en un pulso eléctrico variable en el tiempo, como "corriente y voltaje". Normalmente, un sistema de comunicación básico utiliza estos dos parámetros para transformar el mensaje original y prepararse para establecer la comunicación entre el emisor y receptor. Básicamente se ha mencionado que el transductor realiza las funciones de un traductor, también es importante saber que un sistema de comunicación básico consta de dos de ellos. El primer sistema se denomina "Transductor de entrada o codificador" situado en el extremo del transmisor. El otro se llama "Transductor de salida o decodificador" en el lado de recepción del sistema de comunicación. [15]

2.3.3 Transmisor

En el sistema de comunicación básico, el transductor de entrada generalmente se acopla directamente al medio de transmisión, cuyo propósito es comunicarse entre dos extremos remotos, por lo que el teléfono fijo se puede tomar como un claro ejemplo. Las señales que llegan al receptor requieren el uso de medios de transmisión conocidos, cada uno de los cuales tiene la propiedad específica de producir una respuesta particular en función de la señal que pasa a través de ellos. El transmisor es responsable de generar la señal en base a las características del medio de transmisión y modificar sus parámetros (modulación) para ajustar la señal y sea compatible con el medio de la señal sin pérdidas hacia su destino. [15]

2.3.4 Canal

El término "canal" en telecomunicaciones se refiere a un medio de transmisión a través del cual se envía información a un destino transmitida en forma de impulsos eléctricos u ondas electromagnéticas. El canal de comunicación proporciona características eléctricas y mecánicas para conectar el transmisor y el receptor en un entorno de comunicación. La transferencia de información entre ellos depende del modo de

transmisión utilizado. Esto permite dividir los canales de comunicación en dos grupos: grupo de transmisión guiada (canal telefónico, cable coaxial, cable de fibra óptica). Propagación libre (canal de radio, canal de radio móvil, canal de satélite).[15]

2.3.5 Receptor

El receptor trabaja con el transmisor y restaura la señal enviada por el transmisor para obtener una copia casi exacta de la señal fuente. Cuando se restablece la señal, se envía en secuencia al transductor de salida restaurándolo a su formato original y enviándolo a su destino. En el receptor, es fundamental utilizar un filtro para recuperar la señal. El filtro se aplica como selector de frecuencia. Esto le permite distinguir (filtrar) las frecuencias de interés para obtener una respuesta correcta del sistema. [15]

2.3.6 Ancho de banda y potencia de la señal

Las aplicaciones diseñadas en un sistema básico de comunicación deben tener en cuenta dos parámetros fundamentales como el ancho de banda y la intensidad de la señal radiada. Estos factores determinan su presupuesto de enlace, así que se debe asegurar de optimizar estos recursos sin afectar el rendimiento esperado del sistema. El lema del sistema básico de comunicación es enviar más información en el menor tiempo con el mínimo consumo de energía.

- El ancho de banda representa la porción de frecuencias permitidas en un sistema o aplicación de telecomunicaciones. El ancho de banda debe ser lo suficientemente amplio (ancho) para permitir el paso de la frecuencia que contiene la información de interés.
- La fuente de señal representa la especificación de voltaje o corriente que la fuente proporciona para alimentar la carga. En muchas aplicaciones existe una fuente de alimentación que depende de la distancia para conectar el transmisor y el receptor. Encienda la carga.

Por lo tanto, es necesario conocer la respuesta en frecuencia de las señales presentes en un sistema básico de comunicación. En general, las formas de onda que se encuentran en las señales tienen una característica de tiempo continuo. En otras palabras, puede ver la forma invariante a lo largo del eje del tiempo. Un sistema invariante en el tiempo es un sistema cuyas propiedades y comportamiento permanecen fijos en el tiempo. Por ejemplo, la salida se muestra en la pantalla del osciloscopio cuando se transmite la señal de onda sinusoidal. Para conocer la respuesta en frecuencia de cualquier tipo de señal, con o sin periodicidad, se necesita obtener su

representación fuera del dominio tiempo. Esto se consigue gracias a la teoría planteada por Joseph Fourier. Permitiendo obtener una representación de frecuencia de una señal periódica utilizando un algoritmo llamado serie de Fourier. Para señales no periódicas, el algoritmo introducido se llama transformada de Fourier. Representación espectral de una señal que muestra cómo se distribuye la energía en diferentes componentes de frecuencia. Para señales periódicas, el espectro de frecuencia es discreto y la energía se distribuye en varias frecuencias, las llamadas frecuencias fundamentales, que están directamente correlacionadas con el período de la señal.

Por otro lado, la representación espectral de una señal no periódica es una señal de frecuencia continua Figura 2.2 y Figura 2.3. [16]

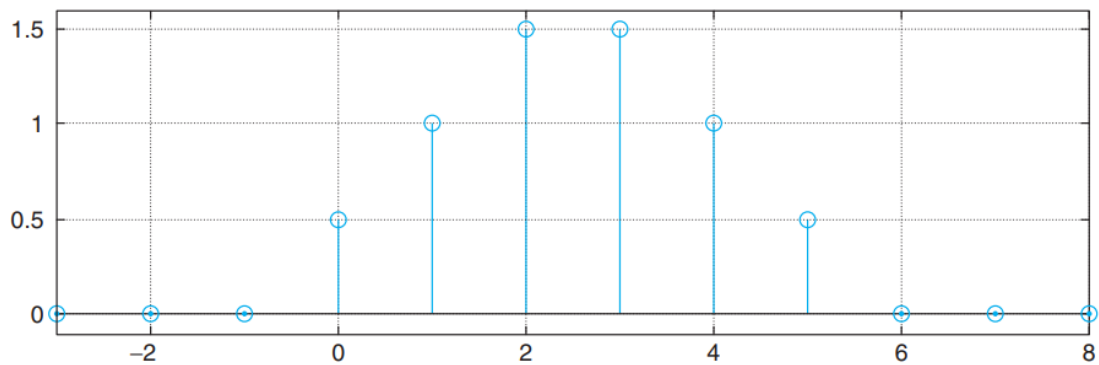


Figura 2.2 Representación del espectro en frecuencia de señal Periódica [16]

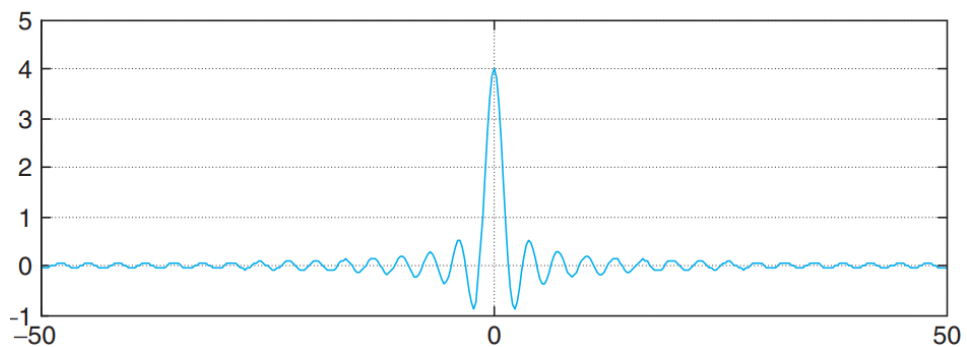


Figura 2.3 Representación del espectro en frecuencia de señal no Periódicas [16]

2.4 Modulación

La modulación es un conjunto de tecnologías desarrolladas para transferir señales de información a través de un canal de transmisión, que es el concepto de modulación. Este no es siempre el caso, pero generalmente se trata de convertir frecuencias bajas en frecuencias altas. Técnicamente, la modulación modifica los parámetros (amplitud, frecuencia, fase) de una señal llamada portadora (generalmente una onda sinusoidal) en proporción a la información o a la señal moduladora.[17]

2.4.1 ¿Por qué se modula una señal?

Es necesario modular las señales por diferentes razones:

- 1) Si todos los usuarios transmiten a una misma frecuencia la señal original o moduladora, no será posible que en el receptor se reconozca la información contenida en dicha señal, debido a que el receptor captará todas las señales sintonizadas en su misma frecuencia lo que genera interferencia entre las señales transmitidas por diferentes usuarios.
- 2) A altas frecuencias se mejora la eficiencia en la transmisión y se logra proteger la señal inteligible de la presencia de ruido, de acuerdo al medio de transmisión que se emplee.
- 3) Se optimiza el espectro electromagnético, ya que permite la multiplicación por frecuencias aprovechando así el canal de comunicación enviando más información por el mismo canal.
- 4) En caso de transmisión inalámbrica, logramos reducir el tamaño de las antenas a medidas más razonables.

2.5 Modos de Transmisión

- Los sistemas básicos de telecomunicaciones se pueden diseñar con la intención de manejar la transmisión en una sola dirección, en ambas direcciones, pero sólo uno a la vez, o ambas direcciones al mismo tiempo. A esto se le conoce como modos de transmisión que se lo define a continuación:[18]

2.5.1 Simplex

La transmisión se realiza en una sola dirección, como una calle de una sola dirección. Los sistemas simplex o sistemas de un sentido, se usan en canales dedicados a sólo recibir o sólo transmitir mensajes. El mensaje puede ubicarse en un transmisor o un receptor, pero no en ambos. Los ejemplos de transmisión simplex son aplicaciones de radiodifusión como la radio comercial o la televisión; la estación de radio siempre transmite y el usuario siempre recibe. El modo simplex puede usar todo el ancho de banda disponible en el canal para enviar datos en una dirección.

En la Figura 2.4 se puede observar un ejemplo de la transmisión simplex. [18]



Figura 2.4 Modelo de Tx SIMPLEX [18]

2.5.2 Half Dúplex o Semi Dúplex

La transmisión se realiza en ambas direcciones, pero no de manera simultánea, cuando un dispositivo está enviando el otro solo puede recibir y viceversa, podríamos hacer la analogía de este modo de transmisión al compararlo con una calle de un solo carril y tráfico en dos direcciones. Estos sistemas son llamados también: sistemas con alternativa de los sentidos, cualquier sentido, o cambio y fuera. El mensaje puede ubicarse en un transmisor o en un receptor, pero no en los dos al mismo tiempo. Por ejemplo, los sistemas de radio de doble sentido que utilizan los botones Push-to-talk (PTT) “oprima para hablar”, para operar sus transmisores, como los radios de banda civil y de banda policiaca o sistemas de comunicación de taxistas. En este modo de transmisión todo el ancho de banda del canal es usado por la estación que se encuentre transmitiendo. [18]

En la Figura 2.5 se puede observar un ejemplo de la transmisión Half Dúplex.

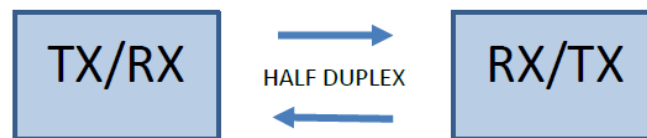


Figura 2.5 Modelo de Tx HALF DUPLEX O SEMIDUPLEX [18]

2.5.3 Full Dúplex

La transmisión se realiza de manera simultánea en ambas direcciones, llamadas también líneas simultáneas de doble sentido, dúplex o de ambos sentidos.

El mensaje puede ubicarse en transmisor y receptor simultáneamente; sin embargo, la comunicación debe ser orientada a un mismo canal; es decir que la estación a la que se está transmitiendo también debe ser la estación de la cual está recibiendo. Por ejemplo, un sistema telefónico estándar. En este modo de transmisión el ancho de banda del canal está dividido entre ambas estaciones. [18]

En la Figura 2.6 se puede observar un ejemplo de la transmisión Full Duplex.

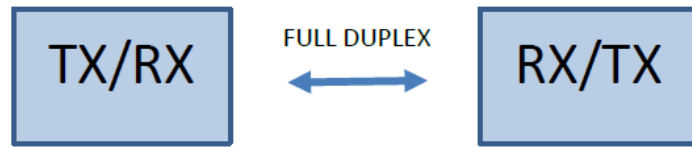


Figura 2.6 Gráfico del Modelo de Tx FULL DUPLEX [18]

2.6 Demodulación

Es el proceso de revertir los cambios hechos a la portadora analógica, con el fin de recuperar la señal de información originada en la fuente. La demodulación se realiza en el receptor, en un circuito llamado demodulador o detector como también se le conoce.

El detector o demodulador debe ser capaz de recibir y amplificar una señal. Un detector también debe tener la capacidad de limitar las bandas del espectro total de la señal modulada a una banda específica de frecuencias.

La modulación consiste básicamente en hacer una traslación de espectros o lo que es lo mismo, hacer convolucionar los espectros tanto de la señal modulante como de la señal moduladora, se puede obtener un demodulador haciendo convolucionar nuevamente estos espectros con el espectro de la señal portadora y aplicar un filtro pasa bajos centrado en la frecuencia de la señal moduladora. Es necesario centrar el filtro a la frecuencia de la moduladora ya que justamente el espectro de esta señal fue trasladado a la frecuencia de la señal portadora. Con esto se eliminarán las frecuencias más altas, el ejemplo más claro sobre esta teoría se obtiene de la modulación Am de portadora suprimida. La Figura 2.7 ilustra esta teoría.[19]

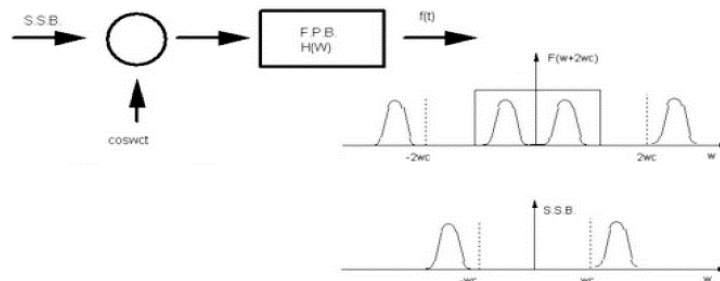


Figura 2.7 Ejemplo de demodulación y traslación de espectros [19]

Los receptores que se utilizan para las señales de modulación angular son muy similares a los que se usan para la recepción de AM de portadora suprimida, excepto por el método utilizado para extraer la información de audio.

Los métodos usados en la recepción de señales son dos: el primero es conocido como Detección Coherente y el otro es conocido como Detector de Envoltente.

En la detección coherente lo que se hace para recuperar la señal es multiplicar la señal modulada por un oscilador local a frecuencia de la portadora y pasar el resultado de esta convolución usando un filtro pasa bajos, centrado a la frecuencia de la señal de información.

En la detección de envoltente, aplicamos simplemente un filtro pasa bajos situado a frecuencia de la portadora y obtenemos la señal enviada.

2.7 Problemas en la recepción de señales

Una vez que la señal llega al receptor se presenta con cierta clase de inconvenientes, los cuales merecen ser profundizados en totalidad, pero no son el objeto de este trabajo. Se puede resumir que debido a que la señal se traslada gran distancia para lograr enlazar dos puntos, disminuyen en ella importantes características, como por ejemplo la amplitud, lo cual impide que sea una réplica exacta de la señal original enviada. Esto sucede cada que atraviesa un medio de transmisión sea guiado o no guiado. A este hecho se lo conoce como atenuación y todos los receptores son diseñados para tolerar un rango mínimo de amplitudes con niveles de potencia o voltaje aceptables, con el objetivo de diferenciarlas del ruido. [15]

ANALISIS DE SEÑALES

2.8 Procesamiento de señales.

Antes de abordar las técnicas propias para el tipo de señales con las que se desea trabajar, es necesario conocer los fundamentos teóricos generales en el que se basa el procesamiento digital de señales como base para determinar de manera adecuada técnica para la extracción de la información

Un claro ejemplo de la importancia del uso de esta herramienta de análisis matemático está en el estudio de la respuesta en régimen permanente de sistemas excitados o que producen señales periódicas (sinusoidales o no-sinusoidales) como[20]:

Rectificadores electrónicos: Los rectificadores electrónicos no filtrados alimentados por una fuente sinusoidal producen formas de ondas no sinusoidales pero que son periódicas, por lo tanto, gracias a las Series de Fourier se puede obtener una expresión

matemática que permite estudiar fenómenos tales como la saturación magnética (máquinas y transformadores): Figura 2.8

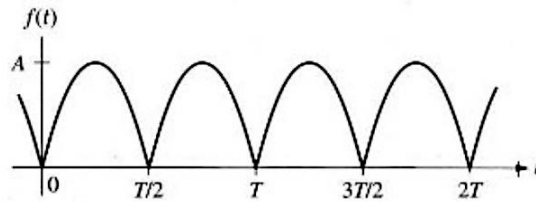


Figura 2.8 Onda sinusoidal rectificada [20]

Generadores de señales: Los generadores de este tipo tienen mayor presencia en laboratorios y sistemas electrónicos donde se precisa de señales periódicas para el estudio de la respuesta o el funcionamiento de diversos tipos de sistemas, y como en el caso anterior mediante el uso de Series de Fourier se puede obtener expresiones matemáticas para analizar tales sistemas como si fueran excitados por un conjunto de señales sinusoidales relacionadas armónicamente.

Generadores de potencia: Este tipo de sistema, aunque idealmente debería producir señales sinusoidales puras, en la práctica la señal que produce no es una senoide pura por lo que para una adecuada comprensión del comportamiento del sistema necesitamos conocer los componentes adicionales presentes en la señal de potencia, y esto se puede lograr mediante el uso de las Series de Fourier.

Análisis de sistemas no eléctricos: vibraciones mecánicas, flujo de fluidos, flujo calorífico, etc. Las magnitudes presentes en este tipo de sistemas son periódicas por lo que puede operar con ellas mediante el uso de sus representaciones en Series de Fourier.

2.9 Serie de Fourier

La idea básica y central de las series de Fourier es que toda función periódica de período T , puede ser expresada como una suma trigonométrica de senos y cosenos dentro del mismo período T . [21]

Una serie de Fourier es una serie infinita que converge puntualmente a una función periódica y continua a trozos (o por partes). Las series de Fourier constituyen la herramienta matemática básica del análisis de Fourier empleada para analizar funciones periódicas a través de la descomposición de dicha función en una suma infinita de funciones sinusoidales mucho más simples (como combinación de senos y

cosenos con frecuencias enteras). El nombre se debe al matemático francés Jean-Baptiste Joseph Fourier, que desarrolló la teoría cuando estudiaba la ecuación del calor. Fue el primero que estudió tales series sistemáticamente y publicó sus resultados iniciales en 1807 y 1811.

Es una expresión utilizada en varias ramas de la ingeniería, además de ser una herramienta sumamente útil en la teoría matemática abstracta. Áreas en las que se aplican series de Fourier incluyen análisis vibratorio, acústica, óptica, procesamiento de imágenes, señales y compresión de datos. En ingeniería, para el caso de los sistemas de telecomunicaciones y a través del uso de los componentes espectrales de frecuencia de una señal dada, se puede optimizar el diseño de un sistema para la señal portadora del mismo. Refiérase al uso de un analizador de espectros.

2.9.1 Serie Trigonométrica.

Se parte de que el siguiente conjunto de funciones trigonométricas

$$\left\{ 1, \cos \frac{\pi}{p} x, \cos \frac{2\pi}{p} x, \cos \frac{3\pi}{p} x, \dots, \operatorname{sen} \frac{\pi}{p} x, \operatorname{sen} \frac{2\pi}{p} x, \operatorname{sen} \frac{3\pi}{p} x, \dots \right\}$$

Ecuación 2-1

Es ortogonal en el intervalo $[-p, p]$. Surge la necesidad de desarrollar una función f definida sobre $[-p, p]$ en una serie ortogonal que consista en las funciones trigonométricas dadas en (2.1), es decir,

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\alpha} \left(a_n \cos \left(\frac{n\pi}{p} x \right) + b_n \operatorname{sen} \left(\frac{n\pi}{p} x \right) \right)$$

Ecuación 2-2

Al integrar ambos lados de (2.2) desde $-p$ a p resulta:

$$\int_{-p}^p f(x) dx = \frac{a_0}{2} \int_{-p}^p dx + \sum_{n=1}^{\alpha} \left(a_n \int_{-p}^p \cos \left(\frac{n\pi}{p} x \right) dx + b_n \int_{-p}^p \operatorname{sen} \left(\frac{n\pi}{p} x \right) dx \right)$$

Ecuación 2-3

Puesto que $\cos\left(\frac{n\pi}{p}x\right)$ y $\sin\left(\frac{n\pi}{p}x\right)$, $n \geq 1$, son ortogonales a 1 en el intervalo, el segundo miembro de (2.3) se reduce a un solo término:

$$\int_{-p}^p f(x) dx = \frac{a_0}{2} \int_{-p}^p dx = \frac{a_0}{2} x \Big|_{-p}^p = pa_0$$

Ecuación 2-4

Al despejar a_0 se obtiene:

$$a_0 = \frac{1}{p} \int_{-p}^p f(x) dx$$

Ecuación 2-5

Multiplicando (2.2) por $\cos\left(\frac{n\pi}{p}x\right)$ e integrando:

$$\int_{-p}^p f(x) \cos\left(\frac{m\pi}{p}x\right) dx = \frac{a_0}{2} \int_{-p}^p \cos\left(\frac{m\pi}{p}x\right) dx + \sum_{n=1}^{\infty} \left(a_n \int_{-p}^p \cos\left(\frac{m\pi}{p}x\right) \cos\left(\frac{n\pi}{p}x\right) dx + b_n \int_{-p}^p \sin\left(\frac{m\pi}{p}x\right) \cos\left(\frac{n\pi}{p}x\right) dx \right)$$

Ecuación 2-6

Mediante la ortogonalidad, se tiene:

$$\int_{-p}^p \cos\left(\frac{m\pi}{p}x\right) dx = 0, \quad m > 0, \quad \int_{-p}^p \cos\left(\frac{m\pi}{p}x\right) \sin\left(\frac{n\pi}{p}x\right) dx = 0$$

Ecuación 2-7

$$\int_{-p}^p \cos\left(\frac{m\pi}{p}x\right) \cos\left(\frac{n\pi}{p}x\right) dx = \begin{cases} 0, & m \neq n \\ p, & m = n \end{cases}$$

Ecuación 2-8

Por lo tanto, (2.6) se puede simplificar a

$$\int_{-p}^p f(x) \cos\left(\frac{n\pi}{p}x\right) dx = a_n p$$

Ecuación 2-9

y así

$$a_n = \frac{1}{p} \int_{-p}^p f(x) \cos\left(\frac{n\pi}{p}x\right) dx$$

Ecuación 2-10

Por último, multiplicando (2.2) por $\sin\left(\frac{n\pi}{p}x\right)$, integrando y utilizando los resultados:

$$\int_{-p}^p \sin\left(\frac{m\pi}{p}x\right) dx = 0, \quad m > 0, \quad \int_{-p}^p \sin\left(\frac{m\pi}{p}x\right) \sin\left(\frac{n\pi}{p}x\right) dx = 0$$

Ecuación 2-11

$$\int_{-p}^p \sin\left(\frac{m\pi}{p}x\right) \cos\left(\frac{n\pi}{p}x\right) dx = \begin{cases} 0, & m \neq n \\ p, & m = n \end{cases}$$

Ecuación 2-12

Se encuentra que:

$$b_n = \frac{1}{p} \int_{-p}^p f(x) \sin\left(\frac{n\pi}{p}x\right) dx$$

Ecuación 2-13

Se dice que la serie trigonométrica (2) con coeficientes a_0 , a_n y b_n definidos por (2.5), (2.7) y (2.8), respectivamente, se conoce como serie de Fourier de la función f . Los coeficientes obtenidos a partir de (2.5), (2.7) y (2.8) se conocen como coeficientes de Fourier de f .

Para calcular los coeficientes a_0 , a_n y b_n , se supone que f es integrable en el intervalo y que (2.2), así como la serie obtenida al multiplicar (2.2) por $\cos\left(\frac{n\pi}{p}x\right)$, converge de tal manera que permite la integración término a término. Hasta que se demuestre que (2.2) es convergente para una función f dada, el signo de igualdad no se tomará en sentido estricto o literal. En algunos textos se utiliza el símbolo \sim en lugar de $=$. En vista de que la mayoría de las funciones incluidas en las aplicaciones son de un tipo que garantiza la convergencia de la serie, aquí utilizaremos el símbolo de igualdad.

Definición de la Serie de Fourier (Zill & Warren, 2015): La serie de Fourier de una función f definida en el intervalo $(-p, p)$, está dada por:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos\left(\frac{n\pi}{p}t\right) + b_n \sin\left(\frac{n\pi}{p}t\right) \right)$$

Ecuación 2-14

donde

$$a_0 = \frac{1}{p} \int_{-p}^p f(x) dx$$

Ecuación 2-15

$$a_n = \frac{1}{p} \int_{-p}^p f(x) \cos\left(\frac{n\pi}{p}x\right) dx$$

Ecuación 2-16

$$b_n = \frac{1}{p} \int_{-p}^p f(x) \operatorname{sen}\left(\frac{n\pi}{p}x\right) dx$$

Ecuación 2-17

La forma canónica de las series de Fourier es la que se ha estado utilizando hasta el momento, donde la función en cuestión está definida sobre el intervalo $[-\pi, \pi]$.

Extensión periódica. Se observa que cada una de las funciones incluidas en el conjunto básico (2.1) tiene un periodo fundamental diferente, es decir, $\frac{2\pi}{n}$, $n \geq 1$; sin embargo, puesto que un múltiplo entero positivo de un periodo es también un periodo, se puede ver que todas las funciones tienen en común el periodo $2p$. En consecuencia, el lado derecho de (2.2) tiene periodo $2p$; de hecho, $2p$ es el periodo fundamental de la suma. Se concluye que una serie de Fourier no solo representa la función en el intervalo $(-p, p)$, sino también proporciona la extensión periódica de f fuera de este intervalo. Se puede suponer desde el principio que la función dada es periódica con periodo $T = 2p$; esto es, $f(x + T) = f(x)$.

En ocasiones es deseable adaptar la forma de una serie de Fourier a funciones periódicas de periodo $T = 2L > 0$ en el intervalo $[-L, L]$. Esto se consigue gracias a un cambio de variable.

Se puede reescribir la serie de Fourier y los coeficientes a_0 , a_n y b_n considerando lo anteriormente expuesto de la siguiente manera:

La serie de Fourier de una función f definida en el periodo $T = 2L$ en el intervalo $(-L, L)$, está dada por:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos\left(\frac{2n\pi}{T}x\right) + b_n \sin\left(\frac{2n\pi}{T}x\right) \right)$$

Ecuación 2-18

Donde

$$a_0 = \frac{2}{T} \int_{-L}^L f(x) dx$$

Ecuación 2-19

$$a_n = \frac{2}{T} \int_{-L}^L f(x) \cos\left(\frac{2n\pi}{T}x\right) dx$$

Ecuación 2-20

$$b_n = \frac{2}{T} \int_{-L}^L f(x) \operatorname{sen}\left(\frac{2n\pi}{T}x\right) dx$$

Ecuación 2-21

Puesto que elaborar un código computacional hace necesario contar con generalidades y símbolos durante la resolución de problemas, la optimización del código es fundamental. Ante ello, se decidió utilizar el tipo de nomenclatura que se ha presentado en (2.18), (2.19), (2.20) y (2.21), ya que soporta intervalos simétricos y asimétricos de cualquier periodo.

2.9.2 Series de Fourier de cosenos y senos

El esfuerzo que se lleva a cabo en la evaluación de los coeficientes a_0 , a_n y b_n al desarrollar una función f en una serie de Fourier se reduce significativamente cuando f es una función par o impar.

Se dice que una función f es:

Par si $f(-x) = f(x)$ e **Impar** si $f(-x) = -f(x)$.

En un intervalo simétrico $(-L, L)$, la gráfica de una función par cuenta con simetría respecto al eje y , mientras que la gráfica de una función impar tiene simetría con relación al origen.

La serie de Fourier de una función Par en el intervalo $(-L, L)$ es la serie de cosenos:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\alpha} a_n \cos\left(\frac{2n\pi}{T} x\right)$$

Ecuación 2-22

donde

$$a_0 = \frac{2}{T} \int_{-L}^L f(x) dx$$
$$a_n = \frac{2}{T} \int_{-L}^L f(x) \cos\left(\frac{2n\pi}{T} x\right) dx$$

Ecuación 2-23

La serie de Fourier de una función Impar en el intervalo $(-L, L)$ es la serie de senos:

$$f(x) = \sum_{n=1}^{\alpha} b_n \sin\left(\frac{2n\pi}{T} x\right)$$

Ecuación 2-24

donde

$$b_n = \frac{2}{T} \int_{-L}^L f(x) \operatorname{sen}\left(\frac{2n\pi}{T} x\right) dx$$

Ecuación 2-25

2.9.3 Fidelidad.

Con el desarrollo de la aplicación se busca que los resultados que se obtengan sean lo más parecidos a la resolución que se elabora en clase. Esto con la finalidad de que los alumnos usuarios sean conscientes del resultado de la resolución en el desarrollo de Series de Fourier.

“Transposición Informática”, frase acuñada por Balacheff (1994) citado en Camacho et. al., (2019), habla del fenómeno que vivimos cotidianamente en las escuelas, en donde se van introduciendo Software en el ámbito de la enseñanza matemática, pudiendo generar conflictos cognitivos, ya que se están reproduciendo conocimientos matemáticos en contextos diferentes a los que fue producida. Esto es que, la representación matemática de las posibles soluciones dependerá de la conceptualización e interpretación del autor del software matemático y no necesariamente de las bases metodológicas del alumno.

Camacho et. al, (2019) nos dicen que: “El concepto de fidelidad es una norma con la que pretendemos que las distorsiones y perturbaciones epistémicas provocadas por la asociación del conocimiento matemático y el software sea mínimo”. Para generar el menor conflicto posible, el software matemático desarrollado, se tiene que adaptar lo suficiente a la resolución en el cuaderno que pueda hacer un estudiante.

2.10 Analizador de Espectro

2.10.1 Definición

Un analizador de espectro es un dispositivo electrónico que puede graficar la potencia de un espectro de frecuencia como una relación amplitud / frecuencia que recibe una señal sinusoidal de amplitud máxima en su entrada y muestra información. Analiza los siguientes componentes basándose en tres características principales: centrado de frecuencia, rango de frecuencia en pantalla y amplitud.

2.11 Clasificación del Analizador de Espectro

2.11.1 Analizador de Espectro Analógico

Este tipo de analizador de espectro utiliza un conjunto de filtros que pueden

determinar el contenido de armónicos en un superheterodino para realizar su función. Además, son costosos debido al material voluminoso formado y no se pueden llenar con múltiples tipos específicos. Es una función, un proceso bastante complicado, y requiere conocimientos previos y la imposibilidad de almacenar información en la memoria interna.

2.11.2 Analizador de Espectro Digital

Los analizadores de espectro digitales para cumplir todo el desarrollo de estudio se basan en el Procesamiento Digital de Señales (DSP), la cual debe cumplir una serie de pasos como: realizar un muestreo de la señal de interés, luego con la información receptada se debe aplicar un algoritmo de la Transformada Discreta de Fourier (TDF) para poder procesar todos los datos, a diferencia de los dispositivos analógicos se almacena toda la información evaluada para un posterior análisis.

2.12 Funcionamiento

El funcionamiento de los analizadores de espectro se enfoca en los modelos digitales a razón de ser los más utilizados y comerciales actualmente, el funcionamiento que desempeñan todos los analizadores de espectro digitales son similares en todos los modelos, teniendo el siguiente proceso: selección de una señal proveniente de un transmisor, la cual posee una ganancia en amplitud establecida por el atenuador, además de otros parámetros como el ancho de banda y la Ventana de visualización horizontal (SPAN) para acondicionar la señal, luego se realiza una conversión analógica a digital de la señal obtenida para el análisis a posterior.

2.12.1 Frecuencia

Este componente es el principio fundamental en el que se enfoca el analizador de espectro, teniendo diferentes parámetros, como puede ser frecuencia inicial, frecuencia central, frecuencia final, amplitud, entre otros, todos estos valores son obtenidos mediante un análisis de Fourier sobre la frecuencia, las cuales sirven como variables para establecer la exactitud del análisis.

Con una frecuencia central se puede obtener características de la descomposición de su espectro, dando lugar al espectro de frecuencia, la cual será representada en un eje de coordenadas donde el eje x tiene el valor de frecuencia y el eje y la amplitud de la señal.

2.12.2 Amplitud

Es un valor que representa los picos altos de la señal recibida con la variación máxima en la propagación, el analizador de espectro muestra la amplitud de cada armónico de la señal como una función de la frecuencia, obteniendo dos valores: la amplitud fundamental que será la de mayor potencia recibida y la amplitud de todos los armónicos con niveles inferiores.

El analizador de espectro permite manipular el nivel de amplitud deseado teniendo cuidado en que la señal ocupe toda la pantalla, pero sin sobrepasar los límites visibles para tener una mejor perspectiva de todo el espectro recibido.

2.12.3 Ventana de visualización horizontal SPAN

Este parámetro es el margen de frecuencias que permite dar un rango de visualización horizontal por el establecimiento de una frecuencia inicial con una frecuencia final, generalmente el SPAN es un valor fijo y coloca automáticamente el límite inferior y superior, permitiendo que todo el espectro se encuentre distribuido simétricamente alrededor de la frecuencia central.

El analizador de espectro debe tener este parámetro como un valor netamente configurable para ser ajustado según las necesidades del usuario.

2.13 Aplicaciones

Los analizadores de espectro permiten realizar un análisis espectral de las señales emitidas en el espectro radioeléctrico, existiendo diferentes modelos que pueden abarcar rangos de frecuencias diferentes, permitiendo conocer las características para mejorar la efectividad de transmisión.

Generalmente son utilizados en el control del espectro radioeléctrico por parte de una agencia reguladora para el monitoreo de las estaciones de radiodifusión,

televisión, fijo, móvil, detección de interferencias, entre otras, pero también son utilizados en campos de estudio e investigación, permitiendo caracterizar frecuentemente amplificadores, filtros, osciladores y mezcladores.

ANALISIS DE SOFTWARE LIBRE

2.14 Software Libre

2.14.1 Introducción

Al hablar de software libre no estamos hablando de precio sino de libertad, la libertad de elegir, compartir y utilizar las herramientas que deseamos. Este software respeta la libertad, privacidad y derecho a construir conocimiento, no tiene restricciones de uso de ningún tipo. Cualquier persona lo puede usar, mejorar y compartir, también puede distribuir las copias que desee sin tener problemas legales, ya que cuentan con licencias con permisos.

2.14.2 Características

Un software es libre si garantiza las siguientes libertades según la Tabla 2.1:

Tabla 2.1 Libertades del Software Libre

Libertad 0	La libertad de usar el programa con cualquier propósito, sea privado, educativo, público, comercial, militar, etc.
Libertad 1	La libertad de estudiar cómo funciona el programa y adaptarlo a tus necesidades, para lo que es necesario poder acceder al código fuente.
Libertad 2	La libertad de distribuir copias, para poder ayudar a quien sea.
Libertad 3	La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

Fuente: Desarrollado por el investigador

2.14.3 Ventajas y Desventajas

Las ventajas que más resaltan en Software Libre son las que se detallan a continuación en la Tabla 2.2

Tabla 2.2 Ventajas del Software Libre.

Bajo costo	El costo de adquisición del software puede ser gratis o de costo muy reducido.
Independencia total	No está atado a las condiciones de mercado impuestas por empresas de software.
Seguridad y privacidad	Gracias al código fuente, se conocerá el funcionamiento interno, se encontrarán y corregirán los posibles errores y fallos de seguridad.
Adaptabilidad	Las modificaciones y correcciones de posibles errores se realizan de forma inmediata.
Calidad	Al ser de dominio público, es usado y depurado por varios desarrolladores que añaden y demandan constantemente nuevas funcionalidades.
Respecto a los estándares	El uso de software libre y sistemas abiertos facilita la interoperabilidad entre distintas organizaciones.
Predistribución	Cualquier cambio que se introduzca en programas de licencia libre debe ser incluido en versiones posteriores y añadido al código fuente.
No hay restricción legal de uso	No hay limitación en el número de licencias ni de copias dentro de la organización.
Continuidad	Se garantiza el derecho de cualquier usuario a continuar el desarrollo.
Facilidad	Se pueden iniciar nuevos proyectos basados en el código de un programa libre.

Fuente: Desarrollado por el investigador

A continuación, una breve descripción de las desventajas de Software Libre como se detalla en la Tabla 2.3

Tabla 2.3 Desventajas del Software Libre.

Costumbre y formación	Tratar de superar la tendencia de los usuarios a seguir usando el entorno de sistema operativo y aplicaciones que han venido usando hasta ahora.
-----------------------	--

Software desarrollado	Existe gran cantidad de software corporativo desarrollado internamente a lo largo del tiempo para entornos Windows que no funcionan en otros sistemas operativos.
Cantidad de aplicaciones	El número de aplicaciones comerciales disponibles para entornos como Windows es superior al de las aplicaciones libres.
Conjunto de hardware soportado	No todos los dispositivos hardware del mercado (monitores, impresoras, tarjetas de video, escáneres) disponen de drivers que permitan su funcionamiento en Linux.
Falta de garantía y soporte	Si se encuentran problemas con el uso de software libre, no se garantiza por parte del autor la resolución de los mismos ya que no se establece ningún vínculo contractual.

Fuente: Desarrollado por el investigador

2.14.4 Comparación de software libre con software propietario

Mediante la siguiente Tabla 2.4 se describe la comparativa de software libre con respecto a software propietario

Tabla 2.4 Software libre vs. Software propietario

	Software Libre	Software Propietario
Limitaciones	Libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado.	Es todo programa informático en el que el usuario tiene limitaciones para usarlo, modificarlo o redistribuirlo.
Libertades del software	Debe cumplir con cuatro tipos de libertades para que un programa pueda ser considerado libre.	Si una o varias libertades de un programa son restringidas por algún motivo es considerado propietario.
Seguridad	Hay más seguridad y fiabilidad.	Generalmente son menos seguras.
Interfaz gráfica	Suele ser muy sencilla y por tanto poco amigable.	Es más completa por lo que es considerada de avanzada.

Costo	Es muy bajo y mayormente gratuito.	Las aplicaciones tienen un costo mayor.
Aplicaciones compatibles con varias plataformas	Compatible con todas las plataformas	Compatibilidad de aplicaciones limitada
Dependencia del productor del software	No depende del creador del software.	Depende del creador del software.

