

Proyecto Final Integrador

Ingeniería en Telecomunicaciones

Comunicación inalámbrica mediante FHSS basado en SDR

Autores: Agostini, Emmanuel. CyT-10262.
Yañez Iglesias, Juan Manuel. CyT-9300.

Tutor: Ing. Carmona, Leandro. UNSAM-ECyT

AÑO 2024

Introducción	1
Contexto del Proyecto	1
Objetivos del Proyecto	2
Anteproyecto	3
Acta constitutiva del proyecto	3
Work Breakdown Structure (WBS)	4
Gestión de tiempos	6
Marco Teórico	9
Comunicación Half Duplex	9
Modulación de una señal	10
Modulaciones Analógicas	10
Modulaciones Digitales	18
Frequency Hopping Spread Spectrum (FHSS)	23
Teorema de muestreo Nyquist-Shannon	27
Señales analógicas y digitales	27
Señales discretas	29
Transformada rápida de Fourier (FFT)	30
Transmisión inalámbrica	31
Problemas comunes	31
Distancia según frecuencia	31
Pérdida de Trayectoria en Espacio Libre	32
GNU Radio	33
Radio Definido por Software	34
SDRSharp	37
Hardware SDR	38
Metodología	41
Simulación en GNU Radio	41
Preparación del laboratorio de trabajo	41
Transmisor	42
Receptor	44
Implementación en hardware (audio)	45

Pruebas de funcionamiento	45
Transmisor en Hardware	47
Receptor en Hardware	49
Separación del transmisor y receptor en Hardware	50
Análisis de la señal recibida (FFT)	53
Ejemplo de funcionamiento de receptor	56
Implementación en hardware (texto e imagen)	57
Transmisión de una imagen	59
Pruebas de distancia	61
Audio	61
Tx Hack Rf - Rx USRP	61
Tx USRP- Rx Hack Rf	62
Texto e imagen	63
Trabajo futuro	64
Encriptación	64
Protocolo de comunicación	64
Transmitir y recibir archivos de mayor tamaño	64
Etapa de potencia	65
Conclusiones	66
Agradecimientos	69
Referencias	71
Anexos	74

Introducción

A continuación se explicará cómo surge la idea de trabajar en un proyecto vinculado a la tecnología Frequency Hopping Spread Spectrum, indicando los usos cotidianos del mismo. A su vez, se plantean los objetivos a cumplir a lo largo del informe.

Contexto del Proyecto

La comunicación inalámbrica ha evolucionado considerablemente en los últimos 20 años, convirtiéndose en un elemento vital en numerosas aplicaciones y entornos. El siguiente proyecto se basa en la comunicación inalámbrica mediante Frequency Hopping Spread Spectrum (FHSS).

Este proyecto surge como resultado de una propuesta del director de la carrera (Ing. Rodolfo Salvatore), quien identificó la relevancia y el potencial innovador de explorar las posibilidades de la comunicación FHSS. Una gran motivación inicial para desarrollar la investigación, fue la oportunidad de profundizar en un tema de gran importancia tecnológica y su potencial para contribuir al avance en sistemas de comunicación inalámbrica.

La fortaleza de este proyecto radica en su capacidad para facilitar la transmisión segura y confiable de datos de forma inalámbrica, incluso en condiciones adversas y entornos ruidosos. La robustez inherente de la modulación FHSS hace que sea especialmente adecuada para aplicaciones donde la integridad de la comunicación es crítica.

La investigación en comunicación FHSS tiene una relevancia significativa en el panorama tecnológico actual. La capacidad de establecer comunicaciones seguras y confiables en entornos inalámbricos es fundamental en áreas como la Internet de las cosas (IoT), comunicaciones militares, sistemas de control industrial, entre otros. El avance en este campo no solo abre nuevas oportunidades en términos de aplicaciones prácticas, sino que también contribuye al desarrollo tecnológico global.

Objetivos del Proyecto

Los objetivos del proyecto final integrador consisten en establecer una comunicación half duplex entre dos placas de desarrollo con capacidad de transmisión y recepción de señales de radiofrecuencias (RF). Esta comunicación se realizará utilizando la tecnología FHSS. Para demostrar la comunicación se transmitirá un audio, un texto y una imagen, utilizando modulación analógica y digital. A través de este proyecto, buscamos no solo demostrar la viabilidad técnica de esta forma de comunicación, sino también explorar sus potenciales aplicaciones. Llevando a cabo, en primera instancia, simulaciones con la tecnología SDR (Radio Definido por Software). Posteriormente, se implementó en dispositivos físicos.

En resumen, se relató los inicios del proyecto y el porqué nos despertó interés la tecnología. Luego se planteó hacia dónde queríamos llegar, teniendo en cuenta las limitaciones del hardware disponible.

En el próximo capítulo, detallaremos el acta constitutiva del proyecto con un “Work Breakdown Structure” que nos permitió organizar las tareas. Luego, nos adentraremos más en temas teóricos.

Anteproyecto

Acta constitutiva del proyecto

Título del proyecto					
Comunicación inalámbrica mediante FHSS basado en SDR					
Estudiantes		Legajos		Firmas	
Emmanuel Agostini Juan Manuel Yañez		CYT-10262 CyT-9300			
Fecha de inicio del proyecto			Fecha tentativa de finalización del Proyecto		
01/04/2024			30/09/2024		
Objetivo General					
Diseñar, implementar y evaluar un sistema de comunicación inalámbrica half duplex utilizando modulación FHSS (Frequency Hopping Spread Spectrum) entre dos dispositivos. Implementando tecnología SDR (Software Defined Radio). Este sistema propone mejorar la seguridad de la comunicación mediante el uso de FHSS. Además, es confiable en entornos de alta interferencia por su forma de transmitir.					
Objetivos Específicos					
<ul style="list-style-type: none">• Implementar el modelo en GNU-Radio, simulando todas las etapas de transmisor y receptor.• Implementar los transmisores y receptores en las placas provistas (USRP B210 y HackRf One.)• Transmisión y recepción de diferentes archivos.					
Diagnóstico					
La técnica de salto de frecuencia se presenta como una solución eficaz para mitigar interferencias y mejorar la resistencia a la interceptación no autorizada. El proyecto final de carrera se centra en la implementación de un sistema de comunicación FHSS entre dos placas de desarrollo. Inicialmente, se realizará una simulación utilizando GNU-Radio para validar el diseño y los algoritmos de comunicación. Posteriormente, se implementará la solución en las placas, evaluando su desempeño en un entorno de laboratorio. Este enfoque proporcionará experiencia práctica en el uso de herramientas y tecnologías SDR.					
Justificación o propósito del Proyecto (Aportes y resultados esperados)					
Se implementa la comunicación mediante la modulación FHSS debido a su alta seguridad y eficiencia. Esta modulación implementa saltos en las múltiples frecuencias, lo cual dificulta la intersección de la información. Además, tiene mayor resistencia a interferencias. Por los motivos anteriores se elige trabajar con esta técnica para comunicar los dispositivos dentro de un laboratorio y que quede implementada la comunicación para futuros proyectos de la universidad.					
Dirigido por:		Leandro Carmona		Firma:	
Aprobado por:				Firma:	

Work Breakdown Structure (WBS)

1. Creación del entorno de ejecución

- 1.1. Definir software de virtualización.
- 1.2. Definir Sistema operativo y versión.
- 1.3. Habilitar puertos 3.0.
- 1.4. Instalación del software GNU Radio.
- 1.5. Instalación del SDR Sharp.

