



Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos

Naturales no Renovables

Carrera de Telecomunicaciones

Desarrollo de Prácticas de Comunicaciones Digitales utilizando Radio Definida por Software (SDR) para el Laboratorio de Telecomunicaciones de la FEIRNNR – UNL

**Trabajo de Integración Curricular
previo a la obtención del título de
Ingeniera en Telecomunicaciones.**

AUTOR:

Jéssica Tatiana Córdova Sari

DIRECTOR:

Ing. Juan Gabriel Ochoa Aldeán, Mg Sc.

Loja – Ecuador

2023

Certificación

Loja, 28 de febrero de 2023

Ing. Juan Gabriel Ochoa Aldeán, Mg Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

CERTIFICO

Que he revisado y orientado todo el proceso de elaboración del Trabajo de Integración Curricular denominado: **Desarrollo de Prácticas de Comunicaciones Digitales utilizando Radio Definida por Software (SDR)** para el **Laboratorio de Telecomunicaciones de la FEIRNNR – UNL**, previo a la obtención del título de **Ingeniera en Telecomunicaciones**, de la autoría del estudiante **Jéssica Tatiana Córdova Sari**, con cédula de identidad Nro. **1104832777**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

1103676688 Firmado digitalmente
por 1103676688
JUAN GABRIEL JUAN GABRIEL
OCHOA OCHOA ALDEAN
ALDEAN Fecha: 2023.10.05
08:22:00 -05'00'

Ing. Juan Gabriel Ochoa Aldeán, Mg Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

Yo, **Jéssica Tatiana Córdova Sari** declaro ser autora del presente Trabajo de Integración Curricular y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos o acciones legales por el contenido del mismo: Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Integración Curricular en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma:



Cédula de Identidad: 1104832777

Fecha: 11 de septiembre de 2023

Correo electrónico: jessica.t.cordova@unl.edu.ec

Teléfono: 0962609762

Carta de autorización por parte de la autora para la consulta, reproducción parcial o total, y/o publicación electrónica del texto completo del Trabajo de Integración Curricular.

Yo, Jéssica Tatiana Córdova Sari, declaro ser la autora del Trabajo de Integración Curricular denominado: Desarrollo de Prácticas de Comunicaciones Digitales utilizando Radio Definida por Software (SDR) para el Laboratorio de Telecomunicaciones de la FEIRNNR - UNL, como requisito para optar por el título de Ingeniera en Telecomunicaciones, autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos muestre la reproducción intelectual, a través de la visibilidad en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Integración Curricular que realice un tercero.

Para constancia de esta autorización, firmo en la ciudad de Loja, a los once días del mes de septiembre del año dos mil veintitrés.

Firma: 

Autora: Jéssica Tatiana Córdova Sari

Cédula de identidad: 1104832777

Dirección: Loja, C/José A. Palacios e Ingapirca

Correo electrónico: jessica.t.cordova@unl.edu.ec

Teléfono: 2587703

DATOS COMPLEMENTARIOS

Director del Trabajo de Titulación: Ing. Juan Gabriel Ochoa Aldeán, Mg Sc

Dedicatoria

Dedico este Trabajo de Integración Curricular con todo cariño a una mujer especial, la Sra. Betty Sari, quien me ha entregado el amor más sincero, el de una madre. Ella ha cuidado siempre de mí sin importar la distancia y las circunstancias, me ha enseñado a ser una buena mujer gracias a su ejemplo y a sus valiosos consejos, y porque en todo momento siempre he recibido su apoyo y aliento a seguir superándome.

A mi querida hermana Cinthya, por estar conmigo siempre en todo momento, sea bueno o malo, gracias por cuidar bien de mí. De ti recuerdo lo fuerte que podemos llegar a ser. Ahora que somos adultas sigues siendo mi mayor apoyo y así mismo cuentas con el mío.

A Stalin Jiménez, quién significa mucho en mi vida, pues su apoyo, su compañía, el creer en mis capacidades, han logrado que muy pronto pueda verme convertida en toda una profesional.

Finalmente, va dedicado al ser más importante que Dios y la vida me han dado, espero que este escalón más que doy pueda servirte de ejemplo. Aunque aún no lo sepas, eres mi gran inspiración y trataré de guiar tus pasos con amor y disciplina mi pequeña Luciana.

Jéssica Tatiana Córdova Sari

Agradecimiento

Mis más sinceros agradecimientos a la Universidad Nacional de Loja, a la planta docente que conforma la Carrera de Telecomunicaciones por abrirme las puertas e instruirme como una profesional a lo largo de esta etapa de formación.

Un especial agradecimiento a mi director de tesis el Ing. Juan Gabriel Ochoa y demás docentes por haberme hecho partícipe de sus conocimientos y sabias enseñanzas que sirvieron como guía para el desarrollo del presente trabajo de investigación.

Agradezco a mi familia por contribuir en mi superación personal y el apoyo moral

Mil gracias a todos por ser parte del cumplimiento de esta meta.

Jéssica Tatiana Córdova Sari

Índice de Contenido

Portada	i
Certificación	ii
Autoría	iii
Carta de autorización	iv
Dedicatoria	v
Agradecimiento	vi
Índice de Contenido	vii
Índice de Tablas	x
Índice de Figuras	xi
Índice de Anexos	xvii
1. Título	1
2. Resumen	2
2.1. Abstract	3
3. Introducción	4
4. Marco teórico	6
4.1. Comunicaciones digitales	6
4.1.1. Elementos de un Sistema de Comunicaciones Digitales	6
4.1.2. Modulaciones digitales	7
4.2. <i>Multiplexación</i>	8
4.2.1. Multiplexación por división de tiempo o TDM	9
4.2.2. Multiplexación por división de frecuencia o FDM	9

4.2.3.	Multiplexación por división de frecuencias ortogonales o OFDM	9
4.2.4.	Multiplexación por división en código o CDM	9
4.3.	Acceso Múltiple	10
4.3.1.	Acceso múltiple por división de tiempo o TDMA	10
4.3.2.	Acceso múltiple por división de frecuencia o FDMA	10
4.3.3.	Acceso múltiple por división de frecuencias ortogonales o OFDMA	10
4.3.4.	Acceso múltiple por división de código o CDMA	10
4.4.	Radio Definido por Software	11
4.4.1.	Aplicaciones de SDR	11
4.4.2.	Componentes funcionales de SDR	11
4.4.3.	Niveles de SDR	13
4.5.	Entornos de software para SDR	14
4.5.1.	GNU – RADIO	14
4.5.2.	Matlab – Simulink	15
4.5.3.	Comparación entre GNU Radio y Matlab – Simulink	16
5.	Metodología	18
5.1.	Hardware y software	18
5.2.	Guías de laboratorio	20
5.2.1.	PRÁCTICA # 1	20
5.2.2.	PRÁCTICA # 2	28
5.2.3.	PRÁCTICA # 3	38
5.2.4.	PRÁCTICA # 4	53

5.2.5. PRÁCTICA # 5	68
6. Resultados	89
6.1. PRÁCTICA #1	89
6.2. PRÁCTICA #2	94
6.3. PRÁCTICA #3	101
6.4. PRÁCTICA #4	102
6.5. PRÁCTICA #5	104
7. Discusión	107
8. Conclusiones	109
9. Recomendaciones	110
10. Referencias Bibliográficas	111
11. Anexos	113

Índice de Tablas:

Tabla 1. Comparación entre los entornos para simulaciones SDRs	17
Tabla 2. Especificaciones técnicas del dispositivo RTL-SDR.....	18
Tabla 3. Especificaciones técnicas del dispositivo LimeMini - SDR	18
Tabla 4. Especificaciones técnicas del dispositivo Adalm Pluto - SDR.....	19

Índice de Figuras:

Figura 1. Diagrama sobre los elementos fundamentales de un sistema de comunicaciones digitales	6
Figura 2. Diagrama de bloque funcionales de SDR.....	12
Figura 3. Metodología para la creación de bloques en GNU Radio	15
Figura 4. Metodología para la creación de bloques usando MATLAB – Simulink	16
Figura 5. Entorno de la página web de VMware Workstation 16 Player para la descarga de la máquina virtual en el sistema operativo de Windows y el sistema operativo de Linux	21
Figura 6. Interfaz al abrir el ícono de VMware Workstation 16.....	22
Figura 7. Características de compatibilidad de hardware para la creación de la máquina virtual.....	22
Figura 8. Instalación del sistema operativo Linux en una máquina virtual: mediante un disco, a través de un archivo de imagen de disco o realizar la instalación más tarde.....	23
Figura 9. Personalización de Linux mediante un usuario y una contraseña necesarios para cualquier configuración como administrador.	24
Figura 10. Elección de un nombre para la máquina virtual y una carpeta de ubicación para la misma.....	24
Figura 11. Conexión del dispositivo NESDR a la antena y al computador para su correcto funcionamiento	25
Figura 12. Interfaz gráfica del programa Matlab de manera general.....	29
Figura 13. Ventana de Add On Explorer que permite encontrar complementos.....	30
Figura 14. Aceptación de términos para instalar el paquete de RTL-SDR.....	30
Figura 15. Paquetes que se han instalado para el dispositivo RTL-SDR Radio.....	31
Figura 16. Instalación exitosa del paquete de Comunicación para el hardware RTL-SDR Radio	31
Figura 17. Conexión del dispositivo RTL-SDR para la instalación de los drivers.....	32
Figura 18. Aplicación Zadig para Windows que instala controladores USB genéricos.....	32
Figura 19. Información de configuración inicial del RTL-SDR.....	33
Figura 20. Interfaz gráfica de Simulink para crear un nuevo modelo en blanco.....	33
Figura 21. Barra de herramientas de Simulink donde se puede encontrar los componentes para la creación del diagrama	34

Figura 22. Explorador de biblioteca de la librería de Simulink permite buscar mediante palabras cortas el elemento deseado.....	34
Figura 23. Bloque RTL-SDR para la configuración de los parámetros para su funcionamiento.	35
Figura 24. Diagrama de bloques que corresponde a funcionar como un sintonizador de frecuencias	36
Figura 25. Archivo DATV - Express_DVB_ Transmitter que se debe descargar.....	40
Figura 26. Forma de extraer y ejecutar el software Express DVB	40
Figura 27. Parámetros a configurar en la opción del Modulador del programa DATV	41
Figura 28. Parámetros para configurar la opción de DVB-S2 del programa DATV.....	41
Figura 29. Parámetros para configurar el formato de video que será transmitido.....	42
Figura 30. Selección del dispositivo que se utilizará para capturar la señal de video y audio respectivamente.....	43
Figura 31. Configuración de los códec de audio y video para la respectiva codificación y decodificación.....	43
Figura 32. Selección del controlador FTDI tomando en cuenta la arquitectura del procesador del computador.	44
Figura 33. Forma de actualizar el controlador de LimeSDR en Windows.....	45
Figura 34. Actualización con éxito del controlador de LimeSDR a la versión descargada	45
Figura 35. Selección del archivo a descargar del programa SDRangel.....	46
Figura 36. Forma de instalación del software SDRangel a través del gestor de archivadores	46
Figura 37. Ventana que indica la opción de instalar SDRangel en Ubuntu.....	46
Figura 38. Estableciendo conexión del dispositivo RTL - SDR a la máquina virtual.....	47
Figura 39. Selección del dispositivo SDR dentro del programa SDRangel	47
Figura 40. Estableciendo el modo de operación que se desea que trabaje el dispositivo RTL-SDR	48
Figura 41. Configuración lista para transmitir señales de video mediante el programa DATV-Express presionando el botón PTT	49
Figura 42. Forma de seleccionar el paquete apropiado para el dispositivo Adalm Pluto.....	54
Figura 43. Ventana de Add On Explorer para buscar el paquete específico para el hardware SDR	55

Figura 44. Actualización del firmware a la versión recomendada 0.34 del dispositivo Adalm PlutoSDR	55
Figura 45. Indicación del procedimiento de la actualización del Firmware	56
Figura 46. Ventana de información sobre el dispositivo Adalm Pluto	56
Figura 47. Parámetros de configuración para la transmisión de texto.....	57
Figura 48. Creación del objeto para interactuar con el hardware Adalm Pluto Radio	57
Figura 49. Variables para controlar el proceso de transmisión.....	57
Figura 50. Ubicación del archivo de texto a transmitir.....	58
Figura 51. Selección del archivo que será procesado	58
Figura 52. Proceso de serialización de los datos representando cada caracter en binario	59
Figura 53. Proceso del codificador de la técnica convolucional.....	59
Figura 54. Configuración de las variables utilizadas en el codificador convolucional	60
Figura 55. Creación del diagrama de Trellis usando la opción polytrellis de Matlab	60
Figura 56. Creación de los diferentes esquemas de modulación digital	61
Figura 57. Creación de cada símbolo OFDM con portadoras de datos	62
Figura 58. Inserción del prefijo cíclico para prevenir la interferencia intersimbólica.	63
Figura 59. Preámbulo 802.11 con sus respectivas secciones.....	64
Figura 60. Normalización de los datos para la posterior transmisión de los datos al canal inalámbrico	65
Figura 61. Forma de seleccionar el paquete para el dispositivo Adalm Pluto.....	70
Figura 62. Ventana Add On Explorer para buscar el paquete específico para el hardware SDR	70
Figura 63. Actualización del firmware a la versión recomendada 0.34 del dispositivo Adalm PlutoSDR	71
Figura 64. Proceso para la actualización del Firmware del dispositivo PlutoSDR	71
Figura 65. Test de conexión entre el dispositivo PlutoSDR y el ordenador	72
Figura 66. Parámetros de configuración para la transmisión de texto y creación del objeto radio que servirá para interactuar con la radio Adalm Pluto SDR.....	72
Figura 67. Creación de variables para controlar la recepción.....	73
Figura 68. Selección del archivo transmitido para la medición del BER	74
Figura 69. Estableciendo conexión con la radio Adalm Pluto para la recepción de datos	75
Figura 70. Recorte de datos para reducir la carga de procesamiento	76

Figura 71. Detección de la trama OFDM	77
Figura 72. Estimación y corrección del desplazamiento de frecuencia.....	78
Figura 73. Proceso de sincronización de tiempo de símbolo.....	79
Figura 74. Corrección y eliminación de las distorsiones de fase residuales.....	80
Figura 75. Proceso de eliminación del Prefijo cíclico	80
Figura 76. Proceso para pasar los símbolos OFDM al dominio de la frecuencia.....	81
Figura 77. Proceso de mantener únicamente las portadoras con los datos de manera que se pueda recuperar los datos del símbolo OFDM	82
Figura 78. Proceso de selección de los diferentes esquemas de modulación	83
Figura 79. Proceso de corrección de errores a nivel de bit	84
Figura 80. Proceso para cuantificar la cantidad de bits erróneos.....	85
Figura 81. Recuperación del texto	85
Figura 82. Detección del dispositivo NESDR en el computador, listo para utilizarse en la nueva máquina virtual Ubuntu	89
Figura 83. Líneas de código para actualizar hardware, software o firmware de la máquina virtual Ubuntu.....	90
Figura 84. Línea de código git core para gestionar las modificaciones del sistema de archivos de un proyecto.....	90
Figura 85. Línea de código cmake utilizado para controlar el proceso de compilación del software usando ficheros de configuración sencillos e independientes de la plataforma.....	90
Figura 86. Línea de código libusb proporciona aplicaciones con acceso para controlar transferencia de datos hacia y desde dispositivos USB	91
Figura 87. Línea de código build-essential para crear paquetes DEB (extensión del formato d paquetes de software de la distribución de Linux) a partir del código fuente de una aplicación. 91	
Figura 88. Línea de código pkg-config para consulta las bibliotecas instaladas con el fin de compilar el software que depende de ellas.	91
Figura 89. Línea de código cd permite cambiar la ruta actual de la terminal y ubicarnos en una carpeta o directorio específico.	92
Figura 90. La línea de código make sirve para instalar y compilar muchas utilidades en este caso de rtl-sdr	92

Figura 91. Línea de código git clone crea una copia o clon del repositorio “ https://github.com/antirez/dump1090.git ” en un nuevo directorio.	93
Figura 92. Línea de código cd para acceder a la ubicación del directorio dump1090 para instalar y compilar las dependencias del programa	93
Figura 93. Parámetros de la aeronave detectada con el programa dump1090.....	93
Figura 94. Seguimiento visual de la aeronave del vuelo RER872 que proporciona la página https://flightaware.com/ e información adicional sobre el vuelo.....	94
Figura 95. Actividad GSM en una frecuencia de 936.85 MHz	95
Figura 96. Analizador de espectro en cascada sintonizando la frecuencia GSM 936 MHz	95
Figura 97. Analizador de espectro de FFT en la frecuencia GSM 936 MHz	96
Figura 98. Actividad GSM en la frecuencia 869.36 MHz	96
Figura 99. Analizador de espectro en cascada sintonizando la frecuencia GSM 869 MHz	97
Figura 100. Analizador de espectro de FFT en la frecuencia GSM 869 MHz	97
Figura 101. Actividad Radio FM en la frecuencia de 88.9 MHz.....	98
Figura 102. Analizador de espectro en cascada sintonizando la frecuencia FM 88.9 MHz	98
Figura 103. Analizador de espectro de FFT en la frecuencia FM 88.9 MHz	99
Figura 104. Actividad Radio FM en la frecuencia de 95.7 MHz.....	99
Figura 105. Analizador de espectro en cascada sintonizando la frecuencia FM 95.7 MHz.....	100
Figura 106. Analizador de espectro de FFT en la frecuencia FM 95.7 MHz	100
Figura 107. Recepción de la señal de video mediante el dispositivo RTL-SDR utilizando el programa SDRangel y bajo la modulación QPSK.....	101
Figura 108. Calidad de recepción de la señal de video mediante el dispositivo RTL-SDR.....	102
Figura 109. Sección para seleccionar el esquema de modulación digital.....	102
Figura 110. Sección para seleccionar el archivo de texto que será transmitido por el dispositivo Adalm PlutoSDR	103
Figura 111. Ejecución de la técnica convolucional estableciendo conexión con el dispositivo Adalm Pluto SDR para la transmisión del archivo de texto.	103
Figura 112. Creación del objeto para interactuar con la radio Adalm Pluto SDR.....	104
Figura 113. Forma para seleccionar el archivo de texto para medir la BER	104
Figura 114. Estableciendo conexión con el hardware de transmisión para la recepción de datos	105

Figura 115. Ejecución del proceso de recepción aplicando la técnica convolucional	105
Figura 116. Recuperación del texto transmitido	106

Índice de Anexos:

Anexo 1 Certificado Traducción Resumen..... 113

1. Título

Desarrollo de Prácticas de Comunicaciones Digitales utilizando Radio Definida por Software (SDR) para el Laboratorio de Telecomunicaciones de la FEIRNNR-UNL

2. Resumen

En el presente proyecto de grado se desarrolló un conjunto de prácticas de Comunicaciones Digitales para el laboratorio de Telecomunicaciones (FEIRNNR-UNL), utilizando la Tecnología Radio Definida por Software (SDR). Este tipo de prácticas permiten a los estudiantes relacionar los conceptos teóricos que han sido impartidos por los docentes y además conocer sobre las capacidades de utilizar SDR. La relación entre teoría y práctica es básicamente de complementariedad. Primero se realizó un análisis de las posibles temáticas que abarcan las comunicaciones digitales y con la sugerencia del tutor del presente proyecto se planteó una guía de prácticas de laboratorio.

El siguiente paso fue indagar sobre las características técnicas, funcionamiento, y el tipo de programas que soportan los dispositivos SDR (RTL-SDR, LIME SDR Mini, Pluto-SDR) los cuales van a ser utilizados en las distintas prácticas. Para el diseño de cada una de ellas se establecieron diferentes programas y a su vez se utilizó el sistema operativo de Windows y el sistema operativo de Linux (en una máquina virtual con la distribución de Ubuntu).

Después de su diseño se desarrollan y se elaboran las guías de práctica, el formato que se empleó fue el mismo modelo estándar que cuenta el laboratorio de Telecomunicaciones, presentando sus objetivos, las actividades a desarrollar, preguntas de control y otros.

Como último paso, se analizaron los resultados para poder presentar las conclusiones y recomendaciones con respecto al trabajo previamente realizado, con la finalidad de que la realización de las guías de práctica sirva como una herramienta de apoyo para el docente o el encargado de laboratorio, así como para los estudiantes de la carrera.

Palabras Clave: Comunicaciones Digitales, RTL-SDR, LimeMini, BladeRFx40, Windows, Linux, prácticas de laboratorio.

2.1. Abstract

In this degree project, a set of Digital Communications practices were developed for the Telecommunications Laboratory (FEIRNNR-UNL), using Software Defined Radio (SDR) technology. This type of practices allow students to relate the theoretical concepts that have been taught by teachers and also to learn about the capabilities of using SDR. The relationship between theory and practice is basically complementary. First, an analysis of the possible topics that cover digital communications was made and with the suggestion of the tutor of this project, a guide of laboratory practices was proposed.

The next step was to inquire about the technical characteristics, its operation, and the type of programs that support SDR devices (RTL-SDR, LIMEsDR Mini, BladeRFx40) which will be used in the different practices. For the design of each of them different programs were established and in turn the Windows operating system and the Linux operating system (in a virtual machine with the Ubuntu distribution) were used.

After their design, the practice guides were developed and elaborated, the format used was the same standard model used in the Telecommunications laboratory, presenting its objectives, the activities to be developed, control questions and others.

As a last step, the results were analyzed in order to present the conclusions and recommendations with respect to the work previously done, with the purpose of making the practice guides serve as a support tool for the teacher or the laboratory manager, as well as for the students of the career.

Keywords: Digital Communications, RTL-SDR, LimeMini, Pluto-SDR, Windows, Linux, laboratory practices.

3. Introducción

Según Mahyar & Sokolov (2018) el crecimiento exponencial de las formas y los medios por los cuales las personas necesitan comunicarse (comunicaciones de datos, comunicaciones de voz, comunicaciones de video, transmisión de mensajes, comunicaciones de comando y control, comunicaciones de respuesta de emergencia, etc.), la modificación de dispositivos de radio de manera fácil y rentable se ha vuelto negocio crítico. La tecnología de radio definida por software (SDR) brinda la flexibilidad, la rentabilidad y la potencia necesaria para impulsar las comunicaciones, con beneficios de gran alcance para los proveedores de servicios y los desarrolladores de productos hasta los usuarios finales.

Por lo tanto, SDR hace que la comunicación de RF se realice mediante software o firmware, para llevar a cabo tareas de procesamiento de señales que normalmente son procesadas por hardware. Este hardware incluye mezcladores, filtros, amplificadores, moduladores, demoduladores, etc. Por consiguiente, SDR solo usa un ADC y DAC para hacer conversión de señal analógica a digital y digital a analógica junto con antenas, sin necesidad de muchos componentes de hardware. (Lakshan, 2020)

La definición de radio del software SDR se puede aplicar a una variedad de situaciones:

- Radioaficionados: La tecnología de radio definida por software ha sido utilizada con bastante eficacia por los radioaficionados para aumentar la eficiencia y la versatilidad del dispositivo.
- Militar: La tecnología de radio definida por software ha sido ampliamente adoptada por los militares, lo que les permite reutilizar el hardware y actualizar las formas de onda de la señal según sea necesario.
- Investigación y desarrollo: Muchos proyectos de investigación se benefician de la radio definida por software o SDR. Sin tener que empezar desde cero, los radios se pueden programar para proporcionar las especificaciones exactas de transmisor y receptor para cualquier aplicación.
- Comunicaciones móviles: En áreas como las comunicaciones móviles, los radios definidos por software son extremadamente útiles. Al actualizar el programa, es posible realizar ajustes a cualquier especificación e incluso incorporar nuevas

formas de onda sin tener que actualizar el hardware. Esto también se puede lograr de forma remota, lo que resulta en un ahorro de costos significativo.

Por esta razón, es necesario que el laboratorio de telecomunicaciones cuente con guías de laboratorio ya que, al disponer con el equipamiento necesario para poder desarrollarlas, se puede dar un mejor aprovechamiento de estos dispositivos. (Princy A. J, 2021)

Los objetivos propuestos para este proyecto de grado son:

Objetivo general

- Diseñar y desarrollar un conjunto de prácticas de Comunicaciones Digitales utilizando dispositivos de Radio Definido por Software (SDR) para la carrera de Telecomunicaciones de la UNL.

Objetivos específicos

- Determinar el entorno de desarrollo de código abierto que se ajuste a los requerimientos del proyecto e identificar el hardware a utilizarse para el correcto establecimiento de la comunicación con el software de SDR
- Diseñar las prácticas de laboratorio acorde a la temática de los contenidos del sílabo de la asignatura de Comunicaciones Digitales y otras similares.
- Implementar los algoritmos diseñados específicos para cada práctica
- Realizar las guías de laboratorio que permitan al docente evaluar de forma sistemática las clases impartidas y a los estudiantes contar con una herramienta de apoyo en el desarrollo de las mismas.

4. Marco teórico

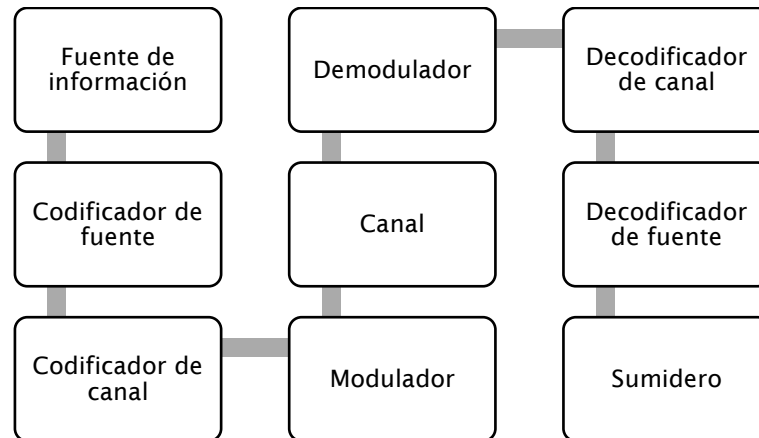
4.1. Comunicaciones digitales

Según Artés Rodríguez et al., (2012) el concepto “comunicaciones digitales” suele evocar la idea de señales de voltaje que cambian entre dos niveles que representan el uno y el cero y que, aunque distorsionadas al ser transmitidas por el canal, pueden ser reconstruidas en el destino gracias a su carácter discreto. Por tanto, en un instante de tiempo dado el transmisor tiene a su disposición un conjunto finito y discreto de formas de onda que puede hacer corresponder con los bits a transmitir. Para un sistema de comunicaciones digitales su objetivo no es otro que comunicar de manera fiable cada bit, con independencia de qué representa.

4.1.1. Elementos de un Sistema de Comunicaciones Digitales

Consta de algunos elementos, el transmisor está compuesto por el codificador de fuente, codificador de canal y modulador. Mientras que, el demodulador, el decodificador de canal y el decodificador de fuente componen el receptor. (Ver Figura 1)

Figura 1. Diagrama sobre los elementos fundamentales de un sistema de comunicaciones digitales



Fuente: Elaboración propia

En primer lugar, tenemos la fuente de información, ésta puede ser analógica o digital. El segundo bloque corresponde a la codificación de fuente, se encarga de representar la información de fuente con la menor cantidad de bits posible; es por ello que en ocasiones se habla de compresión de la información y naturalmente cuanto más redundancia, mayor compresión se puede

alcanzar. Luego, la codificación del canal, protege los bits fruto de la compresión contra errores producidos en el canal, cuanta más redundancia se haya eliminado más importante será cada bit.

La secuencia de bits que sale del codificador del canal debe ser transformada en formas de onda que se adapten a las características del canal, por eso la misión del modulador digital es hacer corresponder las formas de onda de un conjunto finito y discreto con los bits que se tienen a su entrada. El canal, es el medio físico sobre el que transmitimos la señal que sale del modulador. Todos los medios físicos deterioran la señal transmitida al menos de dos maneras: por un lado, distorsionan la forma de onda y por otro corrompen la señal incorporando ruido.

Los bloques del demodulador digital, decodificador de canal y decodificador de fuente realizan las funciones inversas al modulador digital, codificador de canal y codificador de fuente, respectivamente. Finalmente, el bloque sumidero, éste implica la vuelta al mundo analógico en aquellos casos en que la señal a transmitir es de esta naturaleza, y la presentación o almacenamiento en aquellos casos en que la fuente es digital. (Artés Rodríguez et al., 2012)

4.1.2. Modulaciones digitales

La modulación digital es el proceso por el que los símbolos digitales se transforman en señales compatibles con el canal de comunicaciones. El sistema de comunicaciones digital se puede clasificar según la banda del espectro que utiliza.

Si se utiliza la banda de frecuencias más baja, las señales y los canales reciben el nombre de banda base. Si los canales disponibles están en bandas de frecuencia más elevadas, las señales, sistemas o canales que se utilizan reciben el nombre de paso banda.

Existen diferentes tipos de modulaciones, tales como:

Modulación por desplazamiento de amplitud (ASK) o también conocida como On-Off Keying (OKK) es una modulación de amplitud donde la señal moduladora es digital y los valores binarios se representan con dos amplitudes diferentes, de los cuales uno de los dígitos binarios se presenta mediante la presencia de la portadora a amplitud constante, y el otro mediante la ausencia de la señal portadora. La modulación ASK reside en la alteración de la amplitud de la señal seno que es la portadora en función a los pulsos codificados.

Modulación por desplazamiento de fase (PSK), consiste en dos señales de energía finita, que están fuera de fase π radianes (o 180°) para representar el bit 1 y el bit 0, respectivamente, durante un tiempo de bit arbitrario

Modulación por desplazamiento de frecuencia (FSK), en este tipo de modulación la señal portadora cambia para representar los unos y los ceros binarios. La frecuencia de la señal durante la duración del bit es constante y su valor depende de si este es cero o uno. La modulación FSK es poco sensible al ruido eléctrico ya que el cambio en frecuencia es más difícil de alterar que un cambio en la amplitud de la señal. Para conseguir una modulación, se necesita una señal digital binaria de información, con unos niveles de tensión de 0 a 1 voltios y una anchura de bits. A diferencia de la modulación ASK, se utilizan dos señales portadoras de alta frecuencia, ambas de amplitud (A), pero con frecuencias diferentes.

Modulación de amplitud en cuadratura (QAM), es una modulación lineal que consiste en modular en doble banda lateral dos ondas portadoras de la misma frecuencia desfasadas 90° . Estas dos ondas generalmente son señales sinusoidales en la cual una onda es la portadora y la otra es la señal de datos. La señal modulada QAM es el resultado de sumar ambas señales, las cuales pueden operar por el mismo canal sin interferencia mutua porque sus portadoras al tener tal desfase, se dice que están en cuadratura. Al utilizar distintas combinaciones de amplitud y fase, permite obtener para una misma velocidad de modulación una mayor tasa de bits.

Modulación por desplazamiento mínimo gaussiano (GMSK), es un tipo de modulación derivada de la MSK (Minimum Shift Keying). GMSK un esquema de modulación no lineal de fase continua. Se trata de una técnica que consigue suavizar las transiciones de fase entre estados de la señal, consiguiendo por lo tanto reducir los requerimientos de ancho de banda. Con GMSK, los bits de entrada representados de forma rectangular (+1;-1) son transformados en pulsos Gaussianos mediante un filtro Gaussiano, para posteriormente ser suavizados por un modulador de frecuencia. (Quinde, 2017), (Miranda Torres, 2015), (Álvarez, 2014)

4.2. Multiplexación

Este término hace referencia al proceso mediante el cual diferentes mensajes de información (como conversaciones telefónicas, transmisión simultánea de audio + vídeo + datos o de diferentes canales de audio) generados típicamente en una misma ubicación física, se combinan

en una única señal con el fin de compartir un recurso de comunicaciones. Se utiliza esta tecnología con mayor frecuencia en redes de telefonía y en servicios de difusión. Por lo tanto, el objetivo de la multiplexación es compartir la capacidad de transmisión de datos sobre un mismo enlace para aumentar la eficiencia; y también busca minimizar la cantidad de líneas física requeridas y maximizar el uso del ancho de banda de los medios. (Micolau & Tarrés, 2021)

Existen muchas estrategias de multiplexación según el protocolo de comunicación empleado, que puede combinarlas para alcanzar el uso más eficiente; los más utilizados son:

4.2.1. Multiplexación por división de tiempo o TDM

Es una técnica para compartir un canal de transmisión entre varios usuarios. Consiste en asignar a cada usuario, durante unas determinadas “ranuras de tiempo”, la totalidad del ancho de banda disponible.

4.2.2. Multiplexación por división de frecuencia o FDM

Esta técnica consiste en dividir mediante filtros el espectro de frecuencias del canal de transmisión y desplazar la señal a transmitir dentro del margen del espectro correspondiente mediante modulaciones, de tal forma que cada usuario tiene posesión exclusiva de su banda de frecuencias. (Tecnológico Nacional de México, 2020)

4.2.3. Multiplexación por división de frecuencias ortogonales o OFDM

Es una multiplexación que consiste en enviar un conjunto de ondas portadoras de diferentes frecuencias ortogonales entre sí, donde cada una transporta información. En lugar de dejar espacio de guarda entre subportadoras, en OFDM se encuentran cercanas y ortogonales entre sí, haciendo que su ancho de banda se sobreponga.