Fuente: Desarrollado por el investigador

2.15 Arquitectura de software

La arquitectura de software ayuda a diseñar un programa de manera efectiva al enfocarse en cómo interactúan los componentes individuales del programa. El buen desarrollo de una arquitectura de software puede implicar el éxito o el fracaso del proyecto de software. Con el fin de representar la arquitectura de software se utiliza el framework como una estructura conceptual y tecnológica de soporte definido, con artefactos y módulos concretos de software, el cual, sirve de base para la organización y desarrollo del programa. [22]

2.16 Lenguaje de programación Python

En esta sección se muestra una descripción del lenguaje de programación de Python con el fin de verificar la importancia de utilizar este lenguaje de programación para este proyecto de tesis. Según Arturo Fernández Montoro en su libro Python 3 al descubierto define Python como: “un lenguaje de programación de alto nivel, interpretado y multipropósito. En los últimos años su utilización ha ido constantemente creciendo y en la actualidad es uno de los lenguajes de programación más empleados para el desarrollo de software.” Python puede ser usado en varias plataformas y sistemas operativos, entre, como Windows, Mac OSX y Linux. Este lenguaje puede desarrollar software para aplicaciones científicas, para comunicaciones de red, para aplicaciones de escritorio con interfaz gráfica (GUI). Como lo mencionan varios autores (Fernández, 2015; Orbegozo, 2013) las principales razones para elegir Python, sus principales características lo convierten en un lenguaje muy productivo; debido a que es un lenguaje potente, flexible con una sintaxis clara y concisa. Además, no requiere dedicar tiempo a su compilación debido a que es interpretado. Python es open source, es decir, cualquiera puede contribuir a su desarrollo y divulgación. Así mismo,

no se debe pagar ninguna licencia para distribuir un software desarrollado con este lenguaje. (Fernández, 2015) En la actualidad se ha desarrollado tres versiones de Python, cada una de ella con sus actualizaciones. La versión 2 su última actualización es la 2.7.12 y la versión 3 su última versión es la 3.5.2. Estas versiones se encuentran separadas, indicando que, tanto la versión 2.7.12 y la 3.5.2 son estables, pero su lógica difiere con respecto a las versiones. Se sugiere utilizar ambas versiones y no su última versión debido a que ambas son incompatibles una con otra (Fernández, 2015). Para el desarrollo de esta tesis se utilizó la versión 2.7.12, debido a que esta versión se encuentra en su fase final, contiene una gran variedad en sus librerías y la mayoría de las distribuciones de Linux y Mac actuales utilizan esta versión. En cambio, la versión 3.5.2 se encuentra en etapa de desarrollo, por lo tanto, se encuentra sujeta a actualizaciones y no presenta un gran abanico de opciones en su librería como la presenta la versión 2.7.12

2.17 Computación simbólica.

La computación simbólica se ocupa de la computación de objetos matemáticos simbólicamente. Esto significa que los objetos matemáticos se representan exactamente, no aproximadamente, y las expresiones matemáticas con variables no evaluadas se dejan en forma simbólica.

El poder real de un sistema de cálculo simbólico como la biblioteca SymPy es la capacidad de hacer todo tipo de cálculos simbólicamente. SymPy puede simplificar expresiones, calcular derivadas, integrales y límites, resolver ecuaciones, trabajar con matrices y mucho más, y hacerlo todo simbólicamente. Incluye módulos para trazar, imprimir (como la salida impresa en 2D de fórmulas matemáticas, o LATEX), generación de código, física, estadística, combinatoria, teoría de números, geometría, lógica y más.

Un requisito importante es que la aplicación pueda representar de manera simbólica los pasos intermedios y los resultados finales. Para este punto, se decidió utilizar SymPy. Una ventaja de esta librería es que se puede ejecutar en R, Julia y Python.

2.18 Software utilizado para el desarrollo

2.18.1 SymPy.

SymPy es una biblioteca de Python para matemática simbólica. Su objetivo es convertirse en un sistema de álgebra computacional (CAS) con todas las funciones, manteniendo el código lo más simple posible para que sea comprensible y fácilmente extensible. SymPy está escrito completamente en Python.

¿Qué hace que SymPy sea una mejor opción que las alternativas? se puede afirmar lo siguiente:

En primer lugar, SymPy es completamente gratis. Es de código abierto y tiene licencia bajo la licencia BSD liberal, por lo que puede modificar el código fuente e incluso venderlo si lo desea. Esto contrasta con los sistemas comerciales populares como Maple o Mathematica que cuestan cientos de dólares en licencias.

En segundo lugar, SymPy usa Python. La mayoría de los sistemas de álgebra computacional inventan su propio lenguaje. SymPy está escrito completamente en Python y se ejecuta completamente en Python. Esto significa que, si ya conoce Python, es mucho más fácil comenzar a usar SymPy, porque ya conoce la sintaxis (y si no conoce Python, es realmente fácil de aprender). Ya sabemos que Python es un lenguaje bien diseñado y probado en batalla. Los desarrolladores de SymPy confían en sus habilidades para escribir software matemático, pero el diseño del lenguaje de programación es algo completamente diferente. Al reutilizar un lenguaje existente, podemos centrarnos en aquellas cosas que importan: las matemáticas.

Una ventaja de SymPy es que es liviano. Además de ser relativamente pequeño, no tiene otras dependencias que Python, por lo que se puede usar en casi cualquier lugar fácilmente. SymPy, por otro lado, pretende ser un sistema independiente, con todas las características implementadas en SymPy.

Una última característica importante de SymPy es que se puede usar como una biblioteca. Muchos sistemas de álgebra computacional se centran en ser utilizables en entornos interactivos, pero si desea automatizarlos o ampliarlos, es difícil hacerlo. Con SymPy, puede usarlo fácilmente en un entorno interactivo de Python o importarlo en

su propia aplicación Python. SymPy también proporciona API para facilitar la extensión con sus propias funciones personalizadas [23]

2.18.2 Numpy

Según la página web oficial numpy, se define de la siguiente forma: NumPy es el paquete fundamental para la computación científica con Python. Contiene entre otras cosas:

- Un poderoso objeto de matriz N-dimensional.
- Funciones sofisticadas (de transmisión).
- Herramientas para integrar código C / C ++ y Fortran.
- Álgebra lineal útil, transformada de Fourier y capacidades de números aleatorios.

Además de sus usos científicos obvios, NumPy también se puede utilizar como un contenedor eficiente multidimensional de datos genéricos. Se pueden definir tipos de datos arbitrarios. Esto permite que NumPy se integre sin problemas y rápidamente con una amplia variedad de bases de datos.

NumPy tiene licencia bajo la licencia BSD, lo que permite su reutilización con pocas restricciones. [24]

2.18.3 Python.

“Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Y define este como un lenguaje multiparadigma, debido a que soporta orientación a objetos, programación imperativa y en menor medida programación funcional. Es interpretado de tipado dinámico y multiplataforma”.

Python es un lenguaje de propósito general, eso significa que no está orientado a un fin concreto. Con Python se pueden crear páginas web sin tener un alto conocimiento (con Javascript como un poderoso aliado), pero también hacer scripts o software para el sistema operativo Windows.

Python aún no se puede ejecutar en dispositivos móviles, pero se puede usar Kivy o Chaquopy para dicho propósito.

Comentando un poco más las características:

- Python es un lenguaje multiparadigma, porque soporta más de un paradigma (modelo de desarrollo de la programación).
- Es interpretado porque el intérprete va a traducir el código a medida que se necesita.
- Es de tipado dinámico porque permite la mutación (transformación) de variables.
- Es multiplataforma porque se puede ejecutar en cualquier sistema operativo.

Otras funciones o elementos propios de Python:

- De libre distribución.
- Gracias a su popularidad, existen una gran cantidad de librerías y funciones ya hechas, que se pueden utilizar gracias a su extensa biblioteca.
- Tiene soporte para múltiple variedad de bases de datos.
- Tiene un gran soporte gracias a su comunidad. Por ejemplo, la última versión de Visual Studio permite desarrollar en Python, o la comunidad de la página oficial de Python, dónde se pueden ver todas las actividades que hacen en el mundo. [25]

2.18.4 Spyder.

Spyder es un entorno científico gratuito y de código abierto escrito en Python, para Python, y diseñado por y para científicos, ingenieros y analistas de datos. Presenta una combinación única de la funcionalidad avanzada de edición, análisis, depuración y creación de perfiles de una herramienta de desarrollo integral con la exploración de datos, ejecución interactiva, inspección profunda y hermosas capacidades de visualización de un paquete científico. [26]

Componentes básicos del IDE Spyder

➤ Editor

El editor multi-idioma de Spyder integra una serie de poderosas herramientas listas para usar para una experiencia de edición eficiente y fácil de usar. Las características clave del editor incluyen resaltado de sintaxis; análisis de estilo y código en tiempo real; finalización bajo demanda, sugerencias de llamadas y funciones de definición de acceso; un navegador de funciones / clases, división horizontal y vertical, y mucho más.

➤ Consola IPython

La consola de IPython le permite ejecutar comandos e interactuar con datos dentro de los intérpretes de IPython.

➤ Explorador de variables

El Explorador de variables le permite navegar y administrar de forma interactiva los objetos generados al ejecutar su código.

➤ Gráficos

El panel de Gráficos muestra las figuras estáticas e imágenes creadas durante su sesión. Le mostrará gráficos de la Consola IPython , producidos por su código en el Editor o generados por el Explorador de Variables, lo que le permitirá interactuar con ellos de varias formas.

➤ Depurador

La depuración en Spyder es compatible mediante la integración con el ipdb depurador mejorado en la consola IPython . Esto permite que los puntos de interrupción y el flujo de ejecución se vean y controlen directamente desde la GUI de Spyder, así como con todos los comandos familiares de la consola IPython.

➤ Ayuda

Puede utilizar el panel de Ayuda para buscar, representar y mostrar documentación enriquecida para cualquier objeto con una cadena de documentos, incluidos módulos, clases, funciones y métodos. Esto le permite acceder fácilmente a la documentación directamente desde Spyder, sin tener que interrumpir su flujo de trabajo.

2.18.5 Utilidades

A continuación, se detalla las utilidades del dispositivo RTL SDR como muestra la Tabla 2.5

Tabla 2.5 Utilidades RTL-SDR

Utilidades	Escucha conversaciones de control de tráfico de aeronaves.
	Escaneado de conversaciones por radio abiertas.
	Análisis en busca de teléfonos inalámbricos y monitores para bebés.
	Seguimiento y recepción de datos de agencias meteorológica en globos meteorológicos.
	Recepción de los sensores de temperatura inalámbricos y sensores de medidor de potencia inalámbricos.
	Escucha VHF, UHF y parte de HF de radioaficionados.
	Recepción de TV analógica y digital.
	Sniffing de señales GSM.
	Usa el rtl-sdr en dispositivos Android como un escáner de radio portátil.
	Recepción de señales GPS.
	Radioastronomía.
	Usar rtl-sdr como un analizador de espectro.
	Escuchar la radio FM y decodificación de la información RDS.
	Señales de los terminales de datos móviles de taxis.
	Escucha de conversaciones policía, ambulancia, bomberos.

Fuente: Desarrollado por el investigador

2.19 Radio Definido por Software SDR

El SDR es un concepto que se viene utilizando desde principios de los años noventa. Su propósito original era la creación de un dispositivo capaz de emular muchas radios que trabajaban en diferentes frecuencias. Sin embargo, ha evolucionado y sigue haciéndolo hasta convertirse en una herramienta que tiene un uso mucho más amplio.

La radio definida por software es hoy una herramienta que ayuda a la industria de las comunicaciones inalámbricas y móviles en muchos aspectos.[27]

2.19.1 GNU Radio

GNU Radio es un kit de herramientas de software diseñado para permitir a los usuarios crear implementaciones SDR. Proporciona mecanismos y herramientas para crear aplicaciones personalizadas y también una gran cantidad de módulos de código fuente que pueden utilizarse como parte de una radio personalizada, así como ejemplos independientes. GNU Radio es utilizado por una gran comunidad de aficionados, investigadores académicos y empresas comerciales, especialmente para crear prototipos y probar posibles implementaciones.

El software de GNU Radio está dividido en módulos. Estos módulos pueden dividirse en dos grandes grupos. Los módulos que se encargan del procesamiento de la señal necesaria en el sistema están programados en C++. Estos módulos de procesamiento de señal pueden ser filtros de señal, ecualizadores, módulos FFT, etc. El otro grupo de módulos incluye el software necesario para interconectar estos módulos de procesamiento de señales y configurarlos según nuestras necesidades. Estos últimos se programan en Python y actúan como una especie de adhesivo que hace de todo el sistema una unidad [28]. Las posibilidades que ofrece Python para la configuración sin necesidad de compilar cada vez que se hace un cambio en un parámetro son muy convenientes para reconfiguraciones y pruebas rápidas. Los módulos de GNU Radio son capaces de operar con infinitos flujos de datos de un determinado tipo. Los tipos de datos más comunes que utilizaremos son los complejos, los cortos y los flotantes. Muchas aplicaciones de GNU Radio se mantienen en funcionamiento de forma continua, cuando los flujos de datos son infinitos. Si el flujo tiene un número finito de bytes la aplicación terminará una vez que los datos hayan sido consumidos fuera del sistema.

2.19.2 Requisitos de GNU Radio

Las aplicaciones de Radio Definida por Software están pensadas para sustituir la mayor parte del procesamiento de la señal en el sistema con software que se ejecutará en ordenadores básicos, pero todavía necesitan interfaces capaces de enviar y/o recibir datos que viajen de forma inalámbrica a través del aire. Esta sección explica lo que podemos hacer con GNU Radio, y por qué necesitamos hacerlo. En primer lugar, si tenemos una aplicación que se ejecuta infinitamente, esto significa que un procesador

normal tiene que tratar con la señal o señales en tiempo real. Cada aplicación tiene un cierto grado de complejidad de procesamiento de la señal los cuales se traducen en requisitos. Por ello, los requisitos son un factor crítico en SDR y, por tanto, en GNU Radio. El SDR ha ganado popularidad durante los últimos diez años debido al rápido aumento de la velocidad de cálculo en los ordenadores básicos.

Los requisitos de una aplicación sencilla de GNU Radio como la que hemos visto en la Figura 2.9 son básicamente cualquier ordenador, ya que no hay que realizar operaciones complicadas, como transformaciones de Fourier, y una tarjeta de sonido que será la encargada de actuar como interfaz de RF. Sin embargo, aplicaciones más complicadas, como la implementación realizada para este proyecto, tienen requerimientos computacionales mucho mayores, ya que utilizan muchas señales y algoritmos al mismo tiempo y son ellos los que deben trabajar en tiempo real. Las transmisiones a frecuencias más altas también requieren un hardware especial. GNU Radio está pensado para trabajar con un hardware adicional que se encarga de traducir la señal digital generada por el software en una señal analógica centrada en la frecuencia deseada. Para un pequeño número de aplicaciones que trabajan a frecuencias de hasta unos pocos KHz una tarjeta de sonido sería suficiente, pero para la mayoría de las demás aplicaciones se necesita un dispositivo externo para esta función. No hay un mercado muy grande para estos dispositivos, ya que no hay demasiada gente trabajando con SDR. Sin embargo, GNU Radio implementa módulos de software para uno de estos dispositivos, el USRP (Universal Software Radio Peripheral). Con un USRP podremos implementar radios que operen en frecuencias de hasta algunos GHz. En la siguiente subsección explicaré las características y posibilidades del USRP.

Otra cuestión relacionada con los requisitos computacionales de las aplicaciones de GNU Radio es la dualidad de utilizar código generado en Python y C++. Hay una parte de los usuarios de GNU Radio que consideran que el hecho de utilizar ambos lenguajes al mismo tiempo es una causa de pérdida de rendimiento. Según los desarrolladores de GNU Radio, Python sólo debería usarse para crear y configurar el gráfico, y luego el resto del trabajo debería ser realizado por los módulos de C++. Sin embargo, hay una iniciativa en curso que pretende permitir que las aplicaciones de GNU Radio se ejecuten sólo en C++. Esto puede acelerar ciertas aplicaciones, pero aparentemente,

no representaría una mejora muy grande para las aplicaciones bien programadas que utilizan Python y C++.

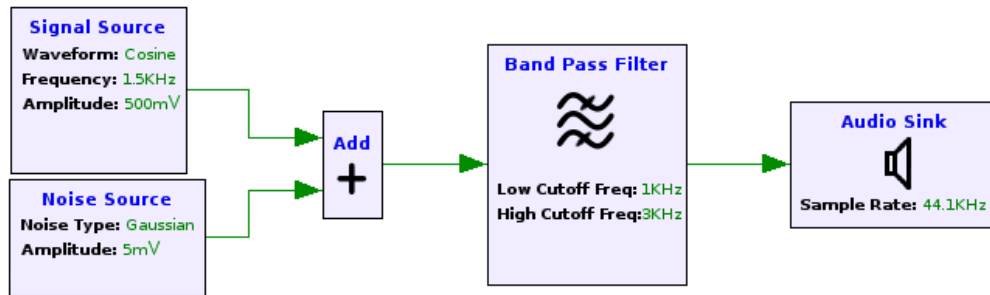


Figura 2.9 Representación gráfica de una aplicación GNU Radio
Fuente: Desarrollado por el investigador

2.19.3 Programación con GNU Radio

Para el desarrollo de una aplicación en GNU Radio se arrastran y se sueltan los bloques desde la biblioteca. Los bloques poseen en sus extremos puertos de entrada, salida o ambos según la función que realicen. Unos bloques pueden contener más de una salida o más de una entrada.

En GNU Radio se pueden agrupar los bloques de la siguiente manera:

- Fuentes: las bibliotecas que poseen generadores de señales, con formas de onda senoidales, cuadradas y otras; señales de audio y también señales que provienen del RTL-SDR.
- Procesado de señal: en este grupo se encuentran a todos los bloques que realizan un procesado de la señal. Por ejemplo: filtros, moduladores, operadores matemáticos, remuestreadores.
- Sumideros: conocidos como Sinks, permiten la visualización de los resultados del procesado de las señales de manera gráfica y/o sonora. También se incluyen los bloques encargados de la conexión con el RTL-SDR.
- Además de los tipos de bloques descritos, existen otros, los cuales permiten una mejor observación y manipulación de los parámetros en tiempo real, estos son sliders, notebooks, cuadros de texto.

Existe la forma de crear nuevos bloques para GNU Radio, la cual es programando directamente en Python, sin embargo, resulta una tarea ardua, por lo que, utilizando la interfaz gráfica de GNU Radio se pueden crear con mayor facilidad [29]. Cada vez que se inicie el programa, éste contará con una frecuencia de muestreo por defecto.

Ésta se encuentra en el bloque Variable con un parámetro llamado `samp_rate` cuyo valor por defecto es 32KHz. Esta frecuencia de muestreo se puede configurar y permite visualizar los resultados obtenidos de la aplicación de una mejor manera. GNU Radio posee la capacidad de trabajar con distintos tipos de datos según los requerimientos de una determinada aplicación. Estos tipos vienen diferenciados por el color en los extremos de los bloques. En la Figura 2.10 se muestra el bloque Throttle y los tipos de datos más usados con los que se puede trabajar en GNU Radio [30].

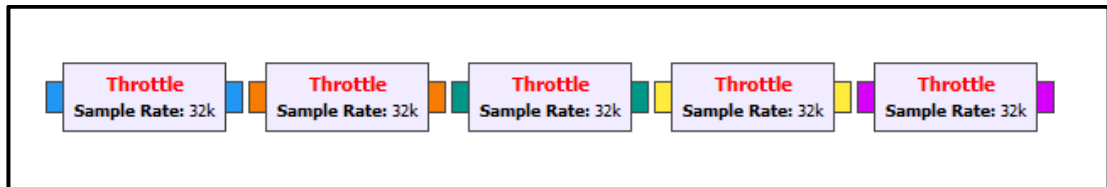


Figura 2.10 Tipos de datos mostrados en el bloque Throttle
Fuente: Desarrollado por el investigador

Como indica la Figura 2.10, los distintos tipos de datos más usados con los que se puede trabajar con los siguientes:

- a) Complex: datos formados por 8 bytes. Tiene parte real y parte imaginaria.
- b) Float: datos formados por 4 bytes de punto flotante.
- c) Int: datos de tipo entero con signo formado por 4 bytes.
- d) Short: datos de tipo entero formado por 2 bytes.
- e) Byte: datos formados por 1 byte (8 bits).

Sin embargo, estos no son los únicos tipos de datos que se pueden encontrar en GNU Radio, en la Figura 2.11 se muestra todos los tipos de datos. Estos se pueden encontrar haciendo clic en el menú Ayuda en la sección de Tipos.

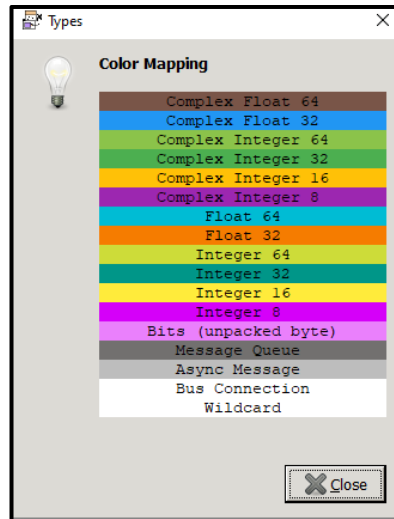


Figura 2.11 Tipos de datos completos en GNU Radio
Fuente: Desarrollado por el investigador

Se debe tomar en cuenta que no se pueden unir los bloques que tengan tipos de datos distintos. El programa muestra un error al hacerlo y el nombre de los bloques se muestran en color rojo. Sin embargo, existen bloques que permiten transformar un tipo de dato a otro. Por ejemplo, se puede transformar el tipo de dato complejo a tipo flotante usando el bloque Complex to Float. En la Figura 2.12 se muestran los bloques ubicados en la biblioteca, los cuales pueden convertir los distintos tipos de datos.

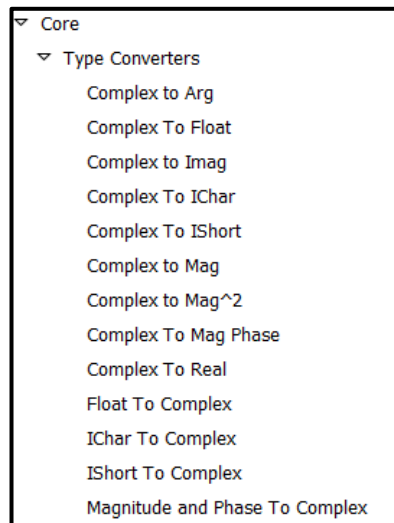














Figura 2.12 Convertidores de tipos de datos disponibles
Fuente: Desarrollado por el investigador

La barra de herramientas posee varios botones mostrados en la Tabla 2.6, dichos botones poseen distintas funciones para la programación en GNU Radio. En la Tabla 2.6 se describirán los más importantes para el desarrollo de aplicaciones.

Tabla 2.6 Descripción de botones de la barra de herramientas

Figura	Nombre	Descripción
	Error Flowgraph	Indica en dónde se encuentra el error en los bloques.
	Generate Flowgraph	Es el compilador de la aplicación.
	Execute Flowgraph	Una vez no haya errores, se encarga de ejecutar el diseño.
	Disable	Permite desconectar/deshabilitar un bloque sin necesidad de eliminarlo del diseño.
	Enable	Permite conectar el bloque antes deshabilitado para que se integre al diseño.
	Bypass	Permite ejecutar el diseño sin tener que deshabilitar o eliminar cierto bloque. Se pasa por alto a dicho bloque.
	Search	Es un buscador, permite la búsqueda de un bloque por su nombre.
	Left rotate	Permite rotar al bloque seleccionado 90° hacia la izquierda
	Right rotate	Permite rotar al bloque seleccionado 90° hacia la derecha
	Toggle Visibility	Permite esconder o mostrar los bloques que se hayan deshabilitado.
	Reload	Permite actualizar la biblioteca de GNU Radio si en ésta se han creado nuevas librerías.
	Open Hierarchical Block	Permite abrir el diagrama del bloque, si éste forma parte de un bloque jerárquico.

Fuente: Desarrollado por el investigador

2.19.4 Uso de variables en GNU Radio

Los bloques de los diagramas poseen parámetros que se pueden modificar. A veces se deben cambiar estos parámetros uno a uno, para evitar esta tarea de reemplazar los valores uno a uno, la cual puede resultar tediosa, se puede usar variables que simplemente colocando su ID en el parámetro a configurar en el bloque, éste ya se encuentra con el valor que tenga dicha variable [30]. En la Figura 2.13 se puede observar el bloque Variable, el cual contiene un ID cuyo parámetro se puede modificar.

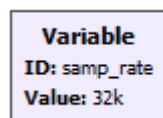


Figura 2.13 Bloque Variable

Fuente: Desarrollado por el investigador

Todo bloque posee su cuadro de propiedades al hacer clic derecho sobre él, luego se elige Propiedades. Otra manera de acceder a las propiedades es realizar doble clic sobre dicho bloque. En la Figura 2.14 Cuadro de propiedades de un bloque dado se muestra las propiedades de uno de los bloques de la librería de GNU Radio.

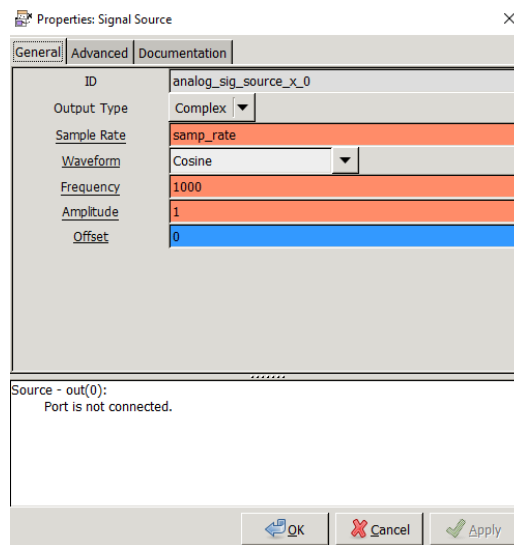


Figura 2.14 Cuadro de propiedades de un bloque dado.

Fuente: Desarrollado por el investigador

En la Figura 2.14 se ha reemplazado el parámetro Sample Rate por el ID de la variable samp_rate mostrada en la Figura 2.13, de este modo está apuntando al valor que contiene dicha variable.

2.19.5 Visualización de Resultados

En GNU Radio se puede observar los resultados mediante dos tipos de interfaces: QT GUI y WX GUI, ambas se pueden elegir en el bloque Options, el cual se encuentra presente en cada archivo nuevo que se ejecute. Además de elegir la interfaz en dicho bloque, se puede completar con la información relacionada con la aplicación a desarrollarse como: título, autor, descripción, tamaño del espacio de trabajo y el tipo de interfaz que se escogerá. Se debe tomar en cuenta que, si se elige un tipo de interfaz, también se debe escoger los bloques asociados a esa interfaz. Por ejemplo, no se puede escoger un bloque WX GUI si en Options está la opción de QT GUI [29]

2.20 Conexión de GNU Radio y el LTR-SDR

Para trabajar con GNU Radio y el LTR-SDR se procede a realizar las configuraciones. Una vez realizada con éxito la conexión de los equipos, se procede a usar los bloques de conexión ubicados en la biblioteca de GNU Radio, los cuales permiten la

comunicación entre el hardware y el software. Estos bloques se les denomina sources. osmosdr_source es la biblioteca que ayuda a la comunicación entre la computadora y todos los modelos de LTR-SDR. Osmosdr_source proporciona el control necesario para el transporte de las formas de onda desde y hacia el LTR-SDR. También ayuda a controlar algunos parámetros como ganancia, frecuencia de muestreo, frecuencia central de la tarjeta de radio del LTR-SDR [34]. Osmosdr_source permite recibir datos del LTR-SDR hacia el computador. En la Figura 2.15 se muestra el bloque que permite la conexión entre el computador y el LTR-SDR.

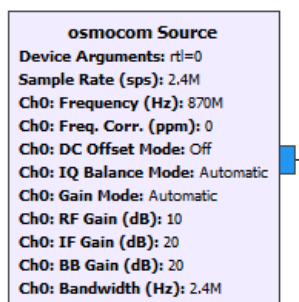


Figura 2.15 Bloque recepción para conexión del LTR-SDR

Fuente: Desarrollado por el investigador

Al realizar doble clic para mostrar el cuadro de propiedades se muestra las pestañas que permiten configurar el bloque según la necesidad de la aplicación. Las más importantes para el diseño de una aplicación son las pestañas de General, RF Options y Documentation. En esta última pestaña se puede encontrar información referente al bloque con el que se está trabajando. Es un tipo de ayuda que ofrece GNU Radio para el uso de los bloques.

En la pestaña de General se configura la parte inalámbrica, es decir, a qué frecuencia trabajará el RTL-SDR, con qué ancho de banda y la ganancia.

2.21 Sistema Operativo Raspberry Pi

Raspberry Pi es una computadora de placa única basada en Linux. Como cualquier otra computadora, la Raspberry Pi también usa un sistema operativo basado en Linux llamado anteriormente Raspbian y hoy en la actualidad Raspberry Pi OS.

Su distribución Debian fue perfeccionada para el hardware del Raspberry Pi, viene con más de 35 mil paquetes pre-compilados para la fácil instalación en el Raspberry Pi.

Linux es la combinación preferida para Raspberry Pi porque es de código abierto y gratuito. El sistema operativo se ejecuta desde una tarjeta flash SD, lo que permite cambiarlo instantáneamente su tarjeta SD. Está destinado a personas de todas las edades y niveles de habilidad, especialmente con fines educativos.

La placa dispone de una expansión de entrada / salida de uso general (GPIO) para Raspberry Pi la cual permite que cualquier periférico externo se conecte a este dispositivo. Esta placa permite controlar diodos emisores de luz (LED), servomotores, sensores y otros componentes electrónicos. La Raspberry Pi dispone de dos interfaces que nos permitirán acceder para controlar cosas.

El primero de ellos es una especie de interfaz estándar que casi se incluye en todas las computadoras modernas, llamada USB o bus serie universal. La segunda de estas interfaces en Raspberry Pi se llama GPIO, que significa Entrada y Salida de Propósito General. Esta parte de la Raspberry Pi permite al usuario controlar fácilmente todo tipo de dispositivos, lo que a su vez ayuda a implementar muchos experimentos electrónicos [30]. En este proyecto de investigación, el puerto USB se utiliza para conectar el dongle RTL-SDR.

ANALISIS DE HARDWARE

2.22 Receptor - Decodificador

2.22.1 Definición

El receptor - decodificador es un dispositivo que cumple las funciones de sintonización, demodulación y descomprensión de señales.

2.22.2 Características del receptor - decodificador

El decodificador recibe señales digitales y analógicas y separa las señales de video, audio y datos que fluyen a través de la señal portadora recibida por la antena. Además, para realizar algunas operaciones necesarias para convertir señales de radiofrecuencia a audio y video, primero modula la señal, la demodula o demultiplexa, descomprime o decodifica el código, y finalmente envía el audio, video y datos a la salida.

2.22.3 Historia de RDS

Los orígenes de la tecnología de radio definida por software están relacionados con el sector militar, o más precisamente con el proyecto SPEAKEasy desarrollado por el Departamento de Defensa de los EE. UU. a principios de la década de 1990. El proyecto fue inicialmente operado en el rango de frecuencia de 2 MHz a 200 MHz con 10 tipos de tecnología de comunicación inalámbrica.

En 1996, Joseph Mitola fundó el SDR Forum, ahora conocido como Wireless Innovation Forum. El foro, junto con otras empresas como la NASA, Motorola Solutions y MIT Lincoln Laboratory, tiene la tarea de llevar la tecnología de Radio Definido por Software y Radio Cognitiva al mercado, cabe destacar que Radio Cognitiva, es un paradigma de las comunicaciones inalámbricas en la cual tanto las redes como los mismos nodos inalámbricos cambian los parámetros particulares de transmisión o de recepción para ejecutar su contenido de forma eficiente sin interferir con los usuarios autorizados.

A mediados de 1997 el ejército americano a fin de terminar con la problemática de interoperabilidad que existía entre los diferentes sistemas de radio creó un programa para el desarrollo de sistemas de radiocomunicación, los cuales fueron creados para ser reconfigurables vía software, este programa fue denominado Joint Tactical Radio System, permitiendo especialmente a los aviones de combate tener la capacidad de realizar una transmisión convergente de voz, datos y video.

En 2000, se lanzó el primer proyecto no militar para radioaficionados, se estableció un foro para probar e informar sobre características recién descubiertas y nuevos problemas por resolver. Al mismo tiempo se inició el proyecto GNU 11 para la implementación del sistema RDS. El más famoso de ellos es una herramienta gratuita llamada GNU Radio.

Los avances en la plataforma de desarrollo y microelectrónica RDS han llevado el mercado de SDR a más de \$ 5,3 billones en 2007. Uno de los proyectos más ambiciosos es OpenBTS. Esto incluye el uso de Asterisk 12 para servicios VoIP con el fin de implementar estaciones base de telefonía GSM en el software.

Actualmente, la radio definida por software se puede adaptar a una variedad de situaciones, lo que la hace ideal para investigar y estudiar las comunicaciones inalámbricas sin la necesidad de modificar el hardware del dispositivo.

2.22.4 Radio definido por software (RDS)

Uno de los principales problemas de los sistemas de comunicación inalámbrica unidireccionales es que incorporan una plataforma de hardware estática diseñada para realizar una o más funciones específicas. Esto significa una inversión y complejidad adicionales cuando se necesitan nuevas funciones. O nueva tecnología.

En la actualidad, la demanda de equipos inalámbricos está creciendo exponencialmente debido tanto a la alta demanda de los usuarios como a la complejidad de las nuevas aplicaciones y sistemas de modulación, que requieren una tecnología de comunicación inalámbrica flexible.

La tecnología de radio definida por software (SDR) permite implementar componentes como mezcladores, amplificadores, transmisores y receptores, proporcionando flexibilidad al construir sistemas de control de comunicaciones inalámbricas. Los parámetros para estos componentes se pueden configurar dinámicamente en software en lugar de hardware, eliminando la necesidad de cambiar el hardware para cada sistema de comunicación inalámbrica implementado.

No existe un estándar oficial para la tecnología RDS, pero hay un grupo de trabajo llamado Wireless Innovation Forum que promueve estándares, especificaciones y difunde la tecnología RDS para una variedad de aplicaciones académicas, militares y comerciales. En colaboración con el grupo P1900.1 Electrical and Electronics Engineers (IEEE), el Wireless Innovation Forum ha elaborado la siguiente definición para establecer una definición coherente de la tecnología RDS y una visión clara: “Radio en el que parte o la totalidad de las funciones de la capa física son definidas por software.”

2.22.5 Arquitectura del RDS

El diagrama básico de la tecnología RDS se muestra en la Figura 2.16 Diagrama de bloques RDS. Consta de tres bloques de funcionamiento: la parte de RF (radiofrecuencia), la parte de FI (frecuencia intermedia) y la parte de banda base.

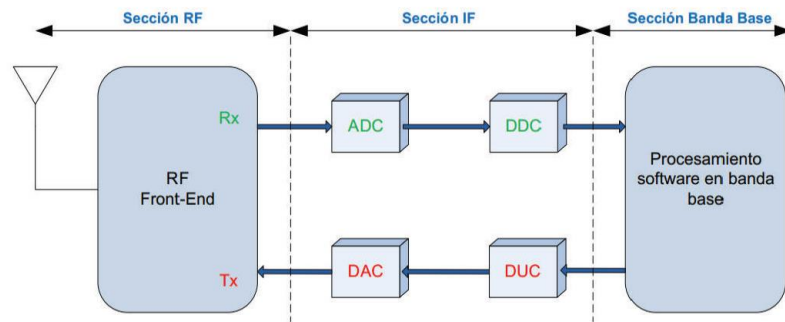


Figura 2.16 Diagrama de bloques RDS[31]

La parte de RF es responsable de enviar y recibir señales de radiofrecuencia. Para la recepción, convierte la señal a frecuencia intermedia, y para la transmisión modular, amplifica la señal a frecuencia intermedia.

En la sección IF se da lugar el concepto de Zero-IF, es decir, que la señal en RF es directamente convertida a la señal en banda base o viceversa. Además, los módulos ADC (convertidor de analógico a digital) y DAC (convertidor de digital a analógico) se utilizan para realizar conversiones de analógico a digital (recepción) y de digital a analógico (transmisión). El convertidor análogo-digital (ADC), como su nombre lo indica es el encargado de realizar el paso de señales analógicas a digitales. Para poder reconstruir fielmente la señal analógica a la salida del ADC, un parámetro importante en el convertidor es la elección de la frecuencia de muestreo, la cual, según la teoría de Nyquist debe ser igual o mayor al doble del ancho de banda de la señal en banda base.

El convertidor digital-análogo (DAC) presenta características similares al ADC, siendo capaz de recibir información digital en forma de una palabra de n bits y transformándola a una señal analógica en un determinado tiempo de muestreo.

El módulo DDC (Digital Down Converter) y el módulo DUC (Digital Up Converter) disminuyen o aumentan la frecuencia de muestreo entre la parte IF y la parte de banda base, respectivamente. Finalmente, la parte de banda base es responsable del procesamiento como la modulación / demodulación y el análisis espectral de la señal. En esta sección, la señal es puramente digital, por lo que el procesamiento solo tiene lugar a nivel de software. Esta es una característica que brinda a los dispositivos RDS la flexibilidad de implementar sistemas de radiocomunicación.

2.22.6 Componentes del receptor – decodificador

2.22.7 RTL-SDR

El módulo RTL-SDR que se utilizará para la realización del presente proyecto se muestra en la Figura 2.17. Este módulo consta principalmente de dos chips, el demodulador RTL2832U y el Sintonizador R820T. Figura 2.18.[32]



Figura 2.17 Módulo RTL-SDR. A [32]

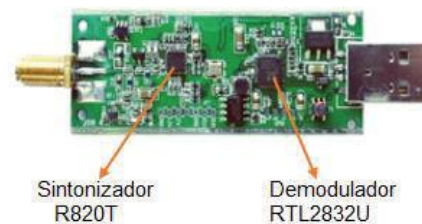


Figura 2.18 Módulo RTL-SDR. B [32]

2.22.8 Comparativa de dispositivos SDR

El módulo RTL-SDR de la Figura 2.17, fue seleccionado por el amplio rango de frecuencias de operación que presenta y principalmente por el bajo costo frente a otros módulos RDS, tal como se puede observar en la Tabla 2.7

Tabla 2.7 Precios de dispositivos RDS.

Dispositivo	Frecuencia de operación [MHz]	transmisión (Sí / No)	Precios [USD]
RTL-SDR	24 – 1766	NO	30
Airspy	24 – 1800	NO	100
SDPlay	0.1 – 2000	NO	149
HackRF	30 – 6000	SI	300
USRP B210	70 – 6000	SI	1100

Fuente: Desarrollado por el investigador

Entre otras características del módulo RTL-SDR se tienen las siguientes:

- La entrada para la antena es de tipo SMA hembra, con una impedancia de 50 ohmios y una pérdida menor a los 0.2 dB.
- La máxima frecuencia de muestreo que presenta es 3.2 MHz, aunque se recomienda que sea 2.8 MHz.
- Tiene una resolución de 8 bits.
- Puede conectarse a cualquier tipo de ordenador, gracias a su puerto USB 2.0.

El módulo RTL-SDR al poder barrer un amplio rango de frecuencias se lo utiliza en varios tipos de aplicaciones, entre ellas se pueden mencionar:

- Escuchar conversaciones de control de tráfico de aeronaves.
- Escuchar radio aficionado.
- Recepción y decodificación de señales GPS.
- Escuchar radio FM.
- Triangulación de la fuente de una señal.
- Búsqueda de fuentes de ruido de RF.
- Recepción de señales de sensores inalámbricos.
- Recepción de señales GSM.
- Decodificación de transmisiones de voz digital cifrada.
- Escuchar conversaciones de la policía y ambulancias.