2. Investigación del funcionamiento del dispositivo USRP B210

- 2.1. Investigación de las características del dispositivo.
- 2.2. Investigación de la implementación en GNU Radio.

3. Implementación de la tecnología FHSS en GNU Radio simulación

- 3.1. Armado del transmisor.
 - 3.1.1. Generación del código Python para los saltos en frecuencia.
- 3.2. Armado del receptor.
- 3.3. Transmitir y recibir un audio.
- 3.4. Documentar pruebas del funcionamiento.

4. Implementación del hardware

- 4.1. Pruebas con USRP B210 y Dongle RTL.
- 4.2. Armado del transmisor.
- 4.3. Armado del receptor.
- 4.4. Transmitir y recibir un audio.
- 4.5. Documentar pruebas del funcionamiento.

5. Generar evidencias del funcionamiento

- 5.1. Subir videos de prueba a un drive.
- 5.2. Subir audios de prueba a un drive.

6. Transmisión de diferentes archivos

- 6.1. Transmitir y recibir un documento de texto.
- 6.2. Preparación de scripts para procesar la imagen transmitida.
- 6.3. Transmitir y recibir una imagen.
- 6.4. Documentar pruebas del funcionamiento.

7. Investigación del funcionamiento del dispositivo HackRF One

- 7.1. Investigación de las características del dispositivo.
- 7.2. Investigación de la implementación en GNU Radio.

8. Separación del transmisor y receptor

- 8.1. Instalación del dispositivo HackRF One en una computadora diferente
- 8.2. Pruebas de transmisión y recepción.
- 8.3. Documentar pruebas de funcionamiento.

9. Pruebas de laboratorio

- 9.1. Investigación y realización de pruebas de distancia y frecuencia

10. Presentación del proyecto

- 10.1 Redacción de un informe con conclusiones
- 10.2 Creación de una presentación formato PowerPoint.
- 10.3 Entrega del informe y el PowerPoint.

Gestión de tiempos

Tarea	Tiempo (Horas)
Creación del entorno de ejecución	12
1.1. Definir software de virtualización.	3
1.2. Definir Sistema operativo y versión.	3
1.3. Habilitar puertos 3.0.	2
1.4. Instalación del software GNU Radio.	2
1.5. Instalación del SDR Sharp.	2
Investigación del funcionamiento del dispositivo USRP B210	16
2.1. Investigación de las características del dispositivo.	8
2.2. Investigación de la implementación en GNU Radio.	8
Implementación de la tecnología FHSS en GNU Radio simulación	43
3.1. Armado del transmisor.	12
3.1.1. Generación del código Python para los saltos en frecuencia.	12
3.2. Armado del receptor.	12
3.3. Transmitir y recibir un audio.	5
3.4. Documentar pruebas del funcionamiento.	2
Implementación del hardware	41
4.1. Pruebas con USRP B210 y Dongle RTL.	6
4.2. Armado del transmisor.	14
4.3. Armado del receptor.	14

4.4. Transmitir y recibir un audio.	5
4.5. Documentar pruebas del funcionamiento.	2
Generar evidencias del funcionamiento	2
5.1. Subir videos de prueba a un drive.	1
5.2. Subir audios de prueba a un drive.	1
Transmisión de diferentes archivos	66
6.1. Transmitir y recibir un documento de texto.	24
6.2. Preparación de scripts para procesar la imagen transmitida.	8
6.3. Transmitir y recibir una imagen.	30
6.4. Documentar pruebas del funcionamiento.	4
Investigación del funcionamiento del dispositivo HackRF One	16
7.1. Investigación de las características del dispositivo.	8
7.2. Investigación de la implementación en GNU Radio.	8
Separación del transmisor y receptor	10
8.1. Instalación del dispositivo HackRF One en una computadora diferente	2
8.2. Pruebas de transmisión y recepción.	6
8.3. Documentar pruebas de funcionamiento.	2
Pruebas de laboratorio	4
9.1. Investigación y realización de pruebas de distancia y frecuencia	4
Presentación del proyecto	26
10.1 Redacción de un informe con conclusiones	20

10.2 Creación de una presentación formato PowerPoint.	6
10.3 Entrega del informe y el PowerPoint.	-
Total	236

El tiempo en horas es aproximado, se tiene que tener en cuenta que las horas no son de corridas. Por motivos personales, profesionales y académicos existieron lapsos de tiempo en donde no se avanzó con el proyecto.

Marco Teórico

A lo largo del capítulo, se expondrá la teoría aplicada al proyecto. Los temas a explicar son: Modos de comunicaciones existentes, la importancia de modular una señal, que es FHSS, el teorema de Nyquist, formas de procesar una señal y la transformada de Fourier. Todos los conocimientos teóricos fueron adquiridos a lo largo de la carrera, profundizando cada uno de ellos en este proyecto.

Esto nos permitió interpretar ciertos aspectos que no son evidentes al realizar la transmisión. Fue fundamental la comprensión de estos conceptos a la hora de depurar los distintos errores que surgieron en el desarrollo de este proyecto.

Comunicación Half Duplex

Existen tres métodos principales de lograr la transmisión de datos entre dos equipos. Estas son, Simplex, Half Duplex (HD), Full Duplex (FD). El primero, solo es posible transmitir la información en un solo sentido, existe un transmisor (Tx) y un receptor (Rx). Un ejemplo son las estaciones de radio comerciales donde con un solo transmisor se comparte la información con varios receptores ubicados en distintos lugares.

El segundo es la comunicación HD en donde ambos dispositivos pueden transmitir y recibir información, pero no de forma simultánea. Un ejemplo son los walkie-talkies.

Por último, la transmisión FD permite que dos dispositivos, simultáneamente, envíen y reciban información. Por ejemplo, las comunicaciones telefónicas por par telefónico.