4.2.4. Multiplexación por división en código o CDM

Basada en el uso de distintas codificaciones para cada canal, que pueden ser transmitidos compartiendo tiempo y frecuencia simultáneamente. En los sistemas CDM todos los usuarios transmiten en el mismo ancho de banda simultáneamente, a los sistemas que utilizan este concepto se les denomina "sistemas de espectro disperso". En lugar de utilizar las ranuras de tiempo o frecuencias, como lo hacen las tecnologías tradicionales, usa códigos matemáticos para transmitir y distinguir entre conversaciones inalámbricas múltiples. (Sánchez, 2019)

4.3. Acceso Múltiple

Es una técnica por medio de la cual los suscriptores pueden compartir el uso de un canal de comunicación casi al mismo tiempo, a pesar de que sus transmisiones individuales puedan originarse en lugares muy diferentes. Dicho de otra manera, el acceso múltiple permite que un gran número de usuarios que buscan comunicarse entre sí compartan los recursos de comunicaciones de un canal.

4.3.1. Acceso múltiple por división de tiempo o TDMA

Se divide el canal en ranuras temporales o *slots*. Una trama consta de N *slots*, asignados a cada uno de los usuarios. Hace uso de una única frecuencia y su uso mayoritario fue en sistemas 2G: GSM

4.3.2. Acceso múltiple por división de frecuencia o FDMA

Cada canal/usuario usa una frecuencia diferente. Se caracteriza por el uso de canales de guarda para evitar la interferencia con canales adyacentes. Es apropiado para las modulaciones analógicas y en los sistemas de comunicaciones móviles 1G.

4.3.3. Acceso múltiple por división de frecuencias ortogonales o OFDMA

Su principal función es dividir el flujo de datos cuando se transmite un gran número de subportadoras ortogonales de banda angosta por medio de la Transformada Inversa de Fourier (IFFT), lo que permite un mayor periodo de símbolo. Esto último, junto con el uso de un intervalo de guarda se adjunta al comienzo de cada símbolo OFDM, esta tecnología proporciona gran robustez contra la transmisión multitrayecto. Si el prefijo cíclico es más largo que el retardo del canal, se evitan las degradaciones por la interferencia entre símbolos (ISI) y la interferencia interportadora (ICI).

4.3.4. Acceso múltiple por división de código o CDMA

Cada señal de banda estrecha se multiplica por la señal de ensanchamiento (*spreading*). Cada usuario tiene su propia secuencia, lo que permite que todos los usuarios transmitan simultáneamente en toda la banda. Las transmisiones del resto de usuarios aparecen como ruido. (Serrano et al., 2015)

4.4. Radio Definido por Software

Un sistema de radio definido por software (SDR) es un sistema de comunicación por radio que utiliza software para la modulación y demodulación de señales de radio. Un SDR realiza cantidades significativas de procesamiento de señales en una computadora de uso general o en una pieza electrónica digital reconfigurable. El objetivo de este diseño es producir una radio que pueda recibir y transmitir una nueva forma de protocolo de radio simplemente ejecutando un nuevo software.

4.4.1. Aplicaciones de SDR

La reconfiguración de los SDR a partir de software hace que sus aplicaciones sean extensas. Siendo una herramienta ideal para investigación y educación.

Algunos estudios han identificado aplicaciones significativas, como: Posicionamiento Dinámico del Espectro, Acceso Múltiple División de Frecuencia (ODMA), Regulación del Espectro y Reducción de. La filosofía SDR empieza a despuntar en áreas de gran impacto dentro de las telecomunicaciones. Es el caso de la Asistencia al Conductor, la Recepción de Señales en el Sistema de Posicionamiento Global (GPS), el Análisis de Propagación en Frecuencias Altas (HF), la Interpretación de Emisiones de Tecnología Celular en particular la modulación OFDM, y la Identificación de Emisiones de Radiofrecuencia. En otros campos visionarios, los experimentos con SDR han proporcionado resultados alentadores que motivan a continuar las investigaciones.

Se están encontrando aplicaciones potenciales en áreas tan diversas como el desarrollo de prototipos de investigaciones microscópicas de la fuerza de la resonancia magnética, las pruebas de aviación, la evaluación de las comunicaciones multitrayectoria, las transmisiones de radiodifusión en entornos móviles multimedia, la diversidad de las redes inalámbricas cooperativas, los prototipos de cruces entre capas de redes inalámbricas, las comunicaciones ópticas cuánticas y, particularmente, en la investigación de la radio cognitiva. (Manchado Fernández, 2014)

4.4.2. Componentes funcionales de SDR

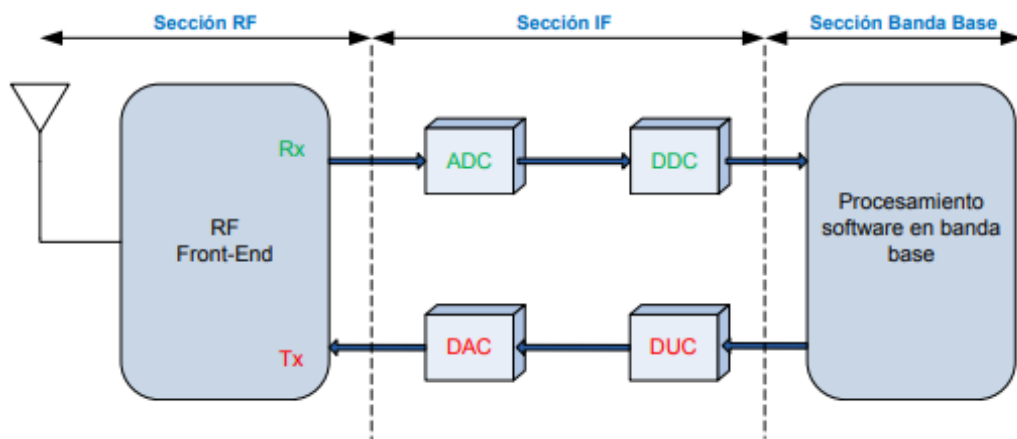
Está compuesta por tres bloques funcionales: sección de radiofrecuencia (RF), sección de frecuencias intermedias (IF) y sección banda base (Ver Figura 2). La parte de RF e IF se implementan en hardware mientras que la sección de banda base en software.

La sección de RF, también denominada RF Front-End o cabecera de RF, es la encargada de transmitir/recibir las señales de radiofrecuencia para adecuarlas y convertirlas en frecuencia intermedia en recepción o amplificar y modular las señales de IF en el caso de transmisión.

La frecuencia intermedia puede ser 0, dando lugar al concepto de Zero-IF; el cual es posible gracias a los avances en los componentes hardware. De igual manera, la sección de IF se encarga de pasar la señal de IF a banda base y digitalizarla en recepción o pasar la señal de banda base a IF y hacer la conversión digital - analógica de la señal en el caso de la transmisión. Las encargadas de la conversión analógica - digital o digital-analógica de la señal son los módulos ADC/DAC. A su vez, se insertan los módulos DDC/DUC para poder bajar/subir, respectivamente, la tasa de muestreo en el sentido de recepción/transmisión, consiguiendo que la tasa de muestras por la interfaz entre IF y banda base sea inferior.

La sección de **banda base** es la encargada de todo el procesamiento en banda base de la señal como modulación/demodulación, análisis espectral de la señal, llevándose a cabo en software. (Pinar Domínguez & Murillo Fuentes, 2011)

Figura 2. Diagrama de bloque funcionales de SDR



Fuente: (Pinar Domínguez & Murillo Fuentes, 2011)

4.4.3. Niveles de SDR

Con el fin de establecer una clasificación de los equipos de radio, el Wireless Innovation Forum ha definido 5 niveles de utilización de software dentro del radio para controlar o realizar funciones de la capa física, determinando la frontera entre hardware y software en el equipo:

- a. **Nivel 0:** Radio construido utilizando hardware solamente, no puede cambiarse por software. En este nivel no hay software, ni en el control ni en la realización propia de las funciones de operación del radio.
- b. **Nivel 1:** Radio controlado por software con limitaciones en cuanto a funciones controlables. Se controlan algunas como nivel de potencia, interconexiones, etc. pero nunca modo o frecuencia.
- c. **Nivel 2:** En este nivel una parte significativa del radio es configurable por software. Usualmente se utiliza el término Radio Controlado por Software (SCR). Existe control de software de ciertos parámetros como frecuencia, modulación, generación/detección de forma de onda, seguridad, etc. La etapa de RF permanece en hardware y no puede ser reconfigurada. Es importante destacar que el software en este tipo de radios sólo controla funciones que están implementadas de modo físico dentro del radio, a hardware únicamente.
- d. **Nivel 3:** Este nivel agrupa todos los radios en los que al menos una de sus funciones está definida por software, incluyendo lo que se conoce como Radio Definido por Software Ideal (ISR) donde la frontera entre la parte configurable y la no configurable se encuentra muy cercana a la antena, y la etapa final de RF es configurable. Se puede decir que el ISR es completamente programable.
- e. **Nivel 4:** Los equipos pertenecientes a esta categoría reciben la denominación de Ultimate Software Radio (USR) y constituyen una etapa superior al ISR puesto que no solo son completamente programables, sino que además son capaces de soportar un amplio rango de funciones y frecuencias simultáneamente. Dentro de este nivel se encuentran los teléfonos celulares con soporte para varios estándares. (García Algora, 2011)

4.5. Entornos de software para SDR

Las plataformas SDR disponibles convierten señales de RF en tiempo real, en muestras en banda base digital y utilizan un mecanismo definido por software para técnicas de modulación y demodulación para transferir datos del mundo real. La radio puede ser configurada con los siguientes parámetros: frecuencia central, frecuencia de muestreo, ancho de banda y otros que operan en la transmisión y recepción de señales. En referencia a las técnicas de modulación y demodulación, éstas cumplen con un proceso que consta de dos pasos:

1. Desarrollar, ajustar y optimizar los algoritmos de modulación y demodulación para una frecuencia de muestreo, un ancho de banda y un entorno específico.
2. Tomar el algoritmo anterior desarrollado para un entorno de producción, basándose en normas de seguridad, de acuerdo al tamaño, peso, potencia y costo de un producto. (Cevallos Durán, 2022)

4.5.1. GNU – RADIO

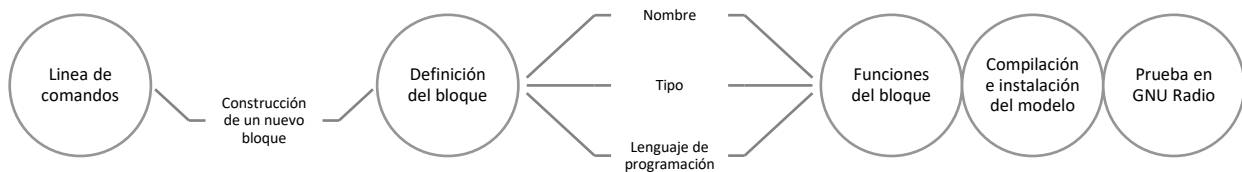
GNU Radio es una herramienta de desarrollo de software de código libre para la implementación de Radios Definidos por Software en plataformas Linux, Mac OS y Windows. Generalmente GNU Radio, se utiliza en ambientes de estudio para el espectro electromagnético en su parte de radio, permite además de recibir estas señales, analizarlas mediante bloques de procesamiento, las señales también pueden ser generadas y transmitidas. Algunos de los principales bloques de hardware disponibles en GNU Radio son: filtros, moduladores, demoduladores, codificadores, decodificadores, capas de sincronización, transmisores y receptores, entre otros elementos que componen un radio.

El código en GNU Radio se desarrolla principalmente en lenguaje de programación Python y cuando se requiere alto rendimiento en la implementación de los algoritmos se utiliza C++. También es posible desarrollar aplicaciones utilizando la interfaz gráfica de usuario, la cual permite implementar sistemas de comunicaciones a partir de bloques que realizan funciones básicas de procesamiento de señales, sin necesidad de escribir ni una sola línea de código. (Fernández & Enriquez, 2015)

- **Creación de bloques para GNU Radio**

Para crear un bloque nuevo en GNU Radio (conocido como Out Of Tree), se emplea un script conocido como gr_modtool preinstalada en Linux, que ayuda a la edición automática de makefiles, en otras palabras, ayuda a generar de manera automática una plantilla. (Ver Figura 3)

Figura 3. Metodología para la creación de bloques en GNU Radio



Fuente: Elaboración propia

Mediante la línea de comandos (terminal) se genera la estructura de un bloque nuevo, pero este no contara con ninguna función, servirá como una plantilla para que una vez avanzado el proceso de creación se le otorgué una funcionalidad, posteriormente se asigna un nombre de identificación, el tipo de información que será capaz de recibir y devolver, y como punto importante, de lo trabajado en línea de comandos se generaran archivos donde será programado el bloque, se dice que es importantes porque de esto depende la correcta función del bloque, por último se guarda, compila e instala en GNU Radio mediante línea de comandos, al finalizar el nuevo bloque se podrá encontrar en la biblioteca de bloques de GNU Radio con su respectivo nombre de identificación.

4.5.2. Matlab – Simulink

Simulink es una herramienta de simulación que corre bajo Matlab, cuenta con su propio lenguaje de programación (llamado M) y proporciona una interfaz gráfica de usuario (GUI) que es usada para crear modelos de sistemas para cualquier operación de procesamiento específico, mediante la realización de simulaciones y análisis de resultados.

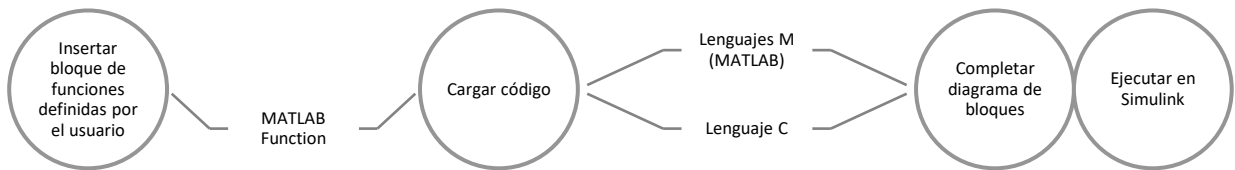
Es compatible con los sistemas lineales y no lineales, modelados en tiempo continuo, el tiempo de la muestra, o un híbrido de los dos. Los sistemas también pueden ser de múltiples velocidades, es decir, tener diferentes partes que se muestrean o actualiza a diferentes velocidades.

Las simulaciones son interactivas, por lo que puede cambiar los parámetros sobre la marcha y ver inmediatamente qué sucede. Tiene acceso instantáneo a todas las herramientas de análisis en MATLAB, por lo que puede tomar los resultados, analizarlos y visualizarlos.

- **Creación de bloques para Matlab – Simulink**

La metodología para la creación de un bloque en Matlab-Simulink, es bastante simple y amigable, consistiendo en cargar un código a un bloque de funciones, esta se muestra más a detalle en la figura 4.

Figura 4. Metodología para la creación de bloques usando MATLAB – Simulink



Fuente: Elaboración propia

Dentro de la interfaz Simulink, se buscará en la librería de componentes el bloque llamado “MATLAB Function” en el apartado “definido por el usuario”, una vez insertado, se carga el código (lenguaje C o M), tomando en cuenta los elementos de entrada y salida, al concluir con la programación del bloque se guarda, para comprobar su funcionamiento se debe integrar a un diagrama de bloques (basta con agregar un bloque de entrada y salida) y ejecutar dicho diagrama.

4.5.3. Comparación entre GNU Radio y Matlab – Simulink

Ambos softwares permiten trabajar con el procesamiento de señales, simulando sistemas de comunicación a nivel de software y hardware esto implica utilizar tarjetas de audio o hardware como los dispositivos SDR. Estos programas son utilizados en diferentes entornos de investigación, académicos, industria y gobiernos. A continuación, se mencionan algunas características de GNU Radio y de Matlab.

Tabla 1. Comparación entre los entornos para simulaciones SDRs

Características	GNU Radio	MATLAB - Simulink
Costo (Licencia)	GPL (General Public License o Licencia Publica General)	Licencia Standard: -Anual: USD 860 -Permanente: USD 2,150 Licencia Educación: -Anual: USD 250 -Permanente: USD 500
Complementos	Code blocks, Sublime Text 3	Ninguno, todo se encuentra en Matlab - Simulink
Interfaz	Gráfica	Gráfica
SDR's	La mayoría de los lanzados al mercado	Ettus research y otros como HackRF sin soporte oficial
Creación de bloques	Línea de comandos	Interfaz gráfica de Matlab - Simulink
Sistema Operativo	Windows, Linux Mac OS	Windows, Linux Mac OS

Fuente: Elaboración propia

5. Metodología

5.1. Hardware y software

En esta sección se describen algunas características técnicas que se consideraron importantes conocer sobre los módulos SDR que fueron utilizados para el desarrollo de las guías de laboratorio. Así mismo, se describen algunos programas apropiados que ofrecen al usuario una interfaz amigable para su operación. A continuación, se mencionan y detallan los dispositivos utilizados en el presente proyecto.

Módulo RTL-SDR

Tabla 2. *Especificaciones técnicas del dispositivo RTL-SDR*

Funcionamiento	Receptor
Rango de Frecuencias	25 MHz a 1.7 GHz
Frecuencia de muestreo	2.56 – 3.2 MSps
Resolución ADC	8 bits y ~7 (ENOB)
Sintonizador	RTL2832
Interfaz USB	2.0

Fuente: Elaboración propia

Módulo LimeMini-SDR

Tabla 3. *Especificaciones técnicas del dispositivo LimeMini – SDR*

Funcionamiento	Transmisor y receptor
Rango de Frecuencias	10 MHz a 3.5 GHz
Frecuencia de muestreo	30.72 MSps
Resolución ADC y DAC	12 bits
Transceptor de RF	LMS7002M
Ancho de banda	40 MHz

Fuente: Elaboración propia

Tabla 4. *Especificaciones técnicas del dispositivo Adalm Pluto – SDR*

Funcionamiento	Transmisor - Receptor
Rango de Frecuencias	325 MHz a 3.8 GHz
Frecuencia de muestreo	3.2 MSps
Resolución ADC y DAC	12 bits
Ancho de banda	20 MHz
Interfaz USB	2.0

Fuente: Elaboración propia

Las herramientas de software que permiten trabajar con los módulos SDR fueron:

- Matlab: La versión empleada en el proyecto es la R2022b Update. Sin embargo, si desea instalar otra versión se recomienda una posterior a 2018-A. Se eligió la antes mencionada porque sus desarrolladores han implementado el paquete de soporte de Communications Toolbox para dispositivos analógicos, permitiendo configurar las características de los equipos SDR tanto para TX/RX de señales. (MATLAB & Simulink^[+], 2023)
- Dump1090: El Dump1090 es un decodificador ADS-B de línea de comandos para usar con dongles RTL-SDR que le permite recibir y mostrar ubicaciones de aeronaves cercanas en un mapa. (Radioescucha, 2021)
- SDRangel: es una aplicación casi en tiempo real que exige potencia de CPU y velocidades de reloj para baja latencia. Es utilizado para aplicaciones SDR y el procesamiento de señales digitales en general, sirve como una interfaz de analizador de señal y Qt5 / OpenGL 3.0+ de código abierto para varios hardware. (doxygen, 2019)

En cuanto a la realización de las guías de laboratorio se define una estructura que se encuentra ya establecida en el laboratorio de telecomunicaciones. Cada una de las prácticas debe contar con todas las siguientes partes: tema, objetivos, materiales, equipos y herramientas,

instrucciones, actividades por desarrollar, marco teórico, resultados obtenidos, discusión, conclusiones, recomendaciones y finalmente las preguntas de control.

5.2. Guías de laboratorio

En la malla curricular de la carrera de Telecomunicaciones la asignatura de Comunicaciones Digitales es impartida en el quinto módulo, debido a esto se desarrollaron cinco prácticas de laboratorio utilizando Radio Definida por Software. Los estudiantes y el docente encargado de la práctica podrán conocer a detalle el procedimiento a seguir de cada una de ellas y analizarán las capacidades que ofrece esta tecnología.

5.2.1. PRÁCTICA # 1

ASIGNATURA: COMUNICACIONES DIGITALES

RESULTADO DE APRENDIZAJE DE LA PRÁCTICA: (VER SÍLABO)

TIEMPO PLANIFICADO EN EL SÍLABO: 2 HORAS

TIEMPO DE LA PRÁCTICA POR GRUPO: 2 HORAS

NÚMERO DE ESTUDIANTES POR GRUPO: 2 ESTUDIANTES

NÚMERO DE ESTACIONES DE TRABAJO: 10

Tema: Configuración de un receptor ADS-B utilizando RTL-SDR

Objetivos:

- Indagar los requisitos previos para implementar un receptor ADS-B
- Instalar los softwares específicos para la práctica
- Implementar un receptor ADS-B basado en SDR

<i>Materiales y Reactivos</i>	<i>Equipo y herramientas</i>
<ul style="list-style-type: none"> • Una antena telescópica sintonizada a 1090 MHz 	<ul style="list-style-type: none"> • Dongle NESDR Smartee • Software de escucha y decodificación (dump1090) • Computadora

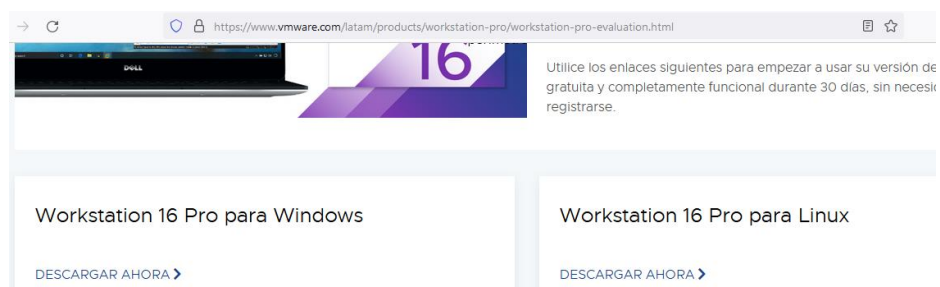
Instrucciones

- Colocar las mochilas en los casilleros.
- Prohibido consumo de alimentos.
- Prohibido equipo de diversión, celulares etc.
- Prohibido jugar.
- Prohibido mover o intercambiar los equipos de los bancos de trabajo.
- Prohibido sacar los equipos del laboratorio sin autorización.
- Ubicar los equipos y accesorios en el lugar dispuesto por el responsable del laboratorio, luego de terminar las prácticas.
- Uso adecuado de equipos.
- Uso obligatorio del mandil.

Actividades por desarrollar

1. Descargar la máquina virtual VMware Workstation 16 Player de <https://www.vmware.com/latam/products/workstation-pro/workstation-pro-evaluation.html>. Una vez seleccionado el instalador según los requisitos de su computador, proceder con la instalación.

Figura 5. Entorno de la página web de VMware Workstation 16 Player para la descarga de la máquina virtual en el sistema operativo de Windows y el sistema operativo de Linux



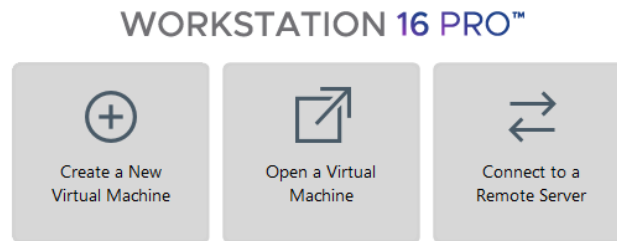
Fuente: Elaboración propia

2. Descargar la distribución GNU/Linux Ubuntu 22.04.1 del repositorio <http://www.ubuntu.com>, e instalarlo en la máquina virtual Workstation 16. Puede

guiarse del siguiente tutorial <https://www.youtube.com/watch?v=-jPVkjjmkXk> o puede realizar el siguiente procedimiento:

- Abrir el ícono de VMware Workstation 16 y crear la máquina virtual

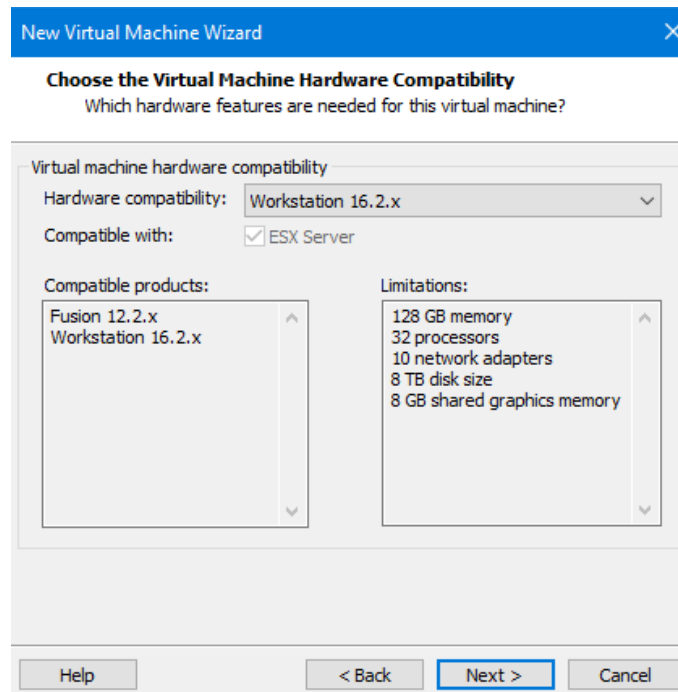
Figura 6. Interfaz al abrir el ícono de VMware Workstation 16



Fuente: Elaboración propia

- Elegir la opción Custom advanced para elegir las características de hardware necesarias para la máquina virtual.

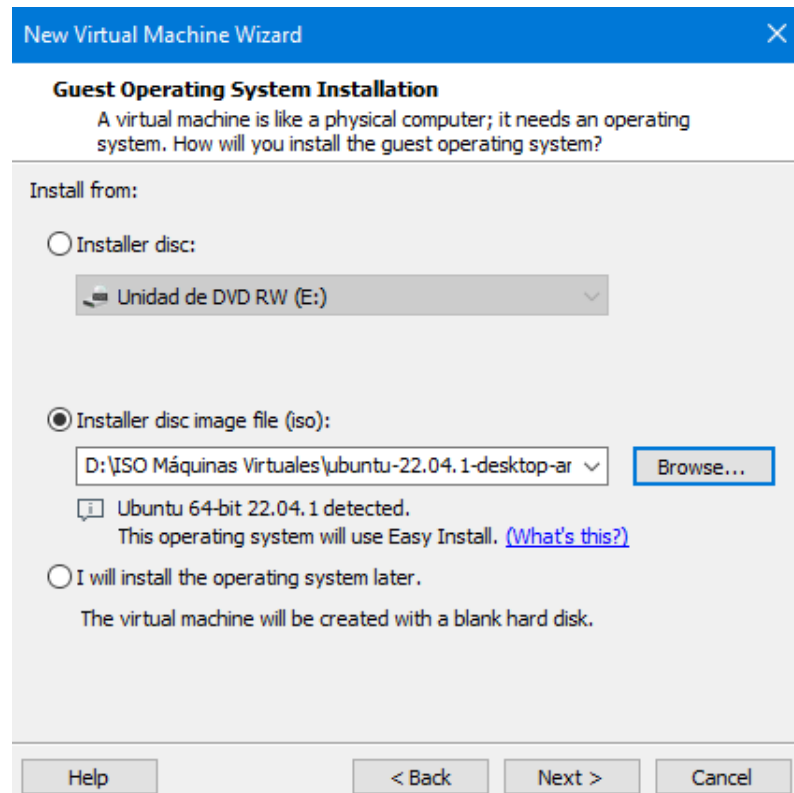
Figura 7. Características de compatibilidad de hardware para la creación de la máquina virtual



Fuente: Elaboración propia

- El sistema operativo se instalará mediante un archivo de imagen .ISO, previamente descargada de la página web de Ubuntu <http://www.ubuntu.com> (última versión Ubuntu 22.04.1)

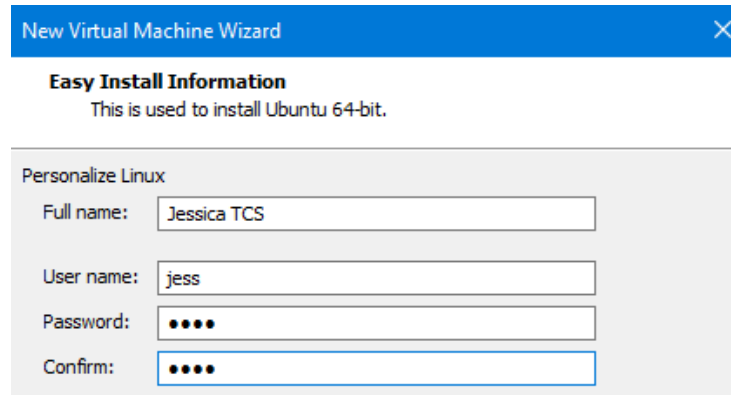
Figura 8. *Instalación del sistema operativo Linux en una máquina virtual: mediante un disco, a través de un archivo de imagen de disco o realizar la instalación más tarde.*



Fuente: Elaboración propia

- Después se personaliza Linux con un nombre de usuario y una contraseña.

Figura 9. Personalización de Linux mediante un usuario y una contraseña necesarios para cualquier configuración como administrador.



New Virtual Machine Wizard

Easy Install Information
This is used to install Ubuntu 64-bit.

Personalize Linux

Full name:

User name:

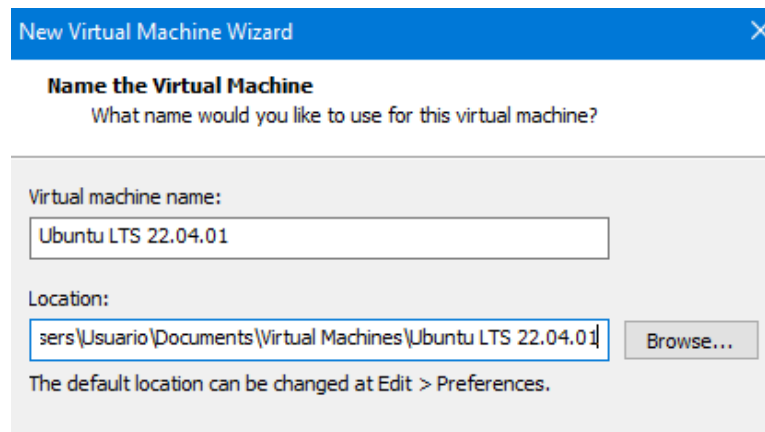
Password:

Confirm:

Fuente: Elaboración propia

- De la misma manera se asigna un nombre que identifique a la máquina virtual

Figura 10. Elección de un nombre para la máquina virtual y una carpeta de ubicación para la misma.



New Virtual Machine Wizard

Name the Virtual Machine
What name would you like to use for this virtual machine?

Virtual machine name:

Location:

The default location can be changed at Edit > Preferences.

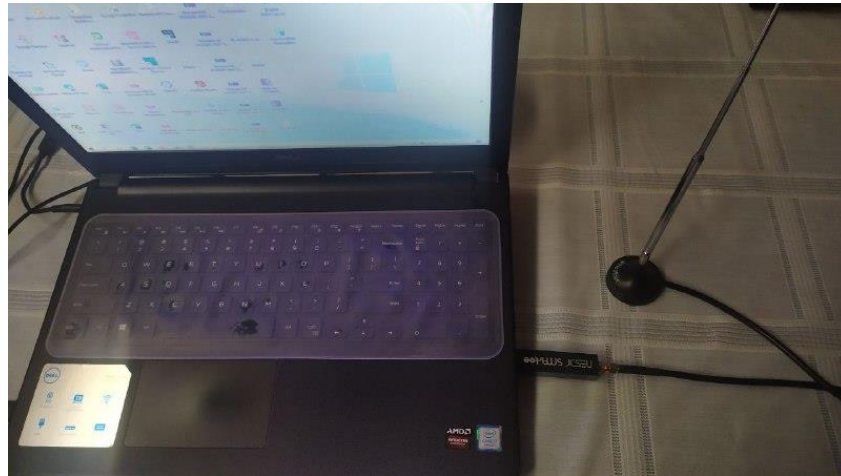
Fuente: Elaboración propia

- Finalmente, se configura la cantidad de procesadores (2), el tamaño de la memoria (4096 MB), el tipo de red (NAT), tipos de controlador de E/S (LSI Logic), tipo de disco virtual (SCSI), la capacidad del disco virtual (25GB) y se reinicia la máquina virtual.
3. Configurar el dispositivo RTL-SDR con los controladores que son dedicados a ser usado como un SDR, ya que el uso predeterminado del dispositivo es recibir

transmisiones de televisión. Se usará como referencia el repositorio de <https://git.osmocom.org/rtl-sdr.git>

- Conectar el dispositivo NESDR Smartee al puerto USB del computador

Figura 11. *Conexión del dispositivo NESDR a la antena y al computador para su correcto funcionamiento*



Fuente: Elaboración propia

- Luego en la ventana de terminal ejecutar los siguientes comandos para que el sistema está actualizado.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

- Después se debe instalar el software requerido para construir el controlador. Utilizar las siguientes líneas de código secuencialmente:

```
sudo apt-get install git-core
```

```
sudo apt-get install cmake
```

```
sudo apt-get install libusb-1.0-0-dev
```

```
sudo apt-get install build-essential
```

```
sudo apt-get install pkg-config
```

```
git clone https://git.osmocom.org/rtl-sdr.git
```

- Finalmente se construye y se instala el driver y finalmente se reinicia la máquina virtual.

```
cd rtl-sdr; mkdir build; cd build
```

```
cmake ../ -DINSTALL_UDEV_RULES=ON -DDETACH_KERNEL_DRIVER=ON
```

```
make
sudo make install
sudo ldconfig
sudo reboot
```

4. Instalar el software Dump1090 que se encargará de rastrear las aeronaves. Al ejecutar el software, el dispositivo RTL-SDR debe estar conectado al puerto USB 2.0 del computador con la antena en 1090 MHz.