2.22.9 Demodulador RTL2832U

El RTL2832U pertenece a la empresa Realtek, es un demodulador Multiplexado por División de Frecuencia Ortogonal Codificada (COFDM) de televisión digital terrestre DVB-T, con puerto USB 2.0. Incluye soporte de radio para FM, DAB y DAB+. El RTL2832U es el primer modulador y esto nos lleva al concepto de RTL-SDR, ya que es la pieza clave para que un receptor de TV pase a ser un receptor SDR.

2.22.10 Aplicaciones

- Dispositivos portátiles
- Dispositivos USB
- MiniCard

2.22.11 Tuner R820T

El R820T pertenece a la empresa Rafael Micro, es un sintonizador de silicio altamente integrado que se basa en un Amplificador de Bajo Ruido (LNA), mezclador, VGA, regulador de voltaje y filtro de seguimiento; eliminando la necesidad de filtros externos. Gracias a la arquitectura LNA, el R820T ofrece el menor costo y una alta solución de rendimiento para aplicaciones de televisión digital.

2.22.12 Características

Actualmente, su campo de desarrollo se ha ido incrementando considerablemente, debido al rendimiento superior que ofrece en linealidad, sensibilidad, inmunidad de canal adyacente, etc.

En la

Tabla 2.8 Características del chip R820T. se muestran las principales características del chip sintonizador.

Tabla 2.8 Características del chip R820T.

Parámetro	Valor Típico	Unidad
Voltaje de entrada	3.3	V
Consumo de corriente	<178	mA
Temperatura de operación	25	°C
Rango de Frecuencia	42-1002	MHz
Figura de Ruido	3.5	dB
Potencia de entrada máxima	+10	dBm
Rechazo de imagen	65	dBc13

Fuente: Desarrollado por el investigador

2.22.13 Aplicaciones

- Televisión Digital Terrestre.
- PCTV, mini-card, y periféricos USB.
- Set Top Box.
- Reproductor de medios portátil.

2.23 Raspberry Pi

2.23.1 Definición

Raspberry Pi o RPI es un ordenador de placa reducida de bajo costo y de bajo consumo de energía, diseñado por la fundación Raspberry Pi Foundation como un ordenador de uso general económico y de altas prestaciones orientado a la enseñanza que posee las mismas características y funcionalidades de un ordenador de mesa.

2.23.2 Generalidades técnicas de un Raspberry Pi

El Raspberry Pi es una placa base que posee una dimensión de 85 x 53 milímetros, en su interior encontramos un chip integrado Broadcom BCM2711, el cual contiene un procesador ARM Cortex de hasta 1.5 GHz de velocidad, además un procesador gráfico VideoCore IV y una memoria RAM de 4GB.

Para su funcionamiento se necesita de un medio de almacenamiento que serán las tarjetas de memoria SD o microSD, luego conectarlo a la corriente utilizando cualquier cargador microUSB y si lo deseamos, guardarlo todo utilizando una carcasa. En función del modelo se dispondrá de más o menos opciones de conexión, aunque siempre se tendrá al menos un puerto de salida de video HDMI y otro de tipo RCA, un mini jack de audio y puertos USB para conectar el teclado y ratón.

Para la conexión de red dispone de un puerto Ethernet 10/100 para enchufar el cable RJ-45 directo al router o utilizar cualquier adaptador inalámbrico WiFi compatible. Para comunicarse con el exterior, tanto para activar elementos como para leer el estado de los mismos consta de un bus de expansión llamado General Purpose Input/Output (GPIO), el cual posee 40 pines, trabaja con 3.3V y puede suministrar hasta 16 mA. (Raspberry Shop, 2011)

2.23.3 Modelos

Existen seis modelos de Raspberry Pi según la Tabla 2.9 que se detalla a continuación:

Tabla 2.9 Especificaciones Técnicas de los modelos de Raspberry PI.

	RPI Modelo A	RPI Modelo A+	RPI Modelo B	RPI Modelo B+	RPI 2 Modelo B	RPI 4 Modelo B
SoC	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2711
CPU	ARMv6 700 MHz.	ARMv6 700 MHz.	ARMv6 700 MHz.	ARMv6 700 MHz.	ARMv7 ARM Cortex-A7 4 núcleos 900 MHz.	ARMv8 ARM Cortex- A72 4 núcleos 1.5 GHz.
GPU	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0
RAM	256 MB LPDDR SDRAM 400 MHz.	256 MB LPDDR SDRAM 400 MHz.	512 MB LPDDR SDRAM 400 MHz.	512 MB LPDDR SDRAM 400 MHz.	1 GB LPDDR2 SDRAM 450 MHz.	4 GB LPDDR4 SDRAM 3200 MHz.
USB 2.0	1	1	2	4	4	2
USB 3.0	0	0	0	0	0	2
Salidas de vídeo	HDMI 1.4 @ 1920x1200 píxeles	HDMI 1.4 @ 1920x1200 píxeles	HDMI 1.4 @ 1920x1200 píxeles	HDMI 1.4 @ 1920x1200 píxeles	HDMI 1.4 @ 1920x1200 píxeles	2 × puertos micro (hasta 4kp60
Almacenamiento	SD/MMC	microSD	SD/MMC	microSD	microSD	microSD
Ethernet	No	No	Sí, 10/100 Mbps	Sí, 10/100 Mbps	Sí, 10/100 Mbps	Sí, 10/100 Mbps
Tamaño	85,60 x 56,5 mm	65 x 56,5 mm	85,60 x 56,5 mm	85,60 x 56,5 mm	85,60 x 56,5 mm	85,60 x 56,5 mm
Peso	45 g.	23 g.	45 g.	45 g.	45 g.	45 g.

Fuente: Desarrollado por el investigador

La Raspberry Pi 4 es sin duda una de las mejores que existe en el mercado debido a su amplia gama de características proporcionadas por el fabricante las cuales se acoplan a las necesidades de la simulación de funciones en un entorno gráfico y procesamiento de señales en tiempo real lo cual hace ideal para dicho prototipo como muestra la Tabla 2.9.

2.23.4 Aplicaciones

Gracias al Raspberry Pi se pueden realizar varios proyectos, entre las aplicaciones más interesantes que se pueden llevar a cabo están:

- Montar un Servidor de Acceso a la Red (NAS) para poder compartir archivos en red.
- Montar un Media Center para reproducir películas en HD, ya que reproduce videos de alta resolución.
- Emulador de juegos, ya que es capaz de ejecutar consolas de juegos.
- Robótica, para conectar sensores y acoplarlos a diferentes sistemas.
- Domótica, para la automatización de una casa por medio de sensores y controlarlos por computadoras o celulares.

Qué es GNU Radio

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 utiliza ampliamente en entornos de investigación, industria, academia, gobierno y aficionados para respaldar tanto la investigación de comunicaciones inalámbricas como los sistemas de radio del mundo real.

Qué es Software Radio

En resumen, una radio de software es un sistema de radio que realiza el procesamiento de señales requerido en software en lugar de utilizar circuitos integrados dedicados en hardware. El beneficio es que, dado que el software se puede reemplazar fácilmente en el sistema de radio, el mismo hardware se puede usar para crear muchos tipos de radios para muchos estándares de comunicaciones diferentes; por lo tanto, una radio de software se puede utilizar para una variedad de aplicaciones.

La radio definida por software (SDR) es un sistema de comunicación por radio en el que los componentes que se han implementado tradicionalmente en el hardware (por ejemplo , mezcladores , filtros , amplificadores , moduladores / demoduladores , detectores , etc.) se implementan en su lugar mediante software en una computadora personal o sistema integrado .^[1] Si bien el concepto de SDR no es nuevo, las capacidades en rápida evolución de la electrónica digital hacen prácticos muchos procesos que antes solo eran teóricamente posibles.

Un sistema SDR básico puede consistir en una computadora personal equipada con una tarjeta de sonido u otro convertidor de analógico a digital , precedido por algún

tipo de interfaz de RF . Cantidades significativas de procesamiento de señales se transfieren al procesador de propósito general, en lugar de realizarse en hardware de propósito especial (circuitos electrónicos). Tal diseño produce una radio que puede recibir y transmitir protocolos de radio muy diferentes (a veces denominados formas de onda) basándose únicamente en el software utilizado.

Las radios de software tienen una utilidad significativa para los servicios militares y de telefonía celular , los cuales deben servir a una amplia variedad de protocolos de radio cambiantes en tiempo real. A largo plazo, los defensores como el Wireless Innovation Forum esperan que las radios definidas por software se conviertan en la tecnología dominante en las comunicaciones por radio. Los SDR, junto con las antenas definidas por software, son los habilitadores de la radio cognitiva .

Una radio definida por software puede ser lo suficientemente flexible como para evitar las suposiciones de "espectro limitado" de los diseñadores de tipos anteriores de radios, de una o más formas que incluyen:

- Las técnicas de espectro ensanchado y banda ultra ancha permiten que varios transmisores transmitan en el mismo lugar en la misma frecuencia con muy poca interferencia, normalmente combinada con una o más técnicas de detección y corrección de errores para corregir todos los errores causados por esa interferencia.
- Las antenas definidas por software se "bloquean" adaptativamente en una señal direccional, de modo que los receptores puedan rechazar mejor la interferencia de otras direcciones, lo que le permite detectar transmisiones más débiles.
- Técnicas de radio cognitiva : cada radio mide el espectro en uso y comunica esa información a otras radios que cooperan, de modo que los transmisores puedan evitar interferencias mutuas seleccionando frecuencias no utilizadas. Alternativamente, cada radio se conecta a una base de datos de geolocalización para obtener información sobre la ocupación del espectro en su ubicación y, de manera flexible, ajusta su frecuencia de operación y / o potencia de transmisión para no causar interferencia a otros servicios inalámbricos.
- Ajuste dinámico de la potencia del transmisor, basado en la información comunicada desde los receptores, reduciendo la potencia de transmisión al

mínimo necesario, reduciendo el problema de cerca-lejos y reduciendo la interferencia a otros, y extendiendo la vida útil de la batería en equipos portátiles.

- Red de malla inalámbrica donde cada radio adicional aumenta la capacidad total y reduce la energía requerida en cualquier nodo. ^[3] Cada nodo transmite usando solo la potencia necesaria para que el mensaje salte al nodo más cercano en esa dirección, reduciendo el problema de cerca y lejos y reduciendo la interferencia a otros.

2.24 Introducción ADF4351

Teniendo en cuenta los requisitos cada vez más altos de los sistemas de comunicación modernos para la pureza espectral, el rango de frecuencia y el ruido de fase de la fuente de frecuencia, se propone una nueva fuente de frecuencia sintonizable de alto rendimiento. Primero, se analizan los principios básicos y las características operativas del sintetizador de frecuencia ADF4351 PLL multiplicador de frecuencia se combinan para generar una fuente de frecuencia de banda ancha, mientras tanto, por medio de un atenuador digital y LNA, se logra una potencia de salida ajustable, así como potencia ajustable, bajo ruido y bajo de fase.

2.25 Sintetizador de Frecuencias

Es un generador de señales cuya frecuencia de salida variable se deriva de una frecuencia standard extremadamente estable, generalmente un oscilador a cristal de precisión que infiere al circuito una gran estabilidad y precisión en frecuencia. También se considera como la generación de señales analógicas utilizando técnicas digitales con operaciones aritméticas binarias (suma, resta, multiplicación y división). Diversos circuitos se encargan de procesar esta señal de referencia que produce una frecuencia de salida que resulta ser un múltiplo entero o fraccionario de la frecuencia original. Por lo tanto, el conjunto de frecuencias de salida es discreto y su número, rango y resolución depende de la topología del sintetizador. Según su funcionamiento los sintetizadores se dividen se dividen en Sintetizador Directo Analógico, Sintetizador Digital (Síntesis Digital Directa - DDS) y Sintetizador Indirecto (con Lazo de enganche de Fase - PLL). En sus orígenes los sintetizadores se utilizaban en las comunicaciones de alta frecuencia, entre 50 MHZ a 10GHZ y luego en baja frecuencia.[33]

Características comunes a distintos sintetizadores:

1.- Ajuste digital de frecuencia por medio de llaves divisoras o divisores programables como el preescaler. El ajuste de f se hace por saltos, los que pueden ser tan pequeños como se necesite, habitualmente saltos más pequeños implican una mayor complejidad en la circuitería, aunque con el desarrollo actual de integración esto se minimiza. Para posibilitar un ajuste continuo de la frecuencia se incluye un interpolador que actúa entre los escalones mínimos de ajuste digital.[33]

2.- Pureza Espectral, se logra con una elevada supresión de ruido y de señales espurias. Un sintetizador debe producir una sola frecuencia de salida sin distorsión u otras impurezas espectrales. [33]

3.- Oscilador Patrón controlado por cristal de cuarzo muy estable. Es clave porque de su estabilidad depende la exactitud de todas las frecuencias a obtener. [33]

4.- Posibilidad de recibir la inyección de una señal patrón externa de mayor exactitud que la del oscilador interno. [33]

5.- Generan señales de RF con modulación de AM, FM, ASK, FSK, pulsos, etc.

6.- Elevada resolución de frecuencia: es considerada como el más pequeño cambio de frecuencia que puede aceptar un sintetizador en un amplio rango que va desde algunos HZ hasta cientos de MGHZ. Normalmente la resolución es una potencia de 10, o sea 1, 10, 100HZ, etc. [33]

7.- La precisión de frecuencia es la diferencia entre la frecuencia real y la frecuencia deseada o marcada. No debe confundirse precisión con resolución. Un sintetizador puede tener muchos dígitos y una resolución de frecuencia muy estrecha, lo cual no significa que la frecuencia real sea la indicada por el último dígito. [33]

8.- Tiempo de enganche: es la cantidad de tiempo desde el instante en que se solicita una nueva frecuencia hasta el momento en que la salida ha asumido esa frecuencia o se obtiene la misma. El tiempo de enganche puede variar desde menos de 1 ciclo de la frecuencia de salida hasta varios segundos para los sintetizadores de elevada precisión. La tecnología actual hace que esto se disminuya al orden de los milisegundos, segundos o nanosegundos según el tipo de síntesis que se realice. Esta característica es crítica cuando se requiere un cambio de frecuencia muy rápido, como es el caso de utilizar un sintetizador para ajustar frecuencias de un receptor y de un transmisor. Cuando se cambia de recepción a transmisión el sintetizador debe suministrar en forma rápida la frecuencia deseada. [33]

9.- Estabilidad de frecuencia: Es algo realmente destacado y está asociado, en general, a un oscilador a cristal. La estabilidad del sintetizador es la estabilidad del oscilador a cristal. Hay diseños en los que se utiliza más de un oscilador a cristal u otros osciladores no controlados a cristal. En todos los casos la estabilidad es una función de los osciladores involucrados. 10.- Señal radiada por el sintetizador: se generan numerosas señales internas de diferentes frecuencias y pueden producir interferencias no deseables. Para evitar esto se hace uso de un blindaje especial, todas las líneas de programación, alimentación y de otros controles están desacopladas y blindadas. La salida del sintetizador se pasa por un Filtro Pasa banda para eliminar cualquier señal no deseada. [33]

2.26 Características módulo generador de señales ADF4351

El módulo generador de señales, fabricado por la empresa Analog Devices, tiene un chip compacto y de bajo consumo ADF4351. Para determinar la frecuencia y potencia se dispone de dos pines, un DATA y un CLK con los cuales se envía la información desde un microcontrolador con los pines MOSI y CLK del bus SPI. Como se observa en la Figura 2.19, el módulo cuenta con 4 pines para salida de la señal A+, A- y B+, B-. Su pantalla OLED permite tener información gráfica del nivel de potencia y la frecuencia; se puede energizar a través de un jack DC 5 V o con un puerto mini USB tipo B. Tiene un pequeño switch on/off para su funcionamiento, 5 botones OK (aceptar), L (izquierda), R (derecha), UP (arriba), DW (abajo), que permiten al usuario modificar los parámetros de frecuencia y potencia.

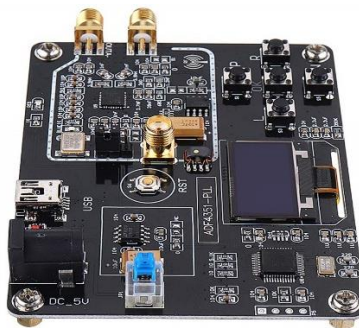


Figura 2.19 Generador de señales ADF4351. [34]

El ADF4351 tiene un oscilador controlado por voltaje integrado (VCO) con una frecuencia de salida fundamental que va desde 2200 MHz a 4400 MHz. Además, los circuitos de división por 1 / -2 / -4 / -8 / -16 / -32 / -64 permiten al usuario generar frecuencias de salida de RF tan bajas como 35 MHz. Para aplicaciones que requieren

aislamiento, la etapa de salida de RF se puede silenciar. La función de silencio se puede controlar tanto mediante pin como mediante software. También está disponible una salida de RF auxiliar, que se puede apagar cuando no se usa.[34]

En la Tabla 2.10 se muestran detalles técnicos útiles proporcionados por el fabricante para crear las condiciones óptimas de funcionamiento del módulo generador de señales ADF4351. Además, se proporciona los rangos de funcionamiento de potencia y frecuencia disponibles en las configuraciones del módulo.

Tabla 2.10 Especificaciones técnicas generador de señales ADF4351

PARÁMETRO	VALOR
Voltaje de operación	5 V
Corriente	130 mAh
Potencia máxima	5 dB
Potencia mínima	2 dB
Frecuencia máxima	4.4 GHz
Frecuencia mínima	35 MHz
Error de frecuencia	± 1 kHz

Fuente: Desarrollado por el investigador

2.27 Comparativa tipos de generador de frecuencia:

Hoy en día en el mercado de la tecnología existe una gran variedad de generados de frecuencia las cuales se detalla la más relevantes:

- **Koolertron DDS:** es generador de forma de onda arbitraria, adopta un circuito integrado FPGA a gran escala y un microprocesador MCU de alta velocidad. El circuito interno adopta el oscilador de cristal activo como punto de referencia. Entonces la estabilidad de la señal se refuerza enormemente. La salida de frecuencia de la onda sinusoidal puede ser de hasta 15 MHz. Frecuencia de muestreo de 200MSa / s. Tiene 60 posiciones para guardar formas de onda definidas por el usuario. La longitud de la forma de onda de cada uno es de 2048 y la resolución vertical es de 14 bits.
- **FG-100 DDS Funtion Generator:** El generador de señal DDS 1HZ-500KHZ es pequeño, pero tiene muchas funciones, fácil de operar y llevar. La fuente de alimentación puede ser un adaptador CA/CC o una batería de litio.

- Se puede utilizar con osciloscopio para probar y depurar circuito electrónico, la respuesta de frecuencia del amplificador de audio y respuesta de impulso, también puede calibrar el factor de tiempo de barrido del osciloscopio.
- **Onyehn XR2206:** Se considera como un generador de alta precisión, puede generar formas de onda seno/triángulo/cuadrado en un rango de frecuencia de salida: 1 Hz-1 MHz y frecuencia y amplitud ajustables.
- **RF Signal Generator ADF4351:** El módulo tiene la capacidad de generar frecuencia de salida de 35MHz-4.4GHz, funciones básicas como frecuencia puntual, barrido de frecuencia, salto de frecuencia. Dispone de una buena interfaz de interacción con pantalla OLED para que los resultados sean claramente visibles.

Por lo tanto, se ha escogido para el prototipo el módulo generador de frecuencia ADF4351 debido a sus características en frecuencia como en su fácil manipulación.

CAPITULO III

MARCO METODOLÓGICO

3.1 Ubicación

El presente trabajo investigativo se lo implemento para el laboratorio de la FISEI de la Universidad Técnica de Ambato

3.2 Equipos y materiales

Las técnicas que se llevaron a cabo en el presente trabajo de investigación se basaron en la aplicación de estudios actualizados, libros de consulta, apegados en software libre, además de un estudio minucioso de las necesidades prácticas que tiene la FISEI y su aportación a dicha necesidad.

3.3 Tipo de investigación

El tipo de investigación aplicada al presente proyecto es explicativa, ya que se fundamenta en teorías, investigaciones e implementaciones que se muestran en artículos científicos de revistas, tesis doctorales y libros basados en publicaciones de alto impacto, la información obtenida sirve de base para fundamentar el presente trabajo de investigación e implementación, permitiendo conocer las herramientas y tecnologías actualizadas de una simulación de aplicaciones con series trigonométricas de Fourier para el área de comunicaciones.

3.4 Hipótesis - pregunta científica – idea a defender

¿La implementación de un prototipo simulador de aplicaciones con series trigonométricas de Fourier para el área de comunicaciones en la carrera de Ingeniería en Electrónica y Comunicaciones de la Universidad Técnica de Ambato maximizara el conocimiento teórico y práctico?

3.5 Población o muestra:

Para la investigación se aplica el muestreo no probabilístico de tipo discrecional para el desarrollo de la propuesta; permitiendo al investigador a juicio personal y de disponibilidad elegir el área o extensión de terreno a ser implementado.

3.6 Recolección de información:

Estos instrumentos son un medio de guía para poder realizar la investigación, y poder obtener información de diferentes puntos de vista o mejor dicho aplicando las

diferentes técnicas que existen para poder observar y evaluar el área de estudio de nuestra investigación.

3.7 Variables respuesta o resultados esperados

Los resultados esperados en este proyecto de investigación es que mediante el desarrollo e implementación de un sistema integrado portátil de simulación de aplicaciones con series trigonométricas de Fourier para el área de comunicaciones se pueda ver graficado lo requerido de acuerdo a los parámetros dados por el usuario y sea visualizado a través de una interfaz portátil, la cual pretende que se haga uso del mismo en las laboratorios de la FISEI de la Universidad Técnica de Ambato cada que se lo requiera y así fortalecer el conocimiento práctico y científico.

CAPITULO IV

RESULTADOS Y DISCUSIÓN

Características técnicas del prototipo

En la siguiente Tabla 4.1 se muestran las características técnicas del prototipo desarrollado e implementado.

Tabla 4.1 Características técnicas del prototipo

Tipo de pantalla	LCD TFT Touch
Tamaño de pantalla	7"
Resolución de pantalla	1024x600 pixeles
Voltaje	5 VDC
Amperaje	2.5 A
Software	Libre (Python, RTL-SDR, GNU Radio)
Tamaño del prototipo	18.75 x 14.5 x 6.5 cm
Conector para Antena	SMA
Demodulador	RTL2832U
Conexiones Externas	USB

Fuente: Desarrollado por el investigador

Componentes del prototipo:

- 1 Raspberry Pi 4 Model B
- 1 Demodulador RTL2832U
- 1 Sintonizador R820T
- 1 Antenas
- 1 Cable SMA
- 1 Cable USB

Arquitectura del prototipo en base al diagrama de bloques de un analizador de espectro y series de Fourier.

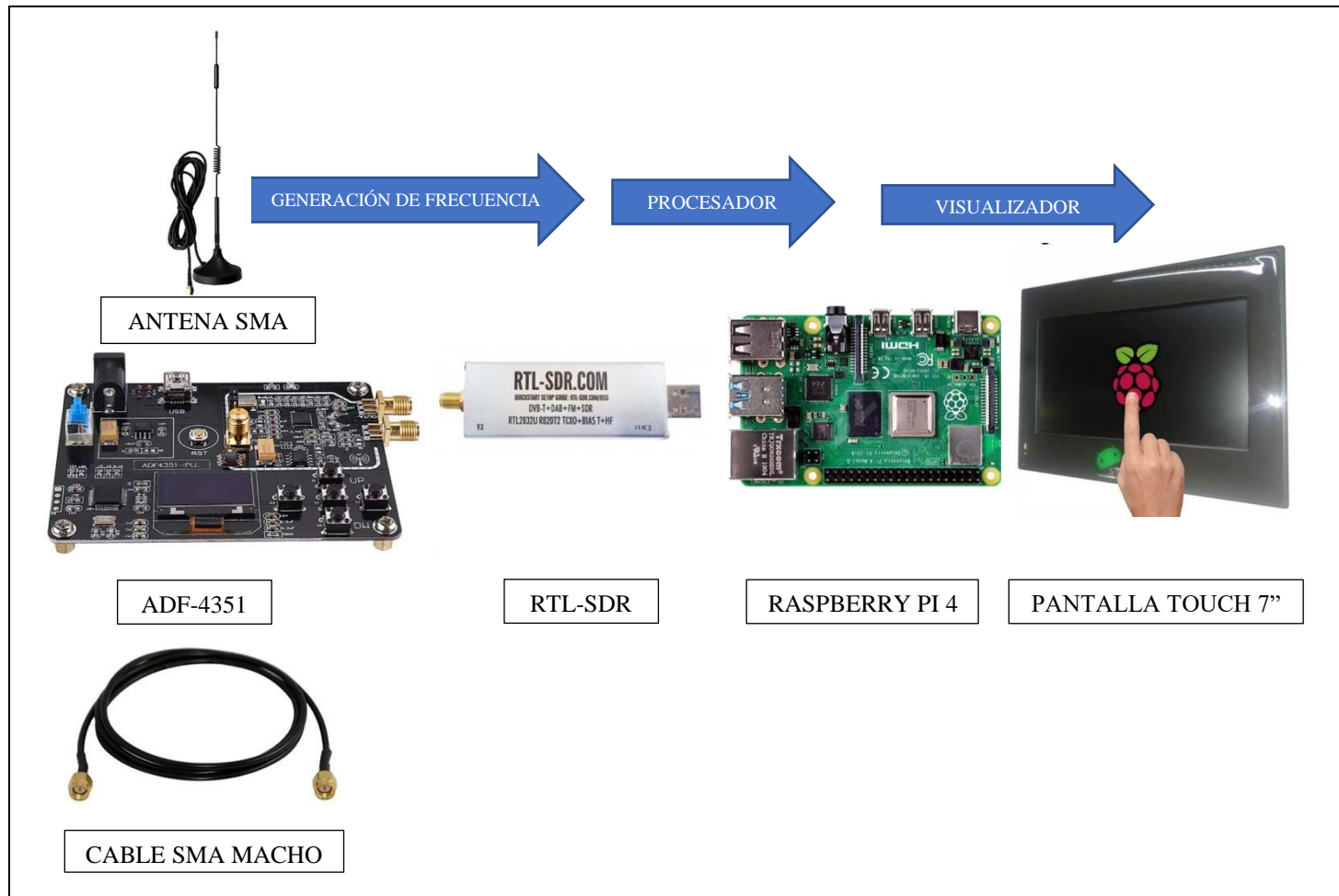


Figura 4.1 Arquitectura del analizador de espectro.

Fuente: Desarrollado por el investigador

Tabla 4.2 Descripción de la arquitectura del prototipo.

Antena	Receptor Input
Cable	Enlace de conexión entre dispositivos
Decodificador RTL2832U y Raspberry Pi 4	Contiene mezclador, detector, oscilador, sintonizador, filtros, demodulador, amplificador y ganancia todo en software.
Pantalla LCD TFT	Display

Fuente: Desarrollado por el investigador

La arquitectura básica de cualquier ordenador, en este caso el Raspberry Pi está compuesto por un procesador, memoria RAM, disco duro, dispositivos de entrada/salida y el software. Figura 4.2

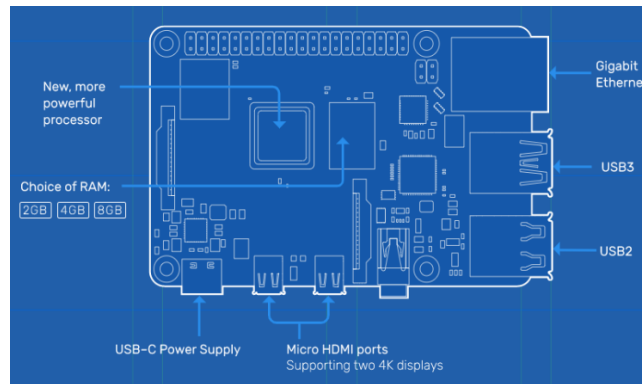


Figura 4.2 Arquitectura de un Raspberry Pi. [35]

La Unidad Central de Procesamiento (CPU) se encarga del funcionamiento de todos los elementos del ordenador, dentro de su estructura constan la Unidad de Control, la Unidad Aritmética Lógica y los Registros Internos, que es el lugar donde se almacenan los resultados de las instrucciones, en el Raspberry Pi es la tarjeta SD la que realiza el papel de disco duro.

La memoria almacena los datos que el CPU va a ejecutar, en el caso del Raspberry Pi 4 la memoria RAM es de 4GB. Los dispositivos de entrada/salida necesitan su propio entorno de funcionamiento para poder interactuar con el usuario ya que son componentes externos y de tecnología distinta.

El Raspberry Pi está diseñado para ejecutar el sistema operativo Linux que soporta procesadores ARM, la distribución más estable para el usuario es Raspbian, pero existen otras como Debian, Pidora, Fedora Remix y Arch Linux, las cuales cada una de ellas tienen algo en común: son de código abierto.

4.1 Sistema de Control

El sistema con el que se controla es la pantalla táctil o touch screen, que es la superficie por la cual desplazamos un dedo o puntero para controlar el movimiento del cursor en la pantalla, esto es, imita las funciones de un mouse, adicional a ello se ha incorporado un teclado y mouse inalámbrico para buscar una mejor comodidad al momento ingresar datos. Las pantallas táctiles resistivas pueden ser usadas con cualquier objeto, son económicas, fiables y versátiles.

La pantalla táctil está hecha de dos capas de material conductor transparente, con una separación entre ambas capas y una pequeña resistencia a la corriente eléctrica. Al entrar en contacto con la capa exterior, el sistema electrónico detecta el contacto entre las dos capas conductoras y mide la resistencia para calcular el punto de contacto.

Los scripts configurados en el prototipo son el software de este proyector de titulación que funciona una vez que RTL-SDR esté instalado en el Raspberry Pi 4, se puede cambiar la frecuencia central, la frecuencia de muestreo y la potencia. Muestra en la pantalla TFT el espectro de RF en tiempo real de la zona sintonizada y adicionalmente puede mostrar el espectro en forma de cascada como muestra la Figura 4.3.

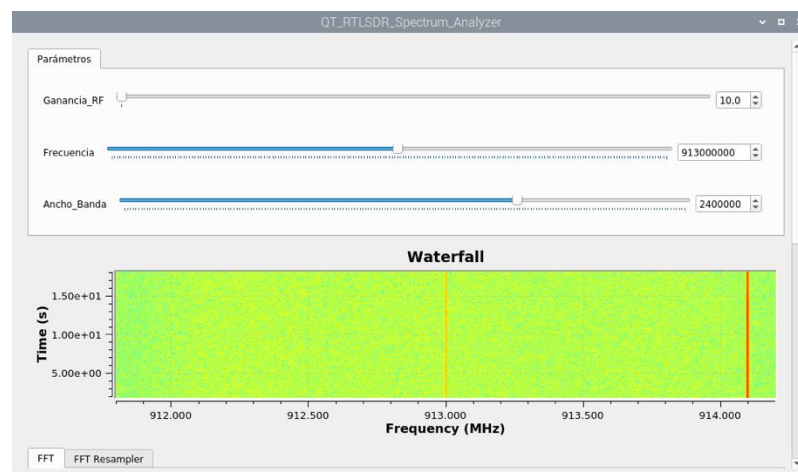


Figura 4.3 Espectro en forma de cascada.

Fuente: Desarrollado por el investigador

GNU Radio: es la herramienta de desarrollo que provee de bloques de procesamiento de señal para implementar sistemas de radio definida por software, las aplicaciones se construyen mediante un entorno gráfico GNU Radio Companion o Python.

4.2 Presentación del Prototipo

Con todos los requerimientos y características tomadas a consideración en el cumplimiento de funciones de recepción para señales en el dominio de la frecuencia, se establece algunos dispositivos tanto software como hardware para forman un conjunto funcional y flexible, el cual tiene como unidad central de procesos un ordenador de placa reducida como Raspberry Pi, siendo soporte en los demás materiales.

El ordenador de placa reducida tiene instalado diferentes softwares, instalados para cumplir funciones de SDR y FFT, además posee conexiones establecidas como las siguientes: se conecta mediante USB un receptor SDR, el mismo conectado a una antena por un conector SMA, otro dispositivo conectado es la pantalla LCD a través de los puertos GPIO, la cual posee como características la comunicación y el control del touch screen.

Además, se puede establecer una conexión directa a través del protocolo de comunicación USB con algún computador para un análisis más exhaustivo, esta conexión posee la facilidad de permitir conexión a alta velocidad.

4.2.1 Conexión Física del Prototipo

El prototipo de analizador de espectro consta de un ordenador de placa reducida, el cual se alimenta mediante la fuente de alimentación de 5V a 2.5A. En los puertos USB se conectan el receptor Wireless USB del teclado y mouse, el decodificador RTL SDR con su respectivo cable SMA, mientras que a los puertos GPIO se conecta la pantalla LCD para la visualización del espectro receptado. En el otro extremo se encuentra el generador de señales ADF listo para ser conectada mediante cable SMA al equipo para su respectivo análisis tal y como muestra la Figura 4.4.



Figura 4.4 Componentes del analizador de espectro.

Fuente: Desarrollado por el investigador

4.2.2 Implementación de prototipo en GNU RADIO

Existe un bloque: Throttle, el cual limita la tasa de datos a la especificada por el programa, por lo tanto, el diagrama diseñado no consumirá la totalidad de los recursos del CPU y evitará que el computador se quede sin responder como se muestra y se señala en la Figura 4.5.

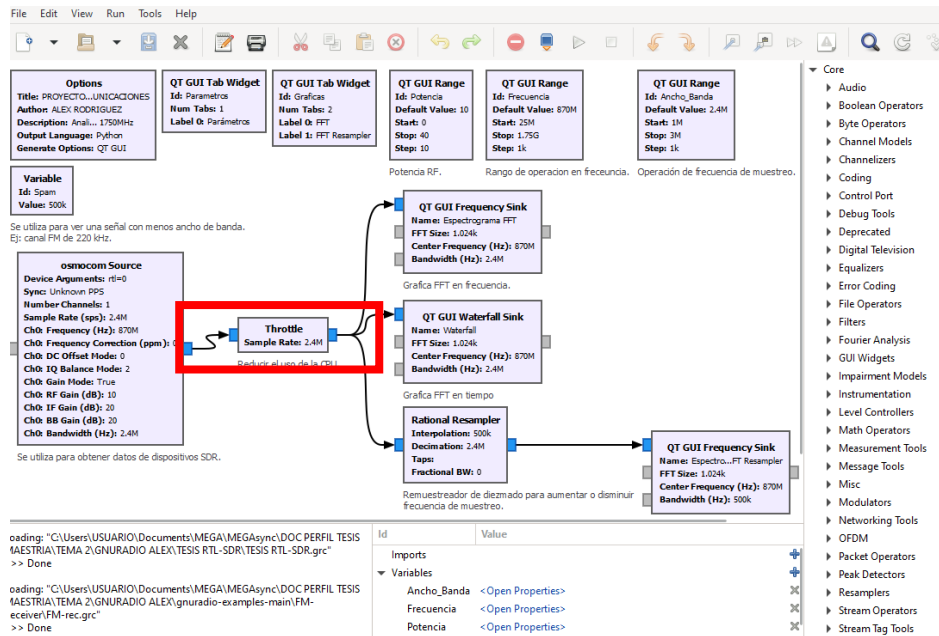


Figura 4.5 Bloque Throttle - programación prototipo.

Fuente: Desarrollado por el investigador

Existen opciones adicionales de operaciones en los bloques: enable, disable y bypass. Las opciones enable y disable son funciones opuestas y permiten habilitar o

deshabilitar un bloque en el diagrama diseñado sin tener que eliminarlo. Al ejecutar el diagrama, el programa no mostrará algún error si un bloque no conectado se encuentra deshabilitado. Un bloque deshabilitado presenta un color más oscuro que un bloque habilitado. La opción bypass permite pasar por alto un bloque sin necesidad de eliminarlo o deshabilitarlo del diagrama. El programa se ejecutará sin tomar en cuenta el bloque que se encuentre con bypass. Un bloque que se encuentre configurado con la opción bypass se presenta de un color amarillo. Estas opciones se pueden configurar haciendo clic derecho sobre el bloque seleccionado como se muestra en la Figura 4.6.

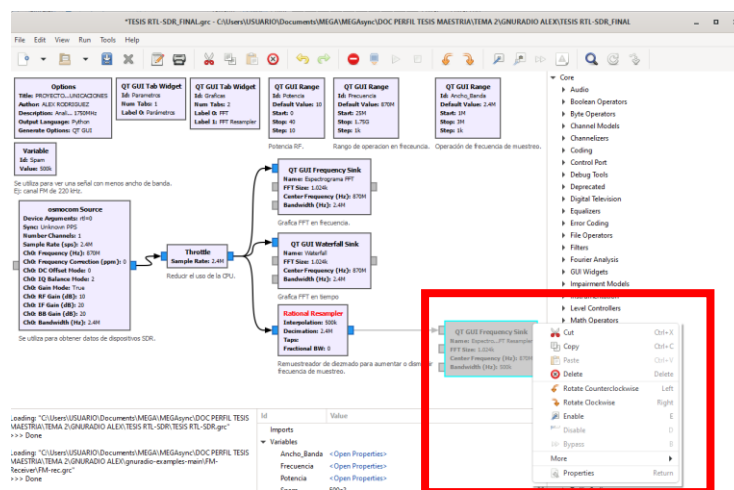


Figura 4.6 Opciones enable, disable y bypass en un diagrama de GNU Radio.

Fuente: Desarrollado por el investigador

El prototipo tiene las aplicaciones desarrolladas por el maestrante en forma preinstaladas de GNU Radio, las cuales permiten probar la funcionalidad y conexión de GNU Radio con el hardware de RF implementado. La primera funcionalidad del prototipo permite analizar radio FM comercial. El procedimiento es conectar el RTL-SDR a la Raspberry el cual se encuentra internamente conectado en el prototipo Figura 4.7, iniciar la aplicación y ajustar a una frecuencia para poder analizar las señales de audio provenientes de la emisora. La segunda funcionalidad permite escanear el espectro de frecuencia que se encuentra en el medio ambiente. Esta aplicación funciona dentro del rango de frecuencias de trabajo del RTL-SDR.



Figura 4.7 Conexión Raspberry y RTL-SDR

Fuente: Desarrollado por el investigador

4.3 Hardware y software para simulación

El hardware y software para la simulación requiere como hardware, un generador de frecuencia, SDR y Raspberry Pi con las especificaciones que se explicaron a anteriormente. Mientras que el software utilizado es GNU Radio para procesar la señal entrante. La simulación se realizará como en la Figura 4.8 a continuación.