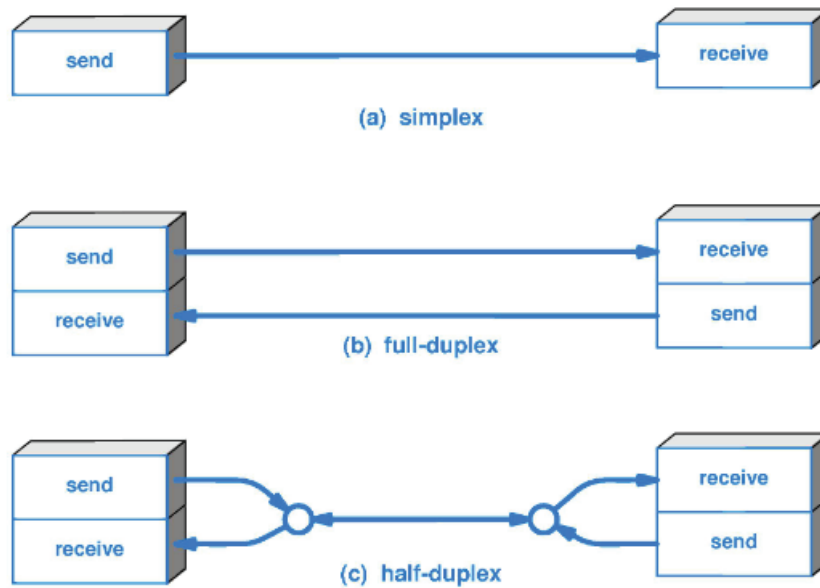


Figura 1- Tipos de comunicación entre dos equipos.

Para este proyecto se optó por establecer una comunicación Half Duplex (HD) entre las placas de desarrollo. Esta decisión se debe a que una de las placas utilizadas, la HackRF One, solo es capaz de operar en modo Half-Duplex.

Modulación de una señal

Modulaciones Analógicas

Modular es el proceso por el cual la señal de información es modificada para ser transmitida. Se lleva a cabo superponiendo la señal de información sobre otra señal, llamada portadora, cuyas características (como frecuencia, amplitud o fase) son modificadas de acuerdo con la señal original. Existen diferentes técnicas de modulación, cada una con sus propias características y aplicaciones. Algunas de las técnicas de modulación más comunes incluyen:

- Modulación de amplitud (AM): Funciona mediante la variación de la amplitud de la señal transmitida en relación con la información que se envía.
- Modulación de frecuencia (FM): Técnica de modulación que permite transmitir información a través de una onda portadora variando su frecuencia.

- Modulación de fase (PM): La fase de la onda portadora varía en forma directamente proporcional de acuerdo con la señal moduladora (información).

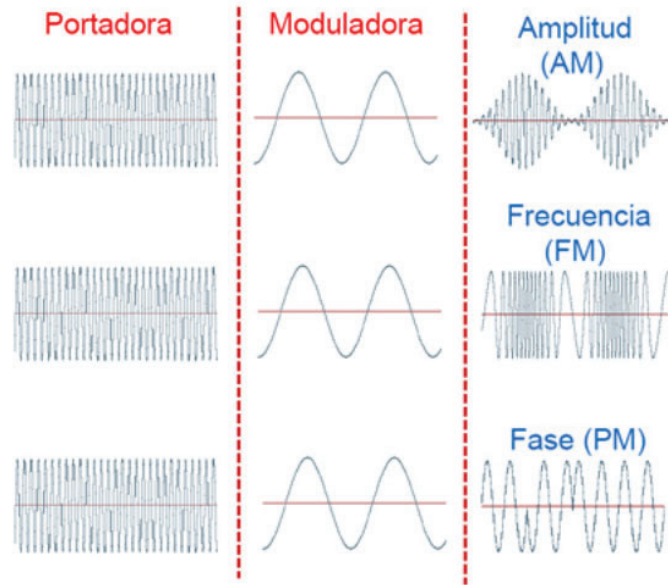


Figura 2- Tipos de modulaciones.

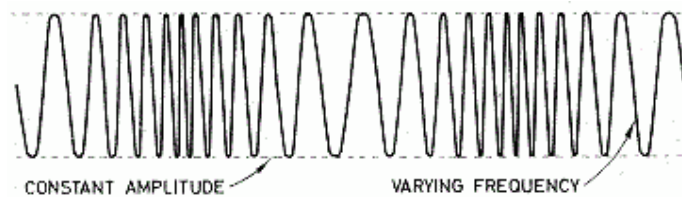


Figura 3- Modulación FM.

Una de las modulaciones utilizadas en el proyecto es la FM. Consiste en variar la frecuencia de la onda portadora de acuerdo con la intensidad de la onda de información. La amplitud de la onda modulada es constante e igual que la de la onda portadora. Gracias a que la amplitud es constante, los ruidos o interferencias que afecten a la señal no son problema. La información es extraída de la variación de frecuencia.

Existen dos métodos para generar señales FM, el primero es el método indirecto y el segundo el directo. El primero utiliza una modulación FM de banda estrecha y una multiplicación en frecuencia para incrementar el nivel de desviación en frecuencia. El

segundo, varía directamente la frecuencia de la portadora de acuerdo con la señal moduladora. Precisamente utilizamos NBFM (Narrowband FM), es decir el método indirecto.

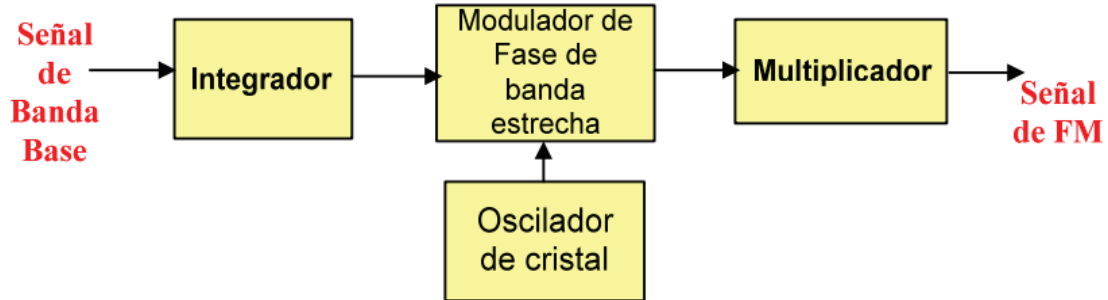


Figura 4- Diagrama en Bloques del modulador NBFM.

En este método, la señal de banda base se integra primero y luego se utiliza para modular en fase un oscilador a cristal, posteriormente, se utiliza una multiplicación de frecuencia para aumentar la desviación de frecuencia al nivel deseado.