- Actualiza el sistema

```
sudo apt-get update
sudo apt-get upgrade
```

- Clonar el software del repositorio del git e instalar

```
git clone https://github.com/antirez/dump1090.git
cd dump1090
make
```

- Ejecutar el software dump1090 en una consola, previamente debe conectar el dispositivo RTL-SDR ya con los paquetes configurados para ser utilizado como un dispositivo SDR

```
./dump1090 -interactive
```

También puede ejecutar dump1090 en una versión basada en web:

```
./dump1090 --interactive --net--net-http-port 8080
```

5. Visualizar la ubicación de las aeronaves rastreadas y verificar la información en la página de tráfico aéreo mundial <https://flightaware.com/>

Marco teórico (a elaborar por el estudiante)

- Funcionamiento del radar de vigilancia secundario SSR

El radar de vigilancia secundario (SSR), también conocido como transpondedor, es un sistema compuesto por dos estaciones: una en tierra, denominada interrogador; y otra en la aeronave, denominada respondedor. El interrogador interroga en la frecuencia de 1030 MHz, a lo que la aeronave responde con pulsos electromagnéticos en la frecuencia de 1090 MHz.

- Tecnología ADS-B

ADS-B significa Difusión de vigilancia dependiente automática, es una técnica de vigilancia que se basa en aeronaves o vehículos aeroportuarios que transmiten su identidad, posición y otra información derivada de los sistemas a bordo. Esta señal (ADS-B Out) se puede capturar con fines de vigilancia en tierra (ADS-B In) o a bordo de otras aeronaves para facilitar el conocimiento de la situación del tránsito aéreo, el espaciamiento, la separación y la auto separación (ADS-B In).

ADS-B permite a los pilotos y controladores de tráfico aéreo "ver" y controlar aeronaves con más precisión, es automático porque siempre está encendido y no requiere intervención del operador; dependiente porque depende de una señal GNSS precisa para los datos de posición; de vigilancia porque proporciona servicios de vigilancia "similares a radar"; y de difusión porque transmite continuamente la posición de la aeronave y otros datos a cualquier aeronave o estación terrestre equipada para recibir ADS-B

- Protocolos y estándares para la codificación y decodificación de datos

La Organización de Aviación Civil Internacional (OACI) posee dos sistemas ADS-B normalizados:

1090ES: Transpondedor de señales espontáneas extendidas Modo S. Opera en una de las dos frecuencias de enlace de datos aprobadas por la Administración Federal de Aviación (FAA). Opera en 1090 MHz). 1090ES es el estándar internacional; en los Estados Unidos, se requiere para operaciones por encima de 18,000 pies. Si bien el enlace de datos 1090 ES proporciona información de tráfico, no recibe datos meteorológicos.

UAT: Transceptor de Acceso Universal. Opera en una de las dos frecuencias de enlace de datos aprobadas por la Administración Federal de Aviación (FAA). Un UAT opera en 978 MHz (978UAT). Los costos de equipo generalmente son más bajos que para 1090ES, y esta frecuencia puede recibir información de tráfico y clima gratuita. Estados Unidos es el único país que utiliza el estándar UAT.

- Frecuencias de operación para sistemas ADS-B
 - Frecuencia de 1090 MHz
 - Frecuencia de 978 MHz

Resultados (a elaborar por el estudiante)

Discusión (a elaborar por el estudiante)

Conclusiones (a elaborar por el estudiante)

Recomendaciones (a elaborar por el estudiante)

Preguntas de Control (a elaborar por el estudiante)

- a) ¿Quiénes necesitan de un receptor ADS-B y cuáles son sus ventajas?
- b) ¿Qué modo de comunicación se utilizó para el desarrollo del receptor ADS-B?
- c) ¿Qué datos transmitió la(s) aeronave(s) que fueron encontradas con el receptor implementado?
- d) ¿En qué frecuencias operan los sistemas ADS-B?
- e) ¿Cuál es la diferencia entre ADS-B IN & ADS-B OUT?

Bibliografía (a elaborar por el estudiante)

5.2.2. PRÁCTICA # 2

ASIGNATURA: COMUNICACIONES DIGITALES

RESULTADO DE APRENDIZAJE DE LA PRÁCTICA: (VER SÍLABO)

TIEMPO PLANIFICADO EN EL SÍLABO: 2 HORAS

TIEMPO DE LA PRÁCTICA POR GRUPO: 2 HORAS

NÚMERO DE ESTUDIANTES POR GRUPO: 2 ESTUDIANTES

NÚMERO DE ESTACIONES DE TRABAJO: 10

Tema: Sintonizador de frecuencias GSM – FM utilizando el dispositivo RTL - SDR

Objetivos:

- Implementar el sistema SDR propuesto por Mathworks siendo utilizado para medir las señales de las diferentes frecuencias.
- Realizar la configuración adecuada para encontrar las frecuencias respectivas para cada ancho de banda correspondiente al rango de frecuencias para GSM y FM
- Identificar las herramientas importantes SDR de la librería Simulink

<i>Materiales y Reactivos</i>	<i>Equipo y herramientas</i>
<ul style="list-style-type: none"> • Antena telescópica 	<ul style="list-style-type: none"> • RTL NESDR Smartee • Computador • Matlab – Simulink

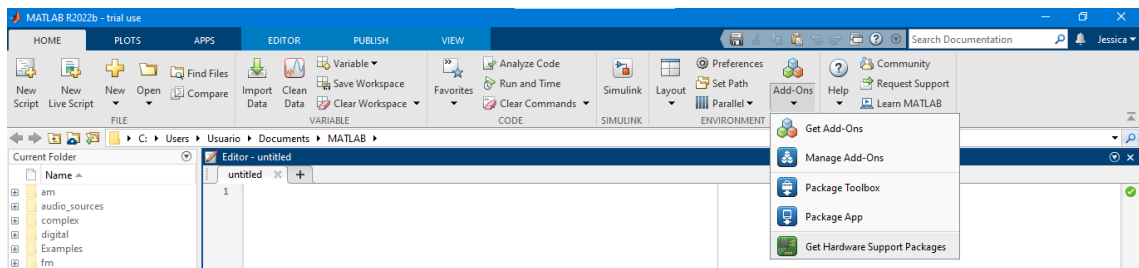
Instrucciones

- Colocar las mochilas en los casilleros.
- Prohibido consumo de alimentos.
- Prohibido equipo de diversión, celulares etc.
- Prohibido jugar.
- Prohibido mover o intercambiar los equipos de los bancos de trabajo.
- Prohibido sacar los equipos del laboratorio sin autorización.
- Ubicar los equipos y accesorios en el lugar dispuesto por el responsable del laboratorio, luego de terminar las prácticas.
- Uso adecuado de equipos.
- Uso obligatorio del mandil.

Actividades por desarrollar

1. Instalar el paquete “Communications Toolbox Support Package for RTL-SDR Radio”
 - Abrir el programa Matlab, en la barra de herramientas ubicada en la parte superior buscar la pestaña “HOME” luego desplegar la opción “Add Ons” y seleccionar “Get Hardware Support Package”

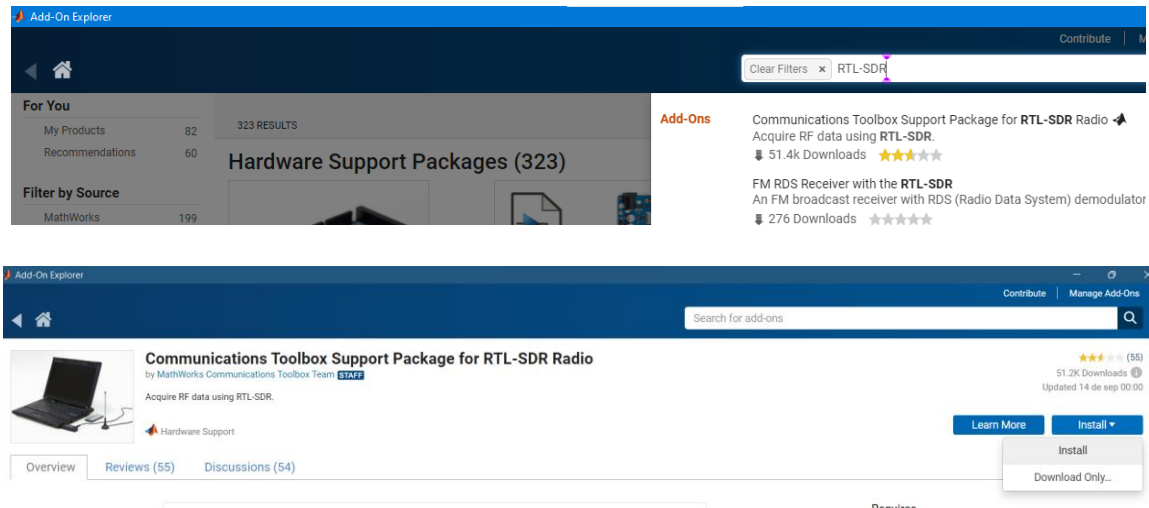
Figura 12. *Interfaz gráfica del programa Matlab de manera general*



Fuente: Elaboración propia

- A continuación, se abre automáticamente la ventana “Add On Explorer” donde se puede buscar el paquete con el nombre de “Communications Toolbox Support Package for RTL-SDR Radio” y al hacer clic se procede con la instalación.

Figura 13. Ventana de Add On Explorer que permite encontrar complementos



Fuente: Elaboración propia

- Se debe aceptar los términos de licencia, después en otra ventana indica la versión actual del dispositivo que se encuentra disponible. Para continuar con la instalación dar clic en “Next”

Figura 14. Aceptación de términos para instalar el paquete de RTL-SDR

MathWorks Auxiliary Software License Agreement

15. HEADINGS. The inclusion of headings is for convenience of reference only and shall not affect the construction or interpretation of this Agreement.

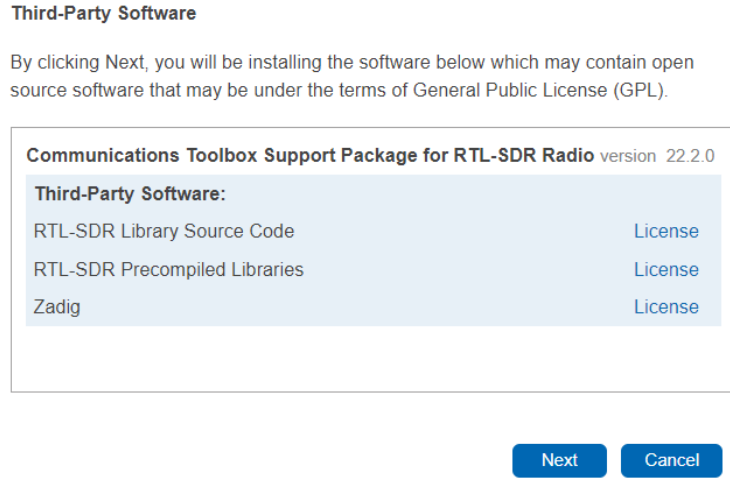
16. ENTIRE AGREEMENT. This Agreement, together with any additional license_addendum.txt file included with the Materials, contains the entire understanding of the parties with regard to the Materials, and may not be modified or amended except by written instrument, executed by authorized representatives of MathWorks and Licensee.

Version 2.0

I Accept Cancel

Fuente: Elaboración propia

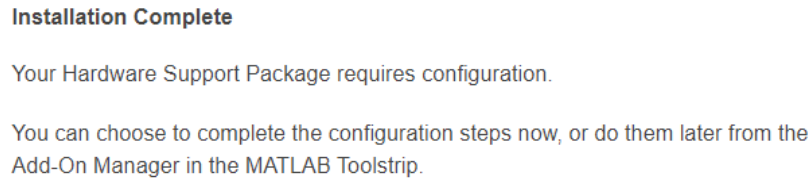
Figura 15. Paquetes que se han instalado para el dispositivo RTL-SDR Radio



Fuente: Elaboración propia

- Finalmente, la instalación del paquete se debe visualizar una ventana como la siguiente.

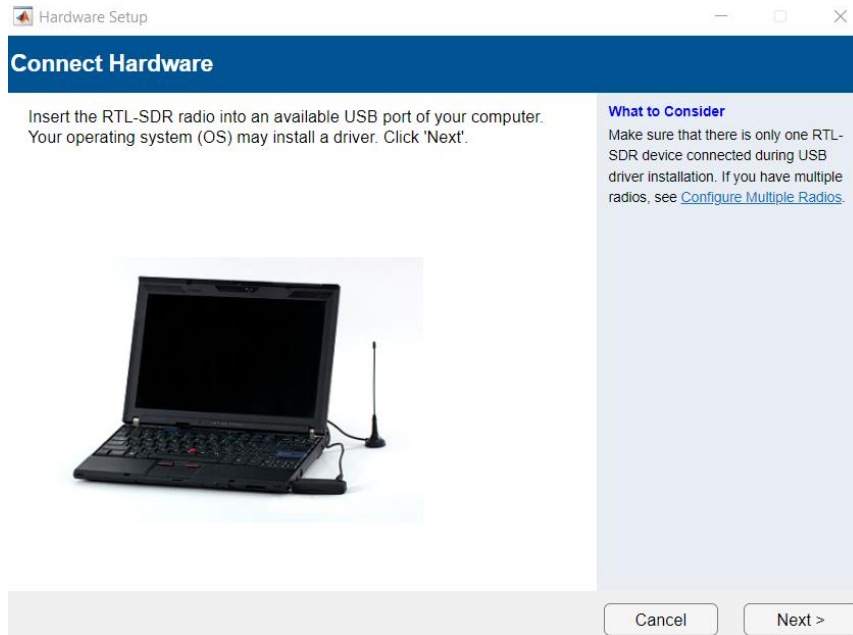
Figura 16. Instalación exitosa del paquete de Comunicación para el hardware RTL-SDR Radio



Fuente: Elaboración propia

2. Para continuar con el proceso, se debe configurar el dispositivo RTL con la instalación de los drivers necesarios para su funcionamiento. Para ello, se conecta el dispositivo al ordenador, luego dar clic en “Next”.

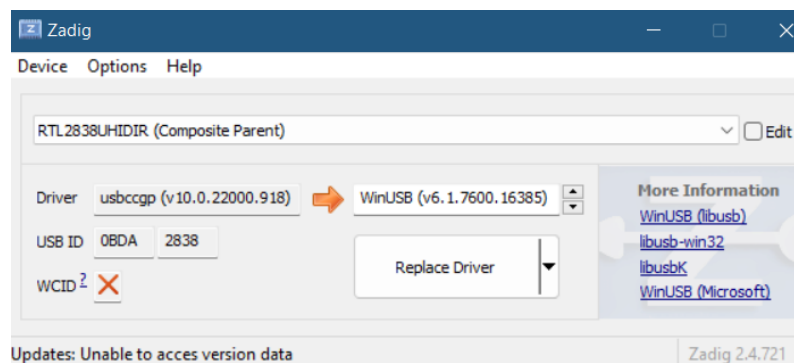
Figura 17. Conexión del dispositivo RTL-SDR para la instalación de los drivers



Fuente: Elaboración propia

- Uno de los drivers que se instalan y se ejecutan automáticamente es el software “Zadig” cuando se abre su interfaz elegir el nombre del dispositivo para que sea reconocido por el ordenador de la siguiente manera.

Figura 18. Aplicación Zadig para Windows que instala controladores USB genéricos



Fuente: Elaboración propia

- Si desea ver la información del dispositivo RTL ubicarse en la ventana de comandos de Matlab y digitar “my_rtlsdr = sdrinfo”

Figura 19. Información de configuración inicial del RTL-SDR

```
Command Window
New to MATLAB? See resources for Getting Started.
>> my_rtlsdr =sdrinfo

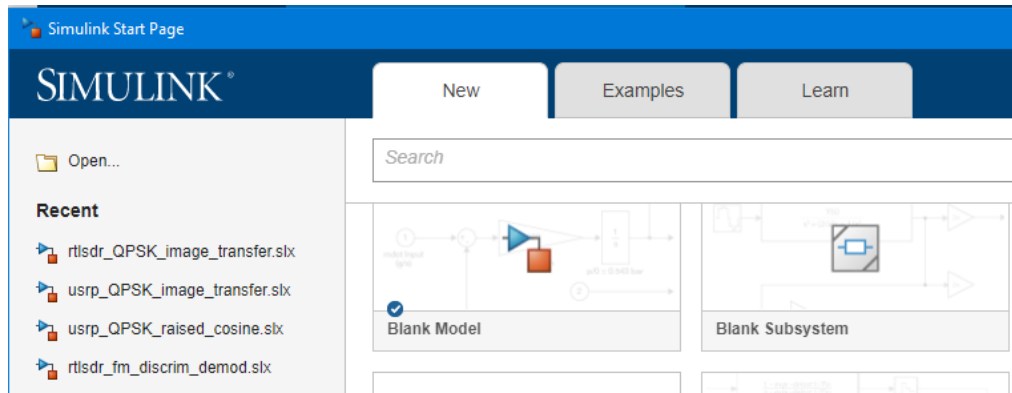
my_rtlsdr =

    RadioName: 'Generic RTL2832U OEM'
    RadioAddress: '0'
    RadioIsOpen: 0
    TunerName: 'R820T'
    Manufacturer: 'Realtek'
    Product: 'RTL2838UHIDIR'
    GainValues: [29x1 double]
    RTLCrystalFrequency: 28800000
    TunerCrystalFrequency: 28800000
    SamplingMode: 'Quadrature'
    OffsetTuning: 'Disabled'
```

Fuente: Elaboración propia

3. Construir el sintonizador de frecuencias.
- Abrir Matlab y buscar en la pestaña de “HOME” la sección de “Simulink” y para crear un nuevo modelo en blanco hacer clic en “Blank Model”

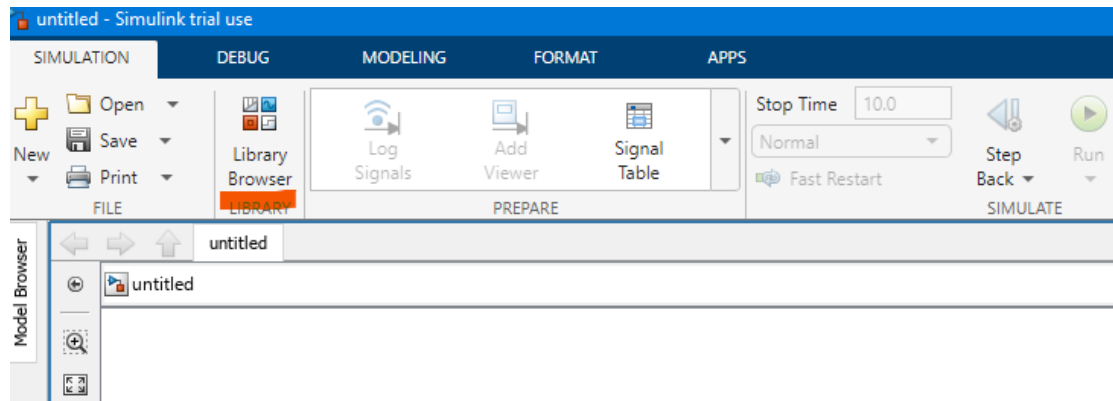
Figura 20. Interfaz gráfica de Simulink para crear un nuevo modelo en blanco



Fuente: Elaboración propia

- Para insertar los diagramas de bloques respectivos ir a la pestaña “SIMULATION” y buscar los diferentes componentes en “Library Browser”

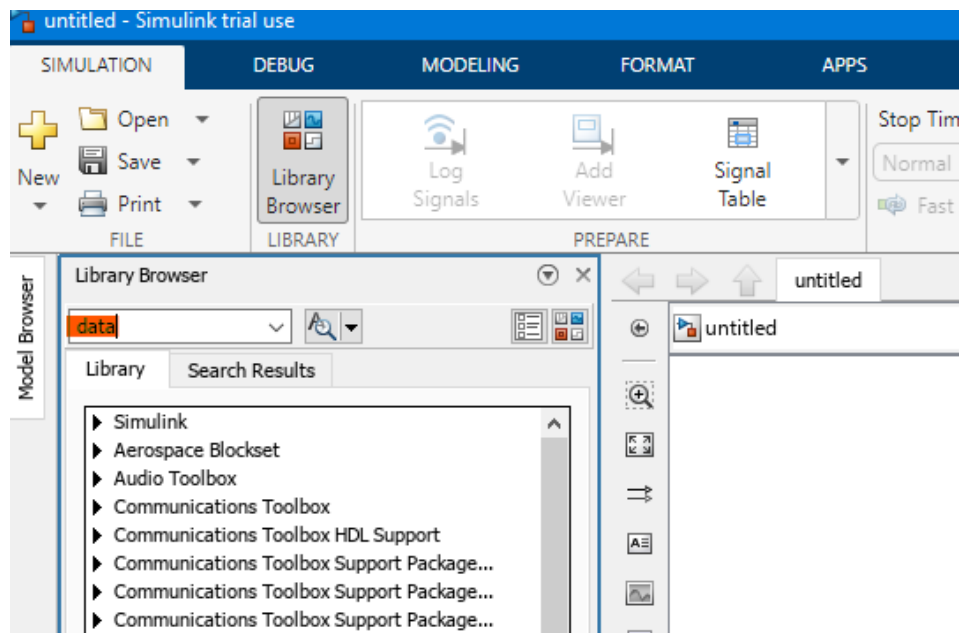
Figura 21. Barra de herramientas de Simulink donde se puede encontrar los componentes para la creación del diagrama



Fuente: Elaboración propia

- Para buscar un elemento de la biblioteca se puede realizar mediante palabras o frases clave, usando el cuadro de búsqueda en la parte superior del Explorador de biblioteca.

Figura 22. Explorador de biblioteca de la librería de Simulink permite buscar mediante palabras cortas el elemento deseado.



Fuente: Elaboración propia

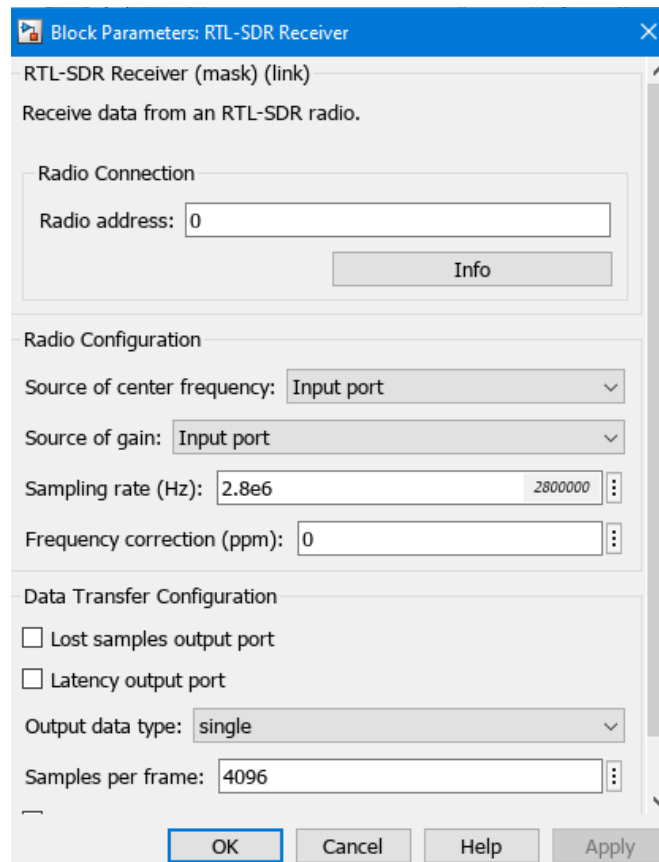
- Realizar el siguiente diagrama de bloques. Nota: el receptor RTL-SDR se configura los siguientes parámetros:

Sampling rate: 2.8 MHz

Output data type: single

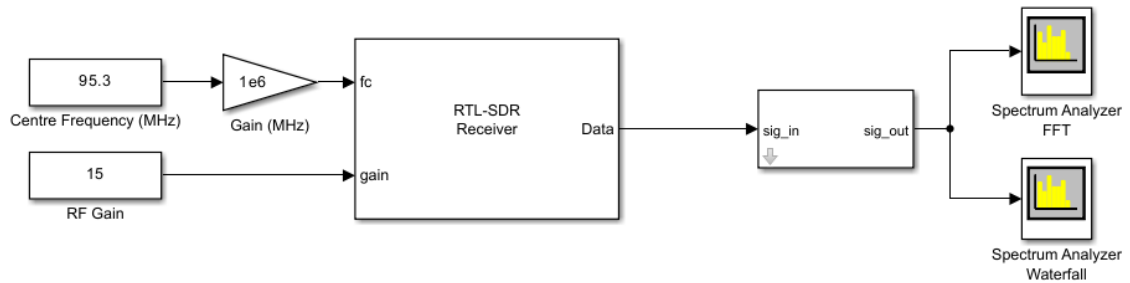
Samples per frame: 4096

Figura 23. *Bloque RTL-SDR para la configuración de los parámetros para su funcionamiento.*



Fuente: Elaboración propia

Figura 24. Diagrama de bloques que corresponde a funcionar como un sintonizador de frecuencias



Fuente: Elaboración propia

Marco teórico (a elaborar por el estudiante)

- ¿Qué es GSM y cómo funciona?

Las siglas de este estándar corresponden a Sistema Global de Comunicaciones Móviles, es conocido como 2G supuso un salto de las comunicaciones analógicas a las digitales. La banda de frecuencia en la que opera el GSM difiere según el territorio. En Europa se utiliza el espectro radioeléctrico de 900 y 1800 MHz, mientras que en Estados Unidos la banda es la de 1900. En la mayor parte de América Central y del Sur utilizan bandas de 1900 y 850. (VIU, 2018)

- Tecnología ADS-B

ADS-B significa Difusión de vigilancia dependiente automática, es una técnica de vigilancia que se basa en aeronaves o vehículos aeroportuarios que transmiten su identidad, posición y otra información derivada de los sistemas a bordo. Esta señal (ADS-B Out) se puede capturar con fines de vigilancia en tierra (ADS-B Out) o a bordo de otras aeronaves para facilitar el conocimiento de la situación del tránsito aéreo, el espaciamiento, la separación y la auto separación (ADS-B In).

ADS-B permite a los pilotos y controladores de tráfico aéreo "ver" y controlar aeronaves con más precisión, es automático porque siempre está encendido y no requiere intervención del operador; dependiente porque depende de una señal GNSS precisa para los datos de posición; de vigilancia porque proporciona servicios de vigilancia "similares a radar"; y de difusión porque

transmite continuamente la posición de la aeronave y otros datos a cualquier aeronave o estación terrestre equipada para recibir ADS-B

- ¿Qué son las estaciones móviles y las estaciones base?

Estaciones móviles, son aquellos teléfonos que necesitan de una tarjeta SIM para poder operar. Cambian de posición geográfica periódicamente. Ese desplazamiento es registrado por el Sistema Automático de Información de Posición (APRS). (Bogotá Positiva, 2019)

Estaciones base, es una torre fija de repetición quién es el usuario que se comunica a través de ella. Esta conexión se realiza a través de ondas de radio. Actúa como puente entre todos los usuarios de móviles de una misma célula, y conecta las llamadas de los móviles con la central de conmutación. (Scientific Committees, 2023)

- ¿Qué frecuencias y bandas son usadas en Ecuador?

En Ecuador se utilizan las siguientes frecuencias y bandas en telefonía móvil:

2G: B2 (1900), B5(850)

3G: B2 (1900), B5(850)

4G: B2 (1900), B4(1700/2100 AWS 1), B28 (700) (MovilesInfo, 2023)

- ¿Cuál es la frecuencia de oscilación de los tonos agudos, graves y medios?

La frecuencia del sonido de los tonos agudos oscila entre los 2000 y los 4000 Hz mientras los graves van desde los 125 a los 250 Hz. Los tonos medios tienen una frecuencia de oscilación entre 500 a 1000 Hz. (FotoNostra, 2022)

Resultados (a elaborar por el estudiante)

Discusión (a elaborar por el estudiante)

Conclusiones (a elaborar por el estudiante)

Recomendaciones (a elaborar por el estudiante)

Preguntas de Control (a elaborar por el estudiante)

- a) ¿Cuáles son los beneficios de GSM?

Uso generalizado en todo el mundo, GSM tiene un espectro armonizado, lo que significa que, aunque diferentes países pueden operar en diferentes bandas de frecuencia, los usuarios pueden transferir sin problemas entre redes y mantener el mismo número. Los usuarios de GSM tienen cobertura esencialmente en más de 218 países.

b) ¿Cuál es la separación de canales Uplink (UL) y Downlink (DL) de GSM?

En la banda GSM 850 hay una separación entre UL (824.0 – 849.0) y DL (869.0 – 894.0) de 45 MHz. Mientras que, en la banda GSM 1950 hay una separación entre UL (1850.0 – 1910.0) y DL (1930.0 – 1990.0) de 80 MHz

c) ¿Qué son los canales ARFCN?

Los canales ARFCN significan “Número de Canal de Frecuencia de Radio Absoluta”, es un número asignado a cada uno de los canales por los que se puede interconectar un móvil hacia la antena transmisora/receptora de GSM, este canal permite identificar por donde se transmite o recibe los datos.

Bibliografía (a elaborar por el estudiante)

5.2.3. PRÁCTICA # 3

ASIGNATURA: COMUNICACIONES DIGITALES

RESULTADO DE APRENDIZAJE DE LA PRÁCTICA: (VER SÍLABO)

TIEMPO PLANIFICADO EN EL SÍLABO: 2 HORAS

TIEMPO DE LA PRÁCTICA POR GRUPO: 2 HORAS

NÚMERO DE ESTUDIANTES POR GRUPO: 2 ESTUDIANTES

NÚMERO DE ESTACIONES DE TRABAJO: 10

Tema: Transmisión de video DATV utilizando LimeSDR - Mini

Objetivos:

- Configurar el dispositivo LimeSDR - Mini para que actúe como transmisor y el dispositivo RTL-SDR como receptor de DATV
- Analizar la importancia de los parámetros a configurar para la transmisión y recepción de DATV.

- Instalar los programas necesarios para el desarrollo de la práctica.

<i>Materiales y Reactivos</i>	<i>Equipo y herramientas</i>
<ul style="list-style-type: none"> • Antena telescópica • 2 antenas omnidireccionales multibanda 800 – 2700 MHz 	<ul style="list-style-type: none"> • LimeSDR - Mini • RTL NESDR Smartee • Software SDRangel • Software DATV Express • 2 computadoras

Instrucciones

- Colocar las mochilas en los casilleros.
- Prohibido consumo de alimentos.
- Prohibido equipo de diversión, celulares etc.
- Prohibido jugar.
- Prohibido mover o intercambiar los equipos de los bancos de trabajo.
- Prohibido sacar los equipos del laboratorio sin autorización.
- Ubicar los equipos y accesorios en el lugar dispuesto por el responsable del laboratorio, luego de terminar las prácticas.
- Uso adecuado de equipos.
- Uso obligatorio del mandil.

Actividades por desarrollar

1. Instalar y configurar el programa DATV – Express para que actúe como el transmisor de video empleando el dispositivo LimeSDR – Mini
 - Descargar el archivo DATV – Express_DVB_Transmitter_v1.25LP14 ZIP disponible en el siguiente enlace: <https://www.datv-express.com/CustomPage/Downloads>

Figura 25. Archivo DATV - Express_DVB_Transmitter que se debe descargar

DATV-Express

USER GUIDE for Windows pdf draft 14 for v1.25LP11 (DATV-Express / LimeSDR / LimeSDR-mini / PLUTO support)

USER GUIDE for Windows pdf (draft 10 French Translation for v1.23)

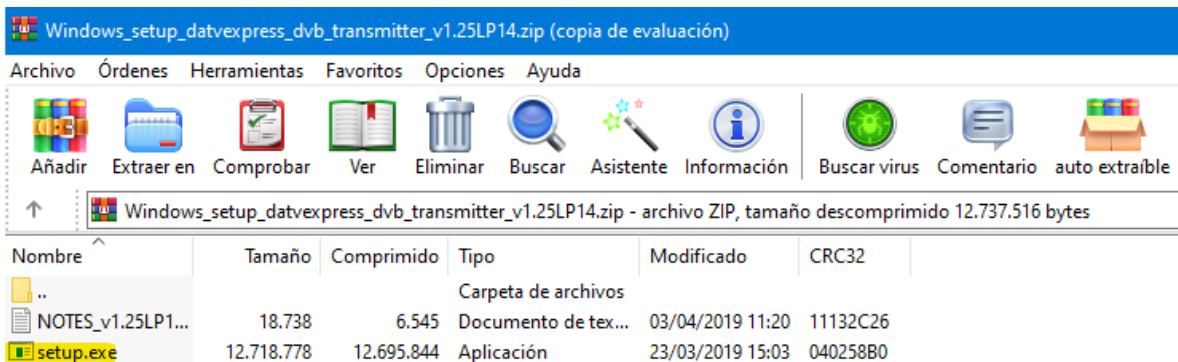
SOFTWARE INSTALL for Windows

DATV-Express_DVB_Transmitter_v1.25LP14 ZIP

Fuente: Elaboración propia

- Para instalar el Software Express DVB Transmitter del archivo descargado en el punto anterior, extraer y ejecutar el archivo “setup.exe” con permisos de administrador.