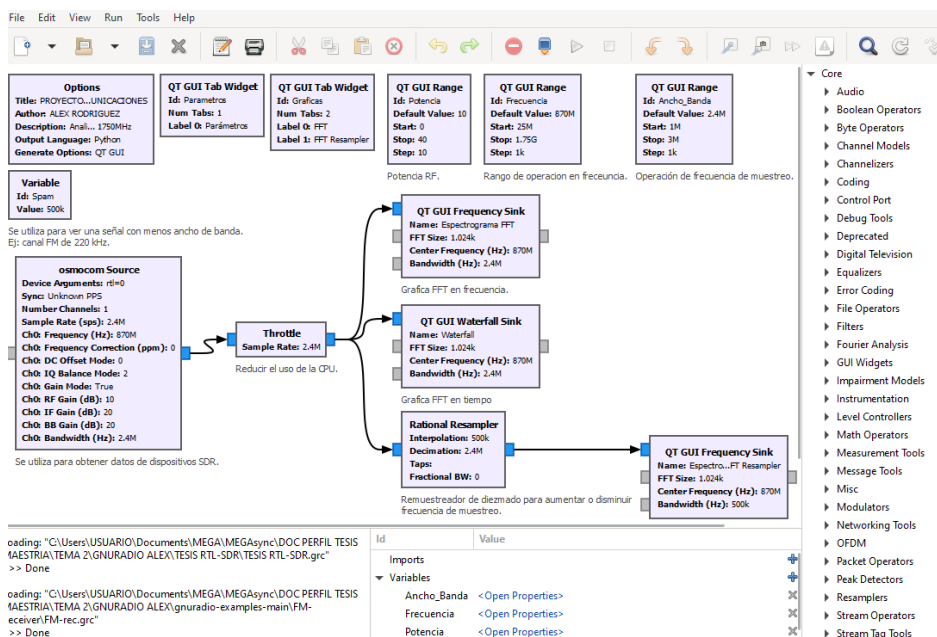


Figura 4.8 Simulación GNU Radio

Fuente: Desarrollado por el investigador

4.3.1 Diseño y Construcción del Hardware

En esta sección se presenta el desarrollo del hardware involucrado en el proyecto, desde los primeros diseños hasta la implementación final.

➤ Requerimientos del prototipo

Los requisitos que debe cumplir el prototipo acorde a los objetivos planteados para la realización del presente proyecto son los siguientes:

- a) Simulación de series trigonométricas de Fourier
- b) Análisis de señales en el dominio de la frecuencia mediante el uso de FFT.
- c) Visualización de las imágenes mediante una PANTALLA 7”.

Adicionalmente, para el cumplimiento de cada uno de los requisitos antes mencionados, el prototipo necesitará ciertos requerimientos de software, los cuales son los siguientes:

- 1) Software RTL-SDR.
- 2) Aplicación GNU Radio Companion.

Estos requisitos de software correrán sobre el sistema operativo Raspberry SO, cuya instalación y configuración se detallada en el Anexo A.

Previamente para el cumplimiento de los requisitos a, b y c, es necesario que en el prototipo se encuentren instalados y configurados todos los requisitos de software. A continuación, se explicará la instalación y configuración de estos requisitos.

4.3.2 Acoplamiento al chasis

El ensamblaje del analizador de espectro se realiza luego de comprobar el correcto funcionamiento de todos los materiales tanto en software como en hardware, dicho elementos que se encuentran dentro del chasis como se observa en la Figura 4.9 y destalla a continuación:

- Raspberry Pi.
- Pantalla.

- Cable micro HDMI a HDI.
- Unión USB hembra.
- Decodificadores de señales TL SDR.
- Antenas.

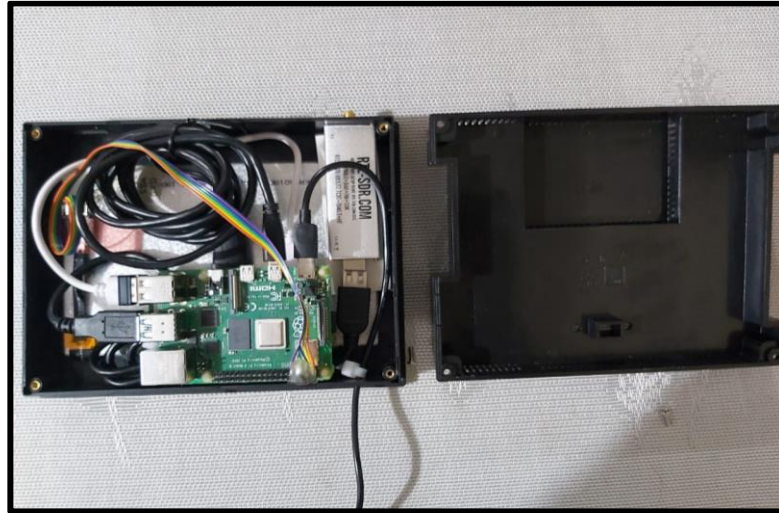


Figura 4.9 Parte interna del prototipo.

Fuente: Desarrollado por el investigador

El prototipo es completamente portátil a continuación se da a conocer diferentes vistas o perspectivas del prototipo como se muestra en la Figura de Vista frontal, Figura Vista posterior, Figura 4.10 Diferentes vistas del prototipo Figura 4.11 y Figura 4.12



Figura 4.10 Vista frontal del prototipo.

Fuente: Desarrollado por el investigador

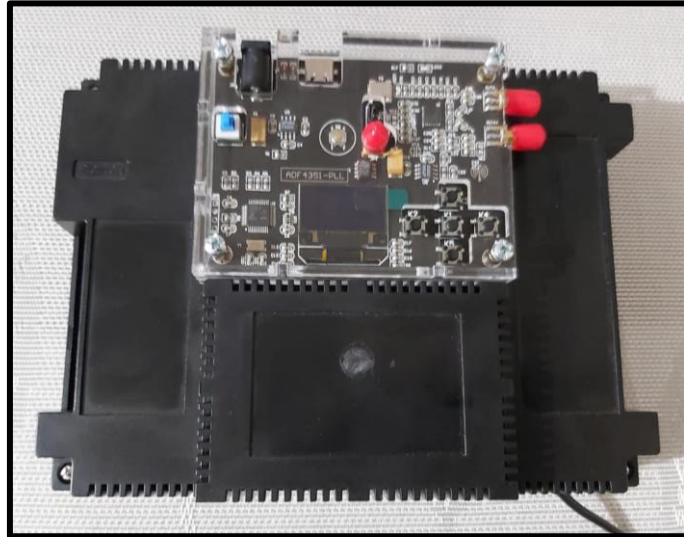
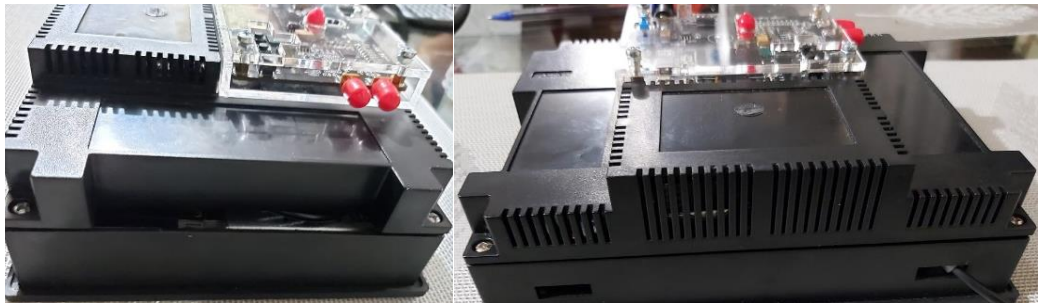


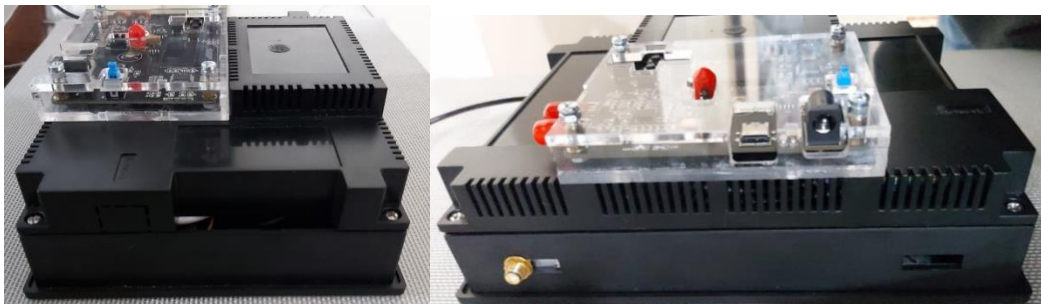
Figura 4.11 Vista posterior del prototipo.

Fuente: Desarrollado por el investigador



a)

b)



c)

d)

Figura 4.12 Diferentes vistas del prototipo.

a.-Vista lateral derecha b.- Vista inferior c.- Vista lateral izquierda d.- Vista superior

Fuente: Desarrollado por el investigador

4.3.3 Pantalla

La pantalla táctil TFT posee las siguientes características: Figura 4.13

- Está diseñado para Raspberry Pi.
- Es una alternativa ideal para monitores HDMI.
- Tecnología IPS de alta calidad y perfecta visualización.
- Resolución de 1024x600.
- Interfaz Hombre-Máquina amigable.
- Soporta sistema Raspbian
- Permite reproducir videos, tomar fotos.
- Interacción del sistema sin teclado y sin ratón.

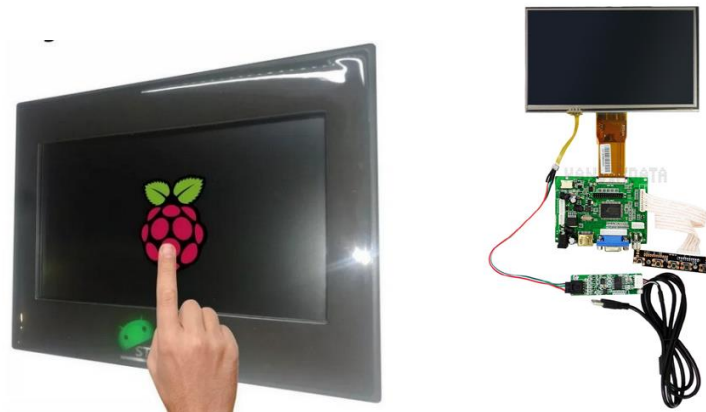


Figura 4.13 Parte frontal de la pantalla.

Fuente: Desarrollado por el investigador

Solución Propuesta

El problema principal es que un analizador de espectros de barrido tiene un gran costo y no permite hacerse de uno con facilidad. Lo que se quiere lograr con este proyecto es la fabricación de un instrumento que permita realizar el mismo análisis utilizando nuevas tecnologías que permitan reducir los costos, el tamaño y que a su vez se pueda utilizar por medio de una pantalla para que su uso pueda ser más simple e interactivo. La mayoría de los productos mencionados anteriormente tienen un rango de frecuencia de análisis mayor al de este prototipo (del orden de GHz). Se hace esta salvedad ya que el prototipo que se diseño tiene un rango de 500Hz-1750MHz, lo que se quiere lograr con el prototipo final en principio es la realización exitosa en baja frecuencia y

dejar abierta la posibilidad a futuro de aumentar el rango de análisis. Para tal tarea se trabajó sobre el diagrama de bloques en la

Figura 4.14. En él se pueden diferenciar 4 bloques (secciones) que se trabajaron por separado. Luego se hará una breve descripción de cada uno ya que en las secciones posteriores se verá en detalle cada uno de ellos:

Diagrama en bloques prototipo

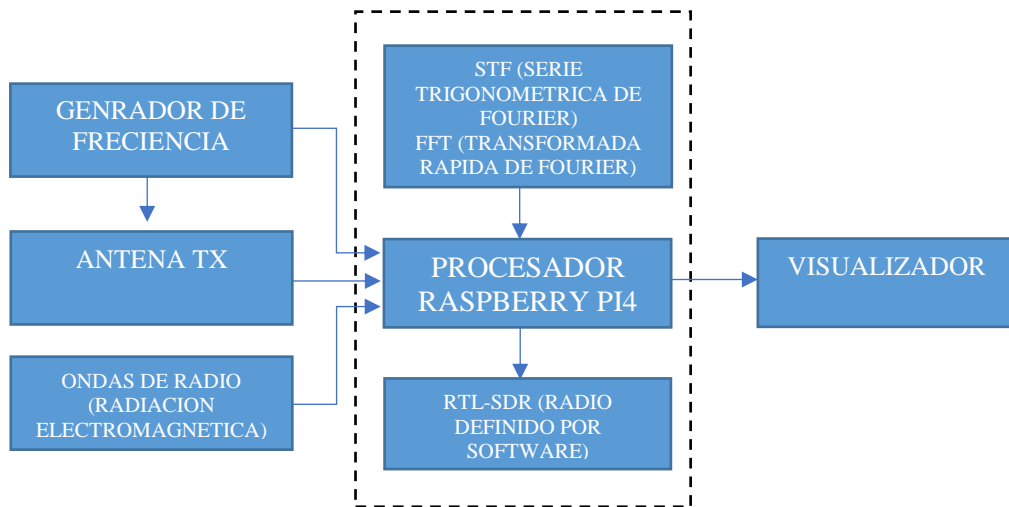


Figura 4.14 Diagrama de bloques prototipo

Fuente: Desarrollado por el investigador

Descripción de los bloques:

➤ Sintetizador de señales basado en un PLL - ADF4351

El más importante de ellos, se encargará de la generación de las ondas senoidales en un rango de 35- 4400 MHz. La importancia fundamental es que este pueda ser programado digitalmente, para poder realizar la conexión hacia el prototipo.

➤ RTL SDR

Se encarga de recibir las señales captadas por el dispositivo RTL-SDR (conectado al Raspberry-Pi).

➤ **Microcontrolador Raspberry Pi 4**

Será el encargado de la comunicación con el generador PLL y de adquirir muestras del RTL SDR para ser enviadas de vuelta a la Raspberry para posterior análisis. Este conformará la tercera placa del proyecto final.

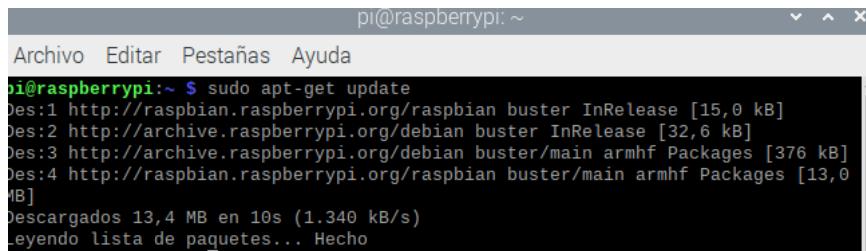
➤ Pantalla 7” Táctil

Contará con una interfaz virtual hecha en GNU RADIO la que permitirá al usuario elegir el rango de frecuencias de funcionamiento.

Instalación del software RTL-SDR

Para la instalación y el correcto funcionamiento del módulo RTL-SDR con la Raspberry Pi se tiene que realizar el procedimiento que se menciona a continuación.

1. Realizar la actualización de todos los paquetes que se encuentran instalados en la Raspberry Pi mediante el comando “sudo apt-get update”, tal como se muestra en la Figura 4.15



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~$ sudo apt-get update  
Des:1 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]  
Des:2 http://archive.raspberrypi.org/debian buster InRelease [32,6 kB]  
Des:3 http://archive.raspberrypi.org/debian buster/main armhf Packages [376 kB]  
Des:4 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13,0  
kB]  
Descargados 13,4 MB en 10s (1.340 kB/s)  
leyendo lista de paquetes... Hecho
```

Figura 4.15 Actualización de paquetes.

Fuente: Desarrollado por el investigador

Se instala el soporte del RTL-SDR para GNU Radio, tal como se muestra en la Figura 4.16.

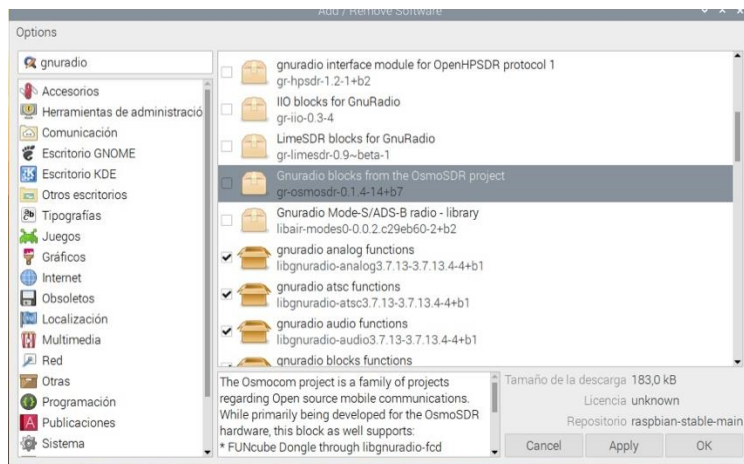


Figura 4.16 Instalación de gr-osmosdr.

Fuente: Desarrollado por el investigador

Ante de instala dicho soporte es posible verificar el número de paquetes, así como se muestra en la Figura 4.17.

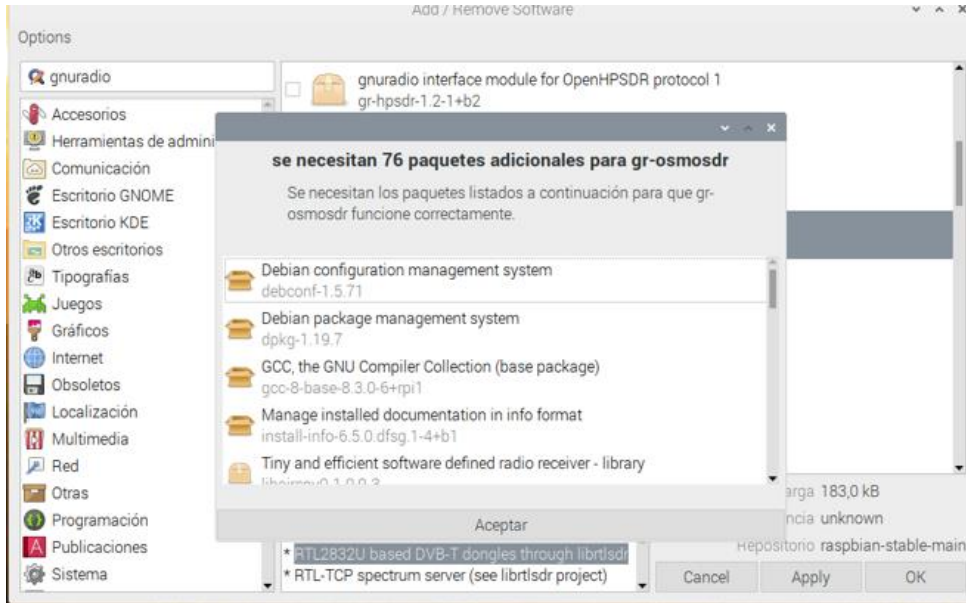


Figura 4.17 Numero de paquetes a instalación gr-osmosdr.

Fuente: Desarrollado por el investigador

Finalmente, para comprobar que la instalación se realizó correctamente y que el dispositivo RTL-SDR está listo para ser utilizado, se escribe en la línea de comandos “rtl_test -s 1024000”, tal como se muestra en la Figura 4.18.

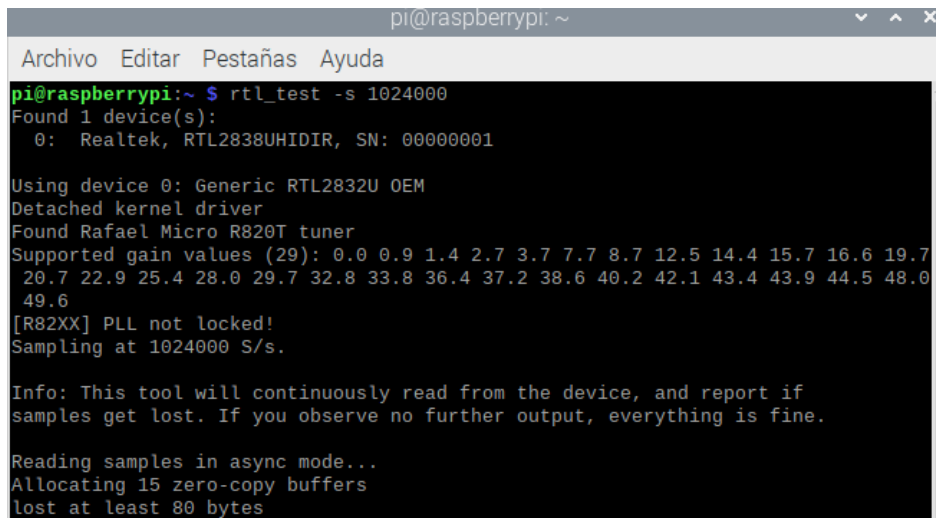


Figura 4.18 Verificación de la instalación del software RTL-SDR.

Fuente: Desarrollado por el investigador

En la Figura se observa que la instalación se realizó correctamente, mostrando algunas de las características que posee el módulo, entre ellas se muestran los dos componentes

principales que son, el chip sintonizador R820T y el chip demodulador RTL2832U, así también los diferentes valores de ganancia que soporta y la velocidad de muestreo.

El espectro de radio es un recurso limitado y debe monitorearse y administrarse para evitar problemas con otros transmisores. Por tanto, es necesario utilizar un analizador de espectro para comprobar el comportamiento de la señal y detectar anomalías que afecten a la calidad de transmisión. Teniendo en cuenta estas consideraciones, el prototipo de analizador de espectro se diseña con una estructura funcional en todos los componentes, respetando los parámetros de recepción, procesamiento, visualización y análisis de la señal recibida. Se crearon cuatro bloques esquemáticos para diseñar un prototipo de analizador de espectro.

El prototipo está constituido por una interfaz gráfica, mediante una pantalla LCD y/o un monitor de computador, lo que facilita ver los registros obtenidos del espectro para un posterior análisis, lo cual permite observar la señal mediante una interfaz gráfica a través de un eje de coordenadas, donde el eje de las abscisas poseerá la frecuencia, mientras que la amplitud se encontrará en el eje de las ordenadas, esto para un análisis manual de los datos obtenidos en pantalla con la comparación de resultados con datos ya obtenidos del analizador de espectro.

Se diseña un analizador de espectro empleando como hardware rtl sdr y el software GNU radio companion versión 3.7.13 en el sistema operativo Raspberry Os los parámetros a configurar están basado en las características técnicas del equipo rtl sdr

En GNU radio companion se crea un nuevo diagrama de flujo con interfaces wx dando doble clic en el bloque option para editar sus propiedades definiendo su ID como `Analizador_Espectro`

Para insertar nuevos bloques se acceda a la opción de búsqueda de bloques, se selecciona y se arrastra a la hoja de trabajo necesitando el bloque Osmocom Source como un interfaz entre el rtl sdr y GNU radio companion

Añadimos el bloque Throttle el cual reduce el uso del CPU *Figura 4.19*

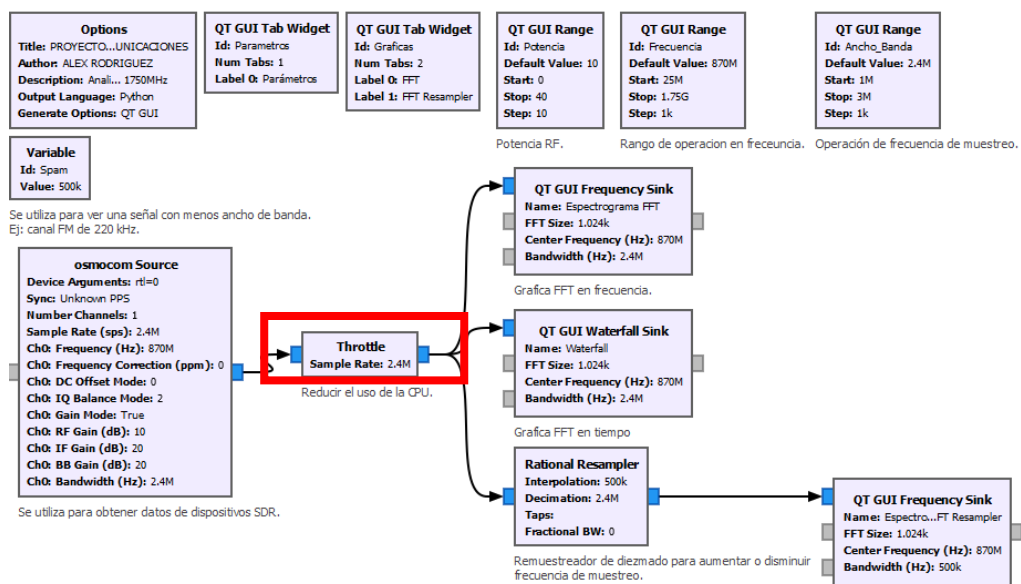


Figura 4.19 Bloque Throttle.

Fuente: Desarrollado por el investigador

El bloque QT GUI range como variable de control deslizante Figura 4.20

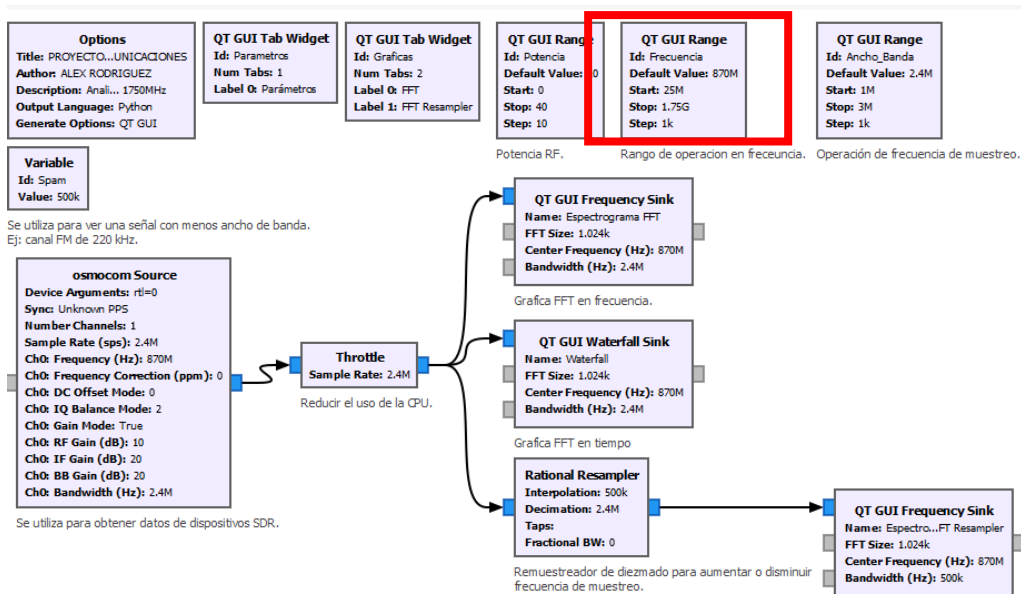


Figura 4.20 Bloque QT GUI range.

Fuente: Desarrollado por el investigador

Se necesita 3 variables de control deslizante por lo que seleccionamos 2 más Figura 4.21

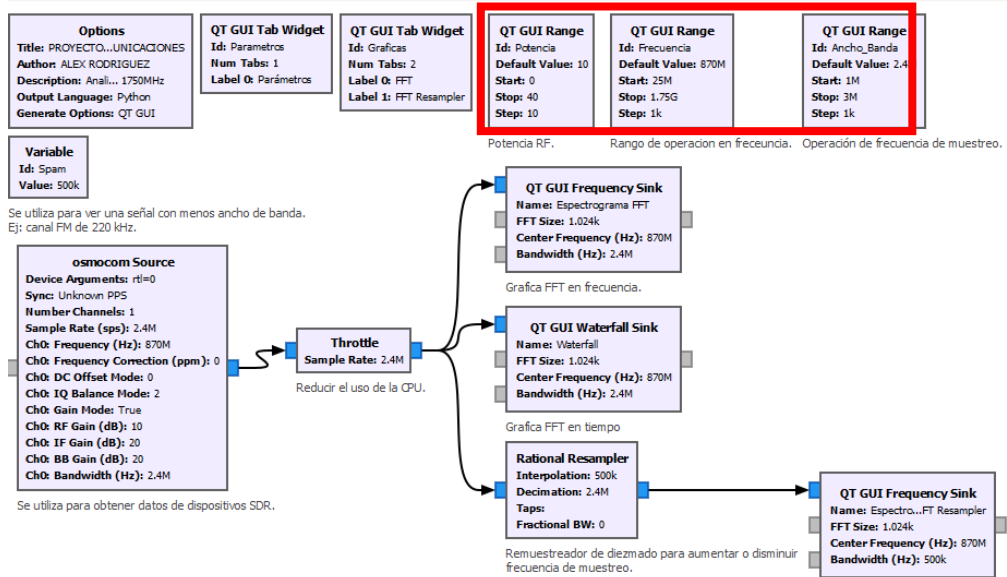


Figura 4.21 Variables de control deslizante.

Fuente: Desarrollado por el investigador

Los bloques se unen entre si dando clic en la entra y salida de cada uno Figura 4.22

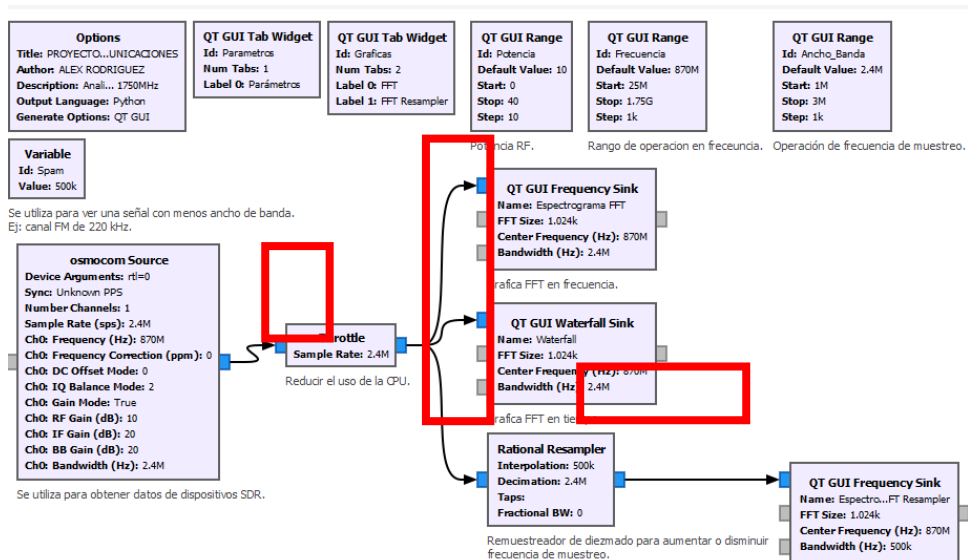


Figura 4.22 Unión entre la entra y salida de los bloques.

Fuente: Desarrollado por el investigador

La primera variable corresponde a la Potencia Figura 4.23

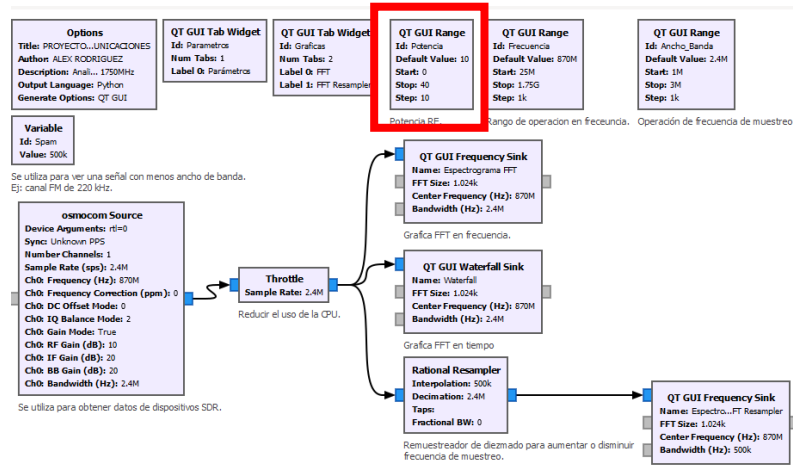


Figura 4.23 Variable Potencia.

Fuente: Desarrollado por el investigador

La segunda variable será Frecuencia, el valor por defecto 95,3 MHz es el valor mínimo 25 MHz, el valor máximo 1700 MHz y el número de pasos 1000. Figura 4.24

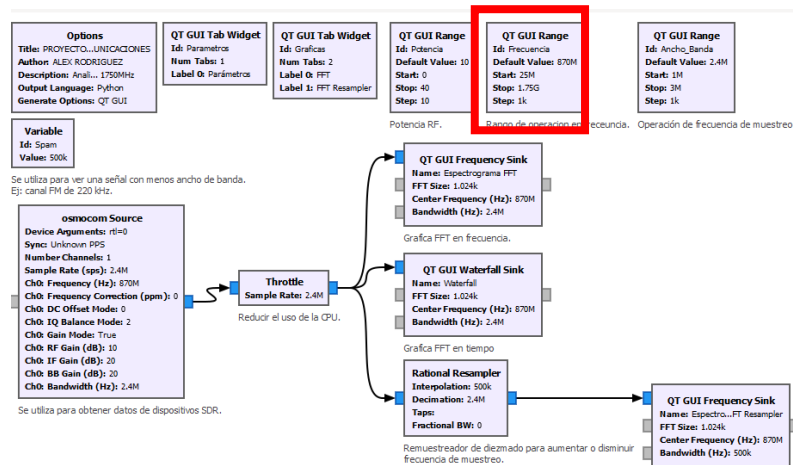


Figura 4.24 Variable Frecuencia

Fuente: Desarrollado por el investigador

La tercera variable será Ancho_Banda con un valor por defecto de 2,4 MHz, el valor mínimo de 1 MHz y el valor máximo de 3 MHz. Figura 4.25

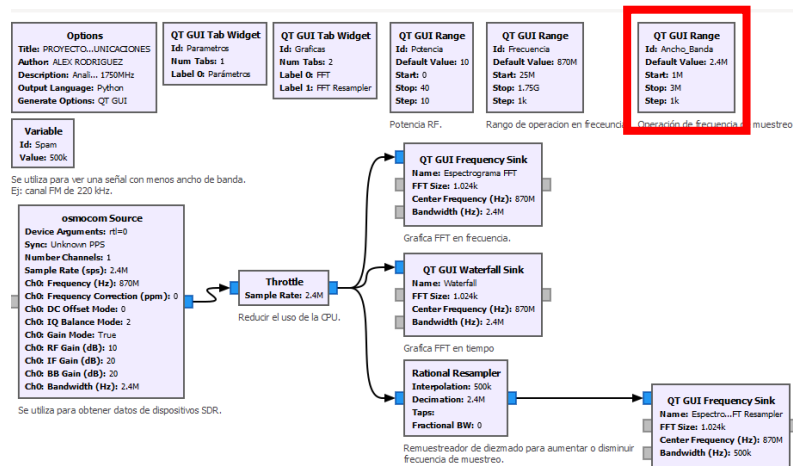


Figura 4.25 Variable Ancho_Banda.

Fuente: Desarrollado por el investigador

Se editan las propiedades de Osmocom Source donde se establece la tasa de muestreo, frecuencia, ganancia RF con las variables establecidas. Además, se fijan los parámetros Modo de Balance IQ y Modo de ganancia en automático respectivamente.

En el bloque Throttle se edita la tasa de muestreo

Finamente se configura los bloques que consumen y nutran información en los bloques WX GUI FFT Sink y WX GUI Waterfall Sink se establecen la tasa de muestreo, frecuencia, el tamaño de la FFT y se configura el promedio de la información en automático.

Para finalizar se ejecuta el diagrama de flujo guardando el archivo y esperando que termine la compilación. Figura 4.26

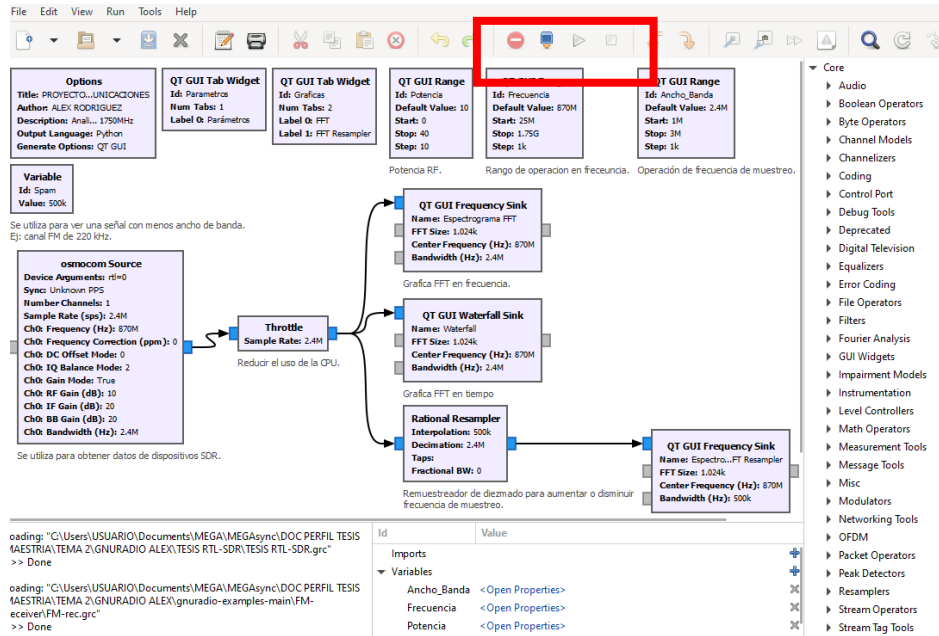


Figura 4.26 Ejecución diagrama de flujo.

Fuente: Desarrollado por el investigador

En la interfaz gráfica se observa las variables con control deslizante Figura 4.27 y el espectro radioeléctrico en razón del tiempo mediante la gráfica “Waterfall Plot” y en razón de la frecuencia con la gráfica FFT Fast Fourier Transform o transformada rápida de Fourier.

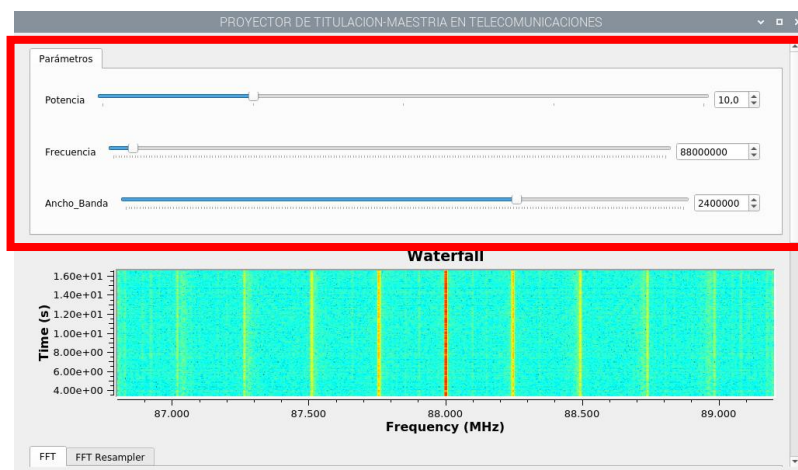


Figura 4.27 Interfaz gráfica variables de control deslizante.

Fuente: Desarrollado por el investigador

Existe dos opciones para modificar el valor de cada variable ya sea mover el indicador de la barra deslizante o ingresar el valor deseado. Se puede variar el valor de la frecuencia, ancho de banda y potencia dentro del rango establecido previamente en cada una de las variables. La variación de la frecuencia central spam y potencia de la señal central se presenta de manera automática en la gráfica FFT y Waterfall.

Para observar la señal en la gráfica FFT a mayor detalle se da clic en Auto Scale. Además, se puede detener y continuar la presentación del análisis. Figura 4.28

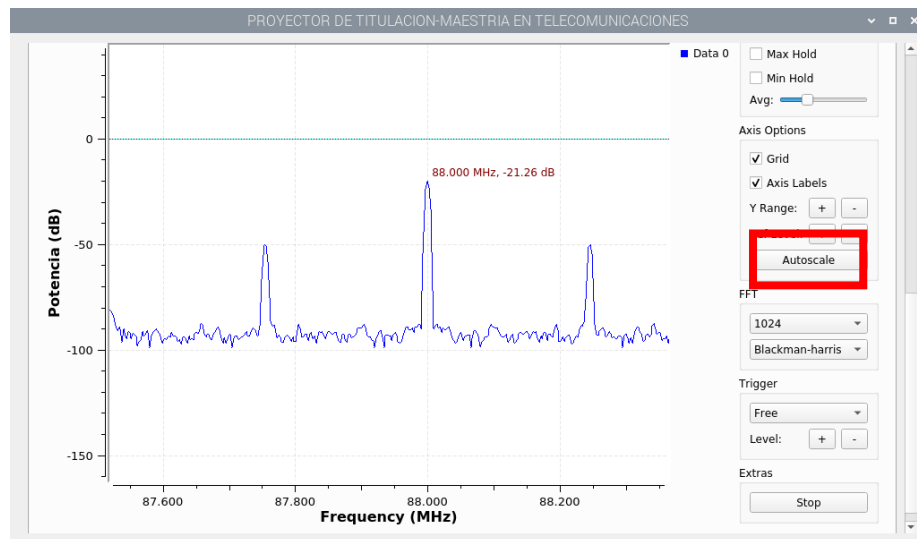


Figura 4.28 Auto Scale.

Fuente: Desarrollado por el investigador

De igual forma en la gráfica Waterfall se tiene la opción autoscale cambiándose el color de la señal en 4 tonos diferentes 3 tonos en RGB y un tono en escalas de grises así mismo se puede detener, continuar y limpiar la presentación.

Colocando el cursor sobre la gráfica FFT en los diferentes picos de la señal se observa permeámetros de frecuencia y potencia mientras que en la gráfica Waterfall únicamente la frecuencia. Figura 4.29

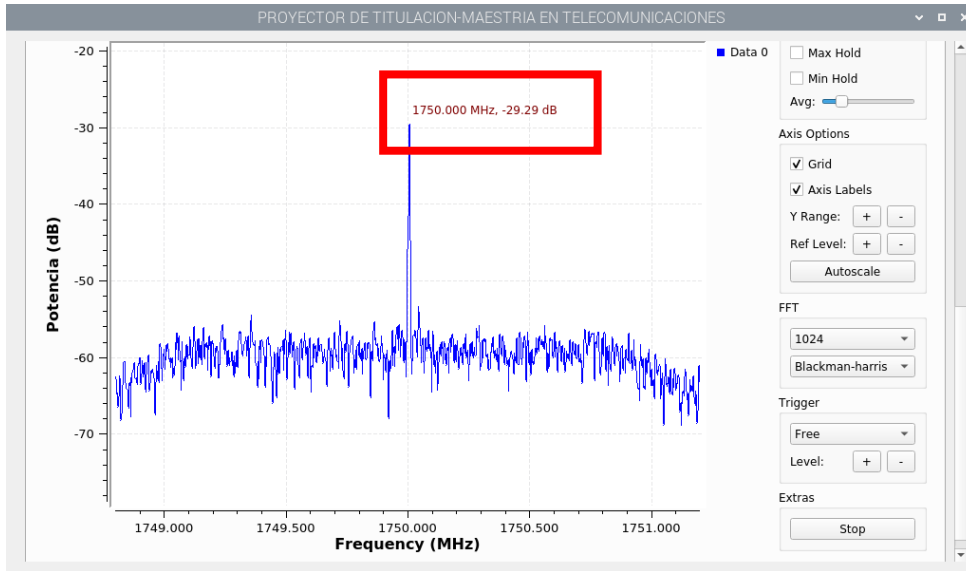


Figura 4.29 permeámetros de frecuencia y potencia

Fuente: Desarrollado por el investigador

Para visualizar la mayor concentración de potencia de la señal en una forma más notoria, se puede activar el Diagrama de Cascada (Waterfall), que permite diferenciar donde se concentra más la energía de cada señal recibida con la escala de colores visualizados, siendo el rojo el de mayor potencia y el verde el de menor potencia.

Figura 4.30

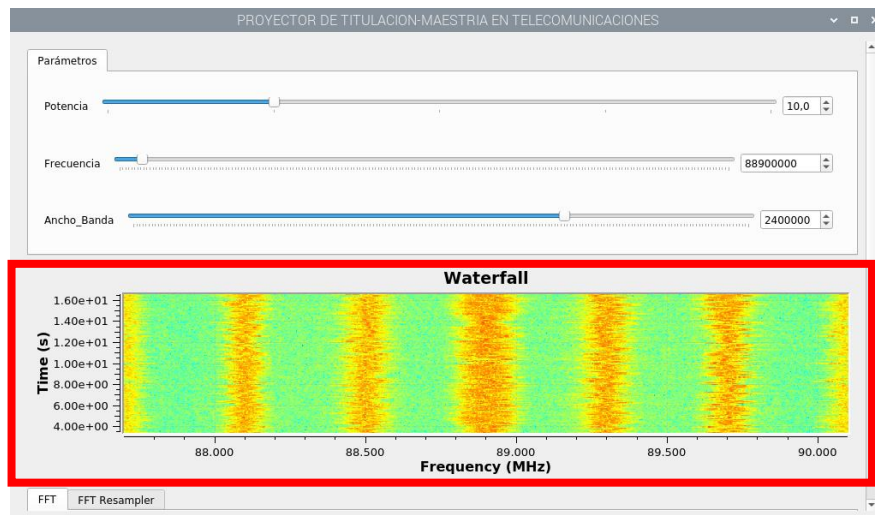


Figura 4.30 Diagrama de Cascada (Waterfall)

Fuente: Desarrollado por el investigador

4.4 Desarrollo de aplicación Serie Trigonométrica de Fourier

Para desarrollar esta aplicación, se revisaron diferentes opciones de lenguajes de programación que pudieran cumplir con las especificaciones ya descritas, como son, el que pueda ejecutarse en un dispositivo portátil, que pueda utilizar cálculo simbólico y que pueda hacer todos los cálculos en el mismo dispositivo portátil a través de una interfaz gráfica. El lenguaje de programación que cumplió con los requisitos de lenguajes de programación para resolver Ecuaciones Diferenciales Ordinarias fue Python el cual cumple a cabalidad los requisitos necesarios para este objetivo.

4.4.1 Interfaz Gráfica.

Esta aplicación móvil tiene una interfaz principal en la que se despliegan las tres opciones de funciones que se pueden introducir, además de un botón para obtener ayuda.

Esta app está diseñada para trabajar con los siguientes tipos de funciones: Función continua definida en un intervalo $[a, b]$:

$$F(x) = \begin{cases} f(x) & \text{si } a \leq x \leq b \end{cases}$$

La segunda vista es, la interfaz para introducir los datos con los que se quiere trabajar. Se pide lo siguiente:

- Función(es) $f(x)$.
- Límites del intervalo donde la función $f(x)$ es válida.
- Número de armónicos para poder realizar la gráfica.

La Figura 4.31 muestra la pantalla para una función.

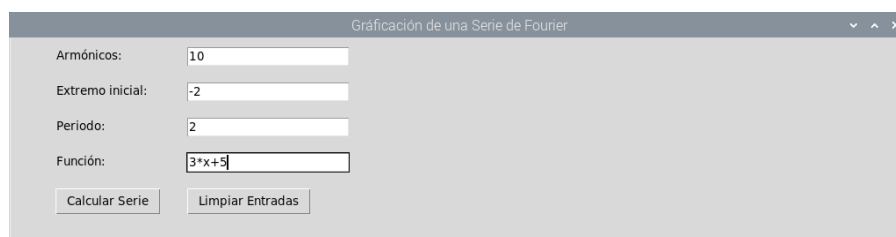


Figura 4.31 Pantalla de la app para una función.

Fuente: Desarrollado por el investigador

Una vez introducidos los valores necesarios, el botón Calcular Serie la serie de Fourier nos muestra el resultado en otra ventana.

La segunda ventana muestra los siguientes elementos:

- El periodo.
- Planteamiento del coeficiente a_0 .
- Planteamiento del coeficiente a_n .
- Planteamiento del coeficiente b_n .
- Planteamiento de la serie de Fourier.
- Serie de Fourier obtenida.
- Gráfica de la serie de Fourier obtenida con el número de armónicos dado.

Estas interfaces están alojadas como Actividades (Activity) en Android. La comunicación se da entre actividades.

4.4.2 Etapas del desarrollo

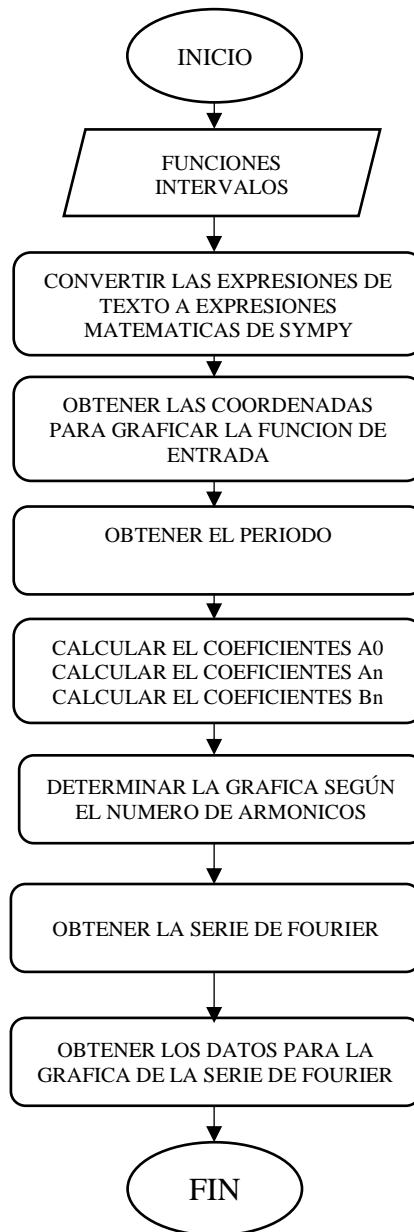


Figura 4.32 Etapas en la resolución del problema:

Fuente: Desarrollado por el investigador

4.4.3 Entrada de datos.

En la interfaz de la aplicación, el usuario teclea su expresión matemática para indicar una función matemática, así como los límites del intervalo donde es válida dicha función. Estos datos se capturan y se almacenan en las variables globales de la clase creada.

4.4.4 Graficar la función de entrada

Para graficar la función de entrada, a cada trozo de la función se le asignan 30 valores como dominio en el intervalo en que es válido, luego se procede a calcular $f(x)$ en cada punto dado. Como se está trabajando con expresiones simbólicas, no se pueden hacer cálculos matemáticos directos, y lo que se quiere hacer, es, obtener 30 valores de $f(x)$, uno para cada uno del dominio asignado. La opción que maneja SymPy para hacer sustituciones de números por variables y evaluar la expresión, es la instrucción `subs`. Esto sería muy lento, y si la expresión tuviera el símbolo π , se tendría que hacer otra sustitución, lo que generaría más código y más lentitud. Por este motivo, se decidió utilizar la función `lamdfify`.

4.4.5 Obtener el período de la función de entrada.

El período donde es válida la función de entrada, ya sea continuo en un intervalo, o discontinua en varios intervalos, es muy importante, ya que lo vamos a utilizar en las fórmulas para obtener el desarrollo de la Serie de Fourier. En este caso, al límite inferior se le cambia de signo y se le suma al límite superior.

4.4.6 Determinar la simetría de la función de entrada.

Determinar la simetría de la función ayuda para ahorrar cálculos innecesarios, como ya se vio anteriormente, si la función es Par entonces la serie de senos será igual a cero, por lo que solo se calculan los coeficientes a_0 y a_n , si la función es impar, la serie de cosenos es igual a cero, por lo que solo se calcula el coeficiente b_n . Solo en caso de que la función no tenga simetría par o impar, se calculan los tres coeficientes.

Para obtener la simetría de la función se parte de la definición de simetría:

$$\begin{aligned} \text{Simetría Par:} \quad & f(-x) = f(x) \\ \text{Simetría Impar:} \quad & f(-x) = -f(x) \end{aligned}$$

Antes de evaluar y comparar cualquier par ordenado, se hace la validación de la diferencia entre el límite superior y el inferior. Si la diferencia es cero, entonces procedemos a verificar la simetría, en caso contrario, sabemos de antemano que la

función o funciones de entrada no tienen simetría, ya que el intervalo dado no es simétrico.

Si la diferencia es cero, se procede a evaluar la función de entrada en 15 valores definidos dentro del intervalo válido de la función de entrada, esto para asegurar la validación en las funciones de entrada de tres trozos.

Se obtienen los pares ordenados y se procede a aplicar las definiciones de simetría, para esto, se toman dos valores en los que se asegura que se apliquen las definiciones de simetría.

En el caso de la función continua, se utiliza el mismo procedimiento para obtener los valores para la gráfica. En el caso de las funciones a trozos, la función *Lambdify* se acompaña con la función *Piecewise*, esto es para obtener los pares ordenados de los diferentes intervalos de cada trozo de función en una sola instrucción, por ejemplo, para una función de dos trozos, tenemos las siguientes instrucciones:

4.5 Cálculo de los coeficientes.

4.5.1 Cálculo del coeficiente a_0 .

Para calcular este coeficiente, se valida que la función tenga simetría Par o que no tenga simetría. Esta parte del algoritmo se divide en 3 partes, para que, como ya se ha comentado, tenga fidelidad con el trabajo del estudiante y del docente en el salón de clase. Estas partes son las siguientes:

```
B_0 = sp.integrate(F/T, (x, I, I+T/2)) + sp.integrate(G/T, (x, I+T/2, I+T))
print('B_0=', B_0)
A_m = []
B_m = []
for i in range (1, m+1):
    A_m.append(sp.integrate(F*(2/T)*sp.sin(i*2*k*x), (x, I, I+T/2))
    + sp.integrate(G*(2/T)*sp.sin(i*2*k*x), (x, I+T/2, I+T)))
    B_m.append(sp.integrate(F*(2/T)*sp.cos(i*2*k*x), (x, I, I+T/2))
    + sp.integrate(G*(2/T)*sp.cos(i*2*k*x), (x, I+T/2, I+T)))
```

Figura 4.33 Programación para el cálculo del coeficiente a_0

Fuente: Desarrollado por el investigador

4.5.2 Cálculo del coeficiente a_n .

Para calcular este coeficiente, se valida que la función tenga simetría Par o que no tenga simetría.

Esta parte del algoritmo se divide en tres partes, para que, como ya se ha comentado, tenga fidelidad con el trabajo del estudiante y del docente en el salón de clase. Estas partes son las siguientes:

Son las mismas que para calcular el coeficiente a_0 , por lo que solo se nombraran:

1. "Planteando el coeficiente a_n ".
2. "Integrando...".
3. "Sustituyendo los límites de las integrales definidas".

En este caso, se hizo la integración definida de la siguiente integral:

$$a_n = \frac{2}{p} \int_a^b \cos\left(\frac{2\pi nx}{p}\right) f dx$$

Ecuación 4-1

```
B_0 = sp.integrate(F/T, (x, I, I+T/2)) + sp.integrate(G/T, (x, I+T/2, I+T))
print('B_0=', B_0)
A_m = []
B_m = []
for i in range(1, m+1):
    A_m.append(sp.integrate(F*(2/T)*sp.sin(i*2*k*x), (x, I, I+T/2))
    + sp.integrate(G*(2/T)*sp.sin(i*2*k*x), (x, I+T/2, I+T)))
    B_m.append(sp.integrate(F*(2/T)*sp.cos(i*2*k*x), (x, I, I+T/2))
    + sp.integrate(G*(2/T)*sp.cos(i*2*k*x), (x, I+T/2, I+T)))
```

Figura 4.34 Programación para el cálculo del coeficiente a_n

Fuente: Desarrollado por el investigador

4.5.3 Cálculo del coeficiente b_n

Para calcular este coeficiente, se valida que la función tenga simetría Impar o que no tenga simetría.

Esta parte del algoritmo se divide en cinco partes, para que, como ya se ha hablado, tenga fidelidad con el trabajo del estudiante y del docente en el salón de clase.

Estas partes son las mismas que para calcular el coeficiente a_n , por lo que solo se nombraran:

1. "Planteando el coeficiente b_n ".
2. "Integrando...".
3. "Sustituyendo los límites de las integrales definidas".

En este caso, se hizo la integración definida de la siguiente integral:

$$b_n = \frac{2}{p} \int_a^b \sin\left(\frac{2\pi nx}{p}\right) f dx$$

Ecuación 4-2

4. "Simplificando senos".
5. "Simplificando cosenos".

Con estos datos se gráfica la función a trozos para confirmar visualmente que estamos trabajando con la función de entrada correcta

Calcula el periodo: Periodo = 2

Después hace una diferencia entre los límites establecidos de los intervalos en el dominio de las funciones de entrada. Si la diferencia es cero, se procede a establecer si tiene simetría o no. Si la diferencia es diferente de cero, se da por hecho que no tiene simetría.

Diferencia de límites = 0

Se muestra el resultado de a_n :

$$a_n = \frac{(-1)^n - 1}{\pi^2 n^2}$$

Ecuación 4-3

Planteando el coeficiente b_n .

$$b_n = \frac{2}{2} \int_{-1}^0 [1] \sin\left(\frac{2n\pi x}{p}\right) dx + \frac{2}{2} \int_0^1 [x] \sin\left(\frac{2n\pi x}{p}\right) dx$$

Ecuación 4-4