Este tipo de FM se utiliza cuando se necesita gran estabilidad en la frecuencia de portadora. Se denomina método indirecto, debido a que realmente se realiza una modulación de fase que indirectamente varía la frecuencia.

La tensión de la señal modulada puede expresarse como:

$$v(t)_{fase} = V_c * \cos[\omega_c * t + \Delta\phi_{max} * x_m(t)]$$

Donde:

$$x_m(t) = \frac{v_m(t)}{|v_m(t)|_{max}} \quad y \quad \Delta\phi_{max} = k_m * |v_m(t)|_{max}$$

Siendo la señal moduladora de un tono de frecuencia $f_m(t)$:

$$x(t)_m = \cos[2 * \pi * f_m * t] = \cos[\omega_m t]$$

La señal modulada en fase:

$$v(t)_{fase} = V_c * \cos[w_c * t + \Delta\phi_{max} * \cos[w_m * t]]$$

Un circuito que provoca la modulación en frecuencia de forma indirecta es el modulador Armstrong.

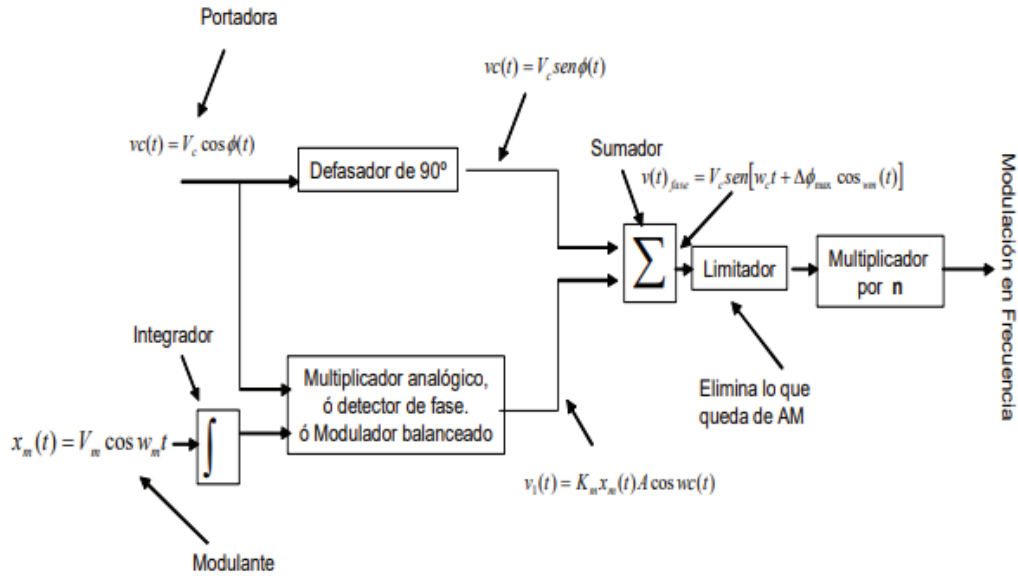


Figura 5- Diagrama en Bloques del modulador NBFM detallado.

Los bloques de la figura 5 son:

- 1) Oscilador de cristal: Genera la señal portadora, que luego se divide en dos caminos.
- 2) Desfasador de 90 grados: Uno de los caminos de la señal portadora pasa por el desfasador, lo que produce una señal idéntica pero con un desfase de 90 grados.
- 3) Modulador en fase: La señal de audio modula una de estas señales, cambiando la fase en función de la señal de audio. Esto crea una pequeña modulación de fase.
- 4) Sumador: Combina la portadora desfasada y la portadora modulada en fase con la señal de audio. Esta suma genera una señal cuya fase varía continuamente generando una modulación en frecuencia.

- 5) Limitador: Se encarga de que la amplitud de la señal se mantenga constante, eliminando cualquier variación de amplitud que pueda haberse introducido accidentalmente.
- 6) Multiplicador de frecuencia: La señal modulada en fase tiene una desviación pequeña, por lo que pasa por un multiplicador de frecuencia para aumentar tanto la frecuencia portadora como la desviación de frecuencia.
- 7) Mezclador y Amplificador: Ajusta la señal a la frecuencia de transmisión y la amplifica para ser transmitida.

La señal portadora generada por el oscilador de cristal es una onda sinusoidal pura que podemos expresar como:

$$c(t) = A * \cos(w_c * t)$$

Donde A es la amplitud, y w_c es la frecuencia angular de la portadora. $w_c = 2 * \pi * f_c$ (f_c es la frecuencia de la portadora).

Al pasar por el desfasador, la señal se convierte en:

$$c(t)_{desfasada} = A * \sen(w_c * t)$$

La señal de audio (o la información que queremos transmitir) modula la fase de la portadora. La señal modulada en fase puede expresarse como:

$$c(t)_{modulada} = A * \cos(w_c * t + \phi(t))$$

Donde $\phi(t)$, es la fase instantánea de la señal modulada, que varía en función de la señal de audio.

La suma de la señal modulada con la desfasada resulta ser:

$$s(t) = A * \cos(w_c * t + \phi(t)) + A * \sen(w_c * t)$$

Para simplificar esta expresión, utilizamos la identidad trigonométrica general para una combinación de senos y cosenos con diferente fase:

$$A * \cos(w_c * t + \phi(t)) + A * \sin(w_c * t) = B * \cos(w_c * t + \Theta(t))$$

llegando a la siguiente ecuación:

$$s(t) = B * \cos(w_c * t + \Theta(t))$$

Donde $\Theta(t)$ es la nueva fase instantánea y B es la nueva amplitud resultante (que puede ser constante).

En esta última expresión podemos llegar a la conclusión de que la señal $s(t)$ ahora está modulada en frecuencia, porque la fase instantánea $\Theta(t)$ está variando en el tiempo. Además, esta variable depende de $\phi(t)$ que está controlada por la señal de audio. Esto es porque la frecuencia está relacionada con la derivada de la fase.

La frecuencia instantánea de una señal es la derivada temporal de la fase instantánea, es decir que cuando la fase $\Theta(t)$ varía en el tiempo debido a $\phi(t)$, la frecuencia de la señal $s(t)$ también varía. Matemáticamente se expresa como:

$$f(t)_{instantánea} = \frac{1}{2 * \pi} * \frac{d\Theta(t)}{dt}$$

Además:

$$\Theta(t) = w_c * t + \beta * \int m(t) dt$$

Donde β es el índice de modulación y $m(t)$ es la señal moduladora. Al hacer la $\frac{d\Theta(t)}{dt}$ llegamos a:

$$\frac{d\Theta(t)}{dt} = w_c + \beta * m(t)$$

Resultando ser que:

$$f(t)_{instantánea} = \frac{1}{2 * \pi} * (w_c + \beta * m(t))$$

$$f(t)_{instantánea} = \frac{w_c}{2 * \pi} * + \frac{\beta * m(t)}{2 * \pi}$$

Recordando que $w_c = 2 * \pi * f_c$:

$$f(t)_{instantánea} = f_c + \frac{\beta * m(t)}{2 * \pi}$$

Esto significa que la frecuencia de la señal final no es fija, sino que varía continuamente en función de la señal de audio (señal moduladora).