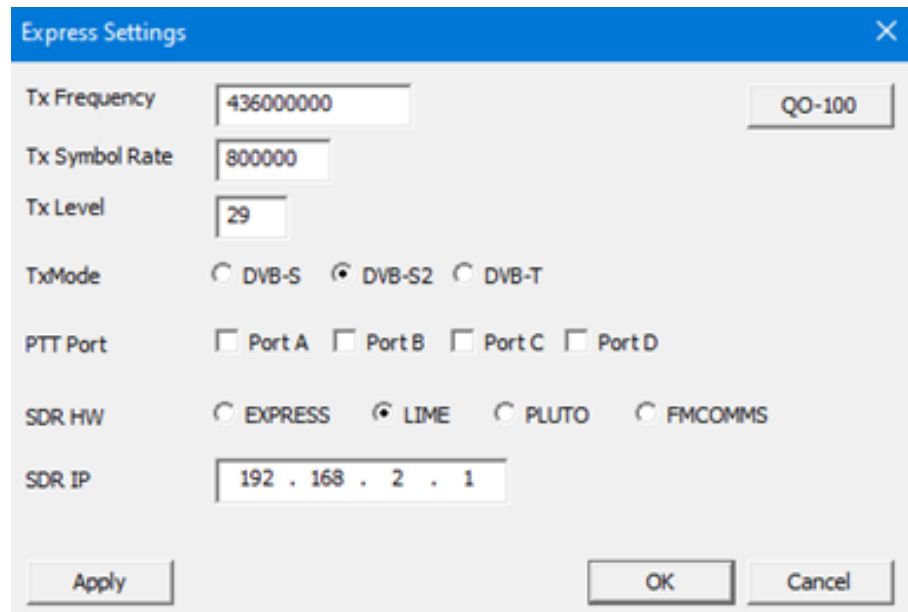
Figura 26. Forma de extraer y ejecutar el software Express DVB



Fuente: Elaboración propia

- Crear un acceso directo del programa Express DVB Transmitter en el escritorio.
- Abrir el programa DATV y seleccionar la opción “Modulator”, luego hacer clic en “Common” y configurar los siguientes parámetros:
 - Frecuencia de transmisión: 436 MHz,
 - Tasa de símbolo: 80 MHz,
 - Nivel de transmisión: 47,
 - Modo de transmisión: DVB - S
 - Tipo de dispositivo a utilizar: Lime

Figura 27. *Parámetros a configurar en la opción del Modulador del programa DATV*



Fuente: Elaboración propia

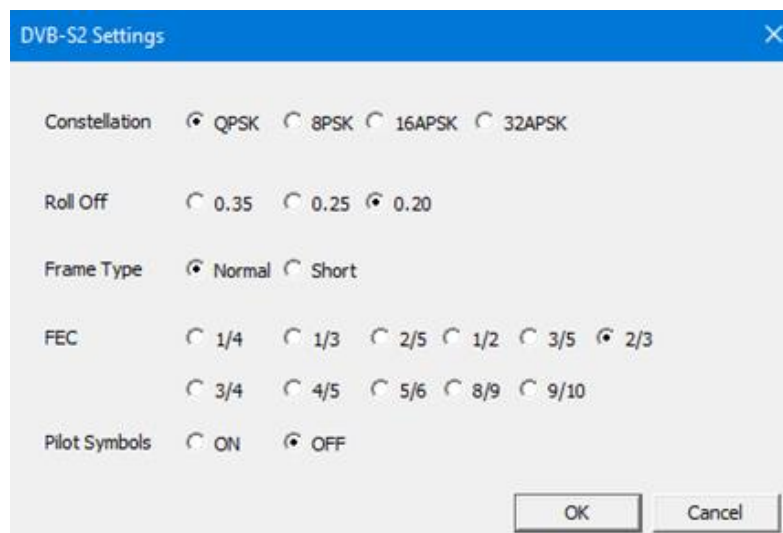
- Ahora configure el estándar de transmisión en la pestaña “Modulator” y seleccione “DVB-S2” y realice la siguiente configuración:

Constelación: QPSK

Roll Off: 0.20

FEC: 2/3

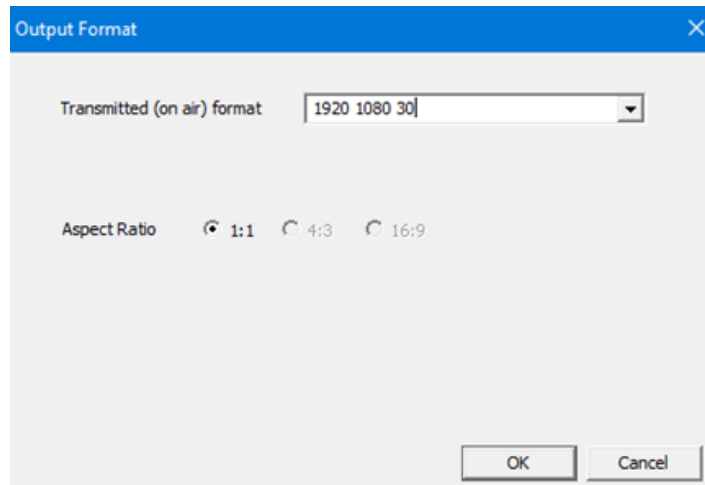
Figura 28. *Parámetros para configurar la opción de DVB-S2 del programa DATV*



Fuente: Elaboración propia

- En el menú desplegable de “Options” hacer clic en “On air format” esto permitirá al usuario seleccionar el formato del video transmitido. Puede seleccionar el mismo formato para la entrada del dispositivo de captura de video o seleccionar uno de los formatos que se muestran en la lista desplegable.

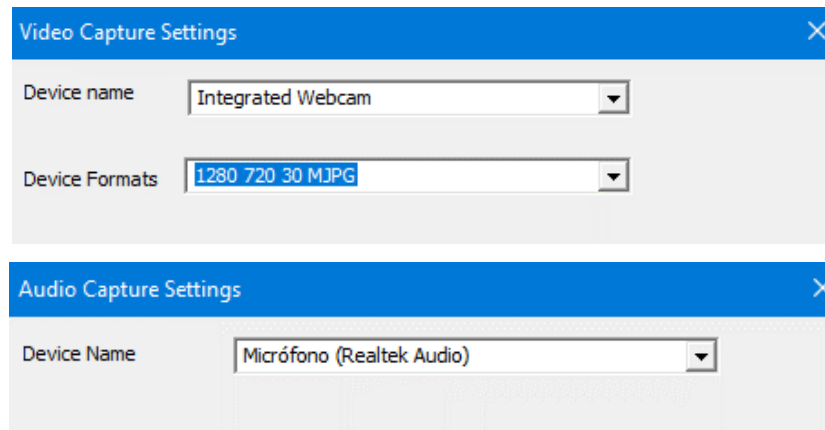
Figura 29. *Parámetros para configurar el formato de video que será transmitido*



Fuente: Elaboración propia

- En la ventana “Source” se encuentran las fuentes que debe utilizar el software para la transmisión de video. Se debe seleccionar el dispositivo que se utilizará para capturar el video y el de audio respectivamente. Nota: Para el desarrollo de la práctica se utilizó la cámara y el micrófono integrados del computador. Puede utilizar dispositivos externos en caso de que lo requiera.

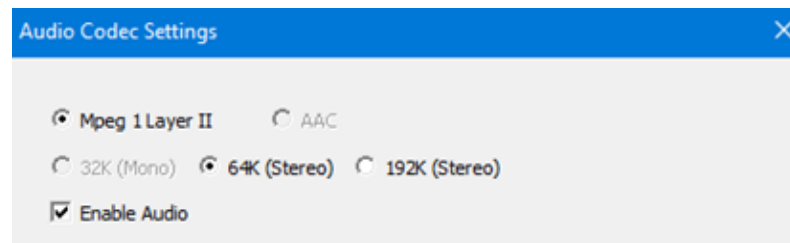
Figura 30. Selección del dispositivo que se utilizará para capturar la señal de video y audio respectivamente.

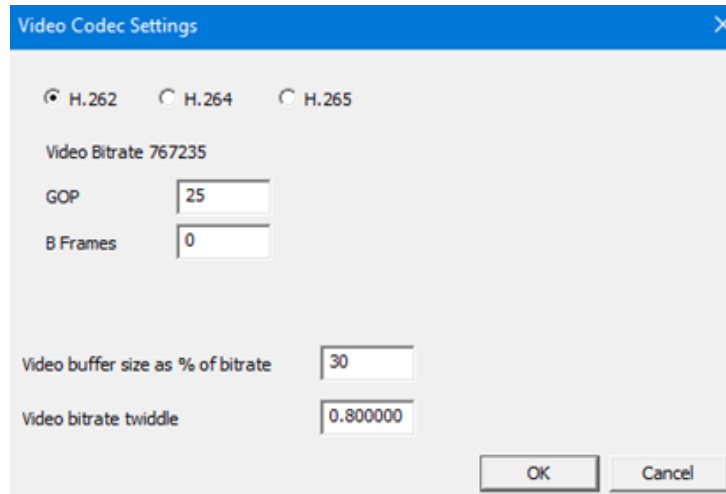


Fuente: Elaboración propia

- También se debe configurar los códecs de audio y video. Para ello, la pestaña “Codec” permite elegir un codificador de VIDEO y un codificador de AUDIO de tal forma se habilita la codificación o decodificación de los flujos de datos de video y audio

Figura 31. Configuración de los códec de audio y video para la respectiva codificación y decodificación





Fuente: Elaboración propia

2. Instalación del controlador LimeSDR – Mini

- Descargar el instalador según la arquitectura del procesador de su computador. La última versión del controlador FTDI se encuentra disponible en el sitio web: <https://ftdichip.com/drivers/d3xx-drivers/>

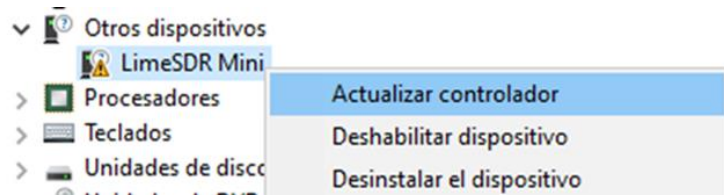
Figura 32. Selección del controlador FTDI tomando en cuenta la arquitectura del procesador del computador.

Operating System	Release Date	Processor Architecture		
		X86 (32-Bit)	X64 (64-Bit)	ARM
Windows*	26 February 2020	1.3.0.4	1.3.0.4	–

Fuente: Elaboración propia

- Conectar el dispositivo LimeSDR al computador, ir a “Administrador de dispositivos” y en la pestaña de “Otros dispositivos” hacer clic derecho en LimeSDR Mini y elegir la opción de “Actualizar controlador” y luego “Examinar mi PC en busca de controladores”

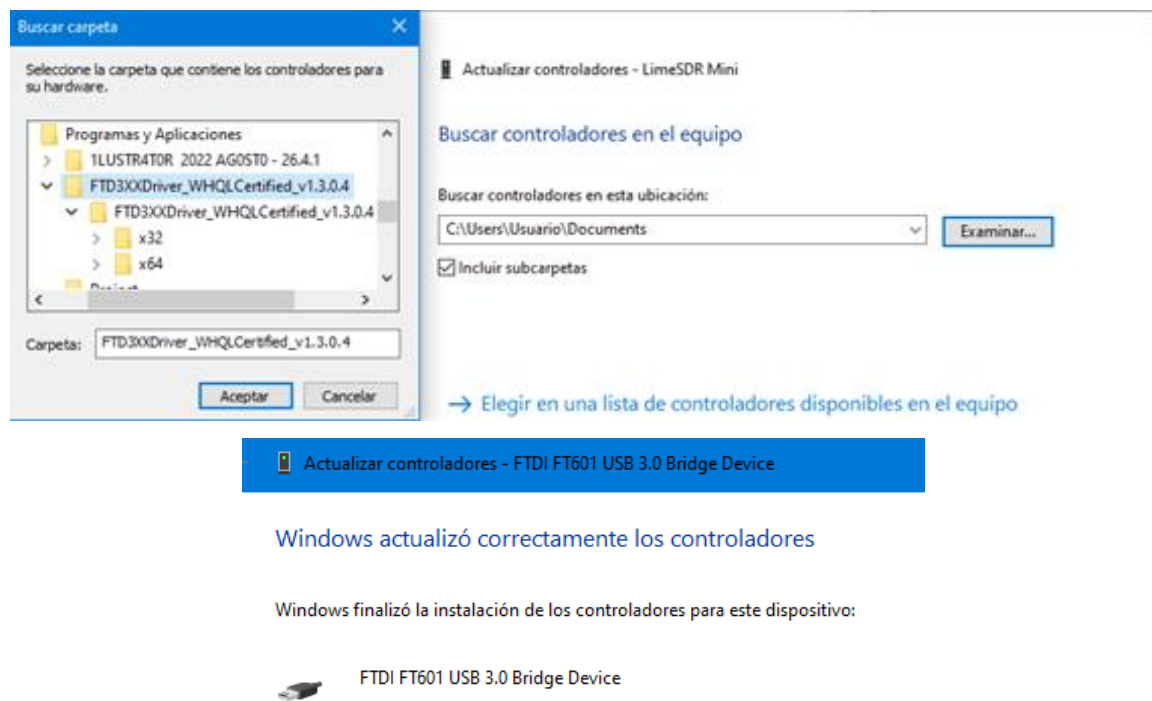
Figura 33. Forma de actualizar el controlador de LimeSDR en Windows



Fuente: Elaboración propia

- En el ícono de examinar, se debe buscar y seleccionar el archivo extraído en el paso 6.9. Finalmente, debe aparecer una ventana indicando que se ha actualizado correctamente los controladores para el dispositivo.

Figura 34. Actualización con éxito del controlador de LimeSDR a la versión descargada



Fuente: Elaboración propia

3. Instalación del programa SDRangel que actuará como receptor de la señal de video.
- En otro computador utilizar la máquina virtual Ubuntu 22.04.01 que hará de receptor. En ésta descargar y guardar en algún disco de almacenamiento local el programa “sdrangel-1864-master.tar.gz” el cual se encuentra disponible en el siguiente enlace: <https://github.com/f4exb/sdrangel/releases>

Figura 35. Selección del archivo a descargar del programa SDRangel

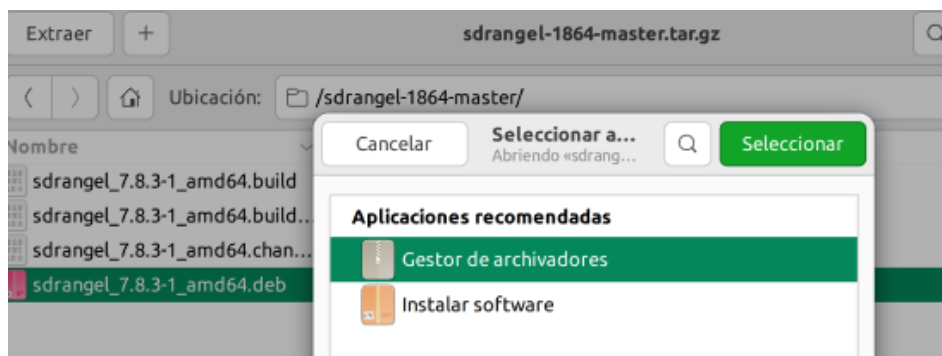
▼ Assets 5

 sdrangel-1864-master.tar.gz	31.3 MB	15 days ago
 sdrangel-7.8.3-macx64.dmg	179 MB	15 days ago
 sdrangel-7.8.3-win64.exe	141 MB	15 days ago
 Source code (zip)		15 days ago
 Source code (tar.gz)		15 days ago

Fuente: Elaboración propia

- Ubicar la carpeta donde se descargó y extraer el archivo. Hacer clic derecho sobre el archivo y abrir con “instalar software”.

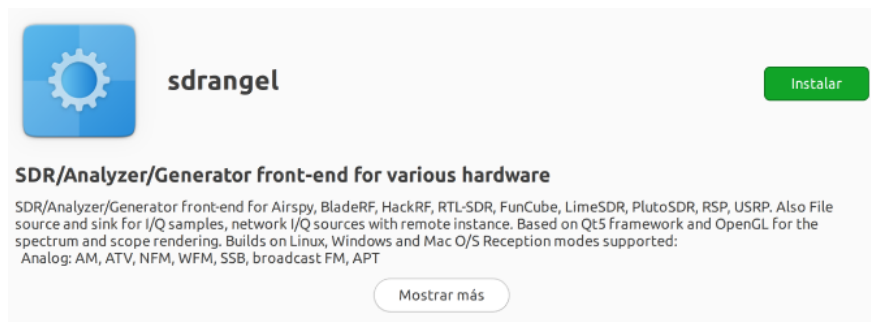
Figura 36. Forma de instalación del software SDRangel a través del gestor de archivadores



Fuente: Elaboración propia

- Aparecerá una imagen indicando la opción de instalar.

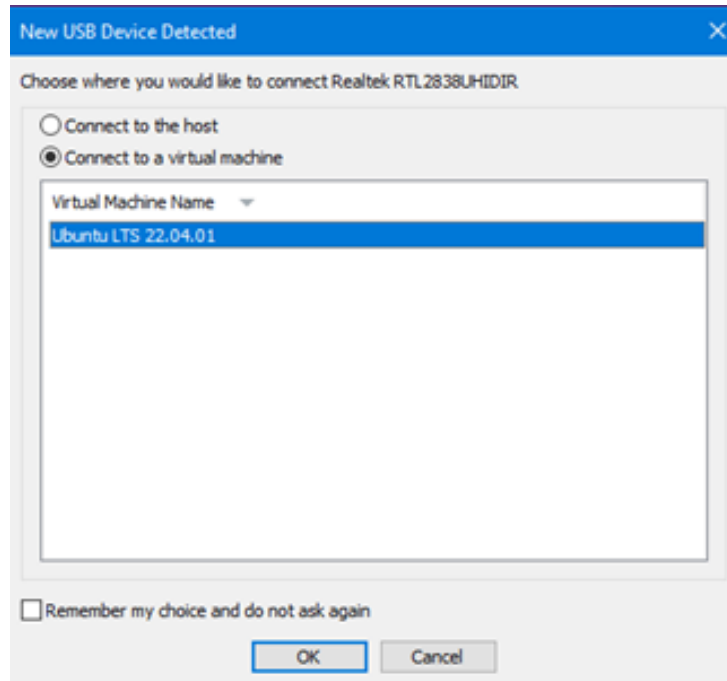
Figura 37. Ventana que indica la opción de instalar SDRangel en Ubuntu



Fuente: Elaboración propia

4. Transmisión y recepción de video con los dispositivos LimeSDR – Mini y RTL – SDR utilizando los programas DATV – Express y SDRangel.
- Conectar el dispositivo RTL-SDR en la máquina virtual.

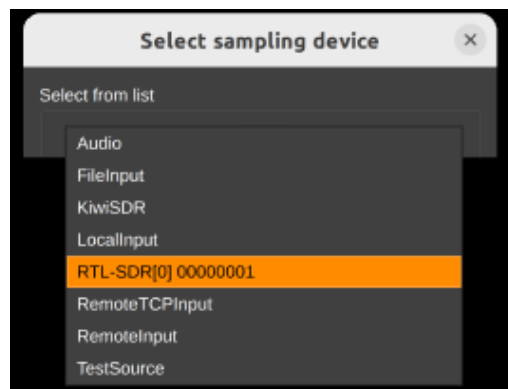
Figura 38. Estableciendo conexión del dispositivo RTL - SDR a la máquina virtual



Fuente: Elaboración propia

- Abrir el programa SDRangel y hacer clic en la pestaña “Add Rx device” y seleccionar el dispositivo RTL-SDR[0] 00000001 y luego hacer clic en “Ok”

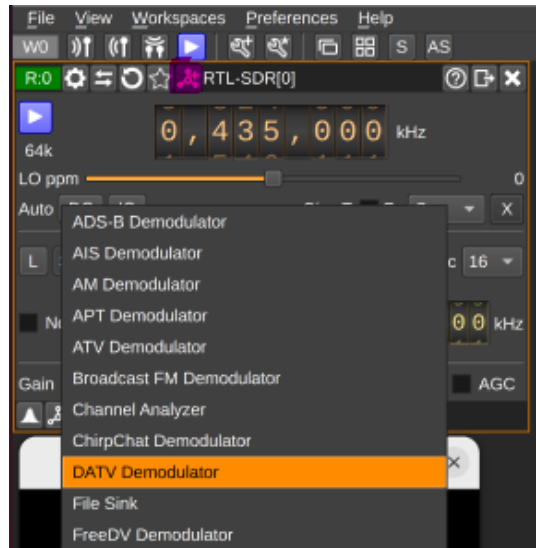
Figura 39. Selección del dispositivo SDR dentro del programa SDRangel



Fuente: Elaboración propia

- En la opción “Add channels” seleccionar “DATV Demodulador” y luego hacer clic en “Apply”

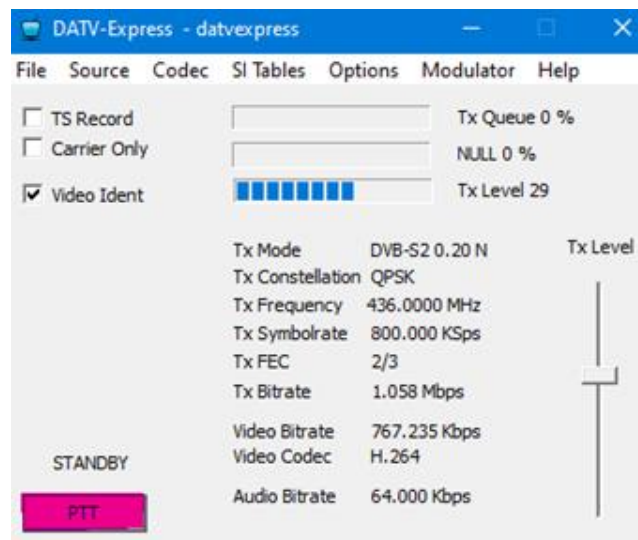
Figura 40. *Estableciendo el modo de operación que se desea que trabaje el dispositivo RTL-SDR*



Fuente: Elaboración propia

- Para transmitir señales de video se debe ir al programa DATV-Express, y presionar el botón PTT.

Figura 41. Configuración lista para transmitir señales de video mediante el programa DATV-Express presionando el botón PTT



Fuente: Elaboración propia

- Receptar el video transmitido en el programa SDRangel

Marco teórico (a elaborar por el estudiante)

- ¿Qué entiende por DATV?

Sus siglas significan Digital Amateur Television, es decir, Television Digital de radioaficionados. Se basa en la transmisión de video y audio con calidad de transmisión en la amplia gama de frecuencias de ondas de radio asignadas para el uso de radioaficionados (Ham). La DATV se utiliza para experimentos no comerciales, placer y eventos de servicio público. (American Radio Relay League., 2018)

- Bandas de frecuencia usadas en DATV

Utiliza las bandas de Frecuencias ultra elevadas (UHF) y la banda de Frecuencias super elevadas (SHF).

Frecuencias ultra elevadas (UHF): Son de propagación directa a través de la baja atmósfera. La distancia cubierta es reducida y se emplea para señales de televisión, radar, sistema de localización y dirección aeronáutica, servicios de tráfico comercial mediante microondas y en comunicación de radioaficionados. Comprenden desde 300 a 3.000 MHz.

Frecuencia super elevadas (SHF): Se propagan en forma directa a través de la baja atmósfera. Su uso está indicado para servicios aeronáuticos y tráfico comercial por microondas. Abarcan desde 3.000 a 30.000 MHz (3 a 30 Giga Hertz). (Bolívar, 2021)

- Estándares de transmisión digital

ATSC (Advanced Television Systems Committee): utiliza una modulación patentada desarrollada por Zenith llamada 8-VSB; como su nombre lo indica, es una técnica de banda lateral vestigial. Esencialmente, 8-VSB es una modulación de amplitud en cuadratura de ocho vías. Este sistema se eligió específicamente para proporcionar la máxima compatibilidad espectral entre la televisión analógica existente y las nuevas estaciones digitales en el sistema de asignaciones de televisión de los Estados Unidos. La modulación 8-VSB admite un flujo de datos digitales de aproximadamente 19,39 Mbit/s, suficiente para un flujo de video de alta definición o varios servicios de definición estándar.

DVB (Digital Video Broadcasting): Desarrollado en Europa, incluye varias versiones dependiendo del medio de transmisión:

DVB-C (cable), adoptado en 1994 es el estándar del consorcio europeo DVB para la transmisión de televisión digital por cable. Este sistema transmite un flujo de audio / video digital de la familia MPEG-2, utilizando una modulación QAM con codificación de canal.

DVB-S (satélite), es el estándar original de modulación y codificación de error de transmisión de video digital para televisión por satélite y se remonta a 1995. Se utiliza a través de satélites que sirven a todos los continentes del mundo, incluida América del Norte. También se han desarrollado DVB-S2, para transmisión más eficiente por satélite que su predecesor.

DVB-T (terrestre), adoptado en 1997 ha sido adoptado en más de cuarenta países en el mundo. Utiliza multiplexación por división de frecuencia ortogonal codificada (COFDM), que utiliza hasta 8000 portadoras independientes, cada una de las cuales transmite datos a una velocidad comparativamente baja. Este sistema fue diseñado para proporcionar una inmunidad superior a la interferencia de múltiples rutas y tiene una variedad de variantes del sistema que permiten velocidades de datos desde 4 MBit / s hasta 24 MBit / s.

ISDB (Integrated Services Digital Broadcasting): Desarrollado en Japón y similar, en algunos aspectos a DVB. Contempla también versiones para satélite (ISDB-S), cable (ISDB-C) y terrestre (ISDB-T). Las transmisiones en este estándar comenzaron en diciembre de 2003 y tiene ciertas características que lo hacen más flexible que DVB. Se divide en 13 subcanales, doce se utilizan para TV, mientras que el último sirve como banda de protección o para el servicio 1seg (ISDB-H).

DTMB (Digital Terrestrial/Television Multimedia Broadcasting): Desarrollado en la República Popular China, aprobado en agosto de 2007, con características diferentes a los otros estándares tanto en el sistema de modulación como de codificación de canal y en el que se funden dos estándares previos también desarrollados en China, ADTB-T, similar al ATSC y desarrollado en la Universidad de Jiaotong de Shanghai y el DMB-T en la Universidad Tsinghua de Beijing.

- Códex usados en DATV

Para D-ATV es el codificador MPEG. MPEG significa Motion Picture Experts Group, un consorcio que desarrolló una serie de estándares para la compresión (digital) de imágenes en movimiento y sonido. Al igual que el estándar JPEG para imágenes fijas, la compresión MPEG puede reducir significativamente la cantidad de datos necesarios para representar una imagen en movimiento. Existen algunos estándares MPEG, incluidos los de audio. Para DATV, se eligió la codificación MPEG-2.

Pero desde mediados del 2020 se utiliza habitualmente en DATV el H.264, que permite mantener la calidad del MPEG-2 usando menores velocidades de transmisión y por tanto necesitando menos Ancho de Banda. Los receptores comerciales DVB- S2 también admiten H.264, pero no están diseñados para recibir velocidades de transmisión inferiores a 1 Ms/s. (Rotolo & N2IRZ, 2004)

- Modulaciones para DATV

La modulación mayoritariamente usada por los radioaficionados para trabajar DATV es la misma que utilizan las emisiones comerciales de TV Digital por Satélite: modulación QPSK compatible con el estándar DVB-S. Un aspecto importante de la DATV es que permite ocupar un ancho de banda notablemente inferior al necesario para una emisión ATV analógica.

Los Anchos de Banda típicos en ATV y en DATV son:

- ATV con modulación AM ocupa 11 MHz de ancho.
- ATV con modulación AM-VSB ocupa 7 MHz de ancho.
- ATV con modulación FM ocupa unos 20 MHz de ancho.
- DATV con modulación QPSK compatible con antiguos Receptores TV Satélite Digitales DVB-S ocupa 4 MHz de ancho (SR = 2 Ms/s)
- DATV con modulación QPSK compatible con actuales Receptores TV Satélite Digitales DVB-S2 ocupa 2 MHz de ancho (SR = 1 Ms/s) (EA7GLU, 2019)

Resultados (a elaborar por el estudiante)

Discusión (a elaborar por el estudiante)

Conclusiones (a elaborar por el estudiante)

Recomendaciones (a elaborar por el estudiante)

Preguntas de Control (a elaborar por el estudiante)

a) ¿Qué es y que significan las siglas FEC?

Conocido como Forward Error Correction por sus siglas en inglés FEC, en español el corrector de errores hacia adelante, es una técnica por la cual se añaden datos redundantes siguiendo unos algoritmos concretos que nos permiten, por un lado, detectar si una determinada transmisión contiene algún error y, por otro, reconstruir la señal original a partir de una señal con ciertos errores.

b) ¿De qué factores depende el bitrate?

El bitrate es la cantidad de bits por segundo que se transmite en un determinado sistema de telecomunicaciones. El bitrate depende de diversos factores, desde el tiempo de símbolo, la constelación en uso, la presencia o no de datos redundantes, etc. En DVB-T, con los factores más importantes definidos de antemano por el estándar. El bitrate depende de tres factores: la modulación, la tasa de código del código convolucional y la duración del tiempo de guarda

c) ¿Es bueno tener un tiempo de guarda mayor?

No, aunque el intervalo de guarda tiene efectos muy beneficiosos, no todo son ventajas: un tiempo de guarda mayor reduce la tasa de símbolo, afectando directamente al bitrate. Tener en

cuenta que el tiempo de guarda se utiliza para evitar interferencias entre las transmisiones, debidas entre otros factores al efecto multicamino y al tiempo de propagación. Los tiempos de guarda suelen expresarse como una fracción del tiempo de símbolo (por ejemplo, 1/32, 1/16, 1/8 ó 1/4 en DVB-T).

Bibliografía (a elaborar por el estudiante)

5.2.4. PRÁCTICA # 4

ASIGNATURA: COMUNICACIONES DIGITALES

RESULTADO DE APRENDIZAJE DE LA PRÁCTICA: (VER SÍLABO)

TIEMPO PLANIFICADO EN EL SÍLABO: 2 HORAS

TIEMPO DE LA PRÁCTICA POR GRUPO: 2 HORAS

NÚMERO DE ESTUDIANTES POR GRUPO: 2 ESTUDIANTES

NÚMERO DE ESTACIONES DE TRABAJO: 10

Tema: Transmisión de un archivo de texto con el dispositivo Adalm Pluto aplicando la técnica de corrección de errores Convolutional.

Objetivos:

- Realizar el estudio de la técnica Convolutional para la detección de errores al momento de enviar información en tiempo real.
- Utilizar el dispositivo Adalm Pluto para que realice la simulación de transmitir un archivo de texto.
- Analizar los tipos de mapeos utilizados en esta práctica.

<i>Materiales y Reactivos</i>	<i>Equipo y herramientas</i>
<ul style="list-style-type: none"> • 2 antenas 	<ul style="list-style-type: none"> • 1 módulo Adalm PlutoSDR • Programa de Matlab • 1 computador

Instrucciones

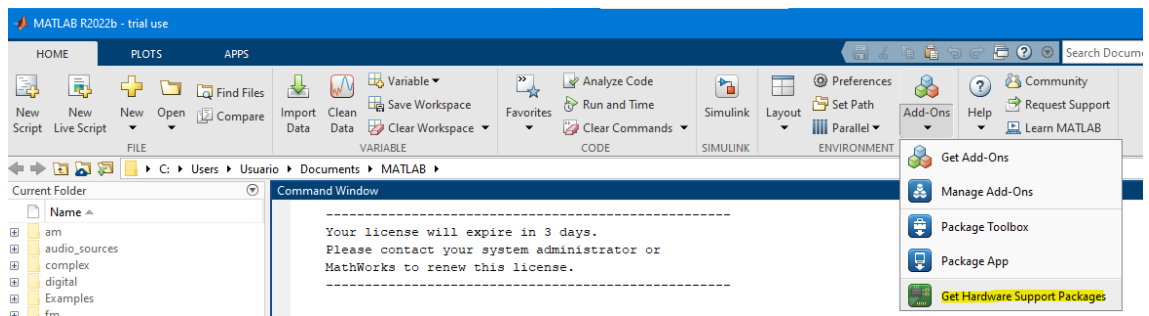
- Colocar las mochilas en los casilleros.

- Prohibido consumo de alimentos.
- Prohibido equipo de diversión, celulares etc.
- Prohibido jugar.
- Prohibido mover o intercambiar los equipos de los bancos de trabajo.
- Prohibido sacar los equipos del laboratorio sin autorización.
- Ubicar los equipos y accesorios en el lugar dispuesto por el responsable del laboratorio, luego de terminar las prácticas.
- Uso adecuado de equipos.
- Uso obligatorio del mandil.

Actividades por desarrollar

1. Instalar el paquete “Communications Toolbox Support Package for Analog Devices ADALM-Pluto Radio”.
 - Abrir Matlab y en la barra de herramientas en la pestaña “HOME” buscar “Add Ons” y seleccionar “Get Hardware Support Packages”.

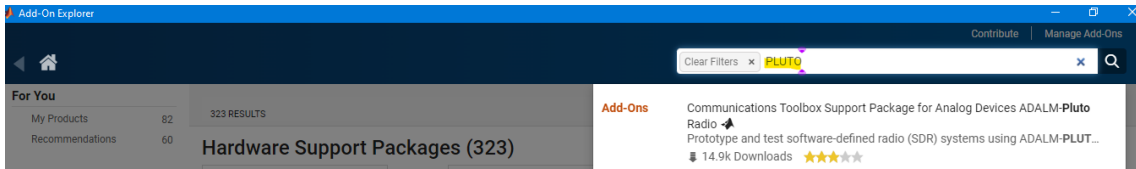
Figura 42. *Forma de seleccionar el paquete apropiado para el dispositivo Adalm Pluto*



Fuente: Elaboración propia

- A continuación, aparece una ventana “Add On Explorer” y en la barra de búsqueda encontrar el paquete para el dispositivo Adalm Pluto y proceder a instalar.