```

B_0 = sp.integrate(F/T, (x, I, I+T/2)) + sp.integrate(G/T, (x, I+T/2, I+T))
print('B_0=', B_0)
A_m = []
B_m = []
for i in range(1, m+1):
    A_m.append(sp.integrate(F*(2/T)*sp.sin(i*2*k*x), (x, I, I+T/2))
    + sp.integrate(G*(2/T)*sp.sin(i*2*k*x), (x, I+T/2, I+T)))
    B_m.append(sp.integrate(F*(2/T)*sp.cos(i*2*k*x), (x, I, I+T/2))
    + sp.integrate(G*(2/T)*sp.cos(i*2*k*x), (x, I+T/2, I+T)))

```

Figura 4.35 Programación para el cálculo del coeficiente b_n

Fuente: Desarrollado por el investigador

4.6 Simulación con diferentes números de armónicas

Como primer paso, y con el fin de evaluar el correcto funcionamiento del programa, se ejecutó este manteniendo fijo el valor del límite superior e inferior -2 a 2.

Se realizaron pruebas para $n = 10, 20$, los resultados se muestran en las figuras posteriormente representadas.

Al momento que se ejecuta el Software “Series de Fourier” se dispone de una pantalla inicial en el cual se debe ingresar datos, tiene dos botones que realizan la función de cálculo y de limpiar los datos ingresados como se muestra en la Figura 4.36.

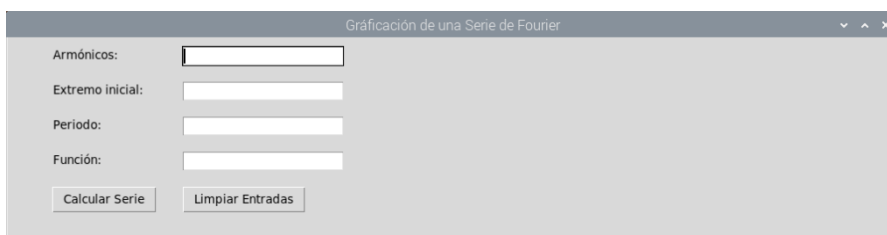


Figura 4.36 Pantalla inicial ejecución del programa “Series de Fourier”.

Fuente: Desarrollado por el investigador

Para los fines prácticos se realizará la simulación de la serie trigonométrica a de Fourier de la $ax+b$ la cual representa una función diente de sierra. Se ingresa como parámetros de numero de armónico igual a 10, extremo inicial igual a -2, periodo igual a 2 y la función diente de igual a $3*x+5$ como se observa en la Figura 4.37.

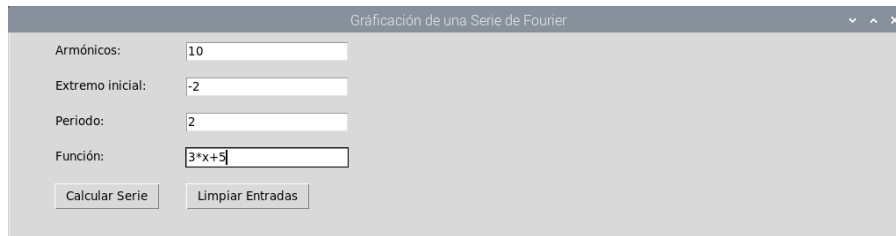


Figura 4.37 Función diente de sierra con 10 armónicos.

Fuente: Desarrollado por el investigador

En la gráfica se puede observar que la función $3x+5$ paso a ser una función diente de sierra con periodo de 2 y 10 armónicos levemente pronunciados las cuales están representadas en la Figura 4.38.

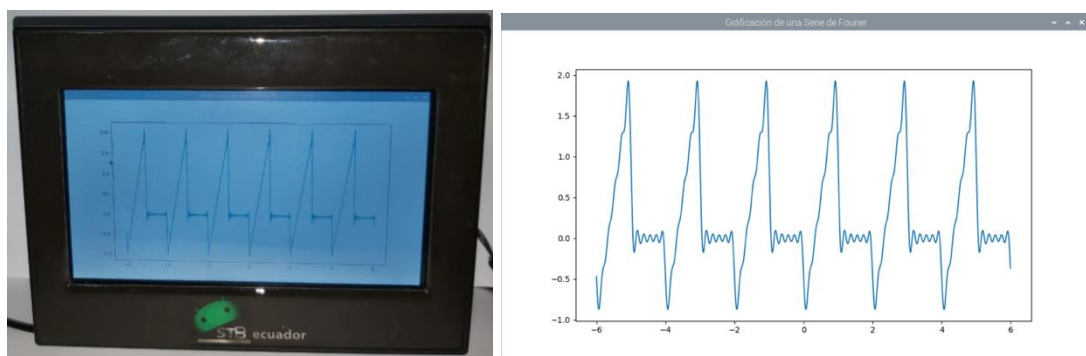


Figura 4.38 Gráfica de la función diente de sierra con 10 armónico.

Fuente: Desarrollado por el investigador

Se pretende seguir incrementando los armónicos para que la función pueda llegar aparecer más línea. Se ingresa como parámetros de numero de armónico igual a 30, extremo inicial igual a -2, periodo igual a 2 y la función impulso igual a 1 como se observa en la Figura 4.39 y Figura 4.40.

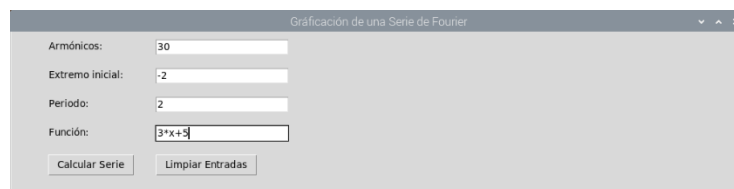


Figura 4.39 Función diente de sierra con 30 armónicos.

Fuente: Desarrollado por el investigador

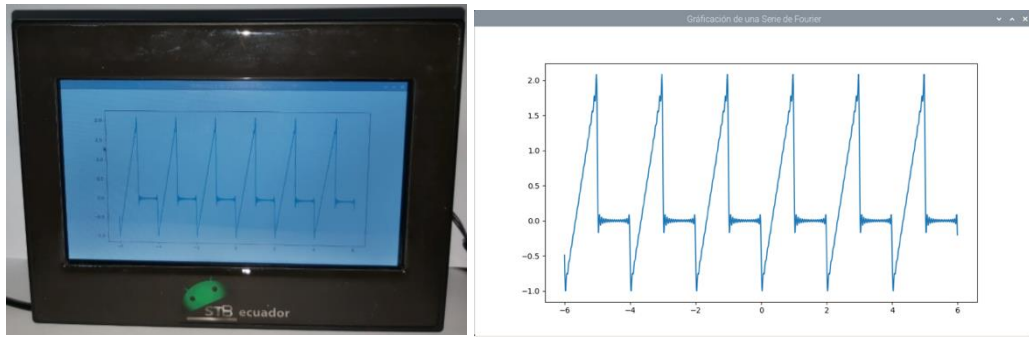


Figura 4.40 Gráfica de la función diente de sierra con 30 armónicos.

Fuente: Desarrollado por el investigador

Como segundo método práctico se representará la serie trigonométrica de Fourier de la función impulso con los siguientes parámetros número de armónico igual a 10, extremo inicial igual a -2, periodo igual a 2 y la función de onda cuadrada igual a 1 como se observa en la Figura 4.41 y Figura 4.42.

Figura 4.41 Función impulso con 10 armónicos.

Fuente: Desarrollado por el investigador

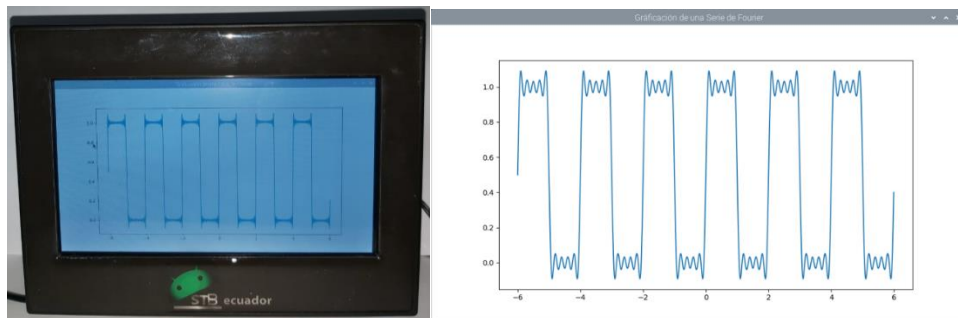


Figura 4.42 Gráfica impulso con 10 armónico.

Fuente: Desarrollado por el investigador

Finalmente se configura los parámetros de ingreso con armónico igual a 30, extremo inicial igual a -2, periodo igual a 2 y la función de onda cuadrada igual a 1 como se observa en la Figura 4.43.

Figura 4.43 Función impulso con 30 armónicos.

Fuente: Desarrollado por el investigador

Comprobando de tal manera que se puede representar cualquier función lineal en una función periódica mediante la serie trigonométrica de Fourier teniendo como resultado una onda cuadrada periódica casi perfecta con un periodo de 2 y una amplitud de 1 como se muestra en la Figura 4.44.

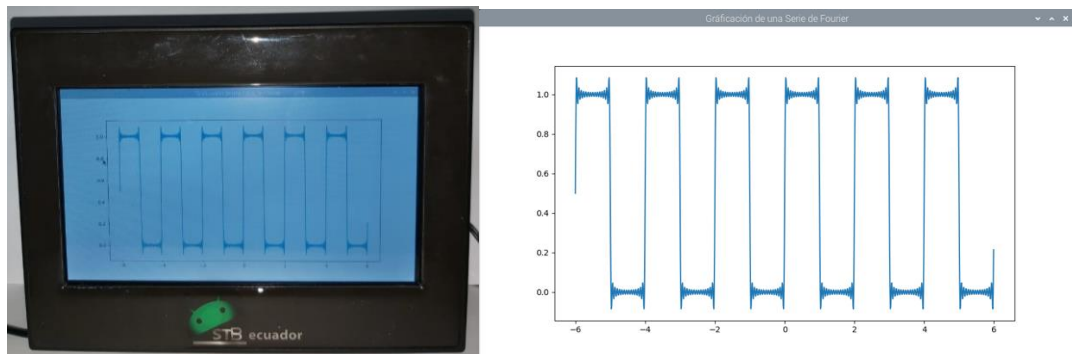


Figura 4.44 Gráfica impulso con 30 armónicos.

Fuente: Desarrollado por el investigador

Puede observarse que el resultado coincide con lo esperado: a mayor número de coeficientes de Fourier, la representación de la señal es más precisa.

4.7 Generación de señales con distintas potencias

Para el análisis espectral de las señales se utilizó la Transformada de Fourier. Esta permite determinar los componentes en frecuencia de una señal dada. El espectro lo componen todas las frecuencias de una señal. Dichas componentes se las puede observar con un gráfico en el dominio de la frecuencia.

Para un ordenador es imposible trabajar con una función continua debido a que esta función posee un número infinito de valores. Por esta razón se ha desarrollado la transformada discreta de Fourier en la cual se muestrea la señal en valores discretos y permite obtener el espectro de una señal, con valores finitos.

Adicional a ello, se desarrolló la transformada rápida de Fourier. Esto elimina la necesidad de cálculos repetitivos y reduce el tiempo requerido para los cálculos. Los sistemas actuales como GNU Radio usan FFT y los bloques de representación están en la biblioteca.

GNU Radio tiene herramientas y bibliotecas que permiten ver el espectro de cualquier tipo de señal sin usar la fórmula de la transformada de Fourier. El prototipo escanea frecuencias detectadas en el entorno. Es fácil de usar, mediante el control deslizante provisto por la aplicación la cual permite ajustar la frecuencia deseada mientras permanece dentro del rango RTL-SDR. Una vez dentro del espacio de trabajo de GNU

Radio, encontrará *WX FFT Sink*, *QT GUI Frequency Sink* y otras bibliotecas que permitirán observar el espectro de la señal en aplicaciones de simulación.

GNU Radio tiene diferentes tipos de interfaces para aplicaciones de monitorización, las más utilizadas son *WXGUI* y *QTGUI*, pero existen otras, pero en la práctica se utilizan estos dos tipos de interfaces. Para usarlo, debe configurar un bloque opcional para seleccionar el tipo de interfaz en la que ejecutar el emulador o la aplicación.

A continuación, se analizarán las diferentes frecuencias que proporciona el generador de frecuencia comercial ADF.

Para poder empezar con el análisis se debe conectar el generador de frecuencia mediante un cable SMA MACHO hacia el prototipo implementado conectándolo hacia la parte superior como se observa en la Figura 4.45.

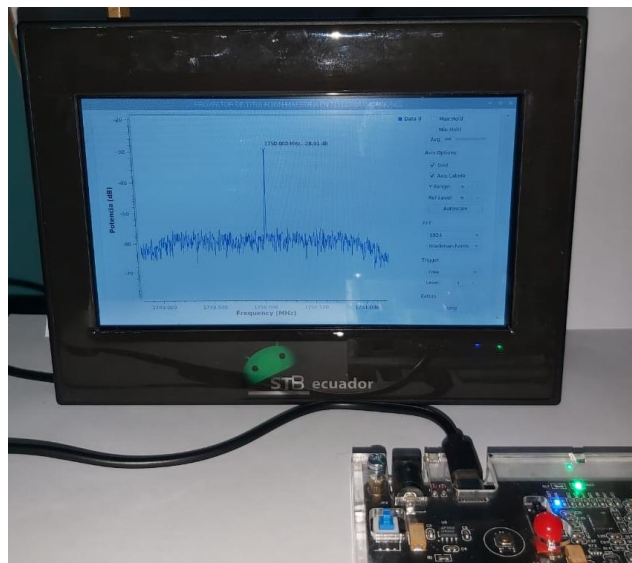


Figura 4.45 Conexión ADF configuración de frecuencia a 88 MHz.

Fuente: Desarrollado por el investigador

Una vez configurada la frecuencia de salida y potencia del generador ADF se procede a colocar los parámetros de ingreso mediante los controles deslizantes para su respectivo análisis como son; potencia de recepción 10 dB, frecuencia central 88 MHz y ancho de banda de 2.4 MHz como se observa en la Figura 4.46

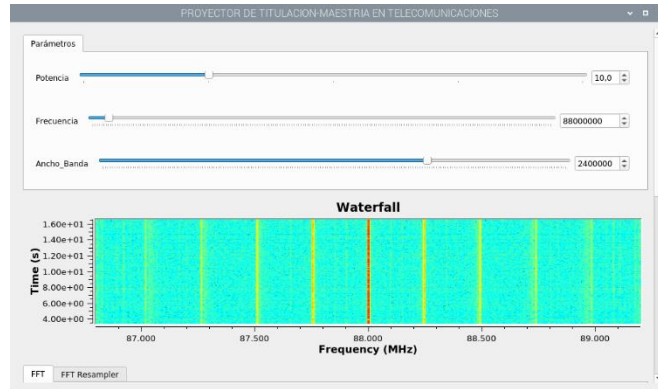


Figura 4.46 Configuración de parámetros-Frecuencia central a 88 MHz.

Fuente: Desarrollado por el investigador

Como resultado de dicha configuración tanto como los parámetros ingresados en el Software como en el dispositivo ADF se tiene la siguiente información que se puede visualizar en la gráfica Waterfall Figura en donde se muestra de manera sencilla y con un color rojo en forma de cascada que quiere decir que es la frecuencia en la que se encuentra resonando. Adicional a ello se tiene una segunda gráfica en donde se observa el espectrograma de dicha configuración antes mencionada dando como resultado en el eje de las x la frecuencia resonante y el eje de las y su potencia de recepción como se muestra en la Figura 4.47.

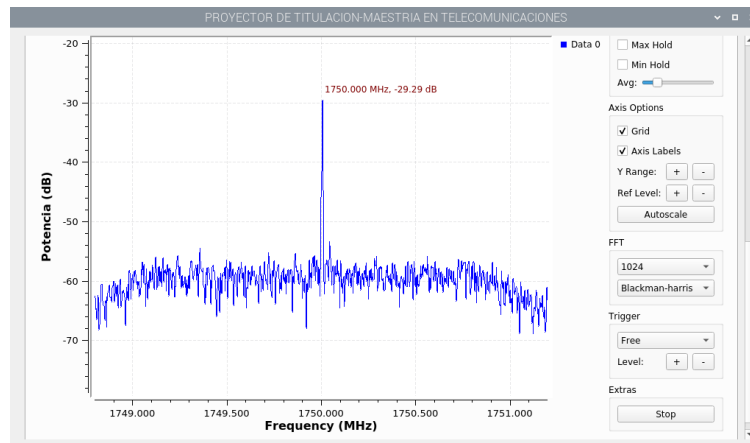


Figura 4.47 Espectrograma con frecuencia resonante a 88 MHz.

Fuente: Desarrollado por el investigador

Se realiza un acercamiento (Zoom) para observar más a detalle la frecuencia de resonancia de acuerdo a los parámetros ingresados como se observa en la Figura 4.48.

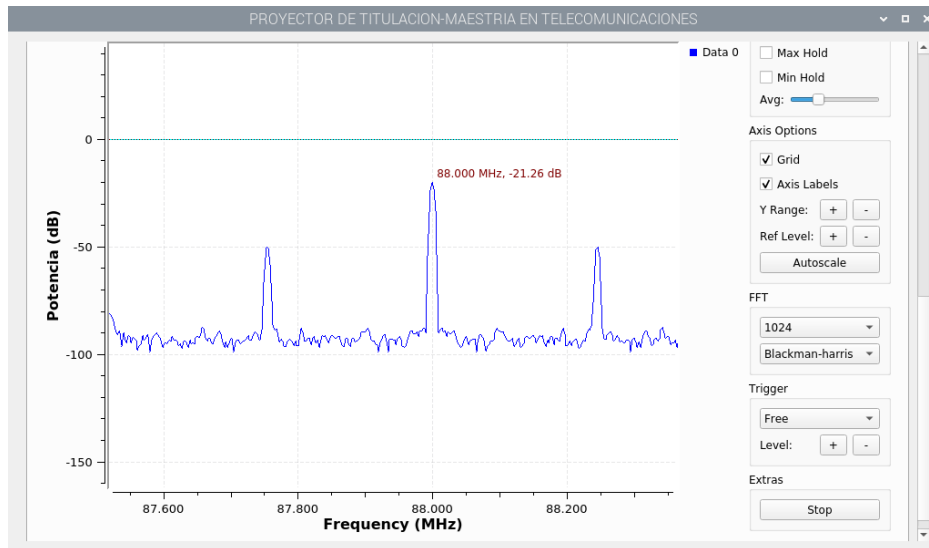


Figura 4.48 Acercamiento (Zoom) frecuencia resonante a 88 MHz

Fuente: Desarrollado por el investigador

Para el segundo análisis se procede nuevamente a configurar el dispositivo ADF a una frecuencia de 500 MHz con una potencia de salida de +5dB como se muestra en la Figura 4.49.

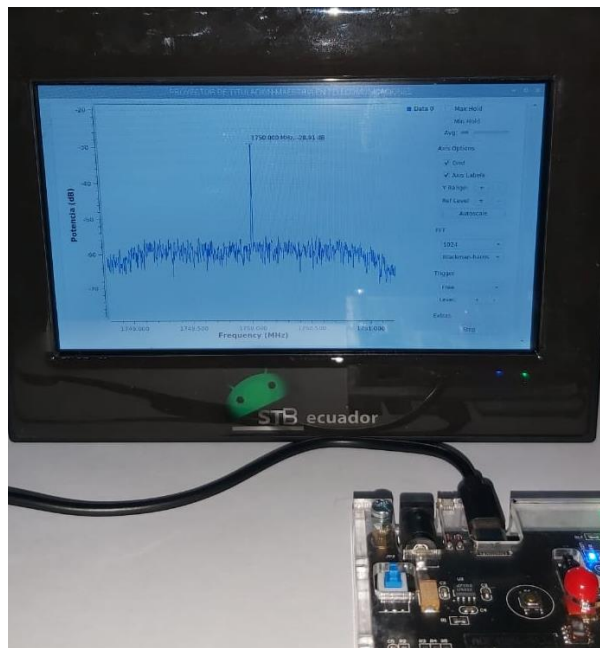


Figura 4.49 Configuración de prototipo – Frecuencia generada a 500MHz.

Fuente: Desarrollado por el investigador

En el software se configura una potencia de recepción de 10 dB, una frecuencia central de 500 MHz y un ancho de banda de 2.4 MHz como se observa en la Figura 4.50.

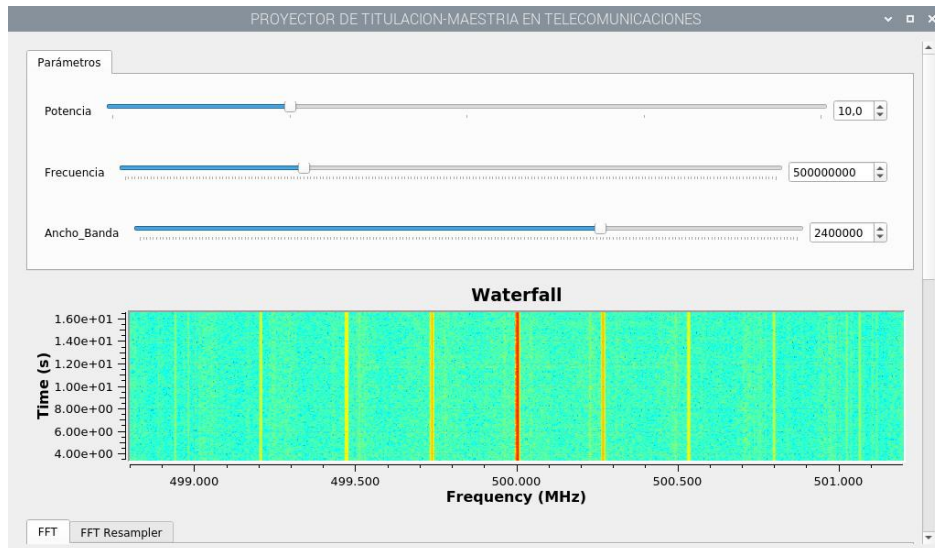


Figura 4.50 Configuración de parámetros-Frecuencia central a 500 MHz.

Fuente: Desarrollado por el investigador

Como resultados se obtiene que tanto en Waterfall como en el espectrograma efectivamente se encuentra resonando a la frecuencia configurada por el generador de frecuencia ADF entregándonos en su gráfica como respuesta una frecuencia de resonancia a 500.000 MHz con una potencia de recepción de -19.08 dB como se observa en la Figura 4.51.

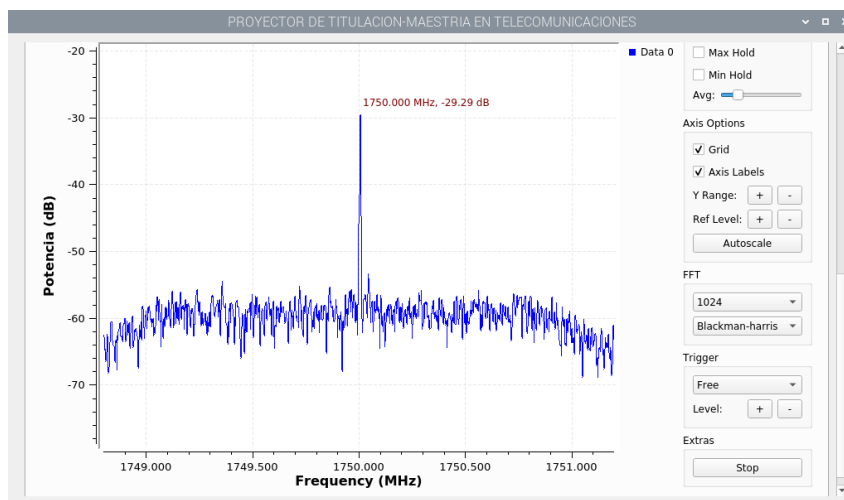


Figura 4.51 Espectrograma con frecuencia resonante a 500 MHz.

Fuente: Desarrollado por el investigador

Como tercer punto de análisis, se configura una frecuencia de 1000 MHz equivalente a 1 GHz con una potencia de + 5dB generado por el dispositivo ADF como se observa en la Figura 4.52.

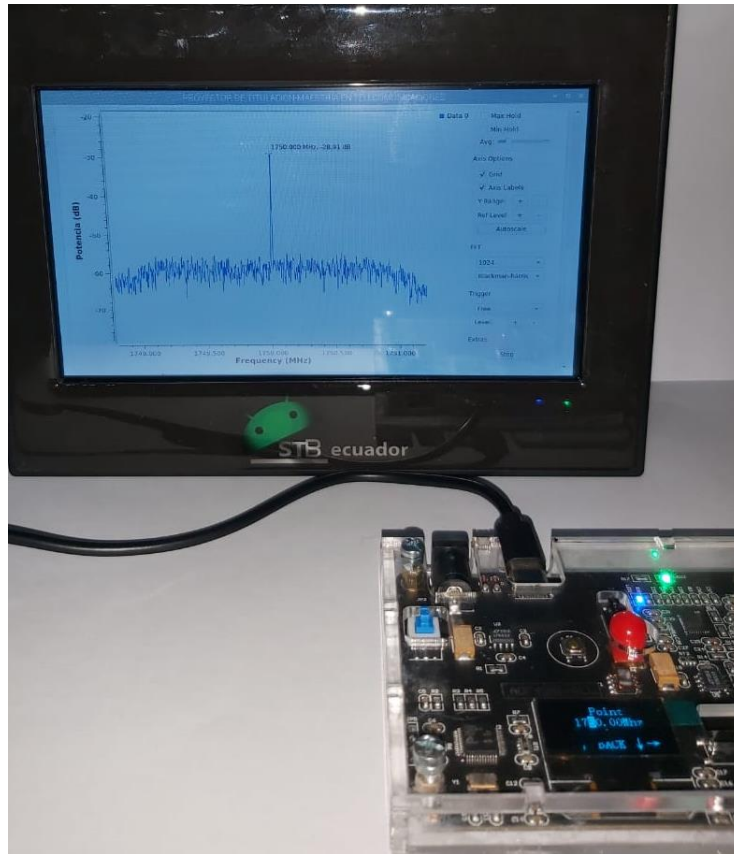


Figura 4.52 Conexión y configuración de prototipo – Frecuencia generada a 1000MHz.

Fuente: Desarrollado por el investigador

De igual manera se realiza el mismo procedimiento de configuración en el software, potencia de recepción de 10 dB, frecuencia central de 1000 MHz y un ancho de banda de 2.4 MHz como se observa en la Figura 4.53.

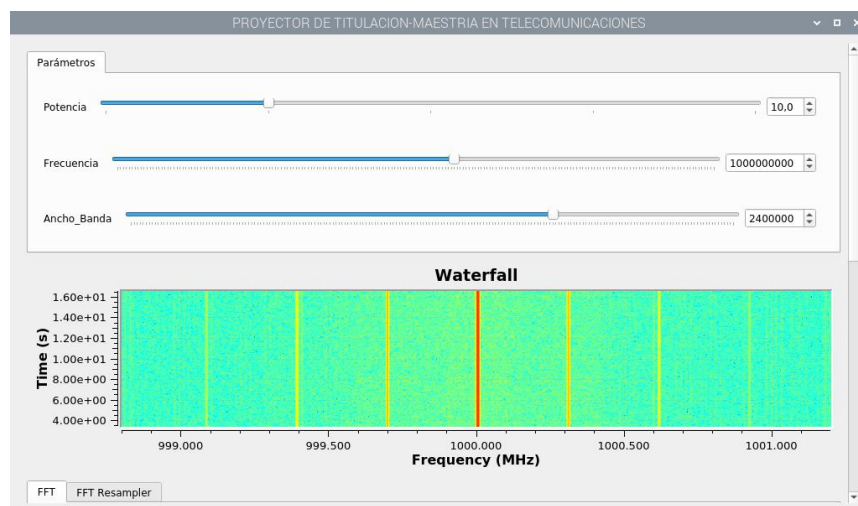


Figura 4.53 Configuración de parámetros-Frecuencia central a 1000 MHz.

Fuente: Desarrollado por el investigador

Se tiene como resultado de dicha generación que, su frecuencia de resonancia en el espectrograma es de 1000 MHz y con una potencia de recepción de -17.8 dB como se observa en la Figura 4.54.

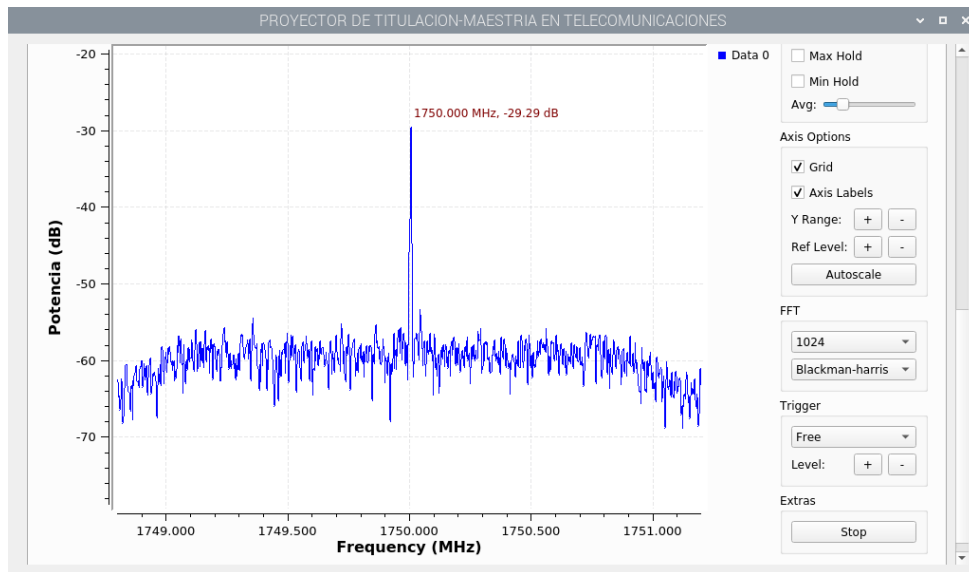


Figura 4.54 Espectrograma con frecuencia resonante a 1000 MHz

Fuente: Desarrollado por el investigador

Finalmente se pretende llevar al límite al dispositivo que se encarga de recibir y procesar las frecuencias, configurando de tal manera al generador de frecuencias a 1750 MHz equivalente a 1.75 GHz con una potencia de + 5 dB. Lo que quiere decir que el dispositivo RTL-SDR va a captar a su máxima configuración de recepción como se observa en la Figura 4.55.

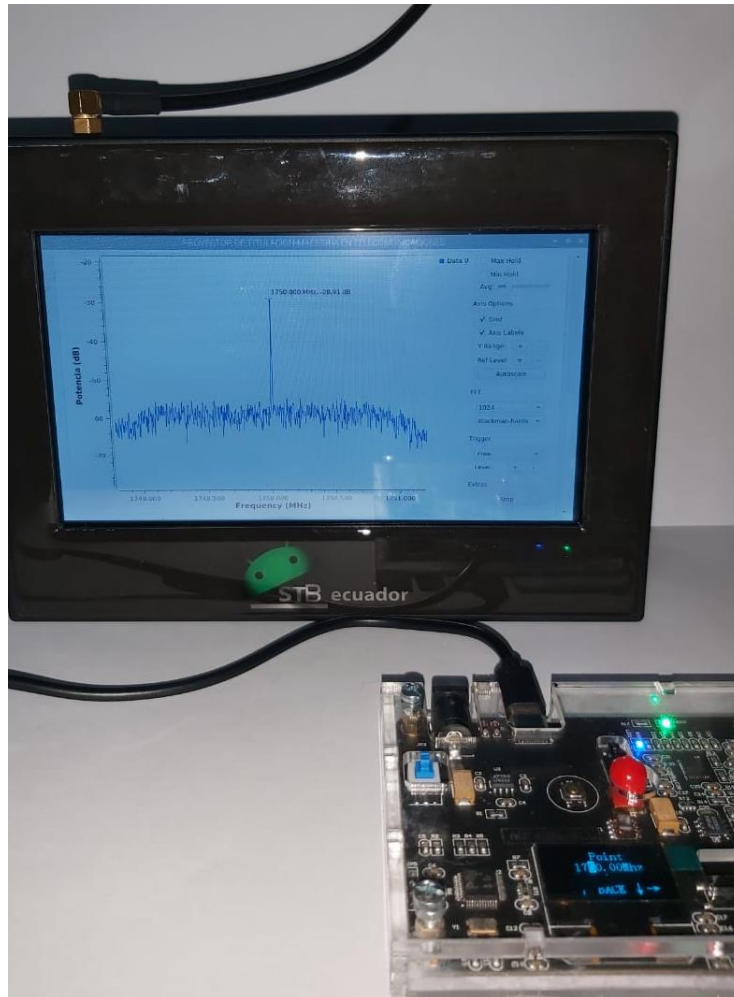


Figura 4.55 Configuración de prototipo – Frecuencia generada a 1750MHz

Fuente: Desarrollado por el investigador

Configuramos los parámetros para visualizar el espectro deseada con una potencia de recepción de 10, frecuencia central de 1750 MHz y ancho de banda de 2,4 MHz. Dicha configuración se detalla en la Figura. Se puede notar en el grafico del Waterfall su resonancia sin desviación alguna, pero con la diferencia que su potencia de recepción es muy escasa con respecto a las mediciones realizadas anteriormente como se observa en la parte inferior de la gráfica Figura 4.56.

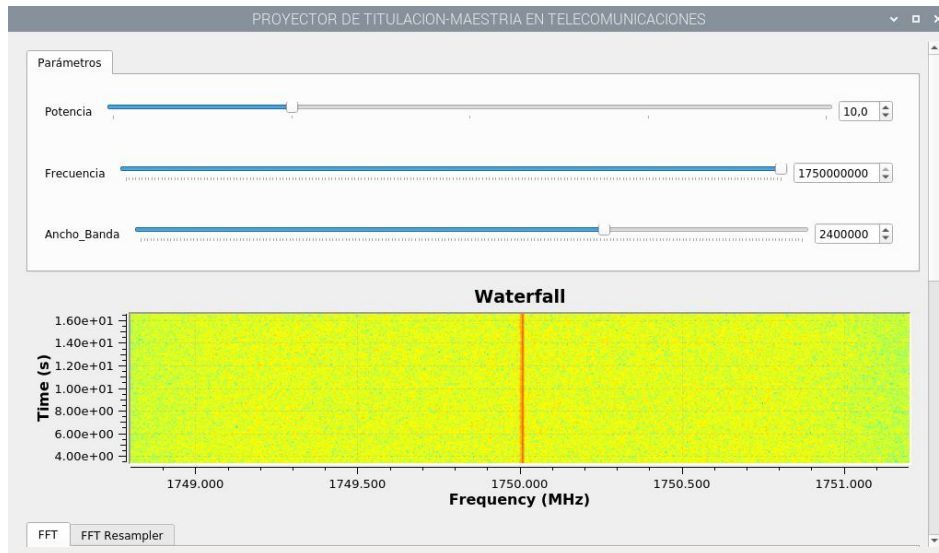


Figura 4.56 Configuración de parámetros-Frecuencia central a 1750 MHz.

Fuente: Desarrollado por el investigador

Se observa que como punto relevante que su potencia de recepción es de -29.29 dB y una frecuencia central de exactamente 1750 MHz como se observa en la Figura 4.57.

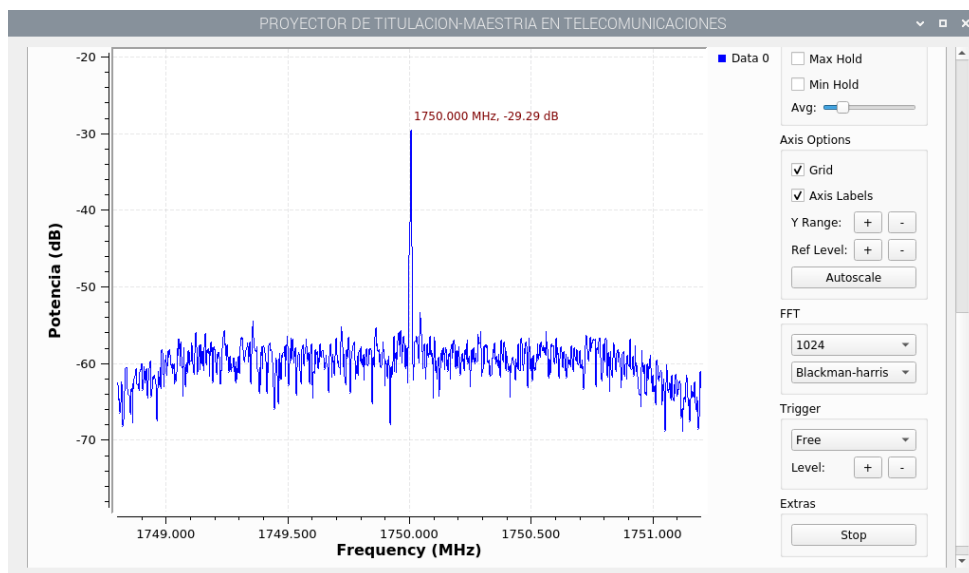


Figura 4.57 Espectrograma con frecuencia resonante a 1750 MHz.

Fuente: Desarrollado por el investigador

De acuerdo a los análisis realizados se pudo determinar que el generador de frecuencias ADF cumple a cabalidad con su propósito generando frecuencias prácticamente exactas las cuales pudieron ser comprobado mediante el Software desarrollado a continuación se detalla un resumen de las mediciones en la Tabla 4.5.

Tabla 4.3 Resumen de mediciones

Frecuencia Generada	Potencia de Tx-ADF	Frecuencia medida	Potencia de Rx-Software
88 MHz	+ 5 dB	88 MHz	-21.26 dB
500 MHz	+ 2 dB	500 MHz	-19.08 dB
1000 MHz	-1 dB	1000 MHz	-17.28 dB
1750 MHz	- 4 dB	1750 MHz	-29.29 dB

Fuente: Desarrollado por el investigador

4.8 Aplicación práctica para detección de frecuencias específicas

El análisis del sistema de transmisión FM de Ecuador es de suma importancia ya que es la mejor tecnología del país y el sistema más utilizado y escuchado desde sus inicios en el país.

De acuerdo a las señales analógicas medidas con fines prácticos, se puede observar el ancho de banda como una longitud, medida en hercios (Hz), de la extensión de frecuencias en la que se concentra la mayor potencia de la señal. Permitiendo calcular a partir de una señal medida mediante el análisis de Fourier que realiza internamente el Software. Las frecuencias que se encuentran entre esos límites se denominan también frecuencias efectivas como se puede observar en la Figura 4.58.

El espectro recibido de la señal se presenta en una gráfica en el dominio de la frecuencia, por lo cual en el prototipo de analizador de espectro se pueden visualizar a través de la pantalla LCD del prototipo como se muestra a continuación en la Figura 4.58 , donde se puede distinguir el patrón de la señal recibida.

A continuación, se realiza un análisis practico con un cierto rango de frecuencias específicas que va de 88 MHz a 90 MHz. Figura 4.58.

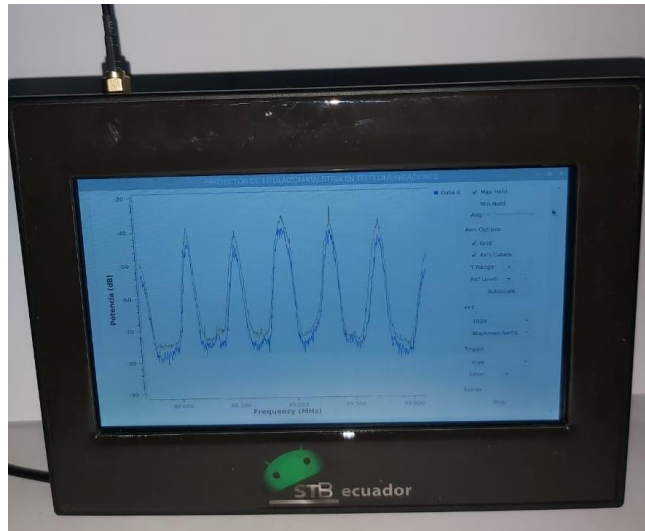


Figura 4.58 Rango de frecuencias específicas de 88 MHz a 90 MHz.

Fuente: Desarrollado por el investigador

Para poder captar las señales con una mejor eficiencia se ha integrado una antena omnidireccional espiral de acero tipo resorte, utiliza un conector SMA macho compatible con el prototipo que utiliza conector SMA hembra. Dicha antena tiene un rango frecuencia de: 698-960/1710-2700 MHz. Ancho de banda: 262/990 MHz. Potencia máxima de entrada: 50 W. Ganancia del amplificador: 12 dBi. V.S.W.R: ≤ 1.5 . Polarización: Vertical. Impedancia de entrada: 50 ohmios. A continuación, una presentación de prototipo Figura 4.59.

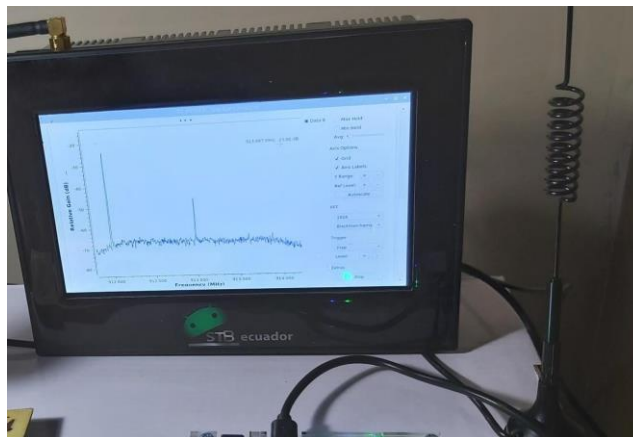


Figura 4.59 Prototipo más antena omnidireccional -captación de radiofrecuencia.

Fuente: Desarrollado por el investigador

Para dicho análisis se necesita configura en el Software la frecuencia central que será de 88.9 MHz como se observa en la Figura 4.60.



Figura 4.60 Configuración de parámetros radiofrecuencia.

Fuente: Desarrollado por el investigador

Una vez configurado se puede observar mediante la gráfica 1 Waterfall que existe 5 frecuencias detectadas correspondiente a la banda de FM con un diagrama en cascada resonante como se muestra en la parte inferior de la Figura 4.61.

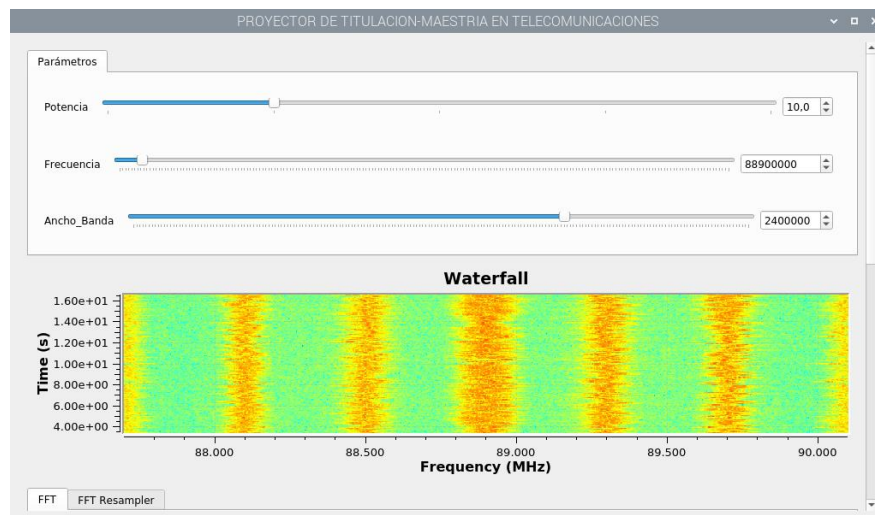


Figura 4.61 Frecuencia resonante en cascada Waterfall.

Fuente: Desarrollado por el investigador

En la gráfica número 2 correspondiente al espectrómetro, se observa el rango de frecuencias resonantes que se analizarán y describirán cada una de las frecuencias detectadas como se observa en la Figura 4.62.

Con el cursor del prototipo nos colocamos en el pico más alto de la primera frecuencia resonante que en este caso es de 88.1 MHz con una potencia de recepción de -38.81

dB cabe destacar que se activado en el panel de control la opción “Max Hol” permitiéndonos tener un dato preciso de la medición como se observa en la Figura 4.62.

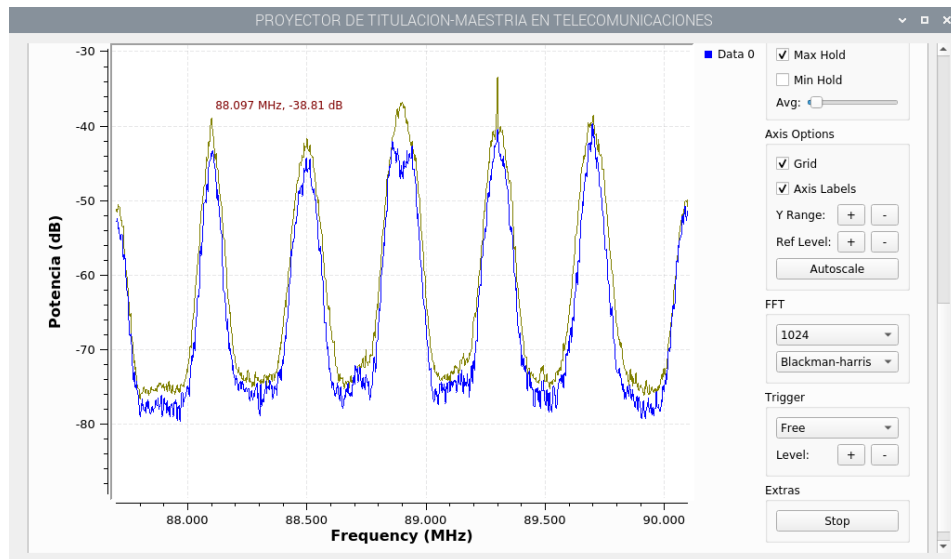


Figura 4.62 Frecuencia resonante detectada a 88.1 MHz.

Fuente: Desarrollado por el investigador

Como respuesta al segundo parámetro detectado se tiene que existe el se tiene el segundo pico de frecuencia a 88.5 MHz con una potencia de recepción de -39,82 dB Figura 4.63.

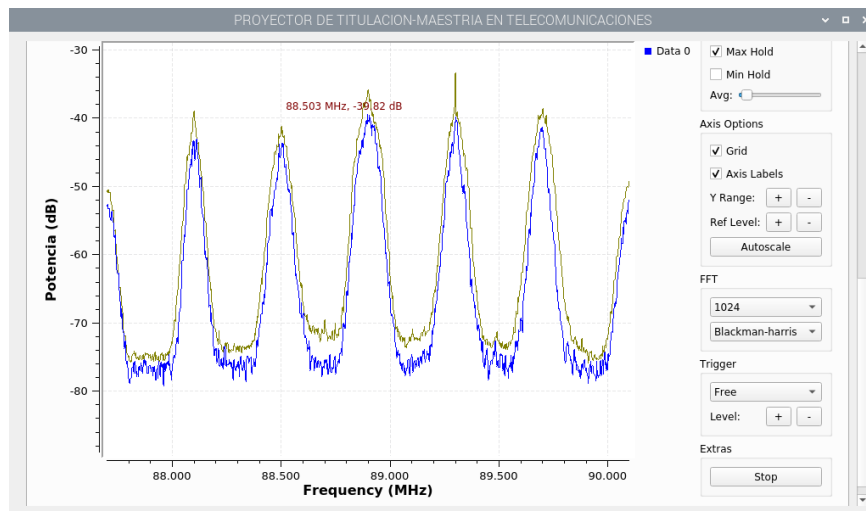


Figura 4.63 Frecuencia resonante detectada a 88.5 MHz.

Fuente: Desarrollado por el investigador

El tercer pico de radiofrecuencia es de 88.9 MHz con una potencia de recepción de -36.07 dB como se observa en la Figura 4.64.

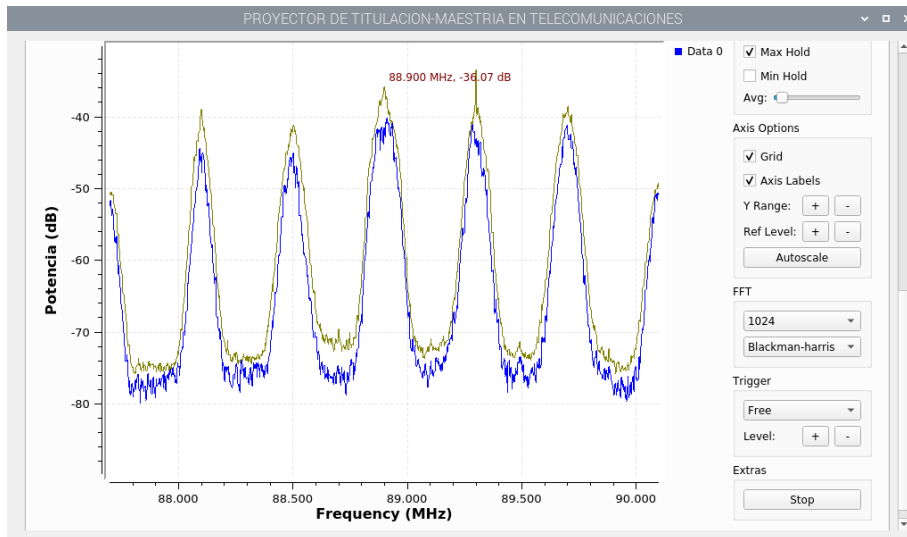


Figura 4.64 Frecuencia resonante detectada a 88.9 MHz.

Fuente: Desarrollado por el investigador

De igual forma se encuentra el cuarto pico con una frecuencia resonante de 89.3 MHz con una potencia de recepción de -33.82 dB como se observa en la Figura 4.65.

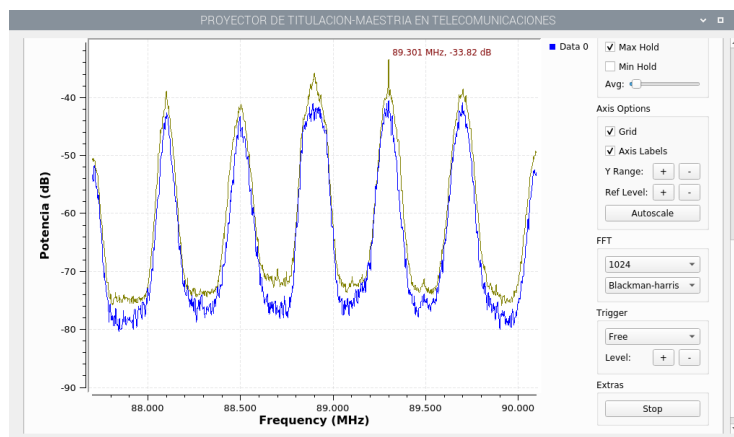


Figura 4.65 Frecuencia resonante detectada a 88.3 MHz.

Fuente: Desarrollado por el investigador

Por ultimo y no menos significativo se tiene la quinta frecuencia resonante que es de 89.7 MHz con una potencia de recepción de -38.11dB como muestra la Figura 4.66.

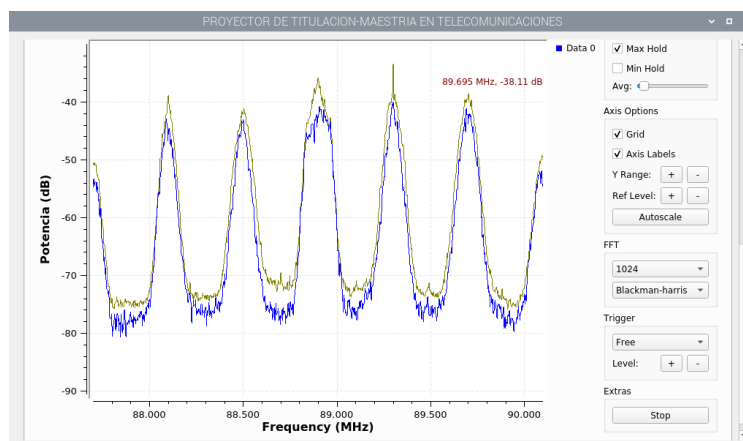


Figura 4.66 Frecuencia resonante detectada a 89.7 MHz

Fuente: Desarrollado por el investigador

Resultados obtenidos

Las características principales de un analizador de espectro es trabajar en el dominio de la frecuencia utilizando diferentes bloques de procesamiento de señales, por lo que se verificó que el prototipo implementado cumpla los parámetros básicos que tiene un analizador de espectro.

Se realizó una medición del ancho de banda de determinadas estaciones radiodifusoras comprendidas entre 88 y 90 MHz con el prototipo de analizador de espectro FFT, para ver el desempeño que tiene el analizador, donde se visualiza el ancho de banda con las mediciones entregadas por el prototipo, dando como resultado los siguientes datos en resumen de la Tabla 4.4:

Tabla 4.4 Resumen de mediciones

Frecuencia	Potencia de Rx	Banda lateral 1	Banda lateral 2	Ancho de Banda
88.1 MHz	-38.81 dB	88.00 MHz	88.20 MHz	200 KHz
88.5 MHz	-39.82 dB	88.40 MHz	88.60 MHz	200 KHz
88.9 MHz	-36.07 dB	88.80 MHz	89.10 MHz	200 KHz
88.3 MHz	-33.82 dB	89.20 MHz	89.40 MHz	200 KHz
89.7 MHz	-38.11dB	89.60 MHz	89.80 MHz	200 KHz

Fuente: Desarrollado por el investigador

4.9 Aplicación prácticas para almacenamiento de energía

La antena que se analiza tiene como único objetivo, transformar de energía electromagnética presente en el medio en diferentes bandas de frecuencia a corriente

directa (DC) o corriente alterna (AC), y su posterior almacenamiento en condensadores y baterías.

Este dispositivo busca proveer de energía a los dispositivos móviles que a diario circulan dentro de las instalaciones de la FISEI, misma que permita la no dependencia de un cargador eléctrico común, a través de estas necesidades de análisis y desarrollo busca la mejor opción en la elaboración de un sistema de antenas capaz de proveer la energía requerida para la alimentación o carga de dispositivos.

Para ello se realizan las mediciones verificando que cumpla con los parámetros simulados para dicho propósito.

Se realiza la conexión del prototipo con la antena denominada “Geometría antena de estructura planar con resonador híbrido (HRPSA)” Figura 4.67



Figura 4.67 Conexión de prototipo a antena HRPSA.

Fuente: Desarrollado por el investigador

Se configura los parámetros para dicho análisis con una potencia de recepción de 10 dB, ancho de banda de 2,4 KHz y una frecuencia central de 914 MHz que representa uno de los parámetros más importantes en el anal ya que esta es la frecuencia de resonancia que opera la antena HRPSA. Como se observa en la parte superior de la Figura 4.68 .

Para visualizar la mayor concentración de potencia de señal en una forma más simple, se activa el Diagrama de Cascada, que permite diferenciar donde se concentra más la energía de cada señal recibida con la escala de colores visualizados, siendo el rojo el

de mayor potencia y el verde el de menor potencia. Como se observa en la parte inferior de la Figura 4.68.

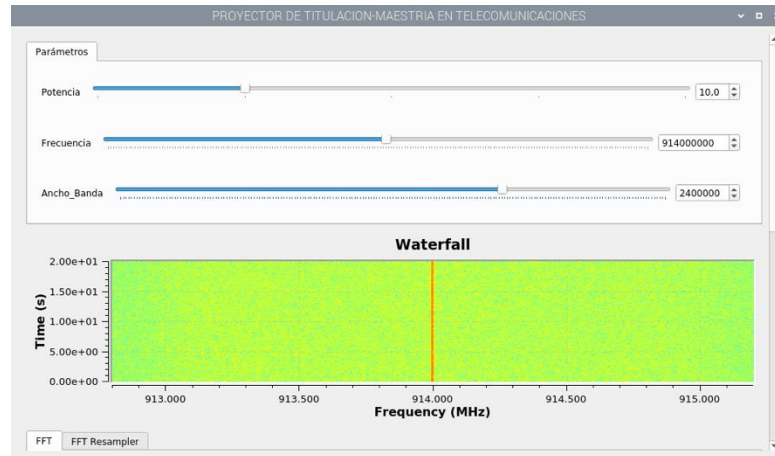


Figura 4.68 Configuración de parámetros antenna HRPSA

Fuente: Desarrollado por el investigador

Una vez configurado los parámetros de recepción se procede a configurar el generador de frecuencia ADF a 914 MHz como se muestra en la Figura 4.69.

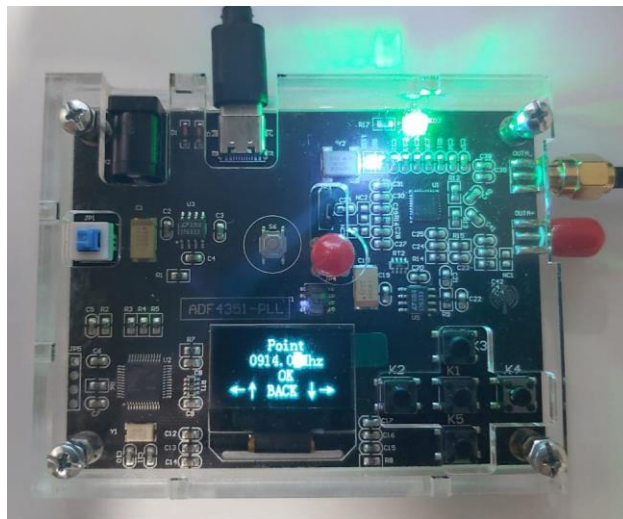


Figura 4.69 Configuración generador de frecuencia ADF

Fuente: Desarrollado por el investigador

posteriormente se coloca la antena omnidireccional a la salida del generador de frecuencia para su transmisión. Cabe recalcar que la antena omnidireccional cumple a cabalidad con las especificaciones técnicas que requiere la antena HRPSA en su recepción ya que su rango de frecuencia es de 698-960/1710-2700 MHz. En la Figura se observa la resonancia presente en la recepción con una frecuencia de 914.0 MHz y potencia de recepción de -32.47 dB. Figura 4.70

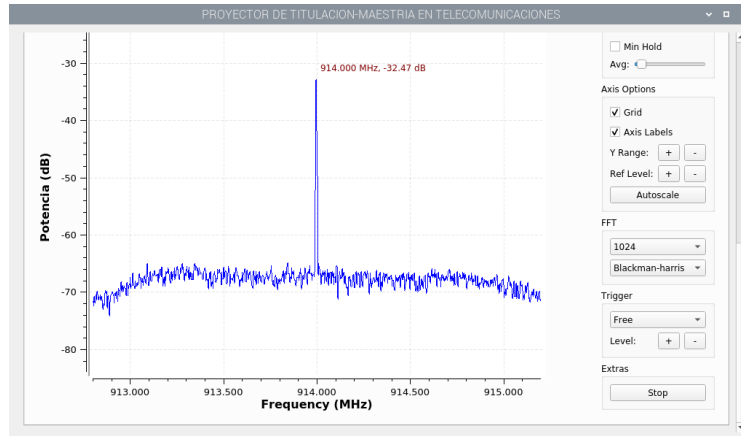


Figura 4.70 Resonancia de antena HRPSA a 914 MHz

Fuente: Desarrollado por el investigador

Se realiza una segunda prueba generando una frecuencia de 914.5 MHz teniendo como resultado una potencia de recepción de -37.19 dB lo cual significa que se encuentra dentro el rango de resonancia según las simulaciones obtenidas en dicha investigación como se visualiza mediante las gráficas de Waterfall y Espectrograma Figura 4.71.

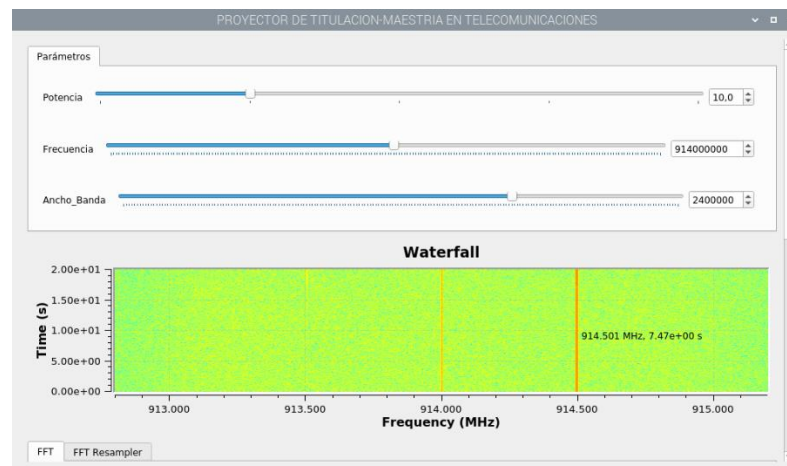


Figura 4.71 Gráfica en cascada Waterfall en resonancia 914.5 MHz.

Fuente: Desarrollado por el investigador

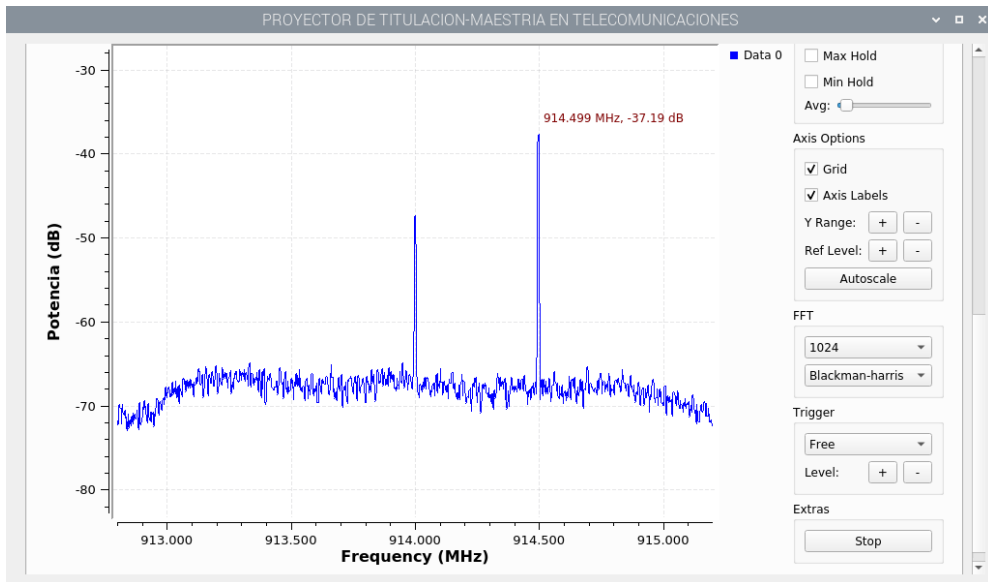


Figura 4.72 Resonancia de antena HRPSA a 914.5 MHz.

Fuente: Desarrollado por el investigador

El tercer análisis se obtiene de igual manera una resonancia de 915 MHz tanto en la gráfica de Waterfall Figura 4.73, como en el Espectrograma Figura 4.74, con un nivel de potencia de recepción de -38.31 dB.

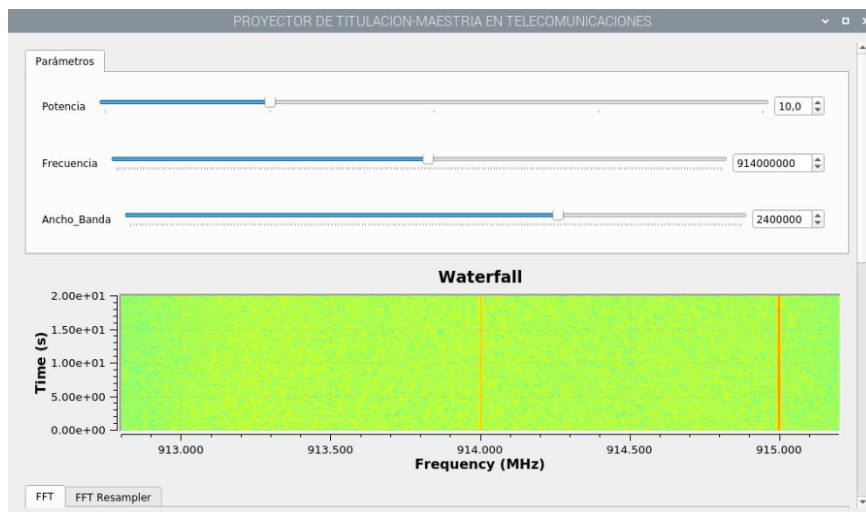


Figura 4.73 Gráfica en cascada Waterfall en resonancia 915 MHz.

Fuente: Desarrollado por el investigador

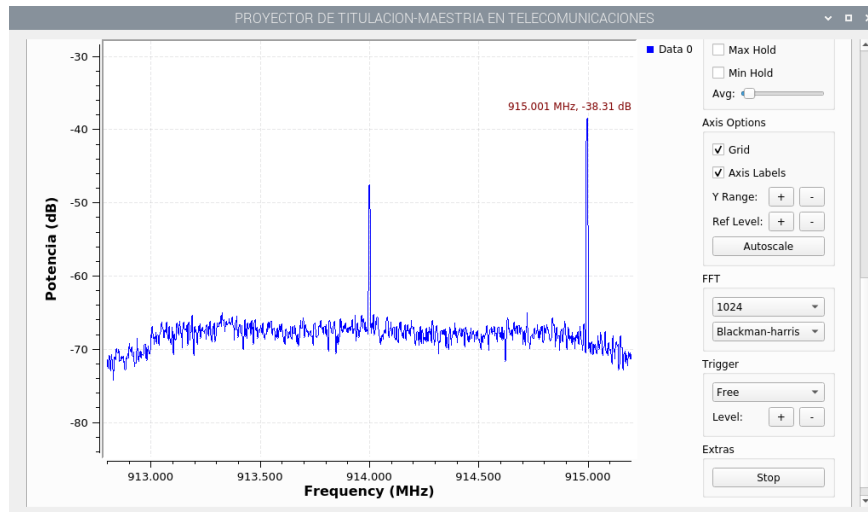


Figura 4.74 Resonancia de antena HRPSA a 915 MHz.

Fuente: Desarrollado por el investigador

Se configura nuevamente en el generador de frecuencias la nueva señal a 913.5 MHz teniendo como resultado la misma frecuencia en resonancia que se ha generado con una potencia de recepción de -36.00 dB tanto en Waterfall Figura 4.75, como en el Espectrograma Figura 4.76.

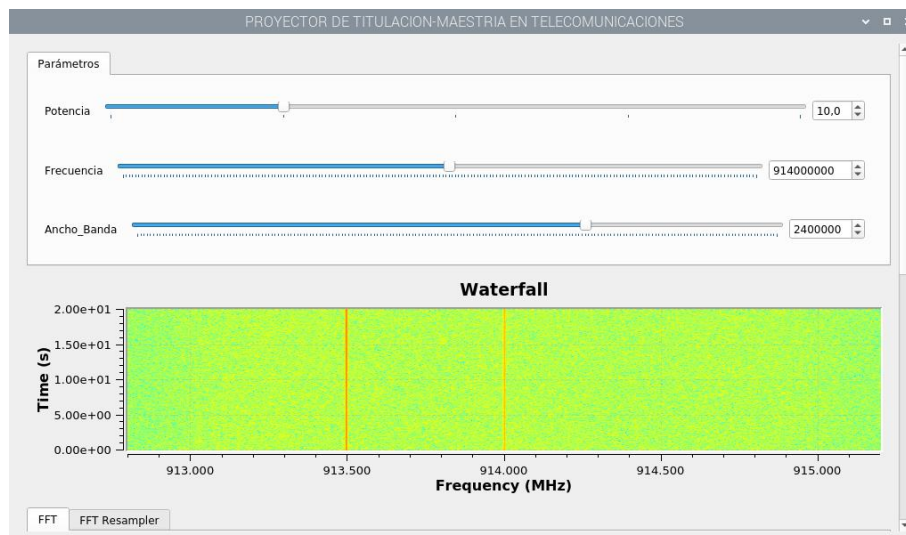


Figura 4.75 Gráfica en cascada Waterfall en resonancia 913.5 MHz.

Fuente: Desarrollado por el investigador

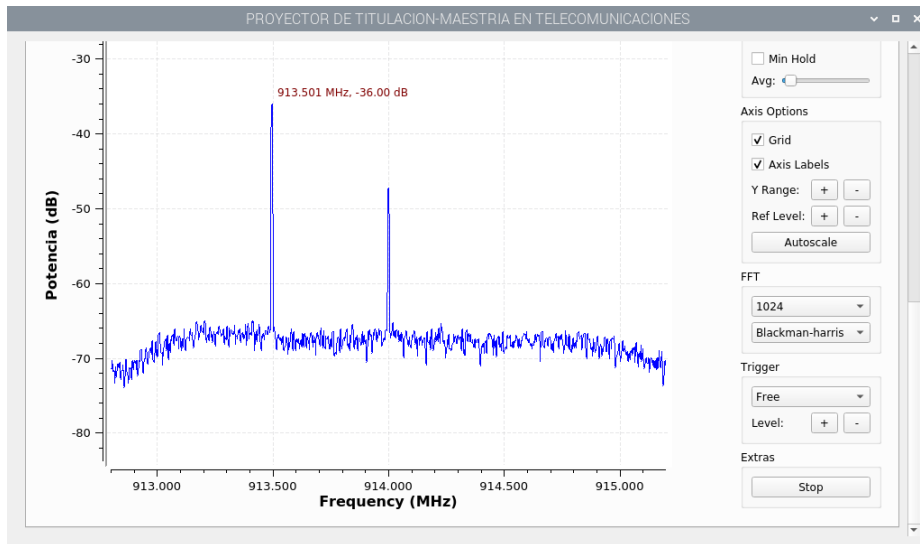


Figura 4.76 Resonancia de antena HRPSA a 913.5 MHz

Fuente: Desarrollado por el investigador

Finalmente se configura a una frecuencia de 913 MHz obteniendo como resultado una recepción de frecuencia a 913 MHz y una potencia de recepción de -37.91 dB como se muestra en la Figura 4.77 y Figura 4.78.