Para demodular la señal, uno de los métodos es utilizar un circuito de fase cerrada (PLL). Este circuito sigue la frecuencia de la señal portadora modulada. Donde se tiene un VCO (Voltage-Controlled Oscillator), como su nombre indica, es un oscilador controlado por voltaje. Cuya frecuencia de salida depende del voltaje de entrada que recibe.

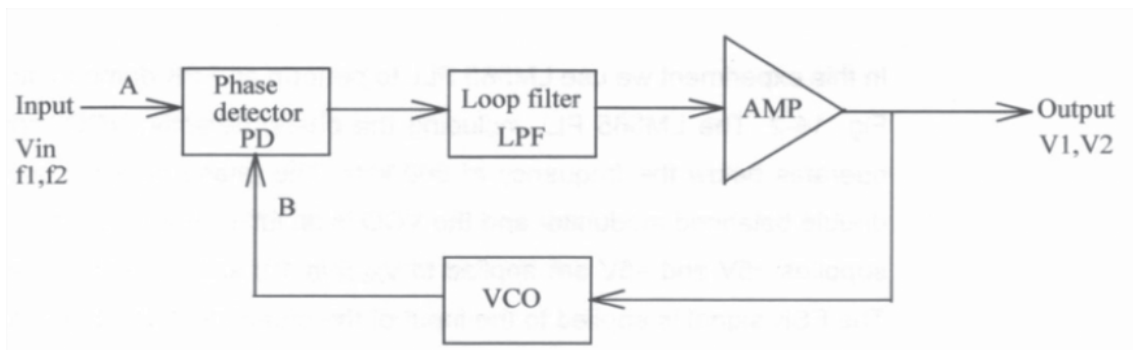


Figura 6- Diagrama en bloques del demodulador FM PLL.

Un demodulador de frecuencia de PLL no requiere de circuitos sintonizados y automáticamente compensa los cambios en la frecuencia de la portadora debido a la estabilidad en el oscilador de transmisión.

Como en la modulación FM la información está en las variaciones de la frecuencia, el PLL ajusta su frecuencia de salida para seguir la señal entrante. Es decir, permite que una señal de referencia externa, controle la frecuencia y la fase de un oscilador. Las variaciones de

frecuencia resultan en variaciones de fase en la salida del PLL, lo que luego se puede traducir en la señal original.

El procedimiento básico sería:

- 1) Enganche de fase: El PLL compara la fase de la señal de entrada (la señal FM) con la de un VCO interno.
- 2) Ajuste del VCO: Si hay una diferencia de fase entre la señal FM y el VCO, el PLL genera un voltaje de error que ajusta la frecuencia del VCO para que coincida con la señal FM.
- 3) Seguimiento de la frecuencia: A medida que la frecuencia de la señal FM cambia, el VCO sigue estos cambios ajustándose continuamente. La señal FM modula la frecuencia del VCO.
- 4) Demodulación: El voltaje de error generado para ajustar el VCO refleja las variaciones de frecuencia de la señal FM, que es justamente la información modulada. Este voltaje es la señal demodulada.

Lo más importante en el PLL para demodular FM es el voltaje de error que se genera. Este voltaje se produce para ajustar el VCO y equiparar su frecuencia con la de la señal de entrada. Ese voltaje de error refleja directamente las variaciones de frecuencia de la señal FM, que es donde se encuentra la información modulada.

Modulaciones Digitales

También existen las modulaciones digitales, que se describen como el proceso de codificar una señal de información digital (discreta) en la amplitud, fase o frecuencia de la señal transmitida. En general, una técnica de modulación codifica varios bits en un solo símbolo. Existen varios tipos de modulaciones digitales: ASK (Amplitude-shift keying), FSK (Frequency Shift Keying), PSK (Phase Shift Keying), QAM (Quadrature amplitude modulation), QPSK (Quadrature Phase Shift Keying), MSK (minimum-shift keying). Para el proyecto se utilizó una variante de la MSK denominada GMSK (Gaussian Minimum Shift Keying).

Es necesario describir la modulación de desplazamiento en frecuencia (FSK) para poder comprender el funcionamiento de sus variantes, en particular la utilizada en este proyecto.

FSK modula los bits de información variando la frecuencia de la señal de la señal portadora, para esto se mapea cada bit, o conjunto bits, a una lista finita de frecuencias que son combinadas para formar la señal que se va a transmitir.

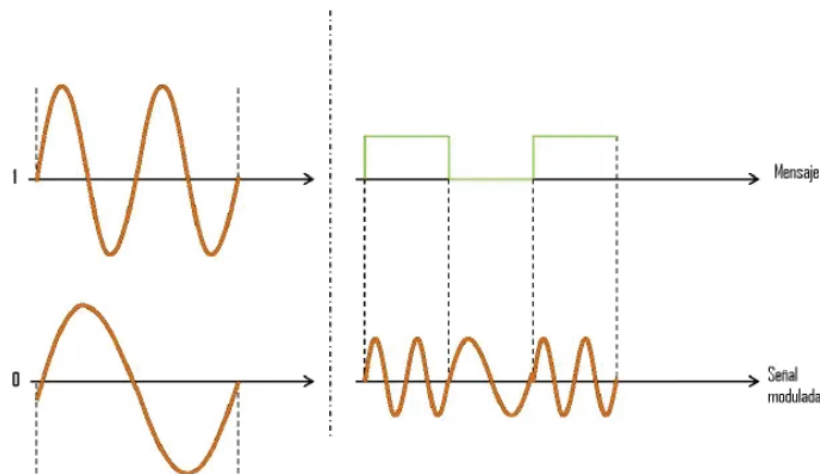


Figura 7- Modulación FSK.

En la figura anterior, se muestran dos frecuencias distintas para el valor 1 y 0 respectivamente. Si el bit es un 1, la señal se emite en la frecuencia más alta. Si el bit es un 0, la señal se emite en la frecuencia más baja.

Un transmisor FSK para una frecuencia por símbolo podría representarse como el de la siguiente figura.