Figura 43. Ventana de Add On Explorer para buscar el paquete específico para el hardware SDR

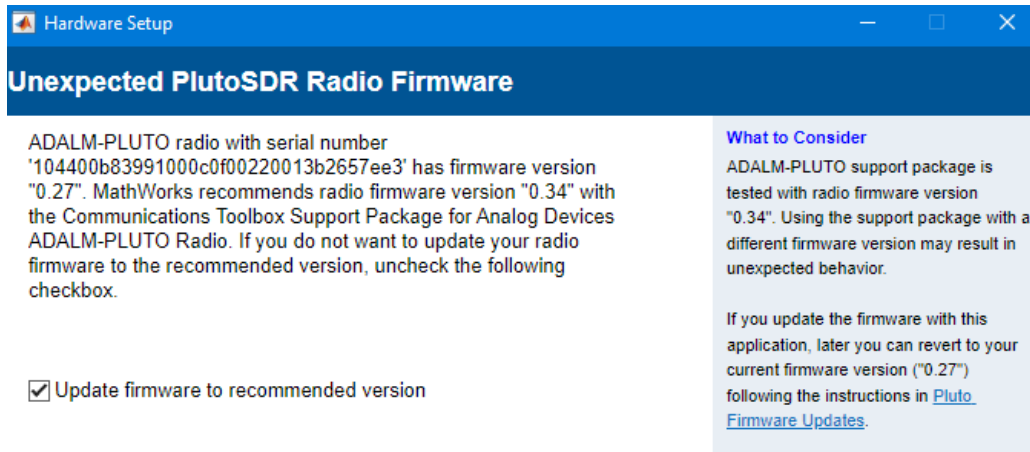


Fuente: Elaboración propia

2. Configurar el dispositivo Adalm Pluto

- Conectar el dispositivo mediante interfaz USB 2.0 para la conexión con SDR. Como primer paso es actualizar el firmware, los drivers y algunas librerías para su buen funcionamiento.

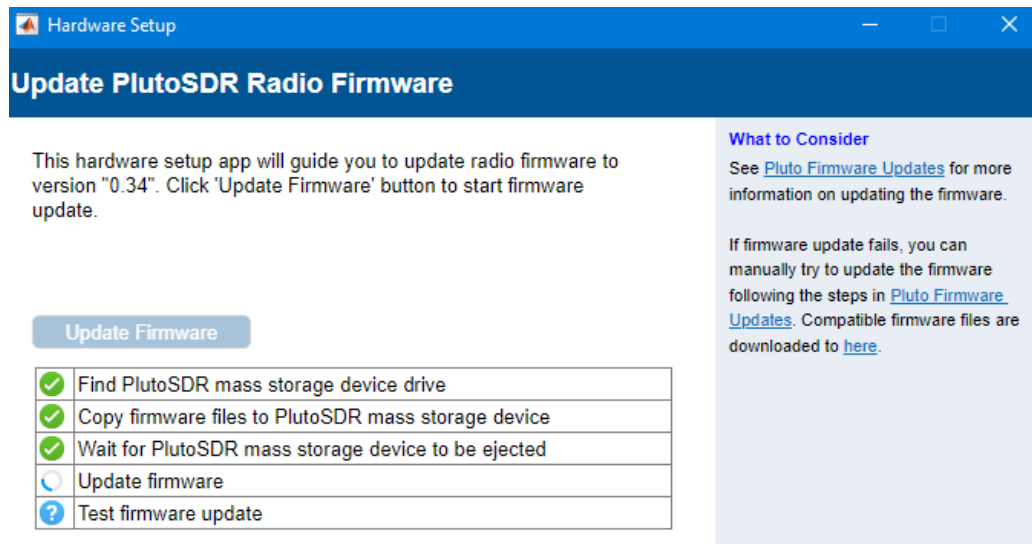
Figura 44. Actualización del firmware a la versión recomendada 0.34 del dispositivo Adalm PlutoSDR



Fuente: Elaboración propia

- Para seguir con el procedimiento dar clic en “Next”

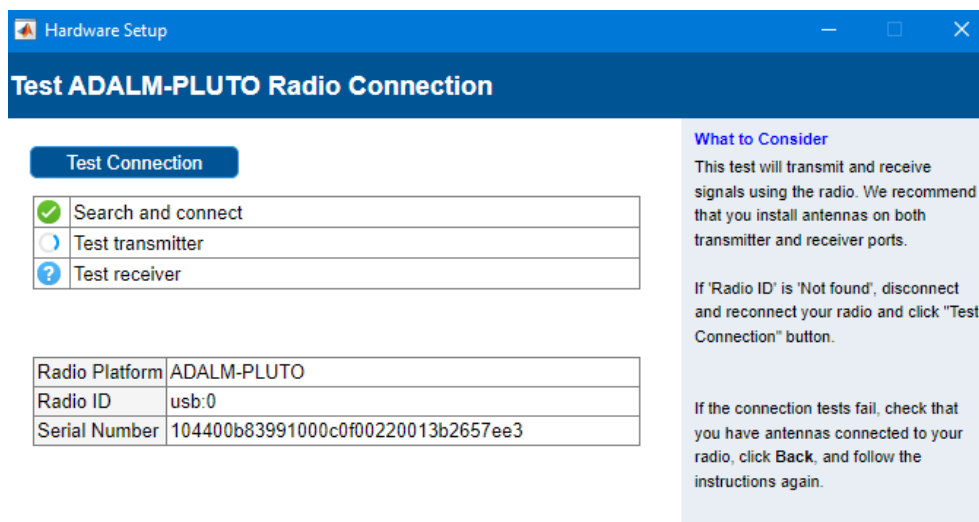
Figura 45. Indicación del procedimiento de la actualización del Firmware



Fuente: Elaboración propia

- Finalmente, dar clic en “Test Connection” para verificar que la configuración realizada se haya completado con éxito.

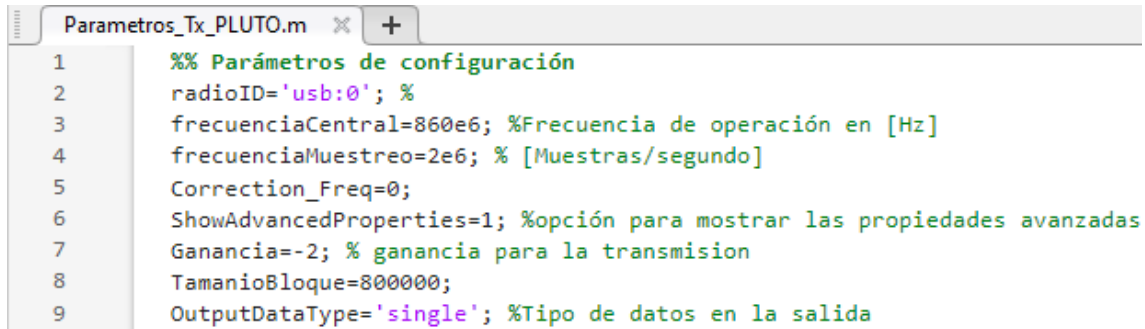
Figura 46. Ventana de información sobre el dispositivo Adalm Pluto



Fuente: Elaboración propia

3. Para esta práctica se realiza un script con las siguientes líneas de código donde se configura el dispositivo para que actúe como transmisor, luego se guarda con el nombre de “Parametros_Tx_PLUTO.m”

Figura 47. Parámetros de configuración para la transmisión de texto

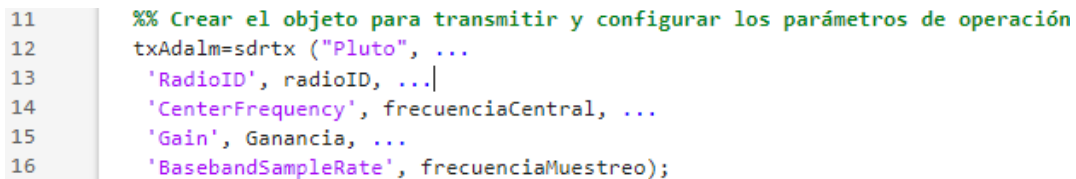


```
Parametros_Tx_PLUTO.m x +
1      %% Parámetros de configuración
2      radioID='usb:0'; %
3      frecuenciaCentral=860e6; %Frecuencia de operación en [Hz]
4      frecuenciaMuestreo=2e6; % [Muestras/segundo]
5      Correction_Freq=0;
6      ShowAdvancedProperties=1; %opción para mostrar las propiedades avanzadas
7      Ganancia=-2; % ganancia para la transmisión
8      TamanoBloque=800000;
9      OutputDataType='single'; %Tipo de datos en la salida
```

Fuente: Elaboración propia

4. En el mismo script se crea el objeto con las variables que se crearon para configurar los parámetros de operación del dispositivo.

Figura 48. Creación del objeto para interactuar con el hardware Adalm Pluto Radio

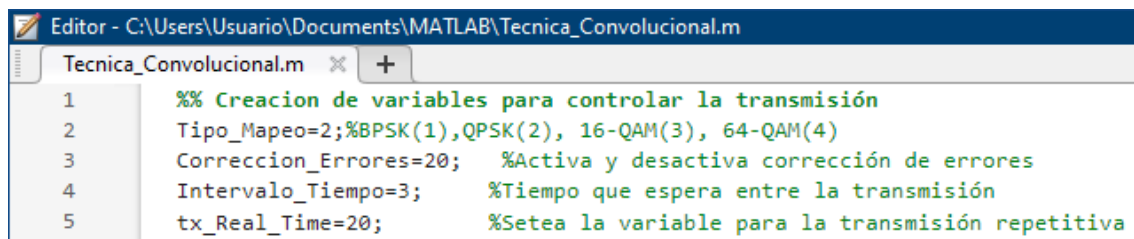


```
11     %% Crear el objeto para transmitir y configurar los parámetros de operación
12     txAdalm=sdrtx ("Pluto", ...
13     'RadioID', radioID, ...|
14     'CenterFrequency', frecuenciaCentral, ...
15     'Gain', Ganancia, ...
16     'BasebandSampleRate', frecuenciaMuestreo);
```

Fuente: Elaboración propia

5. En otro script guardar las siguientes líneas de código bajo el nombre de “Tecnica_Convolucional.m”, el cual servirá para que el dispositivo Adalm Pluto actúe como transmisor. Nota. Todas las líneas de código siguiente se guardarán en el mismo script.
- Se empieza por crear las variables de control, por ejemplo: el tipo de mapeo: BPSK, QPSK, 16-QAM, 64-QAM, el intervalo de tiempo que se espera entre la transmisión y otras como se muestra a continuación.

Figura 49. Variables para controlar el proceso de transmisión

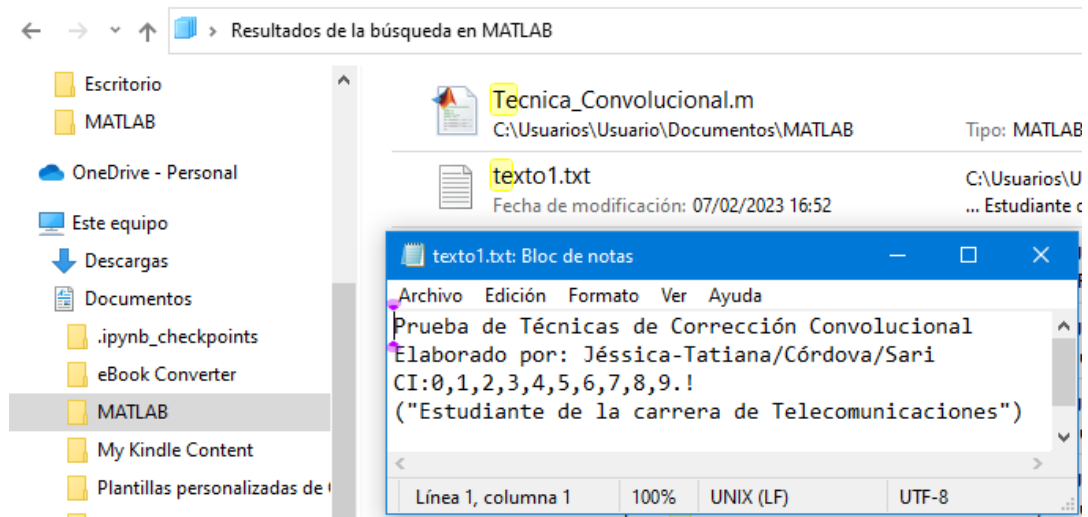


```
Editor - C:\Users\Usuario\Documents\MATLAB\Tecnica_Convolucional.m
Tecnica_Convolucional.m x +
1      %% Creacion de variables para controlar la transmisión
2      Tipo_Mapeo=2;%BPSK(1),QPSK(2), 16-QAM(3), 64-QAM(4)
3      Correccion_Errores=20; %Activa y desactiva corrección de errores
4      Intervalo_Tiempo=3; %Tiempo que espera entre la transmisión
5      tx_Real_Time=20; %Setea la variable para la transmisión repetitiva
```

Fuente: Elaboración propia

- El proceso siguiente es seleccionar y cargar el archivo de texto que se desea procesar. El archivo de texto a transmitir fue creado en un bloc de notas y debe estar guardado en la carpeta donde se ejecutan todos los scripts, de preferencia en la carpeta de Matlab que se crea automáticamente al instalar el programa.

Figura 50. *Ubicación del archivo de texto a transmitir*



Fuente: Elaboración propia

Figura 51. *Selección del archivo que será procesado*

```
7      %% Cargar el archivo de texto para que sea procesado.
8
9      Ingreso_Archivo=uigetfile("*.txt","Seleccionar archivo a transmitir");
10     if Ingreso_Archivo==0 %No se ha seleccionado ningún archivo
11         disp("Archivo no seleccionado");
12     else
13         fID= fopen(Ingreso_Archivo); %Abre el archivo
14         datosAbiertos= fread(fID,"*char");
15         fclose(fID);
```

Fuente: Elaboración propia

- Para continuar se procede a la serialización de datos que hace referencia a representar cada caracter en binario. Por ello, se utilizan dos ciclos “for” para representar los elementos de cada fila y otro ciclo “for” para cada columna. La variable “aux” se encarga de almacenar los datos en cada posición que se encuentre.

Figura 52. Proceso de serialización de los datos representando cada caracter en binario

```
16      %% Serializar datos
17
18      textoBinario=dec2bin(datosAbiertos,8);
19      aux=1;%Inicializar variable auxiliar
20      for i=1:length(textoBinario)
21          for j=1:8
22              Texto_Bits_Serial(aux)=textoBinario(i,j);%Tipo char
23              aux=aux+1;
24          end
25      end
26      for aux=1:length(Texto_Bits_Serial)
27          datos_Serializados(aux)=str2double(Texto_Bits_Serial(aux));
28      end
29      disp("Datos Serializados")
30      datosBina=datos_Serializados; %variable con los datos serializados
```

Fuente: Elaboración propia

- Con la variable que guarda los datos serializados “datosBina” se procede a seguir el proceso que hace el codificador de la técnica convolucional para la corrección de errores a nivel de bit. Nota. Se crean las opciones de mapeo para BPSK, QPSK, 16-QAM, 64-QAM

Figura 53. Proceso del codificador de la técnica convolucional

```
33      %% Codificador Convolucional
34
35      if Correccion_Errores==10      %Sin corrección de errores
36          datosBinario=datosBina;
37      elseif Correccion_Errores==20 %Realiza la corrección de errores
38          NPdatos=40; %Numero de portadorada de datos por símbolo OFDM
39          if (Tipo_Mapeo==1)      %Orden de modulación
40              M=2;
41          elseif(Tipo_Mapeo==2)  %Orden de modulación
42              M=4;
43          elseif(Tipo_Mapeo==3)  %Orden de modulación
44              M=16;
45          elseif(Tipo_Mapeo==4)  %Orden de modulación
46              M=64;
47          end
48          Nbps_M=log2(M);
```

Fuente: Elaboración propia

- Creación de las variables que se utilizan para el codificador

Figura 54. Configuración de las variables utilizadas en el codificador convolucional

```
50      %% Parámetros del codificador convolucional
51
52      nc=2; %longitud por cada ingreso de k bits
53      kc=1; %bits de entrada
54      mc=3; %Tamaño de registro de la memoria
55      g1=5; %Polinomio generador x2+1=5
56      g2=7; %Polinomio generador x2+x+1=7
57      Rcc=kc/nc; %Tasa de codificación
58      Nbits=Nbps M*NPdatos*Rcc;
```

Fuente: Elaboración propia

- Las siguientes líneas de código corresponden para rellenar el vector de entrada y así obtener un vector que sea múltiplo de Nbits. También, se creará la estructura de Trellis, en él se utilizan las variables de los polinomios generadores y será necesaria utilizar la función “poly2trellis”

Figura 55. Creación del diagrama de Trellis usando la opción polytrellis de Matlab

```
60      %% Cálculo para rellenar el vector de entrada
61
62      NumSimb=ceil(length(msg)/Nbits);
63      longitudPadd=Nbits*NumSimb-length(msg);
64      vector_padd=zeros(1,longitudPadd);
65      datosConPadd=[msg vector_padd];
66
67      %% Creación de la estructura de Trellis
68
69      P_Trellis = poly2trellis(mc,[g1 g2]);
70      datosCodConv=[];
71      for i=1:NumSimb %Recorre los símbolos OFDM
72          auxDatosCc=datosConPadd(1,(Nbits*(i-1)+1):Nbits*i);
73          CodConv=convenc(auxDatosCc,P_Trellis);
74          datosCodConv=[datosCodConv CodConv]; %Datos de salida
75      end
76      datosBinario=datosCodConv;
77      disp("Codificación Convolucional Agregada")
78
79      end
```

Fuente: Elaboración propia

- El paso siguiente representa los bits serializados a un esquema de modulación digital que pueden ser: BPSK, QPSK, 16-QAM y 64-QAM. El vector de entrada es “datosBinario” y el de salida es “datosMapeados”.

Figura 56. Creación de los diferentes esquemas de modulación digital

```

81     %% Esquema de modulación digital
82
83     if(Tipo_Mapeo==1)
84         datosColumna=datosBinario.';
85         for i=1:length(datosColumna)
86             if(datosColumna(i)==1)
87                 datosMapeados(i,1)=complex(0.707,0.707);
88             else
89                 datosMapeados(i,1)=complex(-0.707,-0.707);
90             end
91         end
92         disp("Técnica de correccion de errores:Convolucional")
93         disp("Tipo de mapeo:8PSK")
94     elseif(Tipo_Mapeo==2)
95         M=4;
96         Nb=log2(M);
97         long_padd0=ceil(length(datosBinario)/Nb)*Nb-length(datosBinario);
98         paddQ=zeros(1,long_padd0);
99         datosBina=[datosBinario paddQ];
100        dataMatriz=reshape(datosBina, Nb, []).';
101        dataSimb=bi2de(dataMatriz, "left-msb");
102        datosMapeados=(1/sqrt(2))*pskmod(dataSimb,M,pi/4, "gray");
103
104     elseif(Tipo_Mapeo==3)
105         M=16;
106         Nb=log2(M);
107         long_padd0=ceil(length(datosBinario)/Nb)*Nb-length(datosBinario);
108         paddQ=zeros(1,long_padd0);
109         datosBina=[datosBinario paddQ];
110         dataMatriz=reshape(datosBina, Nb, []).';
111         dataSimb=bi2de(dataMatriz, "left-msb");
112         datosMapeados=(1/sqrt(10))*qammod(dataSimb,M);
113         disp("Técnica de corrección de errores:Convolucional")
114         disp("Tipo de mapeo: 16-QAM")
115     elseif(Tipo_Mapeo==4)
116         M=64;
117         Nb=log2(M);
118         long_padd0=ceil(length(datosBinario)/Nb)*Nb-length(datosBinario);
119         paddQ=zeros(1,long_padd0);
120         datosBina=[datosBinario paddQ];
121         dataMatriz=reshape(datosBina, Nb, []).';
122         dataSimb=bi2de(dataMatriz, "left-msb");
123         datosMapeados=(1/sqrt(42))*qammod(dataSimb,M);
124         disp("Técnica de corrección de errores: Convolucional")
125         disp("Tipo de mapeo: 64-QAM")
126     end

```

- El paso siguiente es crear cada símbolo OFDM con sus respectivas portadoras. En el caso de las portadoras “piloto” actuarán como marcadores de sincronismo, para facilitar el proyecto de estimadores de canal.

Figura 57. Creación de cada símbolo OFDM con portadoras de datos

```
127      %% Construcción de cada símbolo OFDM con sus portadoras de datos
128      NPdatos=40; %Número de portadoras de datos
129      longDatos=length(datosMapeados);
130      nSimbOFDM=ceil(longDatos/NPdatos);
131      longPadd=nSimbOFDM*NPdatos-longDatos;
132      datosEntradaPadd=datosMapeados;
133      if(longPadd>0)
134          datosEntradaPadd(end+1:end+longPadd)=datosMapeados(1:longPadd);
135      end
136      numeroSimbolos=1;
137      longDatosPadd=length(datosEntradaPadd);
138      for i=1:NPdatos:longDatosPadd
139          datosAux=datosEntradaPadd(i:i+NPdatos-1);
140          DC=0; %Valor para la subportadora DC
141          p1=1; %Subportadoras piloto
142          p2=1;
143          p3=1;
144          p4=1;
145          simbOFDM=zeros(64,1);
146          simbOFDM(1)=DC; %Portadora DC
147          simbOFDM(2:7)=datosAux(21:26);
148          simbOFDM(8)=p1; %Piloto
149          simbOFDM(9:21)=datosAux(27:39);
150          simbOFDM(22)=p2;
151          simbOFDM(23)=datosAux(40);
152          simbOFDM(43)=datosAux(1); %24 - 42 son de guarda
153          simbOFDM(44)=p3;
154          simbOFDM(45:57)=datosAux(2:14);
155          simbOFDM(58)=p4;
156          simbOFDM(59:64)=datosAux(15:20);
157          simbolosOFDMCreados(1:64,numeroSimbolos)=simbOFDM;
158          numeroSimbolos=numeroSimbolos+1;
159      end
```

Fuente: Elaboración propia

- Luego, los símbolos OFDM pasarán del dominio de la frecuencia al dominio del tiempo y también se realiza la inserción del prefijo cíclico o intervalo de guarda entre los símbolos OFDM para prevenir la interferencia intersimbólica (ISI) y disminuir el atraso de difusión efecto producido por el canal.

Figura 58. Inserción del prefijo cíclico para prevenir la interferencia intersimbólica.

```
160     numeroSimbolos=numeroSimbolos-1;
161     disp("Símbolos OFDM creados");
162
163     %% Símbolos OFDM pasan del dominio de la frecuencia al dominio del tiempo
164     simbolosOFDM=ifft(simbolosOFDMCreados);
165     disp("IFFT implementada");
166     datosEnviar=[];
167     for auxPC=1:numeroSimbolos
168         simbolosOFDMMpc(1:80,auxPC)=[simbolosOFDM(49:64,auxPC);simbolosOFDM(:,auxPC)];
169         datosEnviar=[datosEnviar;simbolosOFDMMpc(:,auxPC)];
170     end
171     disp("Prefijo cíclico (PC) añadido");
```

Fuente: Elaboración propia

- Así mismo, se define el uso de un preámbulo para la sincronización y para configurar el receptor, ajustándolo para cualquier pérdida específica en frecuencia que ocurrió en el canal. Se inicia por definir parte de la sección corta y larga en el dominio de la frecuencia para luego obtener el preámbulo en el dominio del tiempo.

Figura 59. Preámbulo 802.11 con sus respectivas secciones

```
173 %% Añadir preámbulo por el estándar 802.11a
174
175 parteCorta64=[0;0;0;0;-1.4720-1.4720i;0;0;0
176 -1.4720-1.4720i;0;0;0;1.4720+1.4720i;0;0;0
177 1.4720+1.4720i;0;0;0;1.470+1.470i;0;0;0
178 1.4720+1.4720i;0;0;0;0;0;0;0;0;0;0;0;0;0;0
179 1.4720+1.4720i;0;0;0;-1.4720-1.4720i;0;0;0
180 1.4720+1.4720i;0;0;0;-1.4720-1.4720i;0;0;0
181 -1.4720-1.4720i;0;0;0;1.4720+1.4720i;0;0;0];
182 parteLarga=[...
183 0;1;-1;-1;1;1;-1;1;-1;1;-1;-1;-1;-1;-1;1
184 1;-1;-1;1;-1;1;-1;1;1;1;0;0;0;0;0
185 0;0;0;0;0;1;1;-1;-1;1;1;-1;1;-1;1
186 1;1;1;1;-1;-1;1;1;-1;1;-1;1;1;1];
187 parteCorta64Time=ifft(parteCorta64);
188 parteCorta80Time=parteCorta64Time(49:64);
189 parteCorta80Time(17:80)=parteCorta64Time;
190 preambulo=parteCorta80Time;
191 preambulo(81:160)=parteCorta80Time;
192 parteLargaTime=ifft(parteLarga);
193 preambulo(161:192)=parteLargaTime(33:64);
194 preambulo(193:256)=parteLargaTime;
195 preambulo(257:320)=parteLargaTime;
196 datosConPre=[preambulo;datosEnviar];
197 disp("Preambulo 802.11a añadido");
198 auxDat=zeros(TamanoBloque,1);
199 [longDat,~]=size(datosConPre);
200 auxDat(1:longDat,1)=datosConPre;
201 datosEnviarBloque=auxDat;
202 disp("Datos procesados")
```

Fuente: Elaboración propia

- Finalmente, se normalizan los datos, evaluando la parte real y la parte imaginaria de los vectores para encontrar el módulo máximo de la parte real e imaginaria, esto asegura que los diferentes tipos de modulación tengan la misma potencia promedio. Luego se transmiten los datos al canal inalámbrico utilizando el dispositivo Adalm Pluto.

Figura 60. Normalización de los datos para la posterior transmisión de los datos al canal inalámbrico

```
204 %% Normalizar los datos a transmitir
205
206     v_maxR=max(abs(real(datosEnviarBloque)));
207     v_maxI=max(abs(imag(datosEnviarBloque)));
208     if(v_maxR<v_maxI)
209         valorMax=v_maxI;
210     else
211         valorMax=v_maxR;
212     end
213     aux_TxNorm=datosEnviarBloque/valorMax;
214     auxTx=single(aux_TxNorm);
215     while (tx_Real_Time==20)
216         txAdaIm(auxTx(:,1));
217         disp("Datos transmitidos");
218         pause(Intervalo_Tiempo)
219     end
220 end
```

- Después de crear los dos scripts bajo los nombre de: “Parametros_Tx_PLUTO.m” y “Tecnica_Convolucional.m” se deberán ejecutar para comprobar la transmisión del archivo de texto utilizando el dispositivo SDR.

Marco teórico (a elaborar por el estudiante)

- Principio fundamental de las modulaciones multiportadora
Su principio fundamental radica en dividir un canal de banda ancha en canales de banda estrecha ortogonales, a los que se les denomina generalmente subportadoras, de forma que el flujo de bits a transmitir se reparte entre dichos canales de banda estrecha. (Belén, 2020)
- Estándares WLAN y sus principales características técnicas.
 - IEEE 802.11: el estándar que sirve de base en la comunicación de las redes inalámbricas. El primer estándar Wi-Fi del año 1997 permitió transferir datos a 1 Mbps.
 - IEEE 802.11a: se desarrolló sobre la base del estándar IEEE 802.11. Llegó en 1999, funciona en la banda de 5 GHz y alcanzó una velocidad máxima de 54 Mbps.

- IEEE 802.11b: fue el primer estándar desarrollado a finales de los años noventa. Es capaz de transferir datos a un máximo de 11 Mbps en la banda de 2,4 GHz.
 - IEEE 802.11g: también utiliza la banda de 2,4 GHz. Con este estándar, la velocidad máxima de transmisión se incrementó hasta los 54 Mbps. Llegó a partir de 2003.
 - IEEE 802.11n: se ratificó en septiembre de 2009. Funciona tanto en la banda de 2,4 GHz como en la de 5 GHz y alcanza velocidades de hasta 600 Mbps.
 - IEEE 802.11ac: se estandarizó a finales de 2013. Opera en la banda de 5 GHz y puede alcanzar velocidades de 1.300 Mbps.
 - IEEE 802.11ax: un avance importante que alcanza velocidades de hasta 10 Gbps.
 - IEEE 802.11be: será el próximo gran salto en conectividad Wi-Fi. Está previsto para 2024, trabajará en las bandas de 2,4 GHz, 5 GHz y 6 GHz, y promete velocidades de hasta 30 Gbps. (Ros, 2021)
- Tipos de errores

Existen diferentes factores que pueden altera la forma o temporalidad de una señal. Si la señal transporta datos digitales, los cambios pueden modificar el significado de los datos. Los errores posibles son:

Error de bit: Únicamente un bit de una unidad de datos determinada cambia de 1 a 0 o viceversa. Un error de bit altera el significado del dato. Son el tipo de error menos probable en una transmisión de datos serie, puesto que el intervalo de bit es muy breve ($1/\text{frecuencia}$) el ruido tiene que tener una duración muy breve. Sin embargo, si puede ocurrir en una transmisión paralela, en que un cable puede sufrir una perturbación y alterar un bit de cada byte.

Error de ráfaga: El error de ráfaga significa que dos o más bits de la unidad de datos han cambiado. Los errores de ráfaga no significan necesariamente que los errores se produzcan en bits consecutivos. La longitud de la ráfaga se mide desde el primero hasta el último bit correcto, algunos bits intermedios pueden estar bien. Los errores de ráfaga son más probables en transmisiones serie, donde la duración del ruido es normalmente mayor que la duración de un bit, por lo que afectara a un conjunto de bits. (Sites-Google, n.d.)

- Características importantes de la codificación convolucional

Se puede mencionar una propiedad importante de la codificación convolucional es que expande el alfabeto, pero no aumenta el ancho de banda. También opera bit a bit o sobre bloques pequeños y puede considerarse continua. Es decir, opera tanto sobre la información actual, presente a la entrada del codificador, como sobre la información pasada y se trata, por consecuencia, de un proceso con memoria.

Los códigos convolucionales se especifican mediante tres parámetros (n, k, m) donde n es el número de bits a la salida del codificador, k el número de bits de información a la entrada de éste y m , el número de registros de memoria. La relación k/n es la relación o tasa de código, y proporciona una medida de la eficiencia de codificación. (Pérez, 2019)

- ¿Por qué se usa la FFT en OFDM?

La transformada rápida de Fourier (FFT) también tiene su inversa, denotada por IFFT, la cual funciona exactamente con el mismo algoritmo. Teniendo en cuenta que esta transformada toma un número definido de muestras en el dominio del tiempo y da como resultado el mismo número de puntos en el dominio de la frecuencia es posible utilizarla en OFDM.

Sin embargo, antes de aplicar la IFFT al conjunto de puntos, es necesario hacer un mapeo del contenido de cada uno de éstos en algún tipo de modulación como QPSK o QAM; hecho el mapeo se procede a la aplicación de la IFFT para que cada uno de los símbolos sean convertidos al dominio del tiempo, además de asegurar que las muestras producidas sean ortogonales entre sí. (UNICAUCA, 2018)

Resultados (a elaborar por el estudiante)

Discusión (a elaborar por el estudiante)

Conclusiones (a elaborar por el estudiante)

Recomendaciones (a elaborar por el estudiante)

Preguntas de Control (a elaborar por el estudiante)

a) ¿Cómo se elige un intervalo de guarda?

El intervalo de guarda se elige con duración mayor que el máximo retardo de entre todas las subportadoras de forma que las componentes debidas al retardo de un símbolo no interfieran en el siguiente

b) ¿Cómo se evita la interferencia entre portadoras (ICI) y cómo se previene la interferencia intersímbolo (ISI)?

Al haber ICI produce la pérdida de ortogonalidad entre las portadoras por ello para evitar esta interferencia se introduce un prefijo cíclico (PC). Mientras que la ISI puede estar controlada con el intervalo de guarda. (Biblus, 2022)

c) ¿Cuál es la diferencia entre QPSK y BPSK?

BPSK representa la entrada binaria 1 y 0 con un cambio en la fase de la portadora de 180 grados. Mientras que QPSK representa dos bits que usan un símbolo de portador complejo, cada uno con un cambio de 90 grados entre sí.

d) Mencione algunos ejemplos donde los sistemas digitales de banda ancha utilicen la modulación OFDM.

- Telefonía móvil, en el acceso a Internet de banda larga LTE
- Televisión digital terrestre DVB-T, DVB-T2, DVB-H, T-DMB y ISDB-T,
- Red Wi-Fi IEEE 802.11AX
- Power Line Communication (PLC),
- Aplicaciones punto a punto o punto a multipunto.
- Docsis 3.1

Bibliografía (a elaborar por el estudiante)

5.2.5. PRÁCTICA # 5

ASIGNATURA: COMUNICACIONES DIGITALES

RESULTADO DE APRENDIZAJE DE LA PRÁCTICA: (VER SÍLABO)

TIEMPO PLANIFICADO EN EL SÍLABO: 2 HORAS

TIEMPO DE LA PRÁCTICA POR GRUPO: 2 HORAS

NÚMERO DE ESTUDIANTES POR GRUPO: 2 ESTUDIANTES

NÚMERO DE ESTACIONES DE TRABAJO: 10

Tema: Recepción de un archivo de texto con el uso del dispositivo Adalm Pluto implementando la técnica de corrección de errores Convolutional.