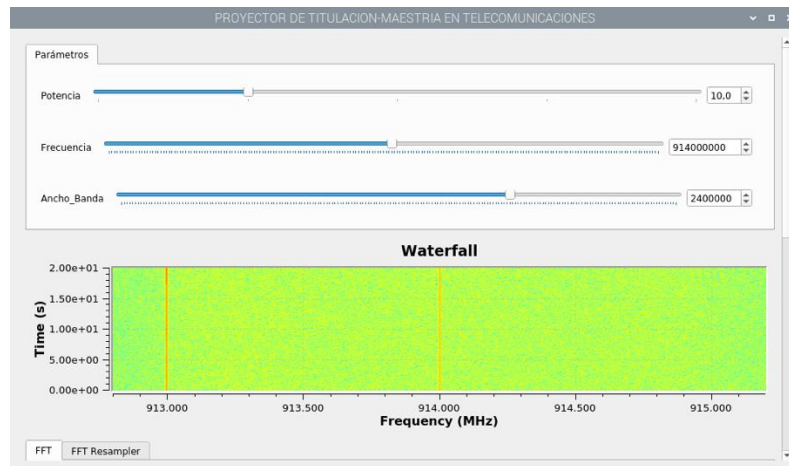


Figura 4.77 Gráfica en cascada Waterfall en resonancia 913 MHz

Fuente: Desarrollado por el investigador

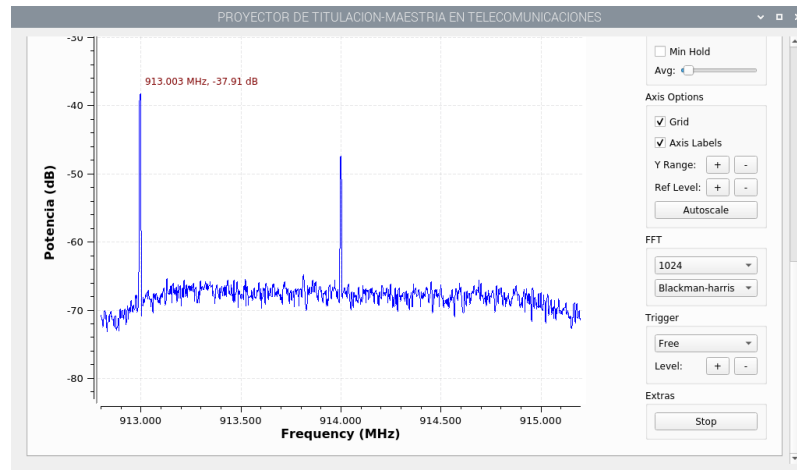


Figura 4.78 Resonancia de antena HRPSA a 913 MHz

Fuente: Desarrollado por el investigador

De acuerdo a los análisis realizados se pudo determinar que la antena HRPSA realmente resuena a la frecuencia que fue diseñada a continuación se detalla un resumen de las mediciones en la Tabla 4.5.

Tabla 4.5 Resumen de mediciones Antena HRPSA

Frecuencia Generada	Potencia de Tx- ADF	Frecuencia medida- Software	Potencia de Rx- Software
913.0 MHz	+ 5 dB	913.0 MHz	-37.91 dB
913.5 MHz	+ 5 dB	913.5 MHz	-36.00 dB
914.0 MHz	+ 5 dB	914.0 MHz	- 32.47 dB
914.5Mhz	+ 5 dB	914.5Mhz	-37.19 dB
915.0 MHz	+ 5 dB	914.5Mhz	-38.31 dB

Fuente: Desarrollado por el investigador

CAPÍTULO V

CONCLUSIONES, RECOMENDACIONES

5.1 Conclusiones

- Mediante la investigación del estado del arte, literatura fundamentada en teorías investigaciones e implementaciones en diferentes artículos científicos y libros basados en publicaciones de alto impacto se logró implementar un prototipo de simulación basadas en la aplicación de series trigonométricas de Fourier con motivo didáctico y a su vez practico. El estudio y caracterización permitió, escoger los recursos de hardware más adecuados para satisfacer los requerimientos del prototipo implementado, mediante un análisis de dispositivos capaces de generar y recibir señales periódicas permitiendo reforzando el conocimiento en el área de comunicaciones.
- El modelo del sistema integrado es capaz de resolver gráficamente y matemáticamente las series de Fourier, mediante su arquitectura y selección de equipos, así como también se puede analizar antenas de propósito específico de captación de energía, logrando un diseño modular, tanto en software como en hardware, que de acuerdo a la selección que realice el usuario, permita modificar la frecuencia de la señal generada por el dispositivo en un intervalo de 500MHz-1750 MHz verificando la funcionalidad del prototipo implementado.
- El prototipo pudo resolver gráficamente y matemáticamente las series de Fourier con funciones periódicas variables en el tiempo, captación de señales de radio frecuencia en el rango de 500 KHz-1.75 GHz, generación de frecuencias con un rango máximo de 35 Mhz-4.4 GHz, así como también se puede analizar antenas de propósito específico de captación de energía con una resonancia en frecuencia de 914.0 MHz, logrando verificar la funcionalidad del prototipo implementado.
- Las mediciones físicas realizadas en el laboratorio permitieron comprobar el desempeño correcto del dispositivo con respecto a la simulación con diferentes números de armónicas, Generación de señales con distintas potencias, Aplicación práctica para detección de frecuencias específicas, Aplicación

prácticas para almacenamiento de energía. Concluyendo que los fundamentos de series trigonométrica de Fourier constituyen la base teórica y didáctica para los estudiantes ya que el aporte y lo novedoso del prototipo implementado son las aplicaciones.

5.2 Recomendaciones

- Mejorar el método de visualización de resultados mediante el estudio con respecto al aplicativo de Series trigonométricas de Fourier utilizando paquetes y librerías especializados, cuya función sea representar datos de resultados científicos de alto nivel capaces de dibujar diversos tipos de diagramas y gráficos 2D/3D.
- Se recomienda mejorar el aplicativo del prototipo para calcular el ancho de banda mediante, la programación y la automatización del análisis de señales y en cuanto a estética y usabilidad, mejorar las pantallas mostradas con botones más interactivos y desplazamientos entre pantallas más fluidas.
- Se recomienda probar otros tipos de decodificadores y receptores para comparar el rendimiento de los dispositivos utilizados requeridos para el análisis de señales periódicas y de radiofrecuencia con un rango de modulación de hasta 4.4GHz, el cual se hace referencia a la frecuencia máxima que opera el generador de frecuencia ADF 4351.
- Se recomienda utilizar este prototipo para verificar la configuración de la señal transmitida por las estaciones de radio. Esto permitirá probar periódicamente la señal respecto a los parámetros establecidos y controlados por ARCOTEL.

BIBLIOGRAFÍA

- [1] A. Adnani, J. Duplicy, y L. Philips, «Spectrum analyzers today and tomorrow: part 1 towards filterbanks-enabled real-time spectrum analysis», *IEEE Instrum. Meas. Mag.*, vol. 16, n.º 5, pp. 6-11, oct. 2013, doi: 10.1109/MIM.2013.6616284.
- [2] T. Lu y F. Chen, «Multiwfn: A multifunctional wavefunction analyzer», *J. Comput. Chem.*, vol. 33, n.º 5, pp. 580-592, feb. 2012, doi: 10.1002/jcc.22885.
- [3] K. Vachhani y R. A. Mallari, «Experimental study on wide band FM receiver using GNURadio and RTL-SDR», en *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Kochi, India, ago. 2015, pp. 1810-1814. doi: 10.1109/ICACCI.2015.7275878.
- [4] P. Fuxjäger *et al.*, «IEEE 802.11p Transmission Using GNURadio», p. 4.
- [5] V. Peluso, P. Vancorenland, A. M. Marques, M. S. J. Steyaert, y W. Sansen, «A 900-mV low-power $\Delta\Sigma$ A/D converter with 77-dB dynamic range», *IEEE J. Solid-State Circuits*, vol. 33, n.º 12, pp. 1887-1897, dic. 1998, doi: 10.1109/4.735528.
- [6] A. A. Del Barrio *et al.*, «HackRF + GNU Radio: A software-defined radio to teach communication theory», *Int. J. Electr. Eng. Educ.*, p. 002072091986814, ago. 2019, doi: 10.1177/0020720919868144.
- [7] «Real-Time Data Transfer Based on Software Defined Radio Technique using Gnu radio/USRP», *Int. J. Eng. Adv. Technol.*, vol. 9, n.º 1, pp. 279-288, oct. 2019, doi: 10.35940/ijeat.A1140.109119.
- [8] «CUIEETGimenez.pdf». Accedido: dic. 04, 2020. [En línea]. Disponible en: <https://digibuo.uniovi.es/dspace/bitstream/handle/10651/48262/CUIEETGimenez.pdf;jsessionid=78BF5FD7EB50171D1E949E53239C9CBA?sequence=1>
- [9] E. A. Zamudio, R. A. A. Ramírez, y M. M. Rivera, «Obtención y Validación de la Serie de Fourier para el Análisis del Contenido Armónico en un Control por Fase», n.º 108, p. 19.
- [10] R. Farfán y F. Romero, «Learning Situation for the Trigonometric Fourier Series from a Socio-epistemological Stand Point | Situación de Aprendizaje para la Serie Trigonométrica de Fourier desde la Teoría Socioepistemológica», vol. 21, pp. 28-48, abr. 2019, doi: 10.17648/acta.scientiae.v21iss2id5019.

- [11] J. Silva, J. Barreto, y S. Espinosa, «SERIES DE FOURIER DE SEÑALES COMPLEJAS», p. 7.
- [12] «tesisJesusDuarte.pdf».
- [13] «software_libre_viable_justo.pdf». Accedido: dic. 04, 2020. [En línea]. Disponible en: http://190.57.147.202:90/xmlui/bitstream/handle/123456789/450/software_libre_viable_justo.pdf?sequence=1
- [14] marvin b, «Introduccion a las telecomunicaciones», *Taringa!*, feb. 19, 2012. https://www.taringa.net/+ciencia_educacion/introduccion-a-las-telecomunicaciones_131af6 (accedido jul. 11, 2021).
- [15] J. Peña, «Principio de las Comunicaciones Jose Briceño ULA-Electrica (1) (1)», Accedido: jul. 11, 2021. [En línea]. Disponible en: https://www.academia.edu/11803894/Principio_de_las_Comunicaciones_Jose_Brice%C3%B1o_ULA_Electrica_1_1_
- [16] L. F. Chaparro, *Signals and systems using MATLAB*. Burlington, MA: Academic Press, 2010.
- [17] W. Tomasi, «Sistemas de comunicaciones electrónicas». Pearson Educación, México, 2010.
- [18] «32381.pdf». Accedido: jul. 12, 2021. [En línea]. Disponible en: <http://bdigital.ula.ve/storage/pdf/32381.pdf>
- [19] «Tema II Técnicas de Modulación de Amplitud - ppt video online descargar». <https://slideplayer.es/slide/25430/> (accedido jul. 12, 2021).
- [20] J. W. Nilsson y S. A. Riedel, *Circuitos eléctricos*. Madrid: Pearson Educación, 2011.
- [21] J. Soto, «Series de Fourier | pimedios». <http://pimedios.jesussoto.es/2011/01/28/series-de-fourier/> (accedido jul. 11, 2021).
- [22] «OrtegaCardenasNicolasDanielAndres2016.pdf». Accedido: jul. 21, 2021. [En línea]. Disponible en: <https://repository.javeriana.edu.co/bitstream/handle/10554/36827/OrtegaCardenasNicolasDanielAndres2016.pdf?sequence=4&isAllowed=y>
- [23] «SymPy». <https://www.sympy.org/en/index.html> (accedido jul. 11, 2021).
- [24] «NumPy». <https://numpy.org/> (accedido jul. 11, 2021).

- [25] «Python básico desde CERO; GRATIS y en ESPAÑOL || Pythones», ene. 23, 2019. <https://pythones.net/> (accedido jul. 11, 2021).
- [26] «Inicio - Spyder IDE». <https://www.spyder-ide.org/> (accedido jul. 21, 2021).
- [27] C. R. Irick, «Enhancing GNU Radio for Hardware Accelerated Radio Design», p. 88.
- [28] «Radio GNU». https://wiki.gnuradio.org/index.php/Main_Page (accedido jul. 20, 2021).
- [29] R. M. Prieto, «Manual de Instalación y uso de GNURadio», p. 26.
- [30] «Getting_Started_with_Raspberry_Pi.pdf». Accedido: jul. 21, 2021. [En línea]. Disponible en: http://www.multimedialab.be/doc/erg/2018-2019/Raspberry_Pi/Getting_Started_with_Raspberry_Pi.pdf
- [31] I. P. Domínguez y J. J. M. Fuentes, «Laboratorio de Comunicaciones Digitales Radio Definida por Software», p. 166.
- [32] «Buy RTL-SDR Dongles (RTL2832U)», *rtl-sdr.com*, abr. 11, 2013. <https://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/> (accedido ago. 05, 2021).
- [33] «NPeiro-TFG-IEE-2016 .pdf». Accedido: jul. 16, 2021. [En línea]. Disponible en: <http://200.0.183.55/bitstream/handle/123456789/321/NPeiro-TFG-IEE-2016%20.pdf?sequence=1&isAllowed=y>
- [34] «ADF4351 Datasheet and Product Info | Analog Devices». <https://www.analog.com/en/products/adf4351.html#product-overview> (accedido jul. 23, 2021).
- [35] T. R. P. Foundation, «Raspberry Pi 4 Model B specifications», *Raspberry Pi*. <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/> (accedido ago. 05, 2021).

ANEXOS

Anexo 1.

Instalación y Configuración de Raspberry Pi Os

Existe una gran cantidad de métodos para instalar Raspberry Pi OS u otros Sistemas Operativos en Raspberry, una de ellas es usando Raspberry Pi Imager. Este método de instalación funciona para cualquier Raspberry Pi, desde la Raspberry Pi 4 a la más antigua Raspberry Pi 1.

Preparación de la tarjeta micro-SD

El primer paso es formatear la tarjeta micro SD en Fat32. Recordar que es recomendable utilizar una tarjeta micro SD de clase 10, la utilización de una tarjeta de calidad inferior hará que la Raspberry Pi funcione más lenta.

Instalación de Raspberry Pi Imager

Raspberry Pi Imager es un software que se puede instalar en Windows, Linux y MacOs. Lo primero será descargarlo desde su página web <https://www.raspberrypi.org/software/> como se muestra en la Figura A 1. O copiando el link y selecciona la descarga de acuerdo a nuestro Sistema Operativo <https://www.raspberrypi.org/downloads/>

Una vez que lo descargamos procedemos ha instalarlo en nuestro sistema.



Figura A 1 Página web oficial

Fuente: Desarrollado por el investigador

Instalación Raspberry Pi OS en Tarjeta Micro SD

Ahora procedemos a abrir Raspberry Pi Imager Figura A 2.



Figura A 2 Software Raspberry Pi Imager

Fuente: Desarrollado por el investigador

Pulsamos sobre el botón CHOOSE OS y seleccionamos la primera opción Raspberry Pi Os como aplicación recomendada Figura A 3.

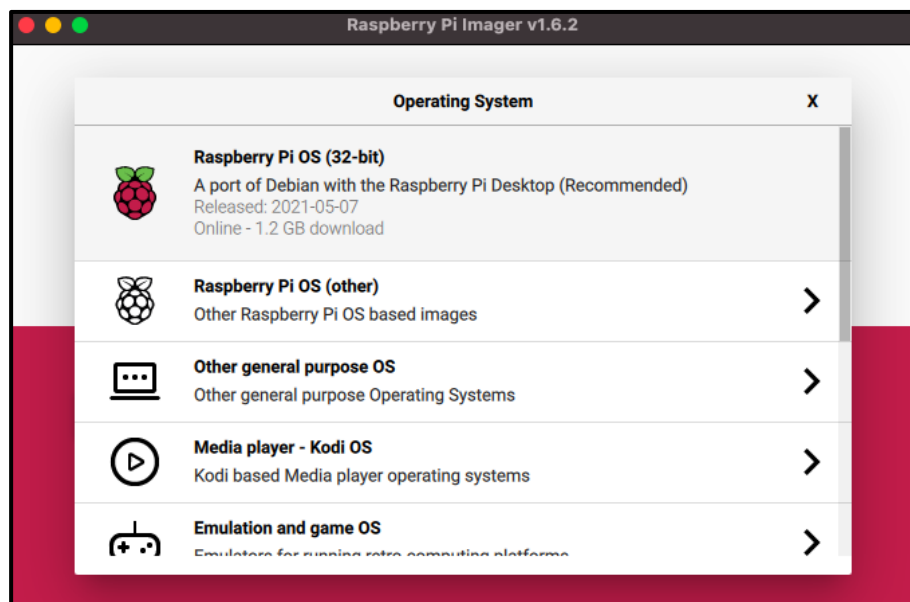


Figura A 3 Selección del sistema operativo

Fuente: Desarrollado por el investigador

Seguidamente pulsamos sobre el botón CHOOSE SD CARD y seleccionamos nuestra tarjeta micro SD Figura A 4.

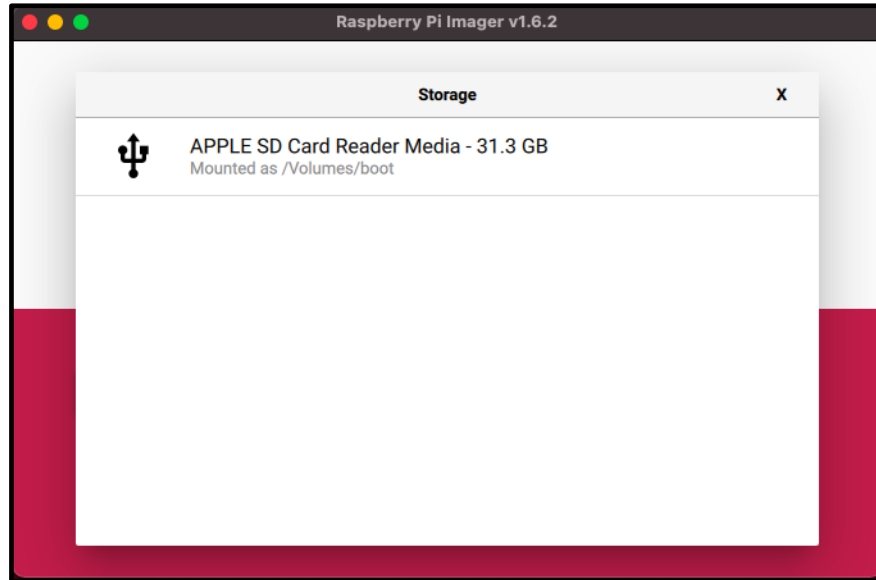


Figura A 4 Selección de tarjeta micro SD

Fuente: Desarrollado por el investigador

Por último, pulsamos sobre el botón WRITE y el programa empezará la descarga de Raspberry Pi OS instalándola en la microSD, este proceso puede tardar varios minutos Figura A 5.

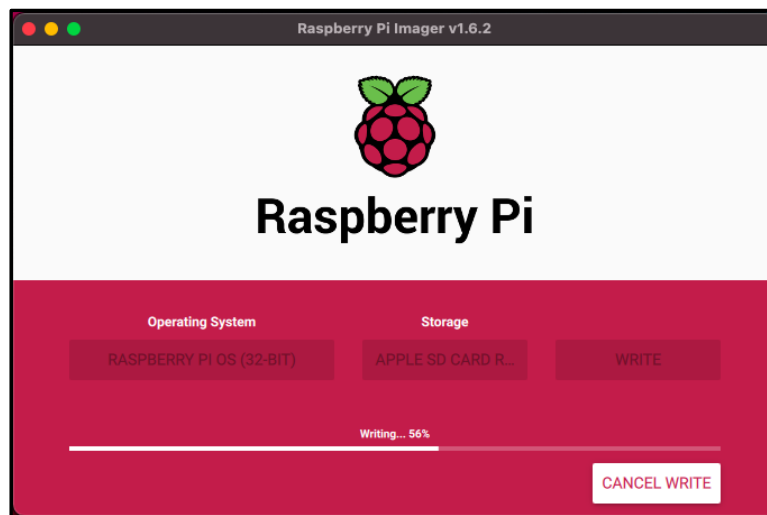


Figura A 5 Descarga de Raspberry Pi OS

Fuente: Desarrollado por el investigador

Cuando termine se mostrará un mensaje marcando que ha finalizado como se muestra en la Figura A 6. Ahora solo queda extraer la tarjeta e introducirla en la Raspberry pi.

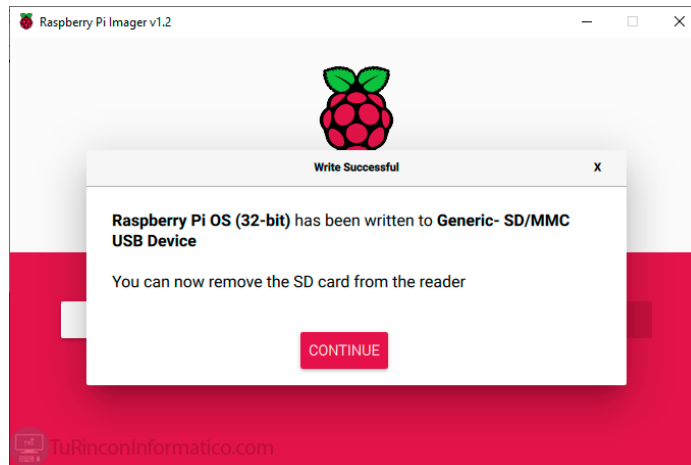


Figura A 6 Mensaje de finalización SD card

Fuente: Desarrollado por el investigador

Como se puede observar, Raspberry Pi Imager es un programa simple y sin ninguna dificultad para que cualquier persona pueda utilizarlo.

Configuración y primeros pasos en Raspberry Pi OS

Cuando arranquemos Raspberry Pi OS mostrará un mensaje de bienvenida Figura A 7, en él se podrá configurar las cosas básicas del sistema.

- Región y Zona Horaria
- Lenguaje
- Ajuste del borde negro de pantalla
- Contraseña
- Conexión a Internet
- Actualización del sistema

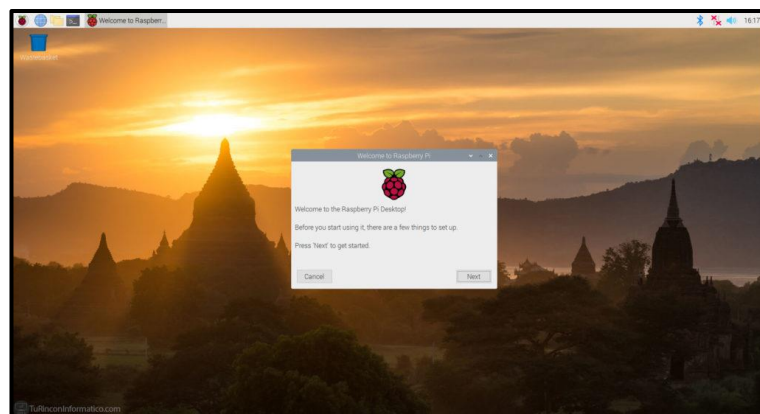


Figura A 7 Mensaje de bienvenida Raspberry Pi Desktop

Fuente: Desarrollado por el investigador

Lo primero que pedirá es que seleccione su región, lenguaje y zona horaria. Cuando termine de seleccionarlo se pulsa Next Figura A 8.



Figura A 8 Selección de parámetros, configuración de inicio Raspberry Pi Desktop

Fuente: Desarrollado por el investigador

El siguiente paso es ponerle una password (contraseña) a Raspberry Pi OS, se introduce la password que desee utilizar, recuerde poner una contraseña segura para mayor seguridad de su Raspberry pi. Una vez introducida la password se pulsa Next Figura A 9.



Figura A 9 Password Raspberry Pi OS

Fuente: Desarrollado por el investigador

Si la pantalla se ve con un borde negro se activa la casilla “*This screen shows a black border around the desktop*”, si no es así, no la active. Seguidamente pulsamos Next Figura A 10.

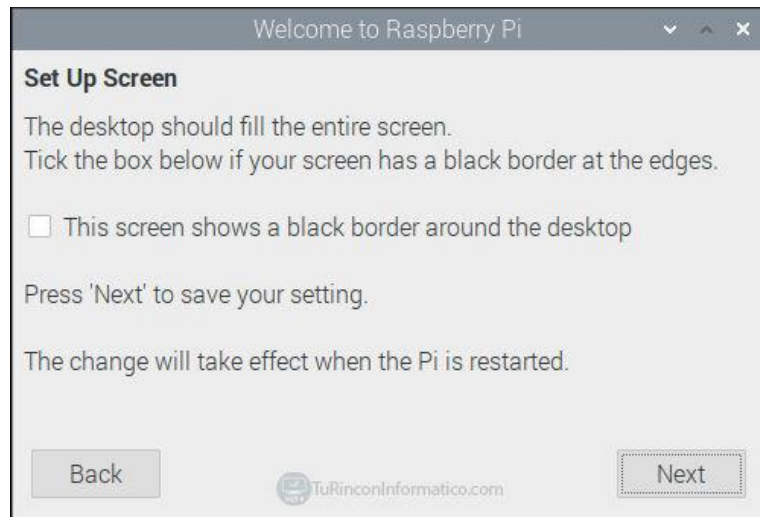


Figura A 10 Casilla de activación “*This screen shows a black border around the desktop*”

Fuente: Desarrollado por el investigador

En esta página podrá seleccionar su red wifi, recuerde que la Raspberry Pi 3 model b+ y la Raspberry Pi 4 tienen wifi 5GHz. Una vez seleccionada la red wifi pulsamos Next, en el caso de no querer conectar la Raspberry Pi por red wifi, conecte un cable de red y pulse Skip Figura A 11.

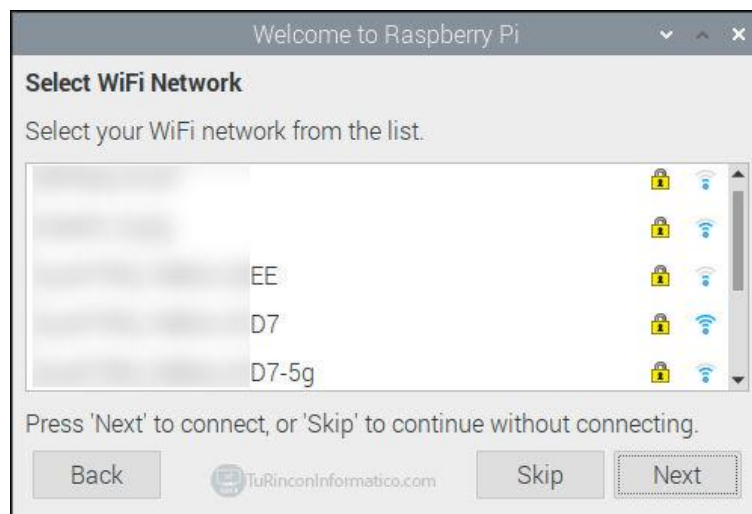


Figura A 11 Selección de red wifi

Fuente: Desarrollado por el investigador

Cuando seleccione la red wifi se introduce la contraseña como indica la Figura A 12. Cuando pulse Next, la Raspberry se conectará a la red y el icono de la red wifi de la parte superior derecha aparecerá en azul.

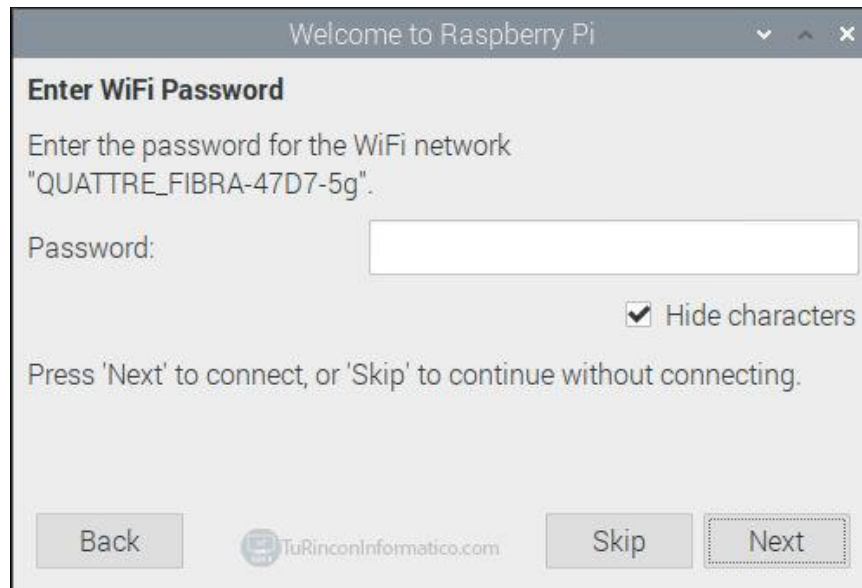


Figura A 12 Ingreso de contraseña red wifi

Fuente: Desarrollado por el investigador

La página nos indica que vamos a realizar actualizaciones del sistema, es recomendable actualizar el sistema para un mejor uso y seguridad. Este proceso puede tardar un tiempo en realizarse Figura A 13.

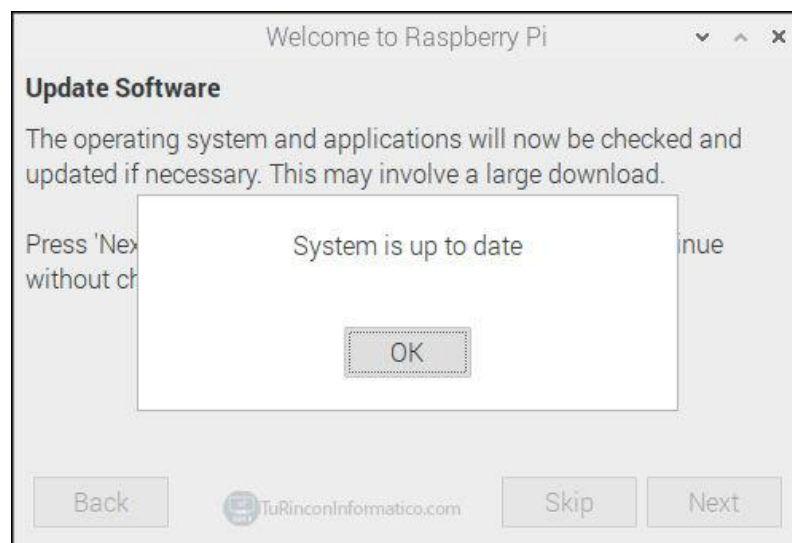


Figura A 13 Actualizaciones del sistema Raspberry Pi

Fuente: Desarrollado por el investigador

Una vez que el proceso ha terminado hay que reiniciar la Raspberry pulsando en Restart Figura A 14.

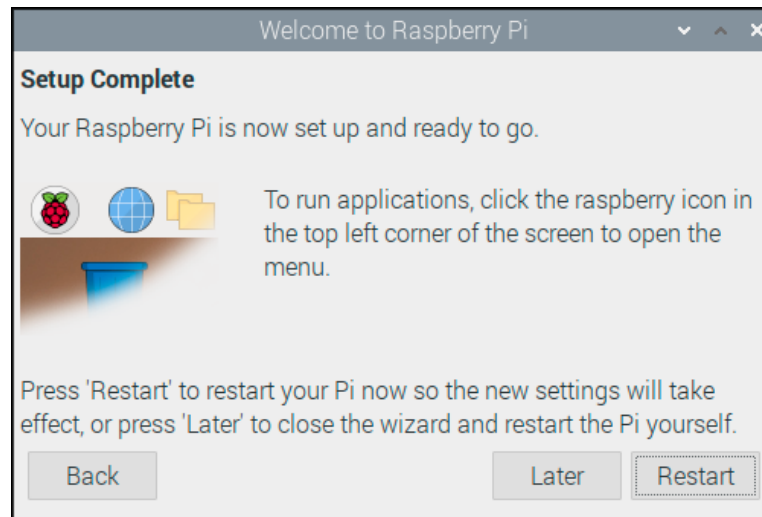


Figura A 14 Configuración completa-Reinicio del sistema

Fuente: Desarrollado por el investigador

Cuando la Raspberry Pi se reinicie podremos observar que el menú de bienvenida ya no aparece, la red wifi está conectada, el sistema actualizado, la pantalla se ha ajustado sin los bordes negros, la fecha y hora es la correcta y el idioma es el que se ha elegido.

Anexo 2.

Instalación de GNU Radio

Para este proyecto se utilizará GNU Radio para Raspberry Pi OS, ya que es el sistema operativo que será usado en el prototipo, el mismo que será manipulado por estudiantes y personal docente. Se lo puede descargar desde un enlace proporcionado en la página oficial. Existe una limitación para la versión de Raspberry Pi OS, solo existe el paquete de instalación para la arquitectura de 32 bits. La arquitectura de 64 bits no está admitida en GNU Radio. A continuación, se descargará un instalador con lo cual se procede a ejecutarlo. Se debe tomar en cuenta que en la página de descarga existen otras compilaciones del programa, e incluso de versiones más actuales pero que se encuentran en fases de prueba. Es recomendable descargar la versión estándar y definitiva para que no se tenga algún inconveniente a la hora de instalar o usar el programa. Cuando se haya descargado el paquete, se lo instala de la misma forma que un programa normal, no requiere de complementos adicionales ni de licencia alguna para su activación. Se muestra el cuadro de diálogo para la instalación de GNU Radio según la Figura A 15.

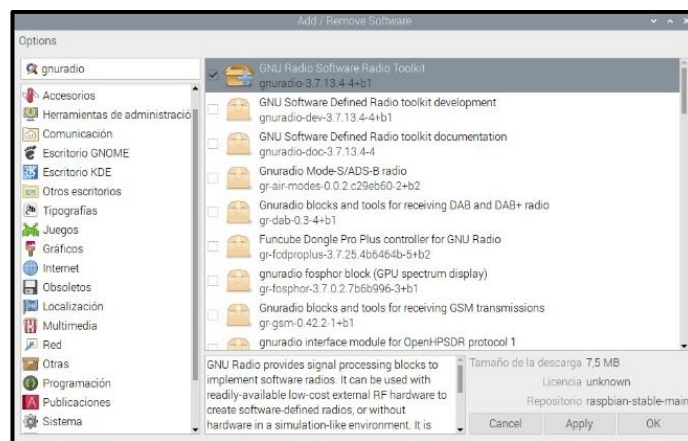


Figura A 15 Ventana de instalación de GNU Radio en Raspberry SO

Fuente: Desarrollado por el investigador

Una vez instalado se procede a ejecutarlo, la interfaz del programa como se muestra en la Figura A 16. La interfaz presenta: en la parte derecha las bibliotecas con las que se podrá trabajar, en la parte superior una barra de herramientas, en el centro el espacio de trabajo conocido como workspace, en la parte inferior la consola de visualización, la cual indica las operaciones que realiza el programa y el resultado si existe algún error en el desarrollo de aplicaciones. Las descripciones de las bibliotecas y barras de

herramientas del programa se lo realizarán en el apartado 2.1. debido a que forma parte del prototipo. En la Figura A 16 se muestra la interfaz de trabajo de GNU Radio para Raspberry SO.

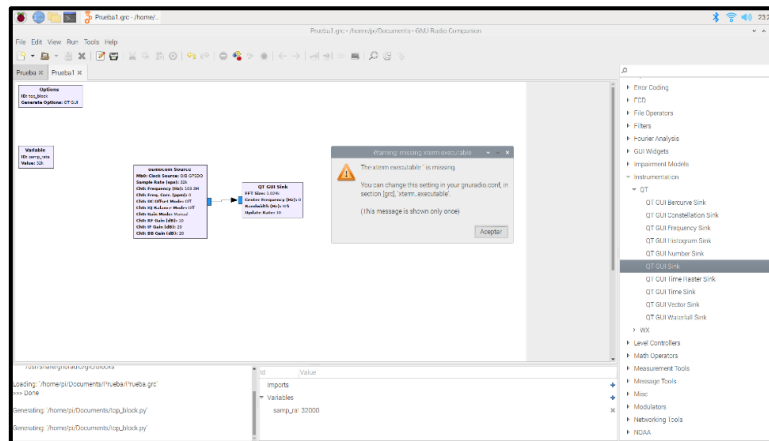


Figura A 16 Pantalla de inicio de GNU Radio

Fuente: Desarrollado por el investigador

La versión más actual que se encuentra en el repositorio es 3.7.13 a la fecha de la elaboración de este proyecto de investigación. Posee una interfaz de trabajo similar a Simulink de Matlab, interconecta los bloques para construir aplicaciones.

Anexo 3.

Manual de usuario

A continuación, se presenta una pequeña guía de uso para el proyecto en donde se hace énfasis en la pantalla y sus botones que utiliza.

Encendido y conexión del dispositivo

Encendido

Para encender el dispositivo se debe conectar la fuente de alimentación del prototipo Serie Trigonométrica de Fourier y analizador de espectro, la cual mostrará en pantalla la carga del sistema operativo como se muestra en la Figura A 17, junto con la activación de los diferentes servicios y dispositivos conectados a los puertos USB. La fuente de alimentación brinda un voltaje de 5V DC a 2.5 A.

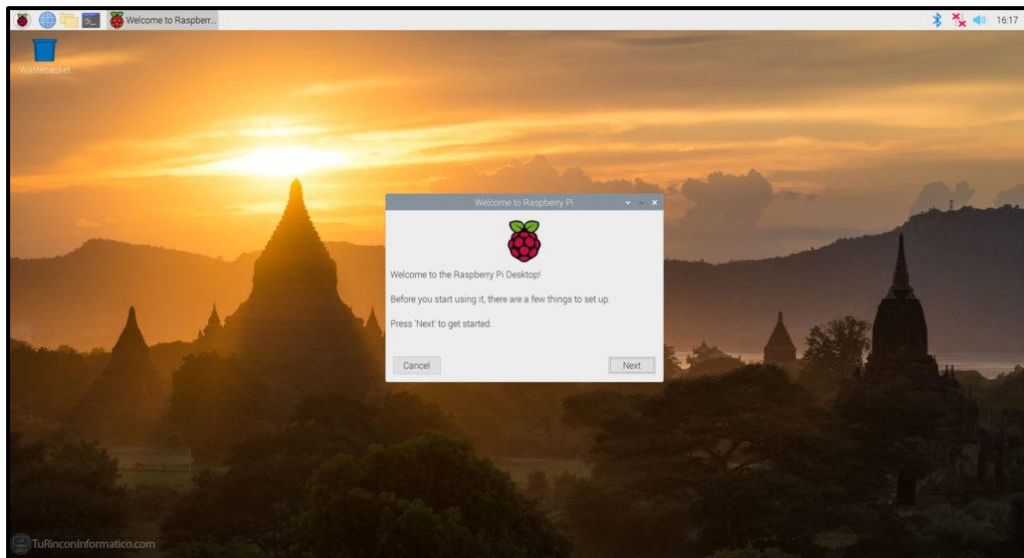


Figura A 17 Encendido del dispositivo- carga del sistema operativo.

Fuente: Desarrollado por el investigador

Ejecución de las aplicaciones

El prototipo dispone de dos tipos de aplicaciones:

1. Serie Trigonométrica de Fourier
2. Analizador de espectro radioeléctrico

Para el aplicativo de Serie Trigonométrica de Fourier consta con una ventana gráfica como se muestra en la Figura A 18. Donde existe el ingreso de 4 parámetros de suma importancia y de carácter obligatorio.

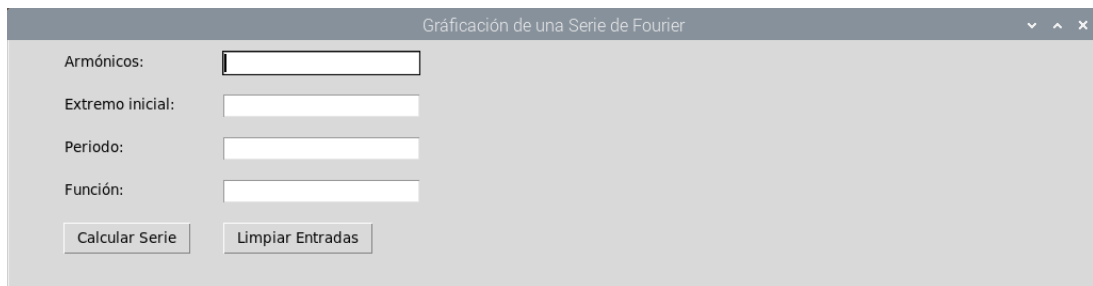


Figura A 18 Pantalla de inicio-Serie Trigonométrica de Fourier

Fuente: Desarrollado por el investigador

En la que se ingresa el número de armónicos, extremo inicial, periodo y función como se muestra a continuación en la Figura A 19

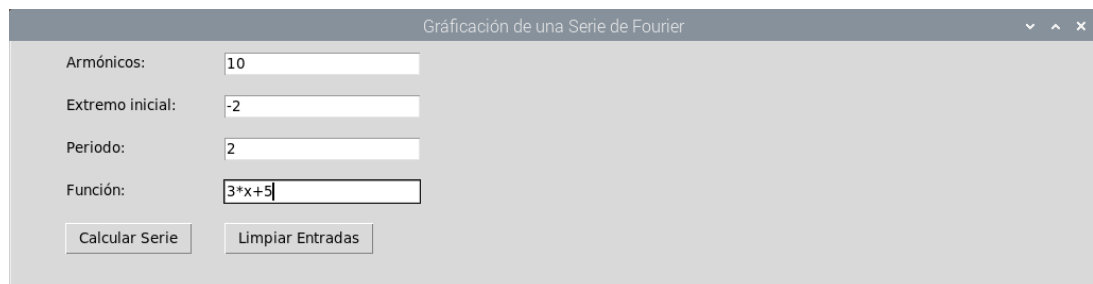


Figura A 19 Ingreso de parámetros- Serie Trigonométrica de Fourier

Fuente: Desarrollado por el investigador

Entregándonos una gráfica como la Figura A 20 para su respectivo análisis según sea la función ingresada

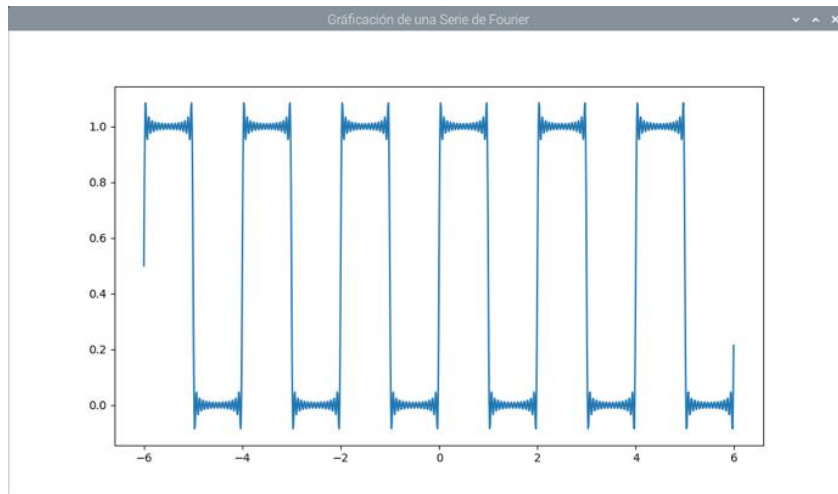


Figura A 20 Representación gráfica aplicativo- Serie Trigonométrica de Fourier

Fuente: Desarrollado por el investigador

Como segundo aplicativo una vez encendido el dispositivo, en la pantalla inicial aparecerá la siguiente aplicación como se muestra en la Figura A 21 y Figura A 22.

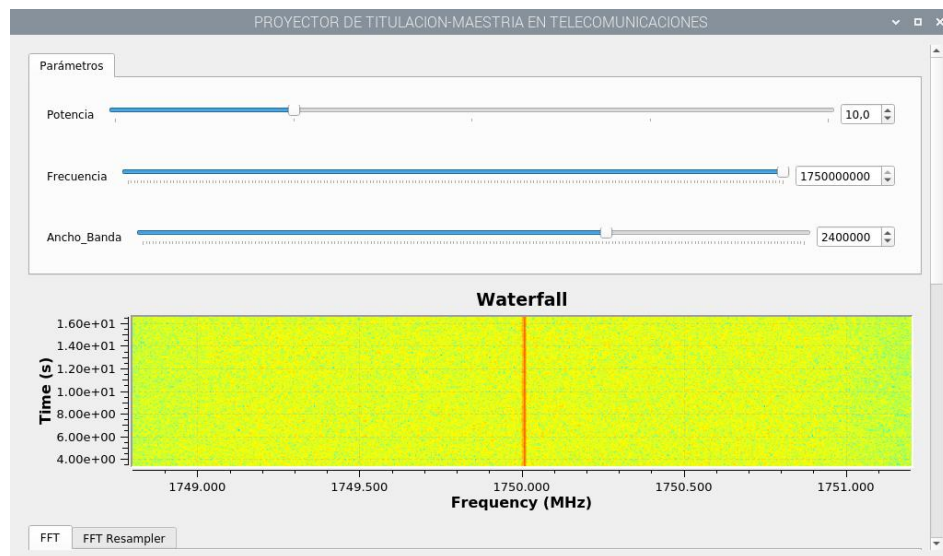


Figura A 21 Pantalla inicial Parte 1.

Fuente: Desarrollado por el investigador

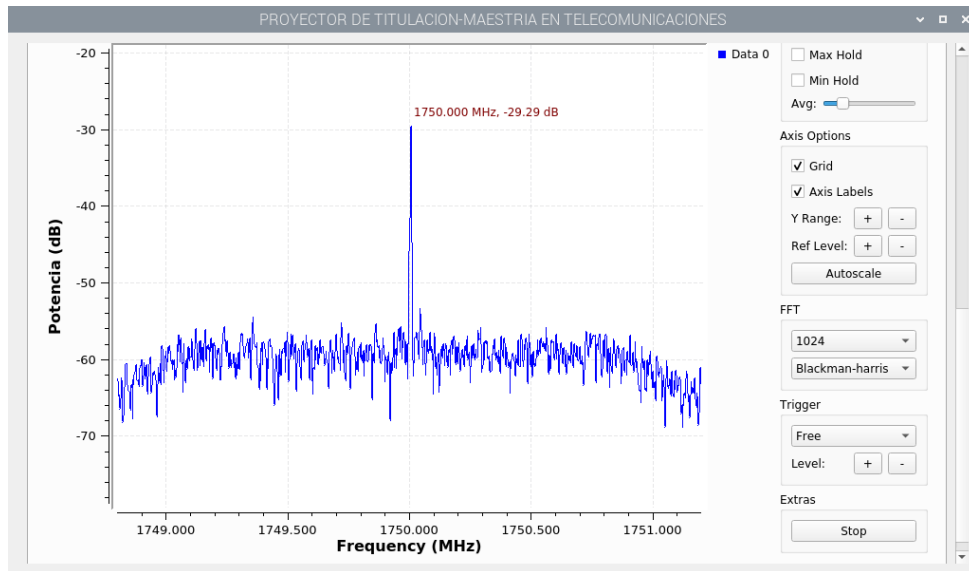


Figura A 22 Pantalla inicial Parte 2.

Fuente: Desarrollado por el investigador

En esta aplicación puede configurar varios parámetros como la potencia de recepción la frecuencia y el ancho de banda. Puede seleccionar la potencia de recepción con el control deslizante a medida que mejora la calidad de la señal, existe también un cuadro de diálogo para ingresar la frecuencia central y ancho de banda como indica la Figura A 23.

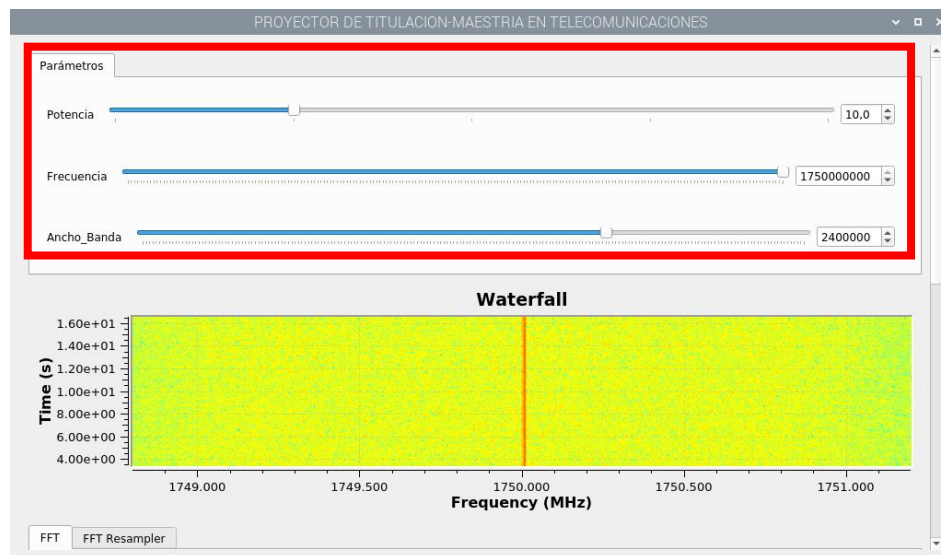


Figura A 23 Panel de configuración

Fuente: Desarrollado por el investigador

Esta aplicación cuenta con diferentes opciones necesarias para el análisis del espectro recibido como Max Hold, Min Hold y Average Figura A 24.

Max Hold: Permite guardar los máximos niveles de potencia recibidos, en una gráfica de color verde, necesario para calcular el ancho de banda de forma manual.

Min Hold: Permite guardar los mínimos niveles de potencia recibidos.

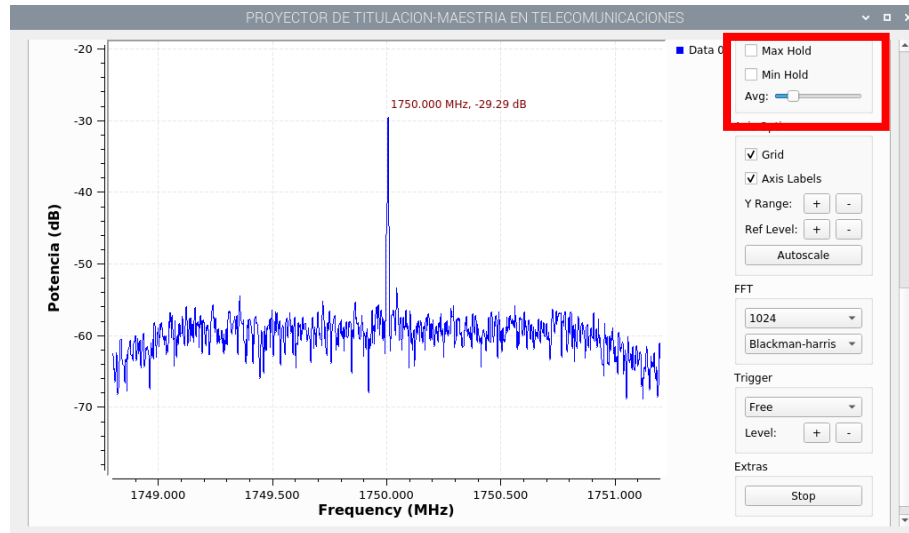


Figura A 24 Panel Max Hold, Min Hold y Average

Fuente: Desarrollado por el investigador

Average: Permite realizar un promedio de la señal recibida Figura A 25.

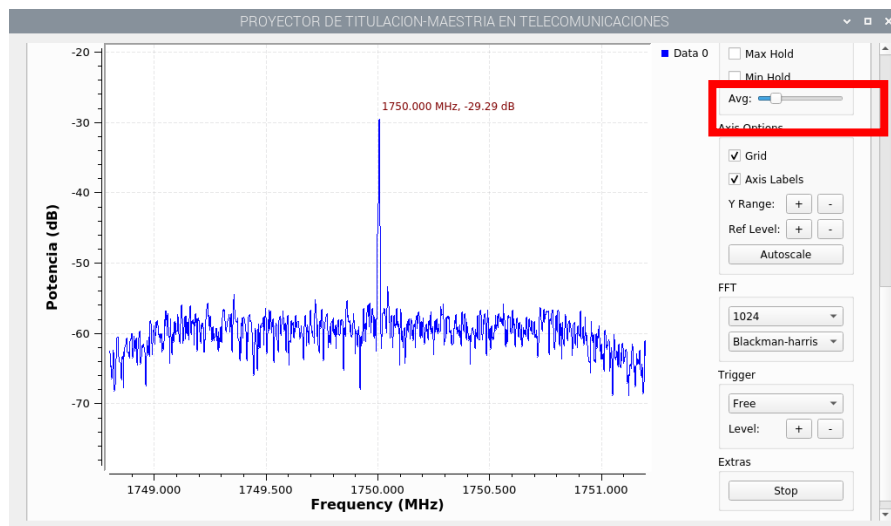


Figura A 25 Configuración del Average

Fuente: Desarrollado por el investigador

También se puede manipular la escala de divisiones de la potencia en el eje de las ordenadas; con la opción “+” se puede aumentar el número de divisiones, mientras con la opción “-” se puede reducir el número de divisiones Figura A 26.

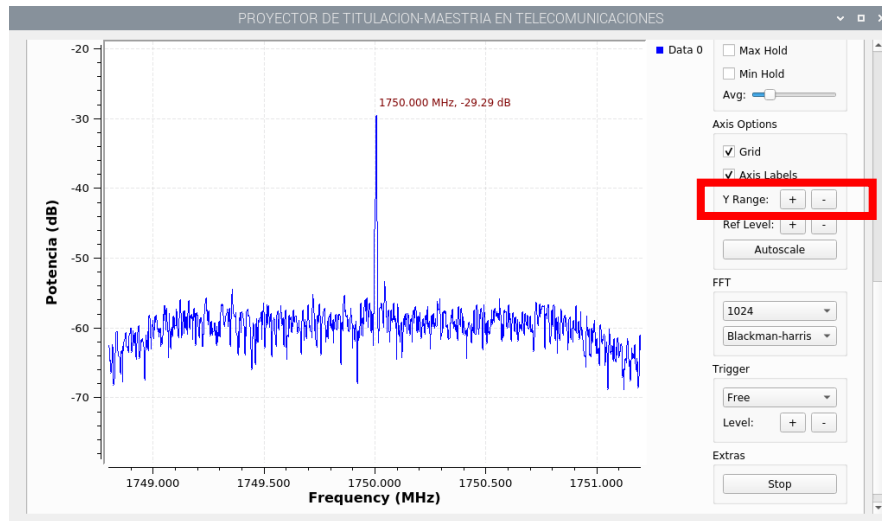


Figura A 26 Configuración del número de divisiones de potencia.

Fuente: Desarrollado por el investigador

Para elegir una mejor posición de la gráfica se puede utilizar la opción Ref Level, que mueve la señal en el eje de las ordenadas para poder centrar en la ventana, sin alterar el número de divisiones Figura A 27.

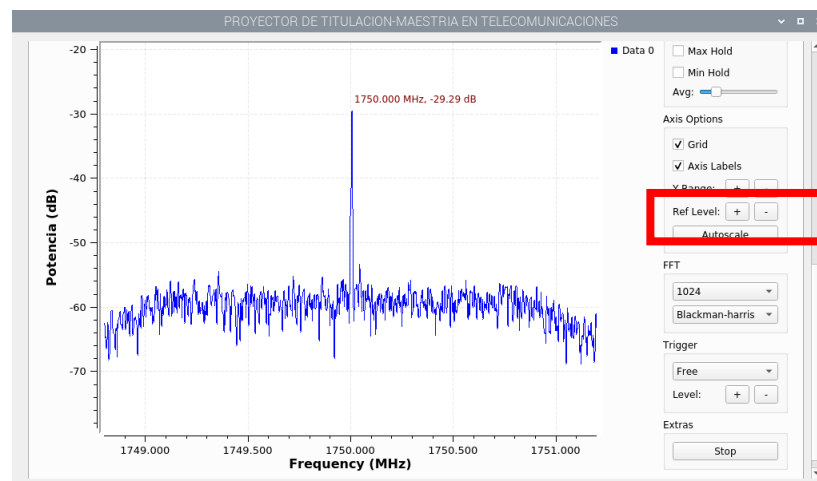


Figura A 27 Configuración del nivel de referencia.

Fuente: Desarrollado por el investigador

La opción Autoscale, ajusta la señal en toda la ventana del analizador de espectro para una visión más amplia de la señal Figura A 28.

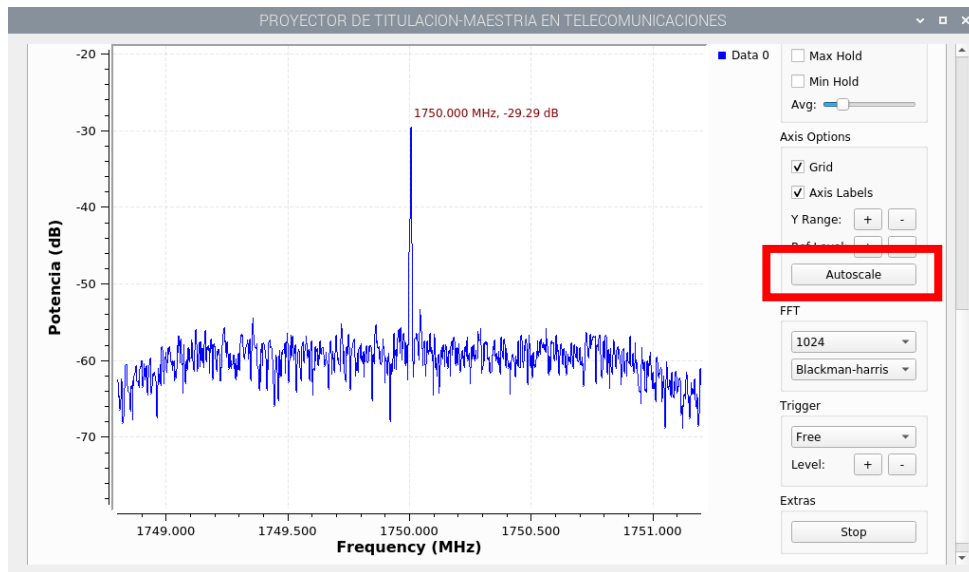


Figura A 28 Activación de la gráfica en toda la pantalla.

Fuente: Desarrollado por el investigador

Anexo 4.-

Esquema controlador Panel táctil capacitivo de 7 pulgadas

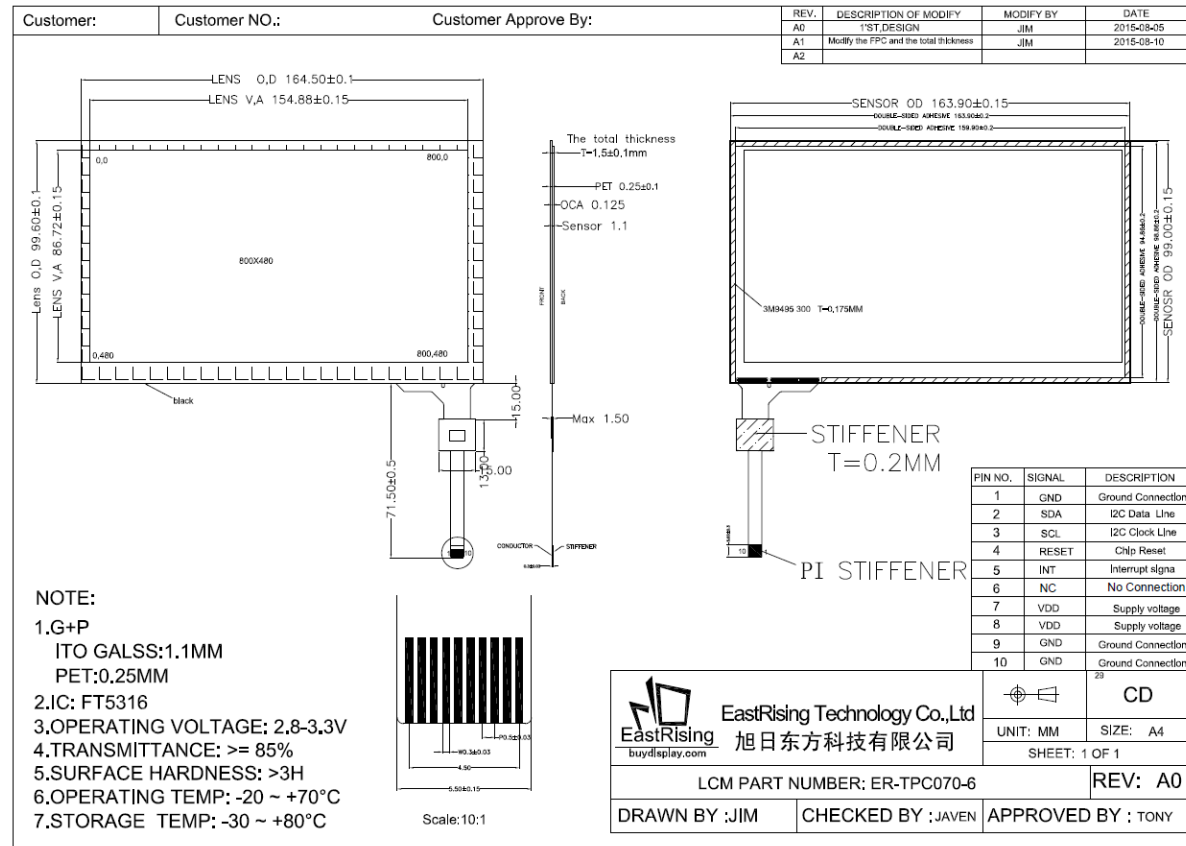


Figura A 29 Diagrama esquemático pantalla táctil.

Fuente: Desarrollado por el investigador.

Anexo 5

Script cálculo de Series Trigonómicas de Fourier