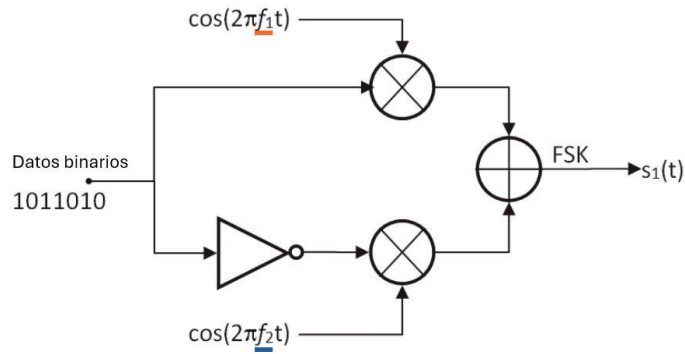


Figura 8- Diagrama en bloques de un modulador FSK simple.

Los datos binarios del transmisor pasan a través de un circuito digital que habilita la frecuencia f_1 o f_2 respectivamente para ser transmitida. Dado que la transición entre las frecuencias es abrupta, por la naturaleza de los datos de entrada, el espectro en frecuencia resultante es el de modular dos pulsos rectangulares con las frecuencias f_1 y f_2 como se ve en la siguiente figura.

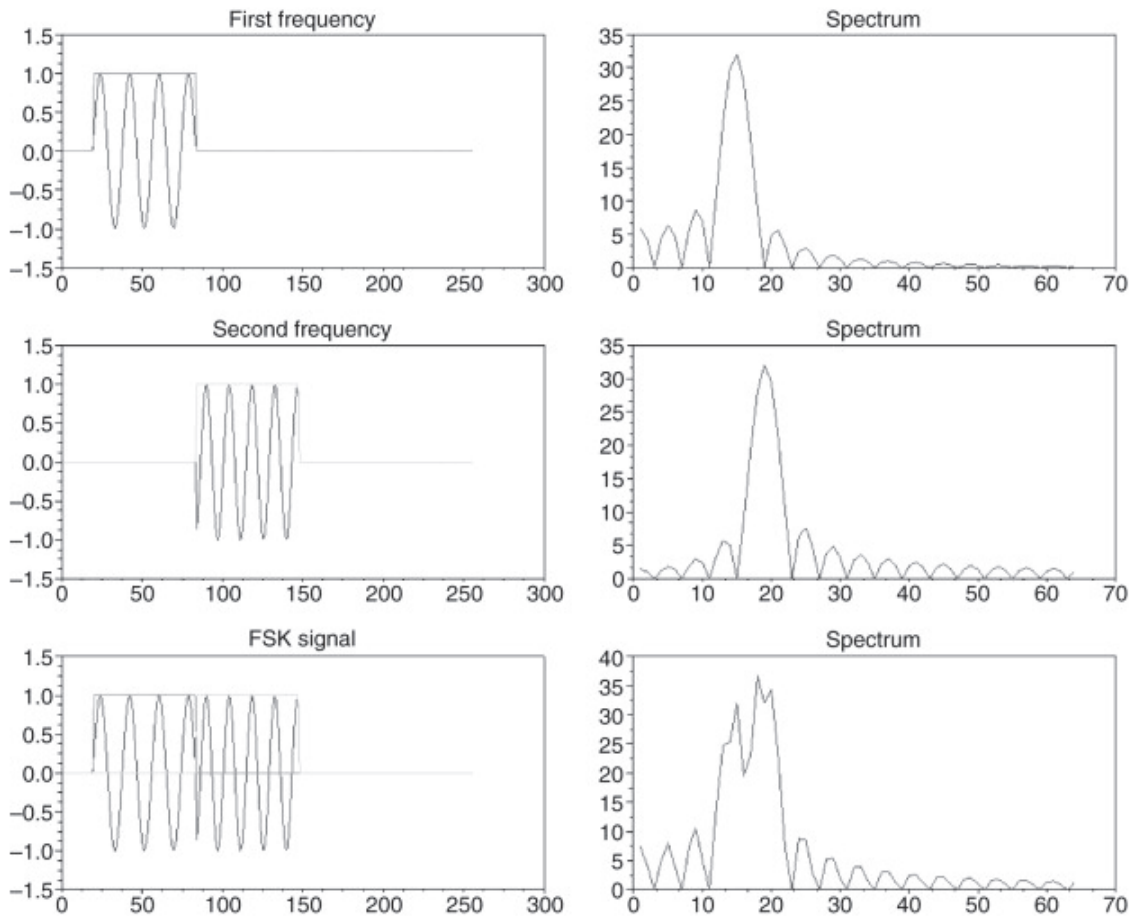


Figura 9- Espectro en frecuencia de la modulación FSK.

Sabemos que el espectro en frecuencia resultante de la modulación de un tono por un pulso rectangular es de la forma

$$f(x) = \frac{\text{Sin}(x)}{x}$$

La distancia óptima entre las dos frecuencias de transmisión es la que asegura que el máximo de la frecuencia 1 se encuentre en alguno de los ceros de la frecuencia 2. Sin embargo, como se ve en la figura anterior, siempre existe interferencia entre las dos frecuencias transmitidas. Este efecto se ve agravado al sumar combinaciones de símbolos y frecuencias. Para mitigar este efecto surge la modulación MSK.

MSK es una modulación de frecuencia basada en un sistema conocido como “modulación por cambio de frecuencia de fase continua”. Esta modulación codifica cada bit como una senoide, ya que se busca dejar el módulo de la señal constante para reducir los problemas de distorsión por cambios abruptos de frecuencia.

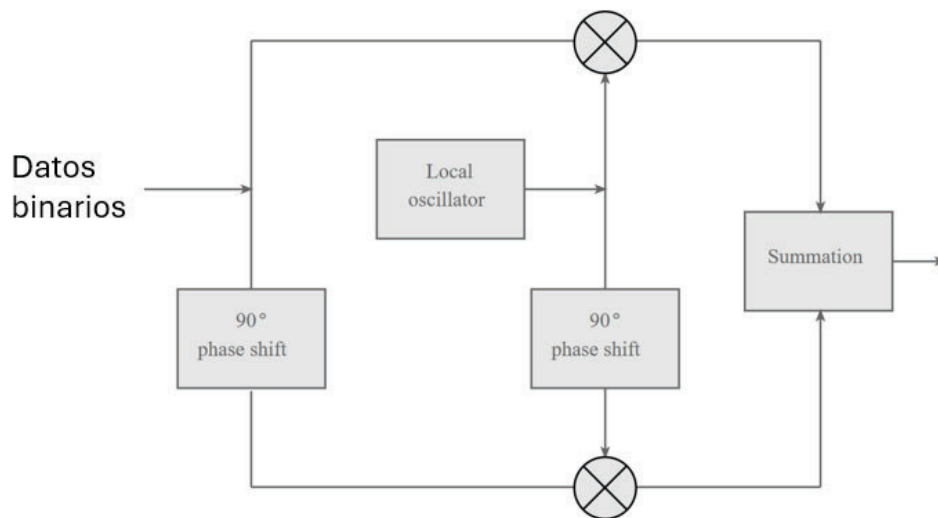


Figura 10- Diagrama en bloques de modulador MSK.