Objetivos:

- Realizar el estudio de la técnica Convolutional para la detección de errores al momento de receptar información en tiempo real.
- Utilizar el dispositivo Adalm Pluto para recibir un archivo de texto en tiempo real.
- Analizar los tipos de mapeos utilizados en esta práctica.

<i>Materiales y Reactivos</i>	<i>Equipo y herramientas</i>
<ul style="list-style-type: none">• 2 antenas	<ul style="list-style-type: none">• 1 módulo Adalm PlutoSDR• Programa de Matlab• 1 computador

Instrucciones

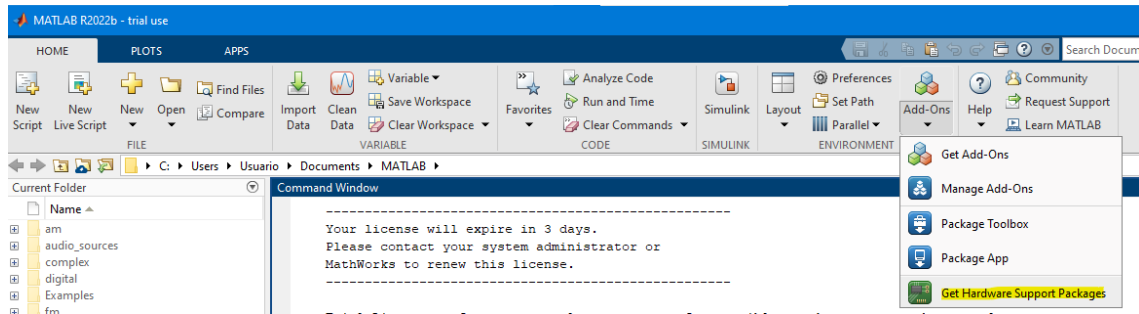
- Colocar las mochilas en los casilleros.
- Prohibido consumo de alimentos.
- Prohibido equipo de diversión, celulares etc.
- Prohibido jugar.
- Prohibido mover o intercambiar los equipos de los bancos de trabajo.
- Prohibido sacar los equipos del laboratorio sin autorización.
- Ubicar los equipos y accesorios en el lugar dispuesto por el responsable del laboratorio, luego de terminar las prácticas.
- Uso adecuado de equipos.
- Uso obligatorio del mandil.

Actividades por desarrollar

1. Instalar el paquete “Communications Toolbox Support Package for Analog Devices ADALM-Pluto Radio”.

- Abrir Matlab y en la barra de herramientas en la pestaña “HOME” buscar “Add Ons” y seleccionar “Get Hardware Support Packages”

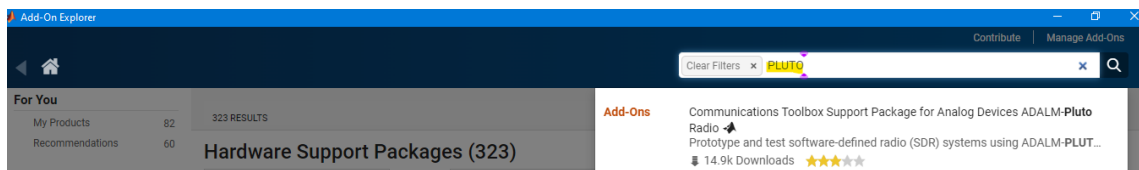
Figura 61. Forma de seleccionar el paquete para el dispositivo Adalm Pluto



Fuente: Elaboración propia

- A continuación, aparece una ventana “Add On Explorer” y en la barra de búsqueda encontrar el paquete para el dispositivo Adalm Pluto y proceder a instalar.

Figura 62. Ventana Add On Explorer para buscar el paquete específico para el hardware SDR

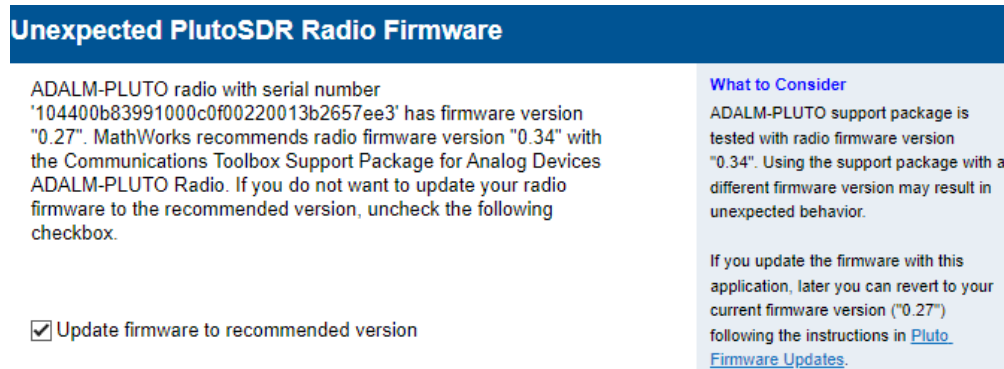


Fuente: Elaboración propia

2. Configurar el dispositivo Adalm Pluto

- Conectar el dispositivo mediante interfaz USB 2.0 para la conexión con SDR. Como primer paso es actualizar el firmware, los drivers y algunas librerías para su buen funcionamiento.

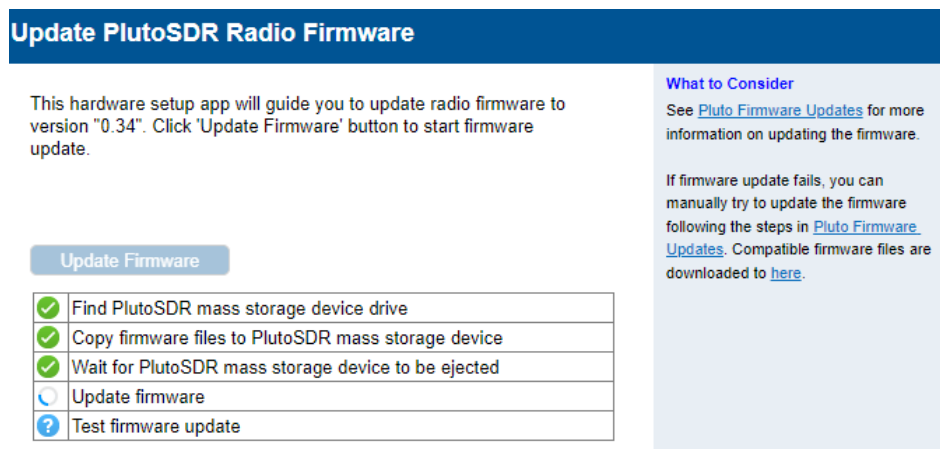
Figura 63. Actualización del firmware a la versión recomendada 0.34 del dispositivo *Adalm PlutoSDR*



Fuente: Elaboración propia

- Para seguir con el procedimiento dar clic en “Next”

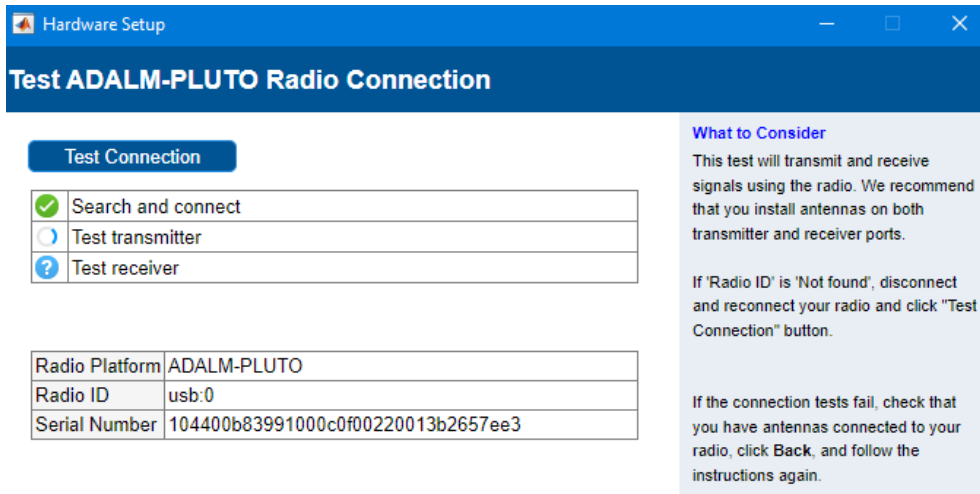
Figura 64. Proceso para la actualización del Firmware del dispositivo *PlutoSDR*



Fuente: Elaboración propia

- Finalmente, dar clic en “Test Connection” para verificar que la configuración realizada se haya completado con éxito.

Figura 65. Test de conexión entre el dispositivo PlutoSDR y el ordenador



Fuente: Elaboración propia

3. Para que el dispositivo SDR actúe como receptor se debe crear un script con las siguientes líneas de código donde se configura el valor de algunas variables que servirán para poder crear un objeto de radio y así poder interactuar con la radio Adalm Pluto. Luego, se guarda con el nombre de “Parametros_Rx_PLUTO.m”

Figura 66. Parámetros de configuración para la transmisión de texto y creación del objeto radio que servirá para interactuar con la radio Adalm Pluto SDR

```
1  %% Parámetros de configuración
2  radioID='usb:0'; %
3  frecuenciaCentral=860e6; %Frecuencia de operación en [Hz]
4  frecuenciaMuestreo=2000e3; % [Muestras/segundo]
5  Correction_Freq=0;
6  ShowAdvancedProperties=1; %opción para mostrar las propiedades avanzadas
7  GainSource='Manual'; % para poder configurar la ganancia en la recepción manualmente
8  Gainrx=20; % ganancia en la recepción
9  TamanoBloque=800000;
10 Num_Muestras_Rec_Tx=TamanoBloque;
11 OutputDataType='single'; %Tipo de datos en la salida
```

```

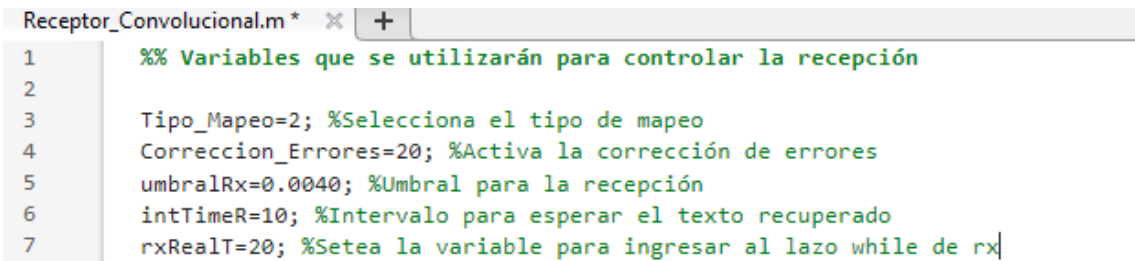
13 %% Creación del objeto para la recepción y su respectiva configuración
14 rxAdalm=sdr_rx('Pluto',...
15 'RadioID',radioID,...
16 'CenterFrequency',frecuenciaCentral,...
17 'GainSource',GainSource,...
18 'Gain',Gainrx,...
19 'BasebandSampleRate',frecuenciaMuestreo,...
20 'OutputDataType',OutputDataType,...
21 'SamplesPerFrame',Num_Muestras_Rec_Tx,...
22 'ShowAdvancedProperties',ShowAdvancedProperties, ...
23 'FrequencyCorrection',Correction_Freq);

```

Fuente: Elaboración propia

4. En otro script con el nombre de “Receptor_Convolucional.m” se agrega todo el procedimiento que implica la técnica Convolutiva. Primero se crearán las variables que permitirán controlar la recepción.

Figura 67. Creación de variables para controlar la recepción



```

Receptor_Convolucional.m * x +
1 %% Variables que se utilizarán para controlar la recepción
2
3 Tipo_Mapeo=2; %Selecciona el tipo de mapeo
4 Correccion_Errores=20; %Activa la corrección de errores
5 umbralRx=0.0040; %Umbral para la recepción
6 intTimeR=10; %Intervalo para esperar el texto recuperado
7 rxRealT=20; %Setea la variable para ingresar al lazo while de rx

```

Fuente: Elaboración propia

- Ahora se debe seleccionar el mismo archivo de texto que se transmitió con la finalidad de poder realizar la medición del BER al poder compararlo con el texto recuperado.

Figura 68. Selección del archivo transmitido para la medición del BER

```
9      %% Selección del archivo que fue enviado para medir el BER
10
11     Ingreso_Archivo=uigetfile("*.txt","Seleccionar archivo a transmitir");
12     if Ingreso_Archivo==0
13         disp("Archivo no seleccionado");
14         actBER=10; %Desactiva la medición del BER
15     else
16         fid=fopen(Ingreso_Archivo);
17         x=fread(fid,"*char");
18         fclose(fid);
19         textoBinario=dec2bin(x,8);
20         tamañoTexto=length(textoBinario);
21         aux=1;
22         for i=1:tamañoTexto
23             for j=1:8
24                 Texto_Bits_Serializado(aux)=textoBinario(i,j);
25                 aux=aux+1;
26             end
27         end
28         datos_SerializadosTxOrig=Texto_Bits_Serializado;
29         for aux=1:length(datos_SerializadosTxOrig)
30             datosMedirBER(aux)=str2double(datos_SerializadosTxOrig(aux));
31         end
32         longDatosBER=length(datosMedirBER);
33         actBER=20; %Activa la medición del BER
34     end
```

Fuente: Elaboración propia

- Luego se empieza a recibir los datos del canal inalámbrico de manera que actuará repetitivamente hasta que pueda detectar los datos transmitidos. Para ello, la creación del objeto radio servirá para que el dispositivo receptor Adalm Pluto establezca comunicación con el dispositivo que hace de transmisor.

Figura 69. Estableciendo conexión con la radio Adalm Pluto para la recepción de datos

```
36 %% Recepción de los datos en tiempo real mediante el dispositivo SDR
37
38 while(rxRealT==20) %Bucle de todos los procesos de recepción en tiempo real
39     clearvars datosRecibidos;
40     for i=0:6
41         vaciar=rxAdalm();
42     end
43     condAux=0;
44     while(condAux==0) %Guarda la señal recuperada del canal inalámbrico
45         datosRxComplejos=rxAdalm();
46         datosRecibidos=double(datosRxComplejos);
47         v_max=max(abs(datosRecibidos));
48         fprintf("Valor máximo de las muestras: %8.5f\n",v_max)
49         if(v_max>umbralRx) %Verifica si se supera el umbral establecido
50             datosRxAux=rxAdalm(); %Guarda la señal recuperada
51             datosRec2=double(datosRxAux);
52             if (max(abs(datosRec2))>umbralRx)
53                 datosRecibidos=[datosRecibidos;datosRec2];
54             end
55             condAux=1;
56         end
57     end
58     disp("Datos recibidos del canal inalámbrico") %Informa de los datos rx
```

Fuente: Elaboración propia

- Como ya se han recibido los datos lo que procede es utilizar el umbral de recepción para poder recortar datos con la finalidad de reducir la carga de procesamiento en las etapas posteriores.

Figura 70. Recorte de datos para reducir la carga de procesamiento

```
60      %% Recortar datos para reducir la carga de procesamiento
61      |
62      tic %Mide el tiempo de procesamiento
63      pos_inic=0;
64      for i=1:length(datosRecibidos)
65          if(max(abs(datosRecibidos(i)))>umbralRx)
66              pos_inic=i;
67              break;
68          end
69      end
70      if(pos_inic>400)
71          datosRec=datosRecibidos(pos_inic-400:end);
72      else
73          datosRec=datosRecibidos;
74      end
75      datosInv=flip(datosRec);
76      pos_fin=0;
77      for i=1:length(datosRec)
78          if(max(abs(datosInv(i)))>umbralRx)
79              pos_fin=i;
80              break
81          end
82      end
83      if(pos_fin>200)
84          datosInvRec=datosInv(pos_fin-200:end);
85      else
86          datosInvRec=datosInv;
87      end
88      datosRecortados=flip(datosInvRec);
89      disp("Datos Recortados")
```

Fuente: Elaboración propia

- Las siguientes líneas de código corresponden a la detección de bloques de datos que pertenecen a una secuencia OFDM. Es necesario que el receptor pueda encontrar las muestras de la señal recibidas perteneciente al bloque OFDM que se quiera demodular sin interferencia de otros bloques y con la fase correcta ya sea para recalculer la estimación de bloque periódicamente, evitar pérdidas de sincronismo y mejorar la estimación del comienzo de cada bloque.

Figura 71. *Detección de la trama OFDM*

```
91     %% Detección de la trama OFDM
92
93     datosRX=datosRecortados;
94     if length(datosRX)>(Num_Muestras_Rec_Tx+600)
95         disp("*****Muestras recibidas inválidas*****");
96         continue
97     end
98     %Al inicio de cada paquete se encuentra el preambulo
99     Mmues=1400; %Primeras muestras que evitan el procesamiento innecesario
100    if(length(datosRX)>Mmues)
101        r=datosRX(1:Mmues);
102
103    else
104        r=datosRX;
105    end
106    L=48; %Múltiplo de la longitud de una secuencia corta
107    inMax=length(r)-2*L+1;
108    m=L-1;
109    M=zeros(Mmues,1);
110    for k=1:inMax %Métrica para comprobar si es un paquete OFDM
111        Numerador=(r(k:k+m)'*r(k+L:k+m+L)); %Sumatorio del numerador
112        Denom= sum((abs(r(k+L:k+m+L))).^2); %Sumatorio del denominador
113        M(k)=Numerador/Denom;
114    end
115    M2=(abs(M)).^2;
116    umbral=0.60; %Umbral para estimar si existe un paquete OFDM >40
117    tramaEncontrada=0; %Trama no encontrada
118    auxET=0;
119    for i=1:length(M2)
120        if(M2(i)>umbral)
121            auxET=i;
122            tramaEncontrada=1;
123            break;
124        end
125    end
126    datosOFDM=[]; %Inicia la variable que contiene la trama OFDM
127    if(tramaEncontrada==1)
128        datosOFDM(:,1)=datosRX(auxET:end);
129    else
130        disp("*****Trama OFDM NO encontrada*****");
131        pause(0.1)
132        continue
133    end
134    disp("Trama OFDM detectada");
```

Fuente: Elaboración propia

- Tras estimar los límites del símbolo en el receptor y detectar si el símbolo está presente, el siguiente paso es estimar y corregir el desplazamiento de frecuencia de los datos que se recibe.

Figura 72. *Estimación y corrección del desplazamiento de frecuencia*

```
135     %% Corrección CFO
136
137     r=datosOFDM;
138     Fs=frecuenciaMuestreo;
139     N=128; %Máximo valor del índice k
140     L=16; %Longitud de una secuencia corta
141     m=L-1; %Máxima longitud que puede tomar m
142     fEst=zeros(N,1); % Inicia vector que almacena la estimación de freq
143     for k=1:N %Bucle que recorre de k igual a 1 hasta N
144         P=r(k:k+m)*r(k+L:k+m+L);
145         fEst(k)=(Fs/(2*pi*L))*(angle(P)); %Estima la freq en la posición k
146     end
147     estFreq=-mean(fEst); %Negativo del promedio del CFO estimado
148     %Corrección de freq con la estimación realizada
149     despFreqObj= comm.PhaseFrequencyOffset("FrequencyOffset", ...
150     estFreq,"SampleRate",Fs);
151     %Aplica la estimación de CFO a las muestras recibidas para corregir
152     datosCorrCFO=despFreqObj(datosOFDM);
153     disp("Correccion CFO terminada");
```

Fuente: Elaboración propia

- Continuando con el proceso, es necesario realizar la sincronización de tiempo de símbolo para determinar la muestra correcta de inicio de símbolo antes de la demodulación mediante FFT, es decir que su objetivo es encontrar el instante de tiempo en el que empieza el símbolo OFDM la posición en la que centrar la ventana de datos correspondiente a un símbolo OFDM.

Figura 73. Proceso de sincronización de tiempo de símbolo

```
156     if length(datosCorrCFO)>5000
157         datSinCFO=datosCorrCFO(1:5000);
158     else
159         datSinCFO=datosCorrCFO;
160     end
161     parteLarga=[...    %parte larga del preámbulo 802.11a (dominio freq)
162         0;1;-1;-1;1;1;-1;1;-1;1;-1;-1;-1;-1;1
163         1;-1;-1;1;-1;1;-1;1;1;1;0;0;0;0;0
164         0;0;0;0;0;1;1;-1;-1;1;1;-1;1;-1;1
165         1;1;1;1;-1;-1;1;1;-1;1;-1;1;1;1];
166     %Conversión del dominio de la frecuencia al tiempo, y adición del PC
167     parteLargaTime=ifft(parteLarga);
168     LLTF=[parteLargaTime(33:64);parteLargaTime;parteLargaTime];
169     secLLTF80=LLTF(1:80);
170     datosCorrRell=(abs(xcorr(datSinCFO,secLLTF80))).^2;
171     %Se corrige en tiempo los datos correlacionados para que
172     %coincidan con las posiciones de los datos de entrada
173     datosCorr=datosCorrRell(length(datSinCFO):end);
174     [~,posPico1]=max(datosCorr);
175     datosCorr(posPico1)=0;
176     [~,posPico2]=max(datosCorr);
177     if abs(posPico2-posPico1)==64
178         inicioLLTF=min([posPico1 posPico2]);
179         inicioLSTF=inicioLLTF-160;
180         if(inicioLSTF>0)
181             datosSincronizados=datosCorrCFO(inicioLSTF:end);
182         else
183             disp("Falla en la sincronización de símbolo (2)")
184             continue
185         end
186     else
187         disp("Falla de sincronización de símbolo")
188         continue
189     end
190     disp("Sincronización de símbolo terminado");
```

Fuente: Elaboración propia

- Una vez detectado el paquete y corregido el CFO, el receptor utilizará el campo largo, denominado Legacy Long Training Field (LLTF) en el estándar IEEE 802.11, que contiene dos secuencias repetidas y un PC. Esta parte del preámbulo está diseñada para utilizarse en la estimación del canal. En general, un receptor OFDM identificará primero el inicio del paquete, corregirá las desviaciones de frecuencia, corregirá los efectos del canal y eliminará las distorsiones de fase residuales.

Figura 74. Corrección y eliminación de las distorsiones de fase residuales

```
192     %% Corrección de Fase
193
194     a=fft(datosSincronizados(257:320));
195     b=fft(datosSincronizados(193:256));
196     angulosAux=[a(2) a(5) a(6) a(8) a(10) a(16) a(17) a(20)...
197               a(53) a(56) a(57) a(59) a(61) a(62) a(63) a(64)...
198               b(2) b(5) b(6) b(8) b(10) b(16) b(17) b(20)...
199               b(53) b(56) b(57) b(59) b(61) b(62) b(63) b(64)];
200     angulos=(angle(angulosAux)*180)/pi;
201     anguloProm=-mean(angulos);
202     desFaseObj= comm.PhaseFrequencyOffset("PhaseOffset",anguloProm,...
203     "SampleRate",Fs);
204     datFase=desFaseObj(datosSincronizados);
205     longDatF=length(datFase);
206     padd80=ceil(longDatF/80)*80-longDatF;
207
208     if(padd80>=1)
209         datosCorrFase=[datFase;datFase(end-padd80+1:end)];
210     else
211         datosCorrFase=datFase;
212     end
213     disp("Corrección de Fase Terminado");
```

Fuente: Elaboración propia

- En este punto se elimina el prefijo cíclico, es decir, las muestras que se agregaron en la transmisión al inicio de cada símbolo OFDM

Figura 75. Proceso de eliminación del Prefijo cíclico

```
215     %% Eliminar Prefijo cíclico
216
217     clearvars simbolosSinPC;
218     nSimOFDM=0;
219     tramaDatos=datosCorrFase(321:end,1);
220
221     for j=1:80:length(tramaDatos)
222         nSimOFDM=nSimOFDM+1;
223         simbolosSinPC(:,nSimOFDM)=tramaDatos(j+16:j+79);
224     end
225     disp("Proceso Eliminación PC terminado");
```

Fuente: Elaboración propia

- Después se deben pasar los símbolos OFDM del dominio del tiempo al dominio de la frecuencia utilizando la transformada rápida de Fourier. Además, se debe realizar la estimación del canal a través de las portadoras piloto. El ecualizador, debe de reducir los efectos del canal y eliminar cualquier desajuste residual de fase o frecuencia que quede en la señal recibida.

Figura 76. Proceso para pasar los símbolos OFDM al dominio de la frecuencia

```
227     %% Transformada Rápida de Fourier
228     simbolosfft=fft(simbolosSinPC);
229     disp("Proceso FFT terminado");
230     %%Estimación de canal y ecualizador
231     umbralF=2;
232     simbolosEcualizados=[];
233     for auxnSimb=1:nSimOFDM
234         simbF=simbolosfft(:,auxnSimb);
235         %%Determina el comportamiento del canal sobre las portadoras piloto
236         v1=[simbF(8) simbF(22)];
237         v2=[simbF(44) simbF(58)];
238         x1=[8 22]; %Puntos de la muestra
239         xq1=(2:23); %Puntos a estimar
240         x2=[44 58]; %Puntos de la muestra
241         xq2=(43:64); %Puntos a estimar
242         val_interp1=(interp1(x1,v1,xq1,"linear","extrap")).';
243         val_interp2=(interp1(x2,v2,xq2,"linear","extrap")).';
244         datEc1=simbF(2:23)./val_interp1; %Interpolación de los datos
245         datEc2=simbF(43:64)./val_interp2;
246         DatosEc=zeros(64,1);
247         DatosEc(2:23)=datEc1; %Datos ecualizados en una nueva variable
248         DatosEc(43:64)=datEc2;
249         simbolosEcualizados(:,auxnSimb)=DatosEc;
250     end
251     nSimOFDM=auxnSimb; %Número de símbolos OFDM
252     disp("Ecualización terminada");
```

Fuente: Elaboración propia

- Como paso siguiente se deben eliminar las portadoras de guarda, las portadoras piloto para mantener únicamente las portadoras con los datos. Este proceso se llama desamblaje de los símbolos OFDM.

Figura 77. *Proceso de mantener únicamente las portadoras con los datos de manera que se pueda recuperar los datos del símbolo OFDM*

```
254     %% Recuperación de datos del símbolo OFDM
255
256     nPdatos=40; %Número de portadoras de datos
257     datosMapeo=zeros(nPdatos*nSimOFDM,1); %datos serializados
258     for aux=1:nSimOFDM %Obtención de los datos OFDM
259         simAux=simbolosEcualesados(:,aux);
260         datosAux=zeros(40,1);
261         datosAux(21:26)=simAux(2:7);
262         datosAux(27:39)=simAux(9:21);
263         datosAux(40)=simAux(23);
264         datosAux(1)=simAux(43);
265         datosAux(2:14)=simAux(45:57);
266         datosAux(15:20)=simAux(59:64);
267         %Serializa los datos
268         datosMapeo((aux-1)*nPdatos+1:aux*nPdatos)=datosAux;
269     end
270     disp("Proceso desamblaje OFDM terminado")
```

Fuente: Elaboración propia

- A continuación, se pasa por los esquemas de modulación BPSK, QPSK, 16-QAM y 64-QAM a bits serializados

Figura 78. Proceso de selección de los diferentes esquemas de modulación

```
272 %% Proceso de demapeo
273
274 if(Tipo_Mapeo==1) %Demapeo BPSK
275     M=2;
276     Nb=log2(M);
277     longitud=length(datosMapeo);
278     datosDemod=zeros(longitud,1);
279     for i=1:longitud
280         if(real(datosMapeo(i))>=0 && imag(datosMapeo(i))>=0)
281             datosDemod(i)=1;
282         elseif(real(datosMapeo(i))<0 && imag(datosMapeo(i))<0)
283             datosDemod(i)=0;
284         end
285     end
286 elseif(Tipo_Mapeo==2) %Demapeo QPSK
287     M=4;
288     Nb=log2(M);
289     long_pM=ceil(length(datosMapeo)/Nb)*Nb-length(datosMapeo);
290     paddM=zeros(long_pM,1);
291     ModDat=(sqrt(2))*[datosMapeo;paddM];
292     datosDemodS=pskdemod(ModDat,M,pi/4,"gray");
293     demodMatriz=de2bi(datosDemodS,Nb,"left-msb");
294     datosDemod=reshape(demodMatriz.',1,[]).';
295 elseif(Tipo_Mapeo==3) %Demapeo 16-QAM
296     M=16;
297     Nb=log2(M);
298     long_pM=ceil(length(datosMapeo)/Nb)*Nb-length(datosMapeo);
299     paddM=zeros(long_pM,1);
300     ModDat=(sqrt(10))*[datosMapeo;paddM];
301     datosDemodS=qamdemod(ModDat,M);
302     demodMatriz=de2bi(datosDemodS,Nb,"left-msb");
303     datosDemod=reshape(demodMatriz.',1,[]).';
304 elseif(Tipo_Mapeo==4) %Demapeo 64-QAM
305     M=64;
306     Nb=log2(M);
307     long_pM=ceil(length(datosMapeo)/Nb)*Nb-length(datosMapeo);
308     paddM=zeros(long_pM,1);
309     ModDat=(sqrt(42))*[datosMapeo;paddM];
310     datosDemodS=qamdemod(ModDat,M);
311     demodMatriz=de2bi(datosDemodS,Nb,"left-msb");
312     datosDemod=reshape(demodMatriz.',1,[]).';
313 else
314     disp("Mapeo no realizado")
315     continue
316 end
317 disp("Datos Demapeados")
```

Fuente: Elaboración propia

- Luego se realiza el proceso de corrección de errores a nivel de bit, debido a que esa es la forma en la que opera la técnica convolucional. El decodificador tiene como

propósito encontrar de entre el conjunta de todas las palabras posibles que pudieran transmitirse la palabra código que más se asemeje a la información recibida.

Figura 79. *Proceso de corrección de errores a nivel de bit*

```
319 %% Decodificador Convolutional
320
321 if Correccion_Errores==20
322     NPdatos=40;
323     nBc=Nb*NPdatos;
324     nGrupos=ceil(length(datosDemod)/nBc);
325     longiPadd=nBc*nGrupos-length(datosDemod);
326     v_relleno=zeros(longiPadd,1);
327     datosDemodRell=[datosDemod;v_relleno];
328     nc=2;
329     kc=1;
330     mc=3;
331     g1=5;
332     g2=7;
333     Rcc=kc/nc;
334     P_Trellis=poly2trellis(mc, [g1 g2]);
335     tb=nBc*Rcc;
336     datosDecodConv=[];
337
338     for i=1:nGrupos
339         auxDecod = datosDemodRell((nBc*(i-1)+1):nBc*i,1);
340         DecodVid=vitdec(auxDecod,P_Trellis,tb,"trunc","hard");
341         datosDecodConv=[datosDecodConv;DecodVit];
342     end
343     disp("Corrección de errores implementado")
344 else
345     datosDecodConv=datosDemod;
346 end
```

Fuente: Elaboración propia

- Posteriormente, se realiza la medición de la tasa de errores (BER) es decir, que se va a cuantificar la cantidad de bits erróneos en relación a la entrada del sistema.

Figura 80. Proceso para cuantificar la cantidad de bits erróneos

```
348     %% Medicion del BER
349
350     if(actBER==20)
351         LongDatDecod=length(datosDecodConv);
352         if(longDatosBER<=LongDatDecod)
353             datosSerializadosRx=datosDecodConv(1:longDatosBER);
354             bitsErrado=0;
355             for aux=1:longDatosBER
356                 if(datosSerializadosRx(aux)~=datosMedirBER(aux))
357                     bitsErrado=bitsErrado+1;
358                 end
359             end
360             disp("Medición del BER:");
361             v_BER=bitsErrado/longDatosBER;
362             fprintf("Bits errados/Bits totales = %g/%g \n",...
363                 bitsErrado,longDatosBER);
364             fprintf("BER = %g \n",v_BER);
365         else
366             disp("Archivo Recuperado, BER no medido")
367             disp("Longitud del archivo recuperado menor al original");
368         end
369     else
370         datosSerializadosRx=datosDecodConv;
371     end
```

Fuente: Elaboración propia

- Para terminar, se recupera el archivo de texto que en un inicio fue transmitido. Para la transmisión y recepción se usaron los mismos dispositivos Adalm Pluto.

Figura 81. Recuperación del texto

```
373     %% Recuperación del texto
374     bPc=8;
375     numC=ceil(length(datosSerializadosRx)/8);
376     LongRellnumC=bPc*numC-length(datosSerializadosRx);
377     rellenoC=zeros(LongRellnumC,1);
378     datos_Serializados=[datosSerializadosRx ; rellenoC];
379     datosMtz=reshape(datos_Serializados,[8 numC]).';
380     datosTexto=[];
381     for i=1:numC
382         datosTexto(1,i)=double(bi2de(datosMtz(i,:), "left-msb"));
383     end
384     Texto_Original=transpose(x)
385     disp("Texto recuperado: ");
386     Texto_Recuperado=char(datosTexto)
387     disp("Proceso de recuperación de texto terminado");
388     toc
389     pause(intTimeR)
390 end
```

Fuente: Elaboración propia

5. Ejecutar los dos scripts creados “Parametros_Rx_PLUTO.m” y “Receptor_Convolucional.m” para comprobar la configuración del dispositivo SDR y el funcionamiento de todo el proceso de la técnica convolucional en el lado receptor.