```
#Autor: Alex Francisco Rodríguez Ortiz
import matplotlib.pyplot as plt
import numpy as np
import sympy as sp
import math as mt

import tkinter

from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg,
NavigationToolbar2Tk
from matplotlib.figure import Figure

from tkinter import scrolledtext #Es para la ventana de texto de los resultados
from tkinter import * #Se importa la libreria Tkinter para Interfaces gráficas
from tkinter import messagebox

ventana = Tk() #Se crea una nueva ventana
ventana.title("Series de Fourier Alex Rodriguez") #Se le pone un titulo a la ventana
ventana.geometry('2000x1000') #Se define un tamaño para la ventana

ventana2 = tkinter.Tk()
ventana2.wm_title("Gráfica Series de Fourier Alex Rodriguez")
ventana2.geometry('2000x1000')

x = sp.symbols("x")

#Un label para lo que necesitaremos ingresar
txt = Label(ventana , text="Series de Fourier ",
font=("Arial Bold" , 15))

lbl = Label(ventana , text="Armónicos: ",
```

```
font=("Arial Bold" , 15))
```

```
lbl2 = Label(ventana , text="Extremo inicial: ",  
font=("Arial Bold" , 15))
```

```
lbl3 = Label(ventana , text="Periodo: ",  
font=("Arial Bold" , 15))
```

```
lbl4 = Label(ventana , text="Función: ",  
font=("Arial Bold" , 15))
```

```
#Para establecer la posicion del label en la ventana
```

```
txt.grid(column=0 , row=0)
```

```
lbl.grid(column=0 , row=5)
```

```
lbl2.grid(column=0 , row=10)
```

```
lbl3.grid(column=0 , row=15)
```

```
lbl4.grid(column=0 , row=20)
```

```
#Para ingresar los datos
```

```
armonicos = Entry(ventana, font=("calibri 20") , width=10)
```

```
armonicos.grid(row=5,column=5,columnspan=2,padx=4,pady=5)
```

```
lim_inf = Entry(ventana, font=("calibri 20") , width=10)
```

```
lim_inf.grid(row=10,column=5,columnspan=2,padx=4,pady=10)
```

```
lim_sup = Entry(ventana, font=("calibri 20") , width=10)
```

```
lim_sup.grid(row=15,column=5,columnspan=2,padx=4,pady=15)
```

```
funcion = Entry(ventana, font=("calibri 20") , width=10)
```

```
funcion.grid(row=20,column=5,columnspan=2,padx=4,pady=10)
```

```
#Botones
```

```
calcular = Button(ventana, text = "Calcular" , width = 7, height = 2 , command =
lambda:calcular_serie())
```

```
calcular.grid(row =5 , column = 10 , padx = 15, pady = 5)
```

```
borrar = Button(ventana, text = "Limpiar" , width = 7, height = 2 , command =
lambda:click_limpiar())
```

```
borrar.grid(row =5 , column = 11 , padx = 15, pady = 5)
```

```
ayuda = Button(ventana, text = "Ayuda" , width = 7, height = 2 , command =
lambda:click_ayuda())
```

```
ayuda.grid(row =5 , column = 12 , padx = 15, pady = 5)
```

```
#Funciones
```

```
def click_ayuda():
```

```
    messagebox.showinfo("Instrucciones" , "--Ejemplos para ingresar una funcion
: "+\n"+"x^2 = x**2"+\n"+"sen(2x) = sin(2*x)"+"\n"+"cos(2x^2) =
cos(2*x**2)"+"\n"--En limite inferior y superior ingresar los limites de
integracion"+\n"--En terminos ingresar el numero de terminos con la que quieres
que se grafique tu serie")
```

```
def Fourier(B_0, A_m, B_m, m, k, y):
```

```
    VA = 0
```

```
    VB = 0
```

```
    for i in range(0, m):
```

```
        VA += A_m[i]*np.sin((i+1)*2*k*y)
```

```
        VB += B_m[i]*np.cos((i+1)*2*k*y)
```

```
    return B_0 + VA + VB
```

```
def calcular_serie():
```

```
    m = int(armonic.get())
```

```
    I = float(lim_inf.get())
```

```
    T = float(lim_sup.get())
```

```
    F = sp.sympify(funcion.get())
```

```
    G = 0
```



```

k = mt.pi/T
B_0 = sp.integrate(F/T, (x, I, I+T/2)) + sp.integrate(G/T, (x, I+T/2, I+T))
A_m = []
B_m = []

for i in range (1, m+1):

    A_m.append(sp.integrate(F*(2/T)*sp.sin(i*2*k*x), (x, I, I+T/2)) +
sp.integrate(G*(2/T)*sp.sin(i*2*k*x), (x, I+T/2, I+T)))

    B_m.append(sp.integrate(F*(2/T)*sp.cos(i*2*k*x), (x, I, I+T/2)) +
sp.integrate(G*(2/T)*sp.cos(i*2*k*x), (x, I+T/2, I+T)))

    y = np.arange(-T*3, T*3, 0.01)

    fig = Figure(figsize=(10, 9), dpi=100)

    fig.add_subplot(111).plot(y, Fourier(B_0, A_m, B_m, m, k, y) )#AÑADIR
"subbplot"

    canvas = FigureCanvasTkAgg(fig, master=ventana2) # CREAR AREA DE
DIBUJO DE TKINTER.

    canvas.draw()

    canvas.get_tk_widget().pack(side=tkinter.TOP, fill=tkinter.BOTH, expand=1)

def click_limpiar():

    armonicos.delete(0,20)

    lim_inf.delete(0,20)

    lim_sup.delete(0,20)

    funcion.delete(0,20)

    return None

tkinter.mainloop() #Esta función reconoce todo lo que estamos haciendo con la
ventana y es necesaria para verla

```