La modulación busca evitar los saltos de fase ya que asegura que los cambios de frecuencia ocurren en los puntos de cruce por cero de la señal portadora como se ve en la siguiente imagen.

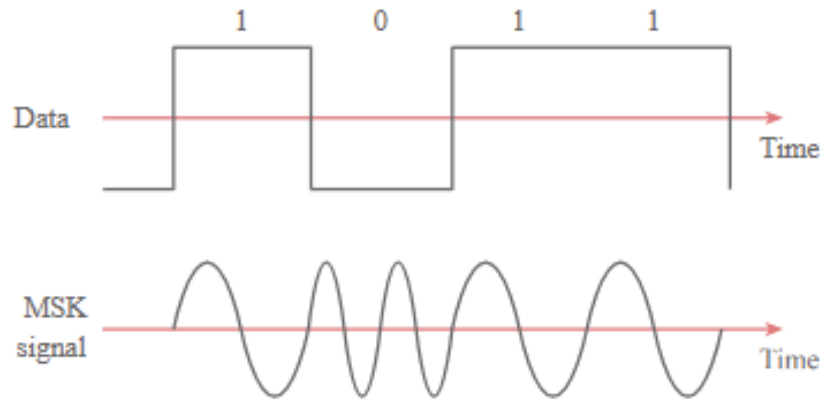


Figura 11- Modulación MSK.

Una mejora posible para reducir más la interferencia entre símbolos es utilizar un filtro pasa bajos en la entrada del modulador. Esto elimina los componentes de más alta frecuencia presentes en los flancos de bits.

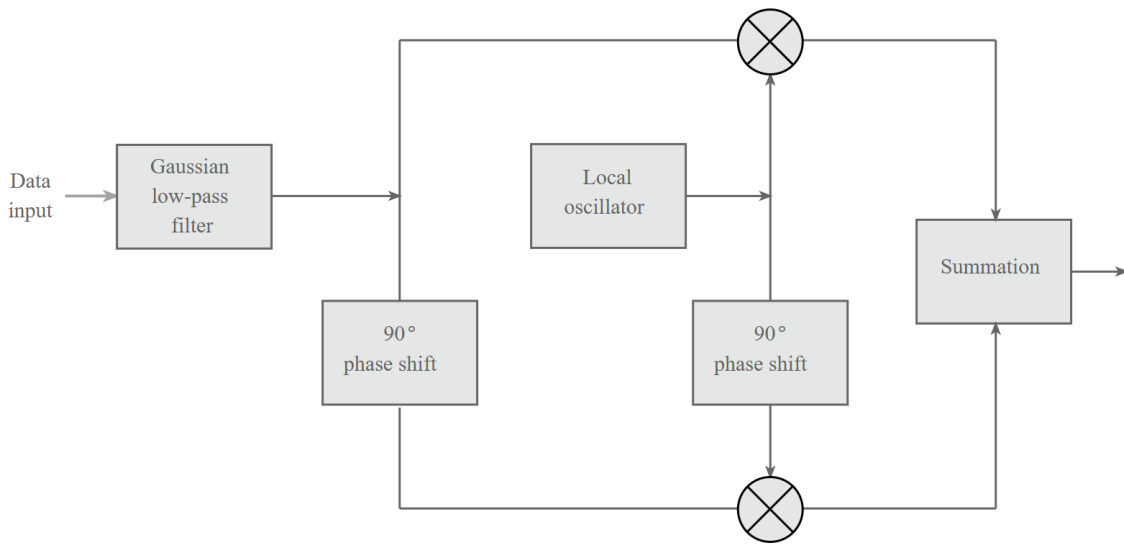


Figura 12- Diagrama en bloques de modulador GMSK.

Se pueden comparar la forma de los datos transmitidos en MSK y GMSK en la siguiente imagen. Eliminar los componentes de alta frecuencia de los datos binarios resulta en las curvas suavizadas que se ven en los cambios de datos.

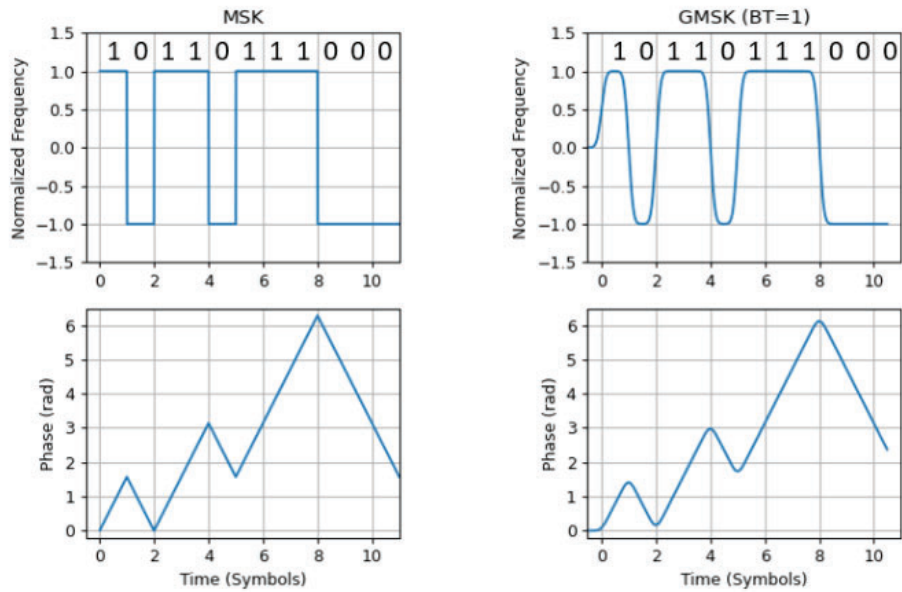


Figura 13- Comparación entre MSK y GMSK.

La mejor manera de comprender el impacto del filtro pasa bajos gaussiano en esta modulación, es viendo el efecto real en el espectro. En esta comparación se ve que la densidad de potencia se reduce de forma mucho más abrupta cuando se utiliza el filtro.