Marco teórico (a elaborar por el estudiante)

- Principio de ortogonalidad

El principio de ortogonalidad permite la transmisión simultánea en un estrecho rango de frecuencias y sin que se produzcan interferencias entre ellas, logrando una eficiencia espectral equivalente a doble de las señales QPSK y QAM, señales en cuya modulación, es basada OFDM. permitiendo que múltiples señales de información sean transmitidas sobre un canal común y detectadas sin interferencia. La pérdida de ortogonalidad en un sistema, da como resultado degradación en la comunicación debido al deterioro de las señales. Algunos esquemas de multiplexación son naturalmente ortogonales como es el caso de la Multiplexación por división de tiempo (TDM). En el dominio de la frecuencia las portadoras en una señal OFDM están espaciadas tan cerca como es teóricamente posible mientras se mantenga la ortogonalidad entre ellas.

- En qué consiste la tasa de bits en error (VER – Bit Error Rate)

El rendimiento de un sistema de comunicaciones digital es medido en términos de la tasa de bits en error, o lo que es lo mismo, la relación de bits erróneos detectados en el receptor. En general, la BER depende del método de modulación, del esquema de codificación, del tipo de forma de onda usada, de la potencia del transmisor, de las características del canal y del esquema de demodulación. La representación convencional del rendimiento de un sistema digital en un canal lineal contaminado por ruido muestra el BER frente a E_b/N_0 , donde E_b representa la energía de bit y N_0 representa la densidad espectral de potencia del ruido introducido en un canal.

Para un nivel de ruido dado, el BER puede ser reducido incrementando la energía asociada a cada bit, transmitiendo con mayor potencia o con un mayor periodo de bit. La meta en las comunicaciones digitales es obtener un determinado valor de BER con la mínima energía de bit posible. (Alsinella, 2008)

- Sincronización de la trama

El término sincronizar en comunicaciones digitales hace referencia a la coincidencia que se debe tener en el tiempo entre las señales de información. Las señales de sincronización permiten recuperar correctamente la información transmitida a través de un proceso de reconocimiento de señales de diferentes tipos, tales como:

Sincronización de bit: También denominada sincronización de reloj, permite distinguir el intervalo correspondiente a cada uno de los bits transmitidos. La sincronización de reloj asegura que el transmisor y el receptor coincidan en un intervalo preciso de tiempo para que se presente un bit. Cuando se recibe una cadena continua de datos, es necesario identificar cuáles bits pertenecen a cuáles caracteres y cuál bit es el menos significativo de los datos, el bit de paridad y el bit de paro.

Sincronización de trama: Permite separar los distintos grupos de bits de una transmisión múltiplex de forma que los bits recibidos se puedan clasificar y dirigir al canal de salida apropiado, evitando pérdidas en la información por el desfase de la señal.

Sincronización de portadora: Recupera la frecuencia y la fase de la portadora utilizada en la señal de transmisión, para la detección de la señal de información. (Vega Rodriguez, 2015)

Resultados (a elaborar por el estudiante)

Discusión (a elaborar por el estudiante)

Conclusiones (a elaborar por el estudiante)

Recomendaciones (a elaborar por el estudiante)

Preguntas de Control (a elaborar por el estudiante)

a) ¿Qué inconveniente se podría dar utilizando portadoras de guarda?

El inconveniente de utilizar portadoras de guarda es que reduce la velocidad de transmisión de datos total.

b) ¿Cuáles son las dos secciones de preámbulo en una trama WLAN 802.11?

La primera sección en la norma IEEE 802.11 es identificada como la parte corta del preámbulo, denominada Legacy Short Training Field (LSTF), que contiene diez copias repetidas de una secuencia corta. Su propósito es realizar la estimación del desplazamiento de la frecuencia portadora (CFO). Y la otra sección es denominada Legacy Long Training Field (LLTF) contiene dos secuencias repetidas y un prefijo

cíclico, esta parte del preámbulo está diseñada para utilizarse en la estimación del canal. Ambos preámbulos se utilizan para corregir la trama recibida.

c) Mencione un problema asociado a la OFDM

Uno de los problemas puede ser el efecto de los desplazamientos de frecuencia. Cuando los símbolos recibidos experimentan un desplazamiento superior del ancho de banda de la subportadora, la señal se vuelve no ortogonal, lo que rompe los supuestos de recuperación asociados a la señal. Por lo tanto, antes de demodular la señal OFDM, es decir procesarla mediante FFT, debe considerarse la corrección de frecuencia.

d) ¿Qué proceso se realiza después de la detección de paquetes y la corrección de frecuencia?

Tras la detección de paquetes y la corrección de frecuencia puede realizarse la temporización de símbolos. Este proceso proporciona una estimación de precisión a nivel de símbolo de las secuencias del preámbulo en el dominio del tiempo. Para proporcionar esta precisión adicional, se realiza una correlación cruzada utilizando la secuencia de preámbulo transmitida. La técnica de correlación cruzada determina el límite entre las secuencias de entrenamiento cortas y largas con una precisión a nivel de muestra, es por ello que se recomienda utilizar las 80 primeras muestras de la secuencia LLTF.

e) ¿Cuándo es necesario utilizar múltiples ecualizadores?

Cuando el canal es dinámico, es decir cuando puede cambiar en pequeños periodos de tiempo en comparación con la longitud de la trama, se hace necesario utilizar múltiples ecualizadores. (Collins et al., 2018)

Bibliografía (a elaborar por el estudiante)

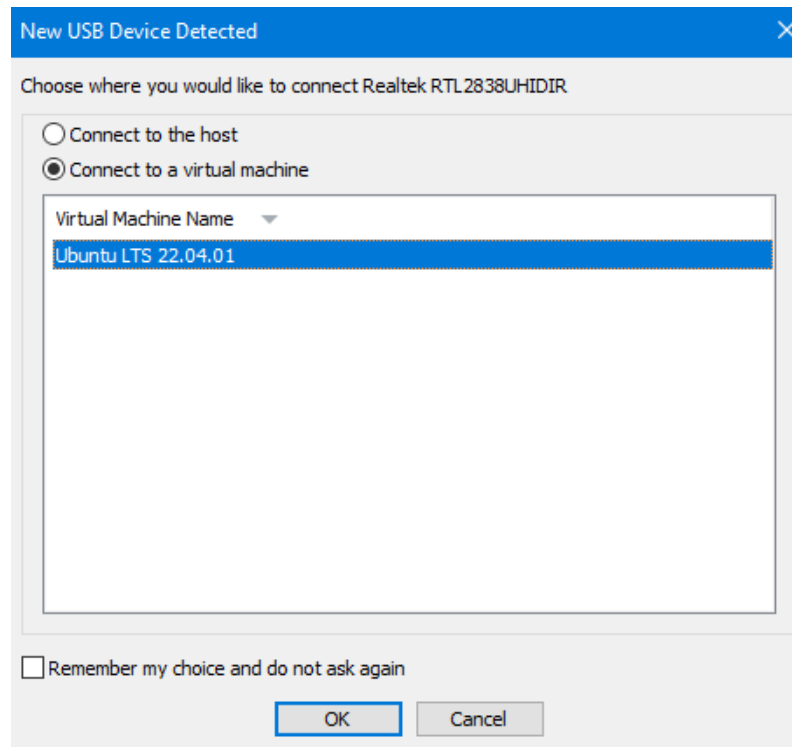
6. Resultados

En esta sección se da a conocer los resultados que fueron obtenidos de cada una de las guías de prácticas realizadas. Es conveniente resaltar que para su implementación se utilizaron diferentes módulos SDR como diferentes softwares que permitieron obtener los siguientes resultados.

6.1. PRÁCTICA #1

Tema: Implementación de un sistema de recepción ADS-B utilizando Radio Definida por Software (SDR)

Figura 82. *Detección del dispositivo NESDR en el computador, listo para utilizarse en la nueva máquina virtual Ubuntu*



Fuente: Elaboración propia

Figura 83. Líneas de código para actualizar hardware, software o firmware de la máquina virtual Ubuntu

```
jess@jess-virtual-machine:~$ sudo apt-get update
Obj:1 http://ec.archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Obj:3 http://ec.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:4 https://dl.google.com/linux/chrome/deb stable InRelease
Obj:5 http://ec.archive.ubuntu.com/ubuntu jammy-backports InRelease
Obj:6 https://ppa.launchpadcontent.net/gnuradio/gnuradio-releases/ubuntu jammy I
nRelease
Leyendo lista de paquetes... Hecho
jess@jess-virtual-machine:~$ sudo apt-get upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
 libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

Fuente: Elaboración propia

Figura 84. Línea de código git core para gestionar las modificaciones del sistema de archivos de un proyecto

```
jess@jess-virtual-machine:~$ sudo apt-get install git-core
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Nota, seleccionando «git» en lugar de «git-core»
git ya está en su versión más reciente (1:2.34.1-1ubuntu1.5).
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
 libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

Fuente: Elaboración propia

Figura 85. Línea de código cmake utilizado para controlar el proceso de compilación del software usando ficheros de configuración sencillos e independientes de la plataforma.

```
jess@jess-virtual-machine:~$ sudo apt-get install cmake
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
cmake ya está en su versión más reciente (3.22.1-1ubuntu1.22.04.1).
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
 libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

Fuente: Elaboración propia

Figura 86. Línea de código *libusb* proporciona aplicaciones con acceso para controlar transferencia de datos hacia y desde dispositivos USB

```
jess@jess-virtual-machine:~$ sudo apt-get install libusb-1.0-0-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
libusb-1.0-0-dev ya está en su versión más reciente (2:1.0.25-1ubuntu2).
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
 libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

Fuente: Elaboración propia

Figura 87. Línea de código *build-essential* para crear paquetes DEB (extensión del formato *d* paquetes de software de la distribución de Linux) a partir del código fuente de una aplicación.

```
jess@jess-virtual-machine:~$ sudo apt-get install build-essential
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
build-essential ya está en su versión más reciente (12.9ubuntu3).
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
 libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

Fuente: Elaboración propia

Figura 88. Línea de código *pkg-config* para consulta las bibliotecas instaladas con el fin de compilar el software que depende de ellas.

```
jess@jess-virtual-machine:~$ sudo apt-get install pkg-config
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
pkg-config ya está en su versión más reciente (0.29.2-1ubuntu3).
fijado pkg-config como instalado manualmente.
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
 libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

Fuente: Elaboración propia

Figura 89. Línea de código `cd` permite cambiar la ruta actual de la terminal y ubicarnos en una carpeta o directorio específico.

```
jess@jess-virtual-machine:~$ cd rtl-sdr
jess@jess-virtual-machine:~/rtl-sdr$ mkdir build
mkdir: no se puede crear el directorio «build»: El archivo ya existe
jess@jess-virtual-machine:~/rtl-sdr$ cd build
jess@jess-virtual-machine:~/rtl-sdr/build$ cmake ../ -DINSTALL_UDEV_RULES=ON
-- Build type not specified: defaulting to release.
-- Extracting version information from git describe...
-- Building with kernel driver detaching enabled
-- Building with usbfs zero-copy support disabled, use -DENABLE_ZEROCOPY=ON to enable
-- Building for version: 0.6.0-34-g5e73 / 0.6git
-- Using install prefix: /usr/local
-- Configuring done
-- Generating done
-- Build files have been written to: /home/jess/rtl-sdr/build
```

Fuente: Elaboración propia

Figura 90. La línea de código `make` sirve para instalar y compilar muchas utilidades en este caso de `rtl-sdr`

```
jess@jess-virtual-machine:~/rtl-sdr/build$ make
Consolidate compiler generated dependencies of target rtlsdr
[ 21%] Built target rtlsdr
Consolidate compiler generated dependencies of target rtlsdr_static
[ 43%] Built target rtlsdr_static
Consolidate compiler generated dependencies of target convenience_static
[ 50%] Built target convenience_static
Consolidate compiler generated dependencies of target rtl_sdr
[ 56%] Built target rtl_sdr
Consolidate compiler generated dependencies of target rtl_tcp
[ 62%] Built target rtl_tcp
Consolidate compiler generated dependencies of target rtl_test
[ 68%] Built target rtl_test
Consolidate compiler generated dependencies of target rtl_fm

jess@jess-virtual-machine:~/rtl-sdr/build$ sudo make install
[sudo] contraseña para jess:
[ 21%] Built target rtlsdr
[ 43%] Built target rtlsdr_static
[ 50%] Built target convenience_static
[ 56%] Built target rtl_sdr
[ 62%] Built target rtl_tcp
[ 68%] Built target rtl_test
[ 75%] Built target rtl_fm
[ 81%] Built target rtl_eeprom
[ 87%] Built target rtl_adsb
[ 93%] Built target rtl_power
[100%] Built target rtl_biast
Install the project...
-- Install configuration: "Release"
-- Installing: /etc/udev/rules.d/rtl-sdr.rules
```

Fuente: Elaboración propia

Figura 91. Línea de código git clone crea una copia o clon del repositorio “<https://github.com/antirez/dump1090.git>” en un nuevo directorio.

```
jess@jess-virtual-machine:~$ git clone https://github.com/antirez/dump1090.git
Clonando en 'dump1090'...
remote: Enumerating objects: 292, done.
remote: Total 292 (delta 0), reused 0 (delta 0), pack-reused 292
Recibiendo objetos: 100% (292/292), 552.76 KiB | 1.03 MiB/s, listo.
Resolviendo deltas: 100% (164/164), listo.
jess@jess-virtual-machine:~$
```

Fuente: Elaboración propia

Figura 92. Línea de código cd para acceder a la ubicación del directorio dump1090 para instalar y compilar las dependencias del programa

```
jess@jess-virtual-machine:~$ cd dump1090
jess@jess-virtual-machine:~/dump1090$ make
cc -O2 -g -Wall -W -I/usr/local/include/ -c dump1090.c
dump1090.c: In function 'interactiveReceiveData':
dump1090.c:1804:17: warning: absolute value function 'abs' given an argument of type 'long
  which may cause truncation of value [-Wabsolute-value]
 1804 |         if (abs(a->even_cpftime - a->odd_cpftime) <= 10000) {
      |             ^
cc -O2 -g -Wall -W -I/usr/local/include/ -c anet.c
cc -g -o dump1090 dump1090.o anet.o -L/usr/local/lib -lrtlsdr -lpthread -lm
jess@jess-virtual-machine:~/dump1090$
```

Fuente: Elaboración propia

Figura 93. Parámetros de la aeronave detectada con el programa dump1090

```
jess@jess-virtual-machine: ~/dump1090
```

Hex	Flight	Altitude	Speed	Lat	Lon	Track	Messages Seen	.
e8408d	RER872	15700	353	-3.941	-79.265	15	50	24 sec

Fuente: Elaboración propia

Figura 94. Seguimiento visual de la aeronave del vuelo RER872 que proporciona la página <https://flightaware.com/> e información adicional sobre el vuelo.



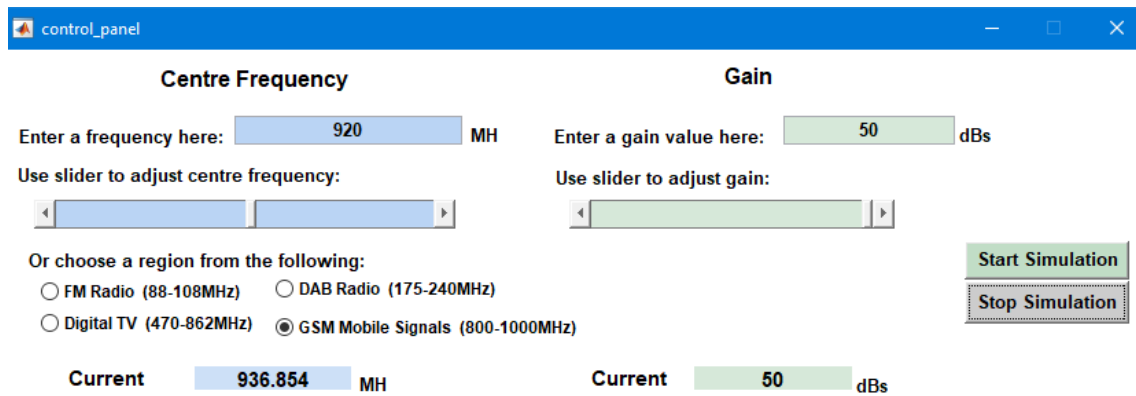
Fuente: Elaboración propia

Como podemos ver en la Figura 93 se detectó el vuelo RER872 cuyo registro o código único Hex se identifica como e8408d. También, presenta una altitud de 15700 pies lo que equivale a 4786.36 metros: va a una velocidad de 353 kts (Nudos), lo que significa que va a unos 653.7 Km/h. Los nudos son una medida de velocidad que utilizan millas náuticas por hora. Esta unidad se utiliza por lo general en el transporte marítimo y en la aviación. Así mismo nos proporciona sus coordenadas geográficas, siendo su latitud de -3.941 y longitud -79.265. Los mensajes enviados fueron alrededor de 50 y fue visto durante 24 segundos.

6.2. PRÁCTICA #2

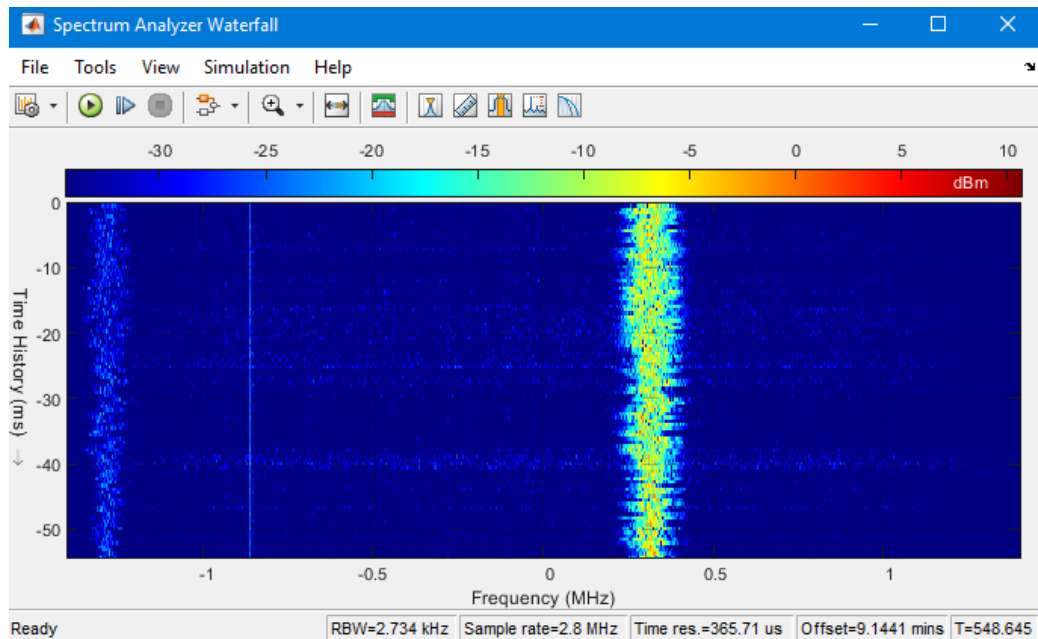
Tema: Sintonizador de frecuencias GSM – FM utilizando el dispositivo RTL – SDR

Figura 95. Actividad GSM en una frecuencia de 936.85 MHz



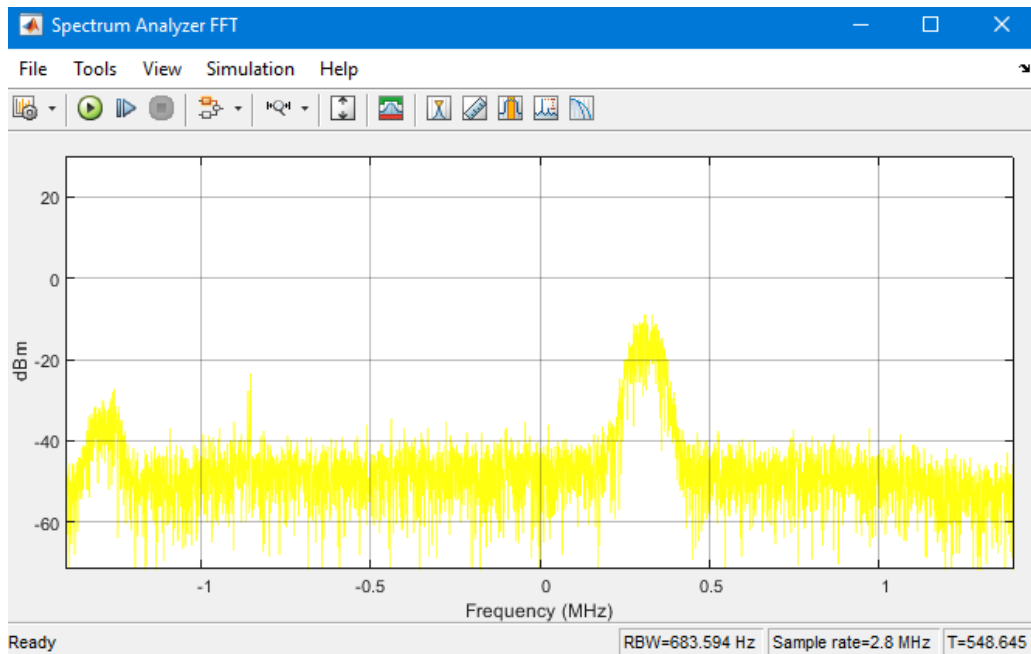
Fuente: Elaboración propia

Figura 96. Analizador de espectro en cascada sintonizando la frecuencia GSM 936 MHz



Fuente: Elaboración propia

Figura 97. Analizador de espectro de FFT en la frecuencia GSM 936 MHz

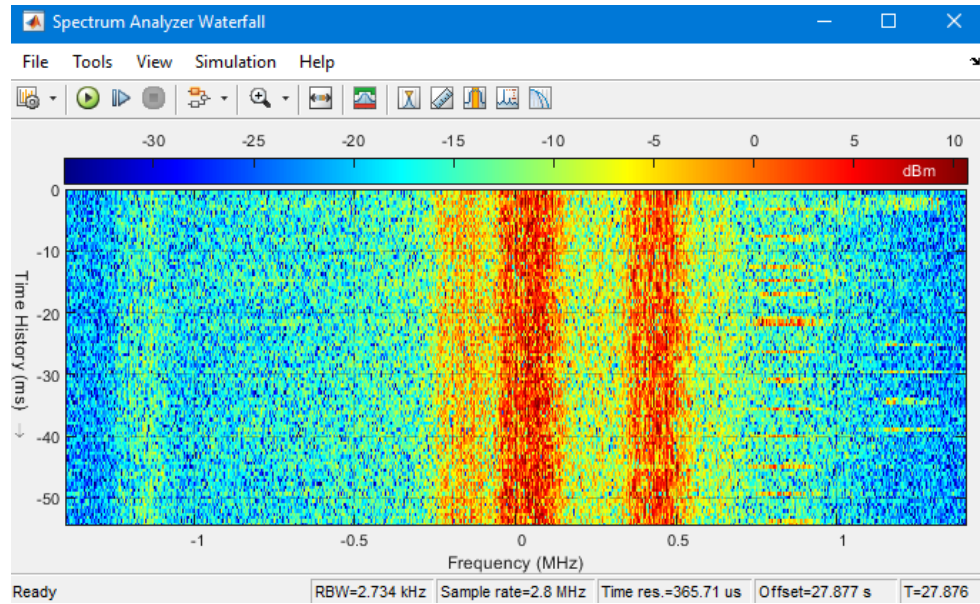


Fuente: Elaboración propia

Figura 98. Actividad GSM en la frecuencia 869.36 MHz

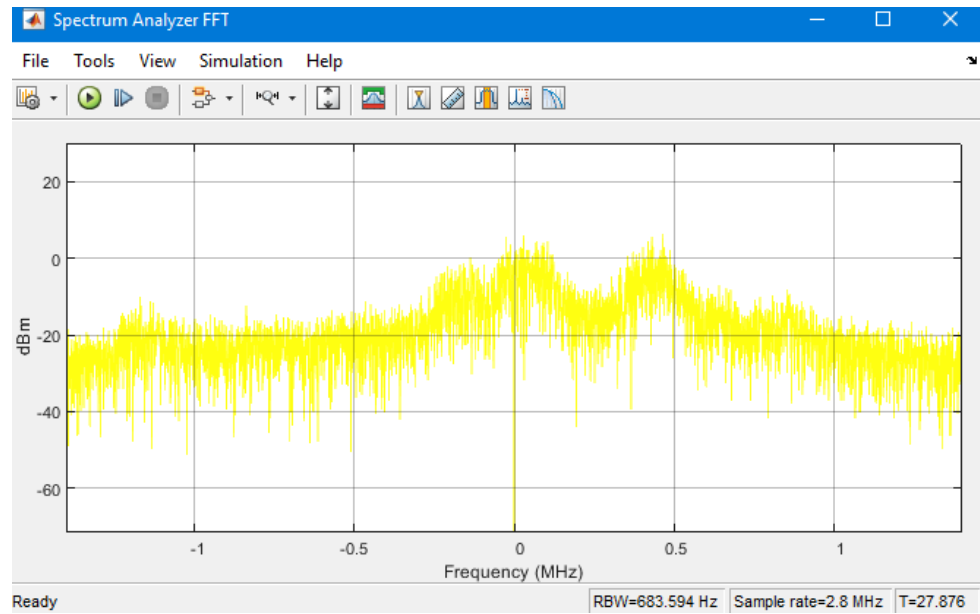
Fuente: Elaboración propia

Figura 99. Analizador de espectro en cascada sintonizando la frecuencia GSM 869 MHz



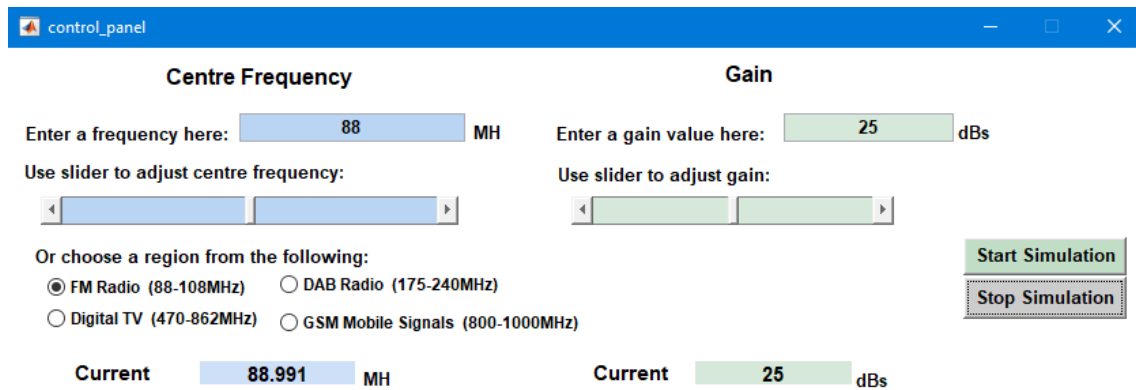
Fuente: Elaboración propia

Figura 100. Analizador de espectro de FFT en la frecuencia GSM 869 MHz



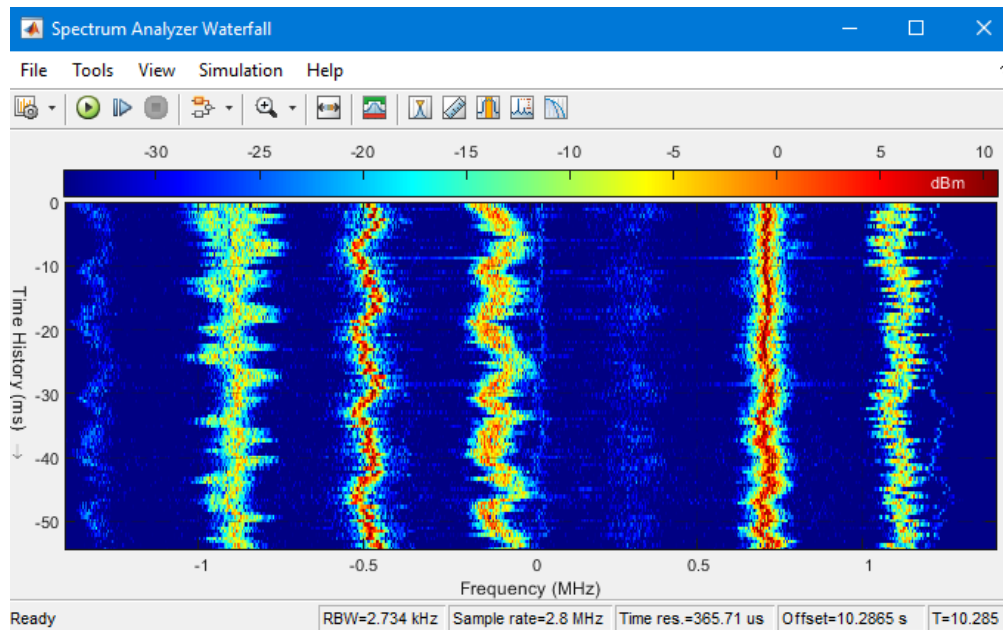
Fuente: Elaboración propia

Figura 101. Actividad Radio FM en la frecuencia de 88.9 MHz



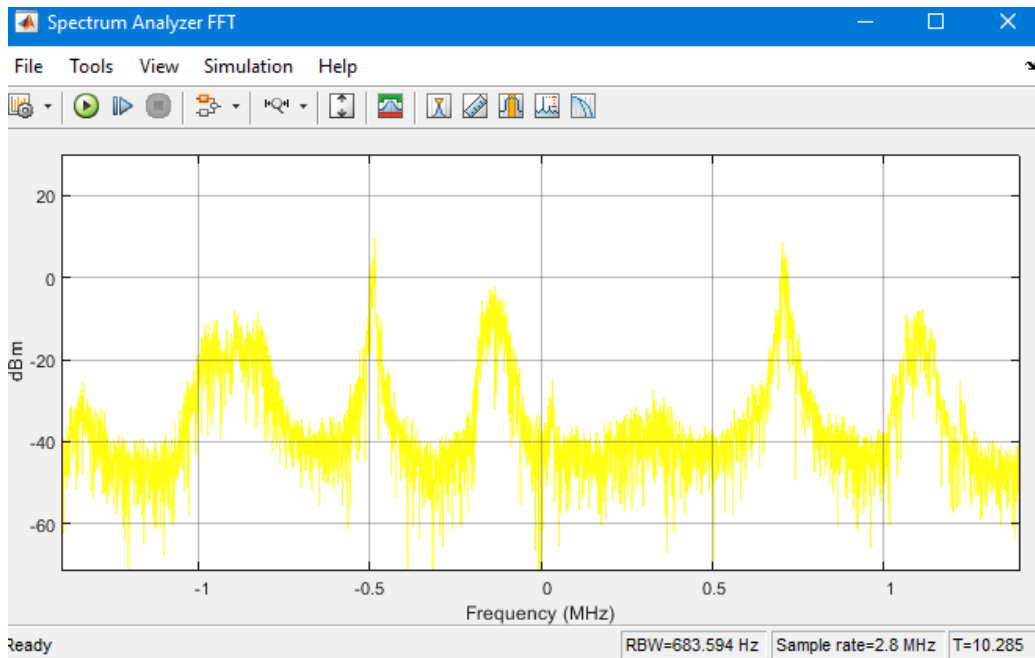
Fuente: Elaboración propia

Figura 102. Analizador de espectro en cascada sintonizando la frecuencia FM 88.9 MHz



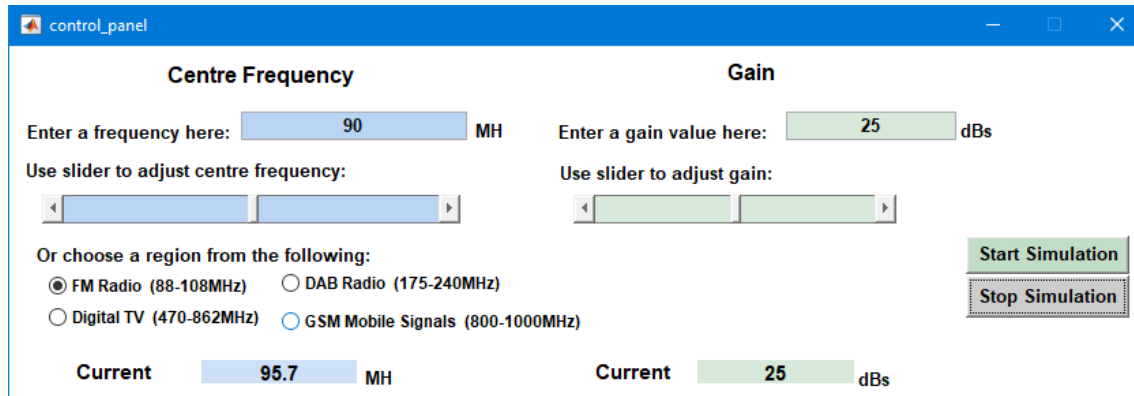
Fuente: Elaboración propia

Figura 103. Analizador de espectro de FFT en la frecuencia FM 88.9 MHz



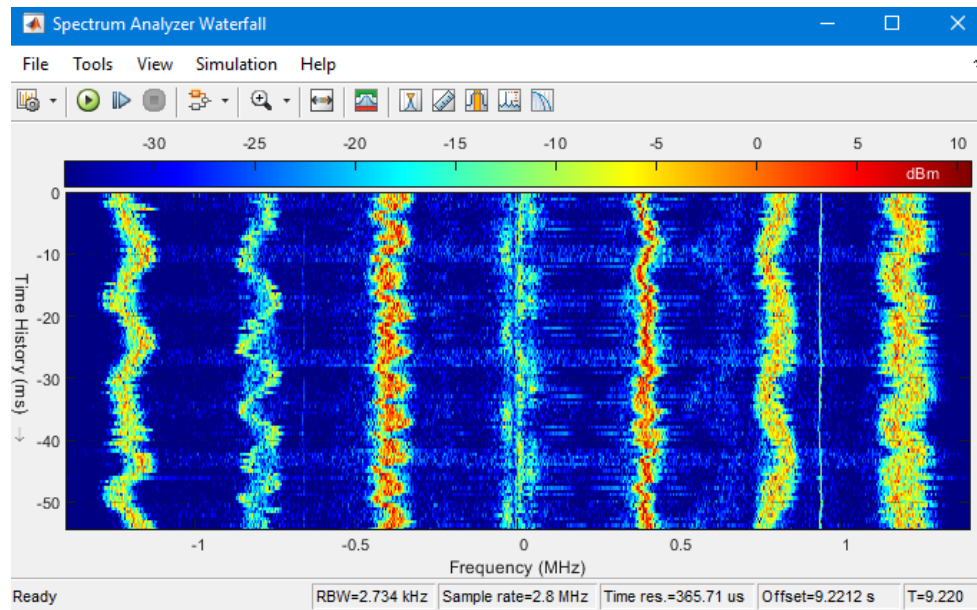
Fuente: Elaboración propia

Figura 104. Actividad Radio FM en la frecuencia de 95.7 MHz



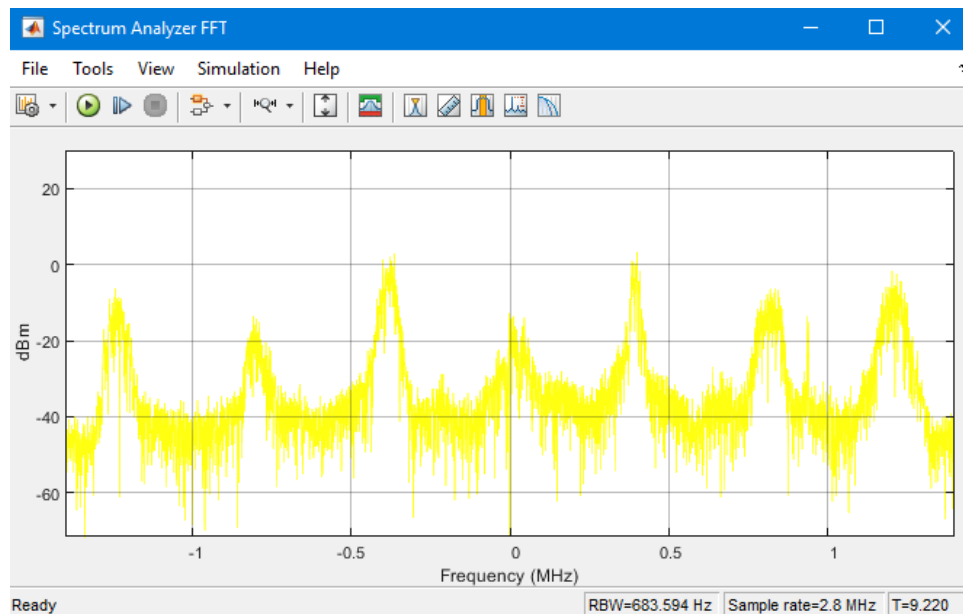
Fuente: Elaboración propia

Figura 105. *Analizador de espectro en cascada sintonizando la frecuencia FM 95.7 MHz*



Fuente: Elaboración propia

Figura 106. *Analizador de espectro de FFT en la frecuencia FM 95.7 MHz*



Fuente: Elaboración propia

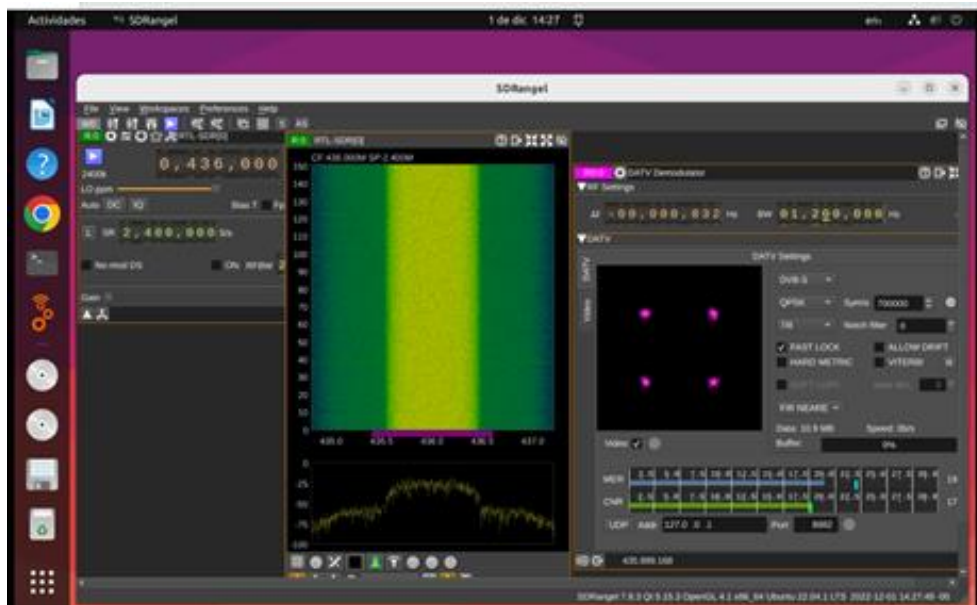
Los resultados mostrados indican que se ha utilizado una frecuencia de muestreo de 2.8 MHz se recomienda no exceder esta tasa de muestreo porque asegura precisión y confiabilidad en el espectro sin pérdida de información. Por cada actividad realizada tanto para GSM y FM se

visualizan tres gráficas, la primera corresponde al panel de control de frecuencia y ganancia la cual se puede ir variando con la finalidad de ir sintonizando señales a una frecuencia diferente, la segunda gráfica muestra un analizador de espectro, es decir el rango de actividad espectral en cascada. Mientras que la última sirve para mostrar la FFT de una señal recibida.

6.3. PRÁCTICA #3

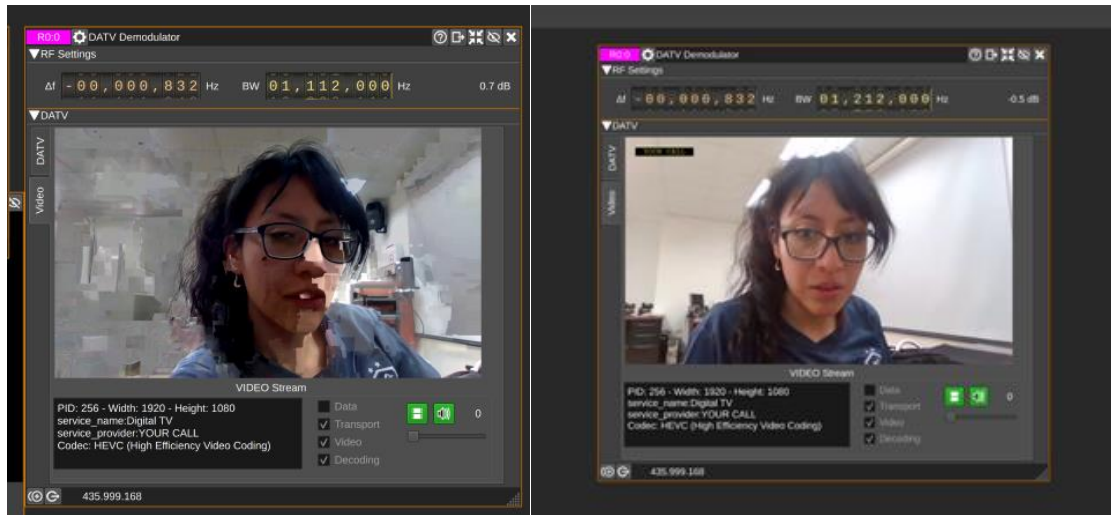
Tema: Transmisión de video DATV utilizando LimeSDR – Mini

Figura 107. Recepción de la señal de video mediante el dispositivo RTL-SDR utilizando el programa SDRangel y bajo la modulación QPSK.



Fuente: Elaboración propia

Figura 108. Calidad de recepción de la señal de video mediante el dispositivo RTL-SDR



Fuente: Elaboración propia

La franja amarilla verdosa (Ver Figura 107) demuestra que se está recibiendo la señal de video y audio a través del dispositivo RTL-SDR que proviene del dispositivo transmisor LimeSDR – Mini a una frecuencia de 436 MHz, utilizando un ancho de banda de 1.2 MHz. También se puede observar el tipo de modulación que se utilizó siendo la QPSK con una tasa de símbolos de 700000 Msps. Así mismo, presenta una frecuencia de muestreo configurada en 2.4 Msps. Se establece la "diezmación" en 1, ya que el demodulador DATV requiere datos sin procesar para hacer lo que hace.

6.4. PRÁCTICA #4

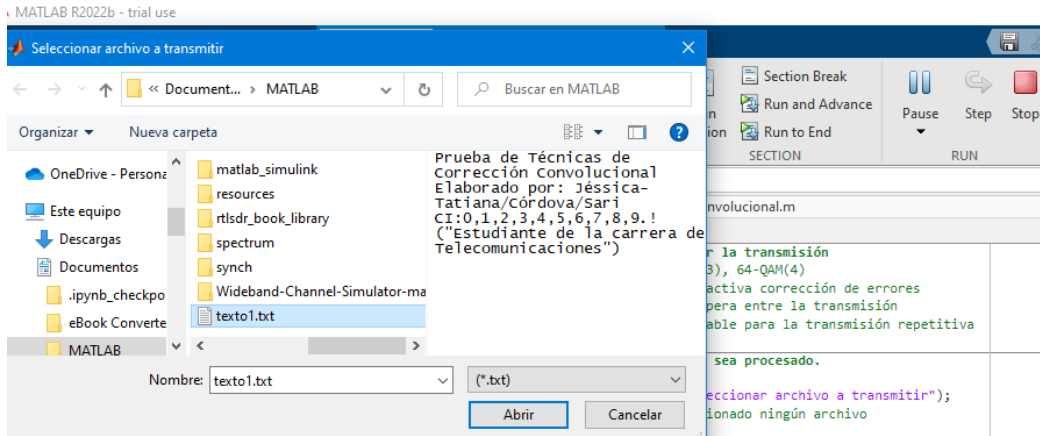
Tema: Transmisión de un archivo de texto con el dispositivo Adalm Pluto aplicando la técnica de corrección de errores Convolutional.

Figura 109. Sección para seleccionar el esquema de modulación digital

```
Editor - C:\Users\Usuario\Documents\MATLAB\Tecnica_Convolutional.m
Parametros_Tx_PLUTO.m  Tecnica_Convolutional.m  +
1      %% Creacion de variables para controlar la transmisión
2      Tipo_Mapeo=2;%BPSK(1),QPSK(2), 16-QAM(3), 64-QAM(4)
3      Correccion_Errores=20;    %Activa y desactiva corrección de errores
4      Intervalo_Tiempo=3;      %Tiempo que espera entre la transmisión
5      tx_Real_Time=20;         %Setea la variable para la transmisión repetitiva
6
```

Fuente: Elaboración propia

Figura 110. Sección para seleccionar el archivo de texto que será transmitido por el dispositivo Adalm PlutoSDR



Fuente: Elaboración propia

Figura 111. Ejecución de la técnica convolucional estableciendo conexión con el dispositivo Adalm Pluto SDR para la transmisión del archivo de texto.

```

Command Window
>> Tecnica_Convolucional
Datos Serializados
Codificación Convolucional Agregada
Símbolos OFDM creados
IFFT implementada
Prefijo cíclico (PC) añadido
Preambulo 802.11a añadido
## Establishing connection to hardware. This process can take several seconds.
Datos transmitidos
Datos transmitidos
Datos transmitidos
Datos transmitidos
Datos transmitidos
Datos transmitidos
Datos transmitidos
Datos transmitidos
Datos transmitidos
Datos transmitidos
Datos transmitidos
Datos transmitidos
Datos transmitidos
fx

```

Fuente: Elaboración propia

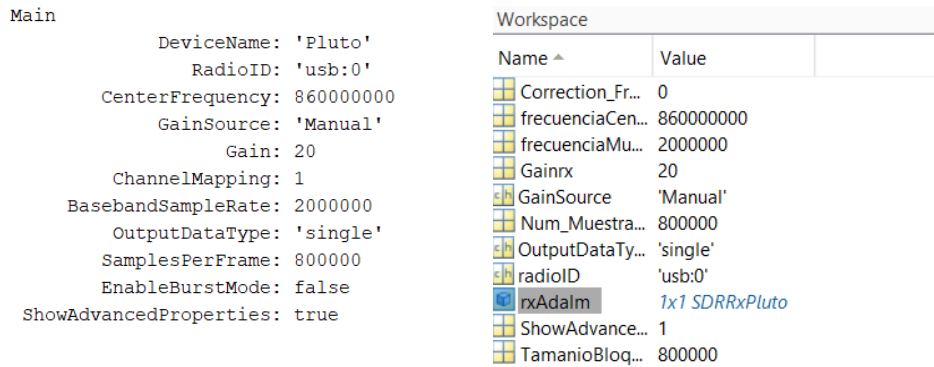
Al ejecutar esta práctica, primero se tuvo que elegir el mapeo con números tal como lo indica la primera gráfica en la segunda línea de código, siendo 1 para la modulación BPSK, 2 para QPSK, 3 para 16-QAM y 4 para 64-QAM. Se comprueba además que se cumple con todo el

procedimiento del script “Tecnica_convolutacional” descrita en la sección de actividades por desarrollar, tal como seleccionar el archivo que se creó para la transmisión. En la última gráfica también se observa cómo los pasos de serializar los datos, la codificación convolutacional, la creación de símbolos OFDM, el uso de la transformada Inversa de Fourier, la inserción del prefijo cíclico y del preámbulo 802.11^a se han completado satisfactoriamente. Y por ello, se pudo establecer conexión con el dispositivo Adalm Pluto para que pueda transmitir en tiempo real.

6.5. PRÁCTICA #5

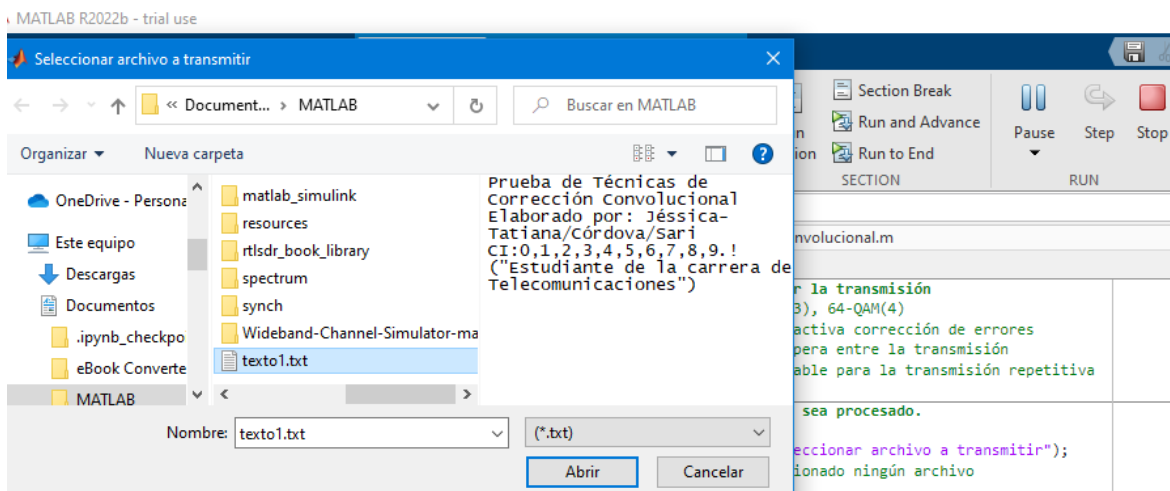
Tema: Recepción de un archivo de texto con el uso del dispositivo Adalm Pluto implementando la técnica de corrección de errores Convolutacional.

Figura 112. Creación del objeto para interactuar con la radio Adalm Pluto SDR



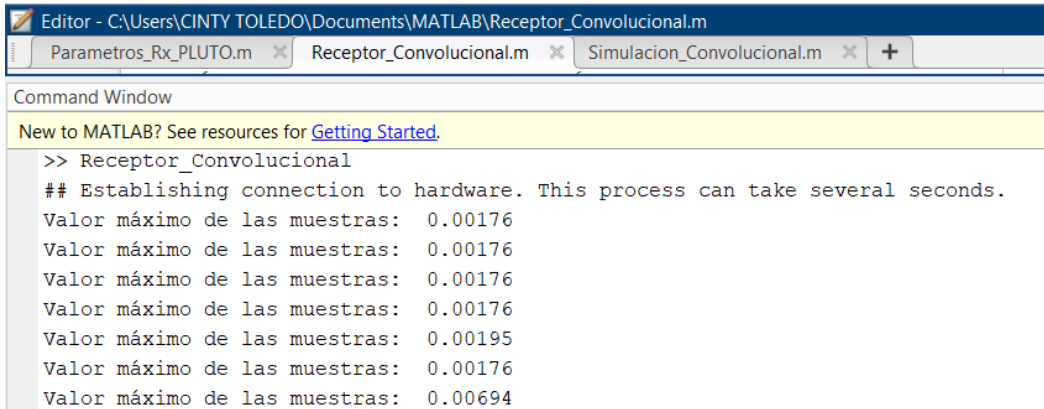
Fuente: Elaboración propia

Figura 113. Forma para seleccionar el archivo de texto para medir la BER



Fuente: Elaboración propia

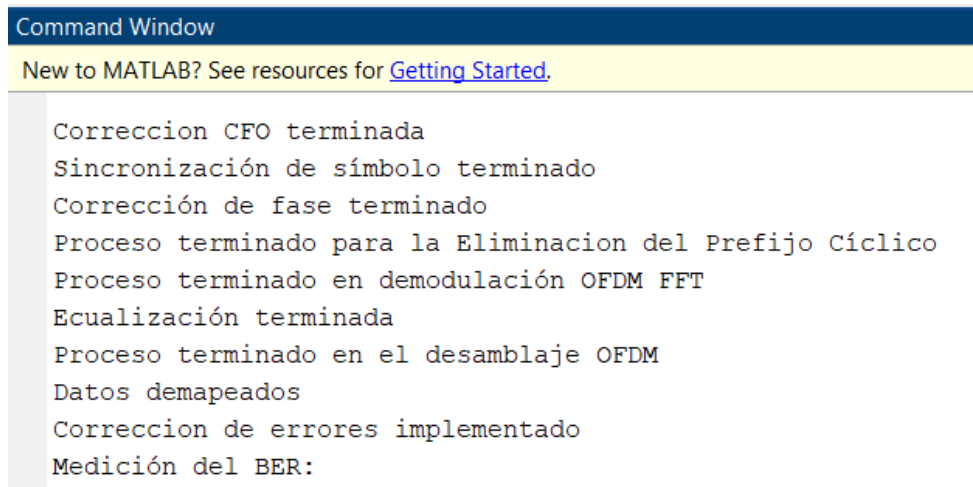
Figura 114. *Estableciendo conexión con el hardware de transmisión para la recepción de datos*



```
Editor - C:\Users\CINTY TOLEDO\Documents\MATLAB\Receptor_Convolucional.m
Parametros_Rx_PLUTO.m x Receptor_Convolucional.m x Simulacion_Convolucional.m x +
Command Window
New to MATLAB? See resources for Getting Started.
>> Receptor_Convolucional
## Establishing connection to hardware. This process can take several seconds.
Valor máximo de las muestras: 0.00176
Valor máximo de las muestras: 0.00176
Valor máximo de las muestras: 0.00176
Valor máximo de las muestras: 0.00176
Valor máximo de las muestras: 0.00195
Valor máximo de las muestras: 0.00176
Valor máximo de las muestras: 0.00694
```

Fuente: Elaboración propia

Figura 115. *Ejecución del proceso de recepción aplicando la técnica convolucional*



```
Command Window
New to MATLAB? See resources for Getting Started.
Correccion CFO terminada
Sincronización de símbolo terminado
Corrección de fase terminado
Proceso terminado para la Eliminación del Prefijo Cíclico
Proceso terminado en demodulación OFDM FFT
Ecuación terminada
Proceso terminado en el desamblaje OFDM
Datos demapeados
Correccion de errores implementado
Medición del BER:
```

Fuente: Elaboración propia

Figura 116. Recuperación del texto transmitido

```
Tipo de mapeo QPSK
Bits Errados: 7/1336
Cantidad de ruido AWGN: 6

Texto_Original =

'Prueba de Técnicas de Corrección Convolutcional
Elaborado por: Jéssica-Tatiana/Córdova/Sari
CI:0,1,2,3,4,5,6,7,8,9.!
("Estudiante de la carrera de Telecomunicaciones")
'

Texto_Recuperado =

'Prueba de Técnicas de Corrección Convoluch/nal
Elaborado por: Jéssica-Tatiana/Córdova/Sari
CI:0,1,2-3,4,5<6,7,8,9.!
("Estudiante fe la carrera de Telecomuoicaciones")
'

Proceso terminado en la recuperación del texto
fx >>
```

Fuente: Elaboración propia

Tras complementar esta práctica con la anterior se pudo obtener el texto recuperado del texto original que en un inicio se tenía previsto transmitir. Los resultados indican el tipo de mapeo que se utilizó en la demodulación siendo la elegida QPSK, mostrando la cantidad de bits errados siendo 7 bits de 1336 y la cantidad de ruido que se agregó. Todo el proceso que se realizó en la recuperación del texto descrita en actividades por desarrollar se ejecutó sin ningún inconveniente.

7. Discusión

En la implementación de cada una de las guías de prácticas se muestra que los módulos SDR usados tienen diferente rendimiento y por ende se pueden utilizar en distintos campos de aplicación. El hardware RTL-SDR permite recibir señales del rango de frecuencia 25 MHz a 1.75 GHz, la capacidad del módulo LimeSDR – Mini es de 10 MHz a 3.5 GHz y el de Adalm PlutoSDR cuenta con una capacidad mayor que los anteriores siendo de 325 a 3.8 GHz. El primer módulo mencionado es el más económico y éste puede actuar solo como receptor a diferencia de los dos restantes que actúan como transmisor y receptor. Se puede resaltar el siguiente análisis por las prácticas que se realizaron.

Para el desarrollo de la primera práctica es indispensable identificar el tipo de programas que permiten decodificar las señales abiertas ADS-B que emiten los aviones; por tal motivo, se selecciona Dump1090 ya que permite visualizar las aeronaves desde la página web de cualquier navegador, las señales recibidas se observan en un mapa propio a través de Localhost:8080. Los enlaces de datos ADS-B funcionan a 1090 MHz o 978 MHz, siendo la primera frecuencia con la que se trabajó utilizando una antena telescópica que ayuda a cubrir la frecuencia requerida (71 MHz – 1GHz). Hay que tener en cuenta que la frecuencia de 978MHz reduce la congestión de mensajes en 1090MHz. Sin embargo, se utiliza la frecuencia de 1090 MHz porque nuestro sistema implementado actúa como receptor. ADS-B en 1090 MHz es esencialmente un transpondedor Modo S modificado. Es decir, un ADS-B que opera en frecuencias UHF y se necesita de línea de visión ya que sólo recibirá aeronaves a la vista de la antena.

Para la segunda práctica el implementar el diagrama de bloques que corresponde a un sistema SDR se puede observar cómo cambian las señales GSM tanto en uplink configurada en el panel de control de frecuencias a 936.85 MHz, y así mismo en downlink 869.36. Las frecuencias configuradas para recibir señales FM (88.9 y 95.7) MHz también muestran cómo aparecen muchas estaciones que el módulo RTL va detectando.

Para la tercera práctica, la transmisión de señal de video y audio se obtuvo al utilizar el dispositivo LimeSDR Mini junto con el programa DATV – Express. La calidad de transmisión va depender del tipo de dispositivo que se utilice para capturar las señales de audio y video. El tipo de códec que se configuró fue el H.264, el cual permite mantener la calidad del MPEG-2 usando menores velocidades de transmisión y por lo tanto necesitando menos ancho de banda, siendo de

1,2 MHz. A pesar de ello, se pudo notar que hubo retrasos en el tiempo de transmisión de 3 a 5 segundos, lo cual puede ser por la potencia que dispone el dispositivo LimeSDR alcanzando una potencia máxima de 10 dBm. Otro factor importante fue elegir la constelación QPSK que permite ocupar menos ancho de banda y además es compatible con Receptores TV Satélite Digitales DVB-S el cual se está utilizando.

Para la cuarta práctica, ésta nos da a conocer en profundidad la técnica de corrección de errores llamada Convolutional, pues ésta opera bajo un sistema con memoria, esto se debe a que en la salida depende de los bits previos. Aunque esta técnica de codificación se pudo representar a través de diferentes formas como el diagrama de estados, registros, diagramas de árbol, se opta por el diagrama de Trellis por ser el más utilizado y por la facilidad que Matlab ofrece gracias a la función `polytrellis`. Así mismo, el script creado hace mención a algunos esquemas de modulación que pueden emplearse y de otros procesos que intervienen en la transmisión que esta técnica de codificación convolutional implica.

Finalmente, la quinta práctica se complementa con la anterior, pues ésta corresponde a recepcionar un archivo de texto mediante el dispositivo Adalm Pluto es interesante comprender todo el procedimiento que se tuvo que realizar. Para establecer la conexión con el transmisor es necesario configurar los parámetros de hardware. Es importante, recibir la información de manera precisa, evitando problemas de interferencia intersimbólica y de portadoras mediante la inserción de prefijo cíclico y los intervalos de guarda. Otro proceso importante a mencionar es la estimación del desplazamiento de la frecuencia portadora (CFO) y la estimación del canal que hacen uso de las secciones corta y larga del preámbulo en la norma IEEE 802.11a (estándar empleado en la práctica). Pues en sí, aunque parezca sencillo recibir datos detrás de ello hay una serie de procesos. A manera resumida se puede decir que el receptor OFDM identificará primero el inicio del paquete, corregirá las desviaciones de frecuencia, corregirá los efectos del canal y también eliminará las distorsiones de fase.

8. Conclusiones

- ✓ En el presente Trabajo de Integración Curricular se desarrolló un conjunto de prácticas con material relacionado a las Comunicaciones Digitales, las mismas que servirán de guía a los docentes que deseen introducir a los estudiantes de la carrera de Telecomunicaciones de la UNL, sobre los fundamentos básicos de la Radio Definida por Software.
- ✓ Mediante el uso de diferentes herramientas informáticas tales como: DATV Express, SDRangel, Matlab – Simulink, se pudo determinar el entorno de programación a utilizar para la correcta comunicación entre el host y los módulos SDR: RTL-SDR, LimeSDR Mini, Adalm PlutoSDR.
- ✓ Con las simulaciones de las prácticas se logró mostrar que SDR proporciona la flexibilidad en el diseño de sistemas de comunicaciones pues cualquier sistema de radiofrecuencia puede ser implementado con previos conocimientos en el procesamiento de las señales.
- ✓ Con la ayuda de los algoritmos creados ya sea mediante bloque o líneas de código se pudo ejecutar cada una de las prácticas propuestas siempre y cuando se haya configurado de manera apropiada los parámetros de los módulos SDR.
- ✓ Para finalizar con el presente trabajo de integración curricular, también se efectuó la elaboración de guías de laboratorio las cuales detallan en orden todo el proceso que se tuvo que realizar por cada una de las prácticas.

9. Recomendaciones

- ✓ En base a los planteamientos y observaciones que permiten la culminación de este proyecto, se recomienda:
- ✓ Revisar las especificaciones técnicas de los módulos SDR que se van a utilizar para que posteriormente los parámetros de operación sean configurados de la manera apropiada y funcionen correctamente.
- ✓ Manipular con precaución el equipo de hardware SDR con la finalidad de evitar inconvenientes que podrían afectar el funcionamiento de los mismos.
- ✓ Utilizar dos computadoras para las prácticas que implican una interacción como transmisor y receptor para mejores resultados ya que requieren de un gran procesamiento computacional.
- ✓ Es necesario ejecutar los códigos tal como se muestran si desea implementar alguna de las prácticas debido a cambios que se realizaron en algunos de ellos con la finalidad de evitar ventanas de errores que podrían modificar de alguna manera el resultado.

10. Referencias Bibliográficas

- Álvarez, G. A. (2014). *Impacto de la distorsión armónica de amplificadores de potencia sobre la distorsión de señales moduladas digitalmente*.
- Artés Rodríguez, A., Pérez González, F., Cid Sueiro, J., López Valcarce, R., Mosquera Nartallo, C., & Pérez Cruz, F. (2012). *Comunicaciones Digitales*.
https://www.tsc.uc3m.es/~antonio/libro_comunicaciones/El_libro_files/comdig_artes_perez.pdf
- Cevallos Durán, J. M. (2022). *Estudio de factibilidad de un radioenlace implementado con equipos de Radio Definido por Software*.
- doxygen. (2019, August 2). *SDRAngel*. <https://rgetz.github.io/sdrangel/>
- Fernández, R., & Enriquez, C. (2015, November 1). *Sistemas de Comunicaciones utilizando GNU Radio*. <https://www.boletin.upiita.ipn.mx/index.php/ciencia/639-cyt-numero-51/1191-sistemas-de-comunicaciones-utilizando-gnu-radio>
- García Algora, M. G. (2011). *Radio Definido por Software usando MATLAB*.
<https://dspace.uclv.edu.cu/bitstream/handle/123456789/4729/Carlos%20Manuel%20Garc%C3%ADa%20Algora.pdf?sequence=1&isAllowed=y>
- Lakshan. (2020, August). *What is SDR and what can you do with SDR?* .
<https://www.seeedstudio.com/blog/2020/05/25/what-is-sdr-and-what-can-you-do-with-sdr/>
- Mahyar, T. D., & Sokolov, V. (2018). *PENETRATION TESTS FOR BLUETOOTH LOW ENERGY AND ZIGBEE USING THE SOFTWARE DEFINED RADIO*.
<https://arxiv.org/ftp/arxiv/papers/1902/1902.08595.pdf>
- Manchado Fernández, J. R. (2014). *View of Software Defined Radio: Basic Principles and Applications*. <https://revistas.uptc.edu.co/index.php/ingenieria/article/view/3160/2852>
- MATLAB & Simulink[®]. (2023). *Communications Toolbox* .
<https://la.mathworks.com/products/communications.html>
- Micolau, F., & Tarrés, F. (2021). *Multiplexación y sistemas de acceso múltiple*. 10–15.

- Miranda Torres, P. R. (2015). *Modulacion, ask, fsk, psk y qam*.
<https://es.slideshare.net/PabloRubenMirandaPonce/9-modulacion-ask-fsk-psk-y-qam>
- Pinar Domínguez, I., & Murillo Fuentes, J. J. (2011). *Laboratorio de Comunicaciones Digitales Radio Definida por Software* (1st ed.).
<https://personal.us.es/murillo/docente/Libros/LibroSDRV7USv8.pdf>
- Princy A. J. (2021, May 13). *Significant Applications of Software Defined Radio Systems*.
ResearchDive. <https://www.researchdive.com/blog/all-you-need-to-know-about-software-defined-radio>
- Quinde, A. G. (2017). *Implementación de Sistemas de Modulaciones digitales en el NI Elvis II+*.
<https://n9.cl/5hj89>
- Radioescucha. (2021, July 14). *DUMP1090*. <https://elradioescucha.net/2021/07/14/dump1090-ahora-disponible-como-una-aplicacion-de-android/>
- Sánchez, C. R. (2019). *MULTIPLEXACION FDM, TDM Y CMD*.
<https://sites.google.com/site/carlosraulsan2987/home/tecnologias-inalambricas/unidad-1/multiplexacion-fdm-tdm-y-cmd>
- Serrano, W. A., Cortes, R. C., & Valdés, S. M. (2015). *Estudio de OFDMA en el enlace de bajada en LTE*. <https://n9.cl/vckdq>
- Tecnológico Nacional de México. (2020). *Fundamentos de Telecomunicaciones: Multiplexación*.
https://enlinea.zacatecas.tecnm.mx/pluginfile.php/24650/mod_resource/content/1/Unidad%20V.pdf

11. Anexos

Anexo 1 Certificado Traducción Resumen



Lic. Karina Yajaira Martínez Luzuriaga

LICENCIADA EN CIENCIAS DE LA EDUCACIÓN MENCIÓN INGLÉS

CERTIFICO:

Yo, Karina Yajaira Martínez Luzuriaga con cédula de identidad Nro. 1104902679, **Licenciada en Ciencias de la Educación Mención Inglés** por la Universidad Técnica Particular de Loja, con número de registro 1031-2022-2574017 en la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación, señalo que el presente documento es fiel traducción del idioma español al idioma inglés del resumen del Trabajo de Integración Curricular denominado **“Desarrollo de Prácticas de Comunicaciones Digitales utilizando Radio Definida por Software (SDR) para el Laboratorio de Telecomunicaciones de la FEIRNNR – UNL”** elaborado por la Srta. Jéssica Tatiana Córdova Sari, con cédula de identidad Nro. 1104832777, estudiante egresada de la carrera de Ingeniería en Telecomunicaciones de la Universidad Nacional de Loja.



Lic. Karina Yajaira Martínez Luzuriaga

C.I. 1104902679

REGISTRO SENESCYT N°: 1031-2022-2574017