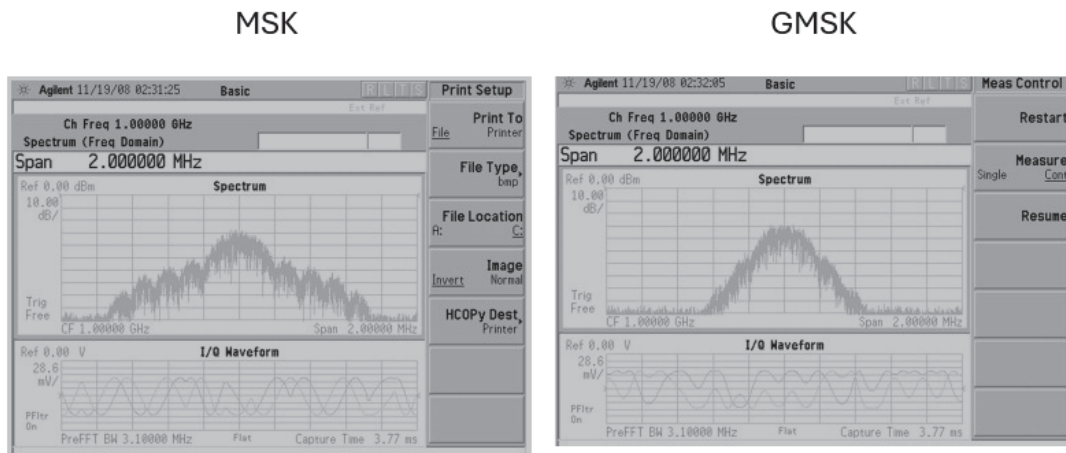


Figura 14- Comparación entre MSK y GMSK en espectro.

Frequency Hopping Spread Spectrum (FHSS)

FHSS es una técnica utilizada en comunicaciones inalámbricas para distribuir la energía de la señal de datos sobre un rango más amplio de frecuencias. En lugar de utilizar una única frecuencia, como hacen las técnicas de modulación tradicionales, opera en una banda de frecuencia más amplia.

La clave está en dividir la información en paquetes discretos, cada uno de ellos se transmite a través de una serie aparentemente aleatoria de frecuencias, cada paquete se transmite saltando de frecuencia en frecuencia a intervalos fijos. Estos saltos de frecuencias siguen el patrón de un código pseudoaleatorio generado tanto en el transmisor como en el receptor. Este código puede tener n bits de largo, al diseñarse el código debe tenerse en mente que las frecuencias a transmitir van a ser 2^n slots de frecuencias distintas.

El transmisor y el receptor deben estar sincronizados y tener precargados los códigos de generación de tal manera que el receptor sintonice a la siguiente frecuencia al mismo tiempo que lo hace el transmisor.

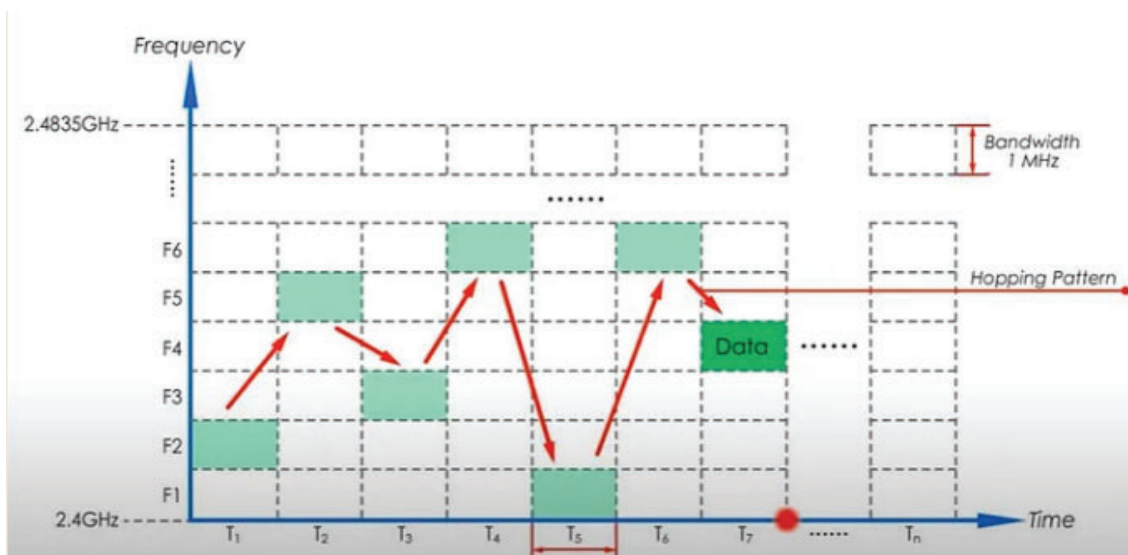


Figura 15- Saltos de paquetes en diferentes frecuencias.

Las ventajas más importantes de esta modulación son:

- Resistencia a interferencias: Al “saltar” entre diferentes frecuencias, FHSS es menos susceptible a las interferencias causadas por dispositivos que operen en la misma banda de frecuencia.

- Seguridad: Gracias a que los paquetes no viajan en la misma frecuencia, dificulta la interceptación de ellos.
- Robustez: El sistema es capaz de adaptarse a entornos inalámbricos cambiantes, como áreas con interferencia electromagnética.
- Uso compartido de frecuencias: Varios usuarios pueden compartir el mismo ancho de banda sin causar interferencia mutua, ya que cada usuario opera con una secuencia pseudoaleatoria única.

Las aplicaciones más comunes de esta tecnología son:

- Redes inalámbricas: Más precisamente en redes locales inalámbricas (WLAN), como Bluetooth y algunas implementaciones de Wi-Fi (IoT).
- Comunicaciones militares: Gracias a su robustez y seguridad, garantiza confidencialidad y resistencia a interferencias.
- Telecomunicaciones industriales: En estos entornos donde hay muchas fuentes de interferencias, mantiene la conectividad con sistemas de control y monitoreo inalámbricos.

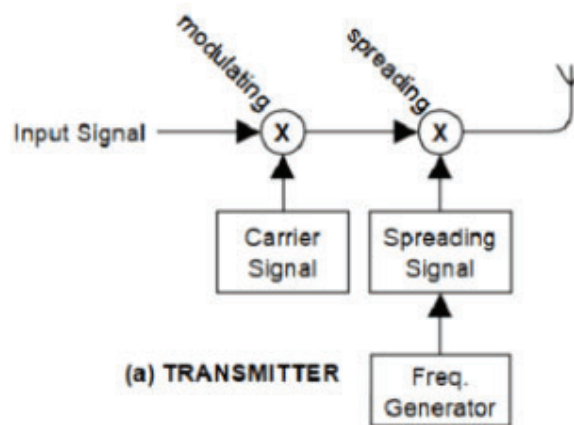


Figura 16- Diagrama de bloques de un transmisor FHSS típico.

Para generar la la transmisión de la información en diferentes frecuencias, como se mencionó anteriormente, se utiliza un generador de código pseudoaleatorio

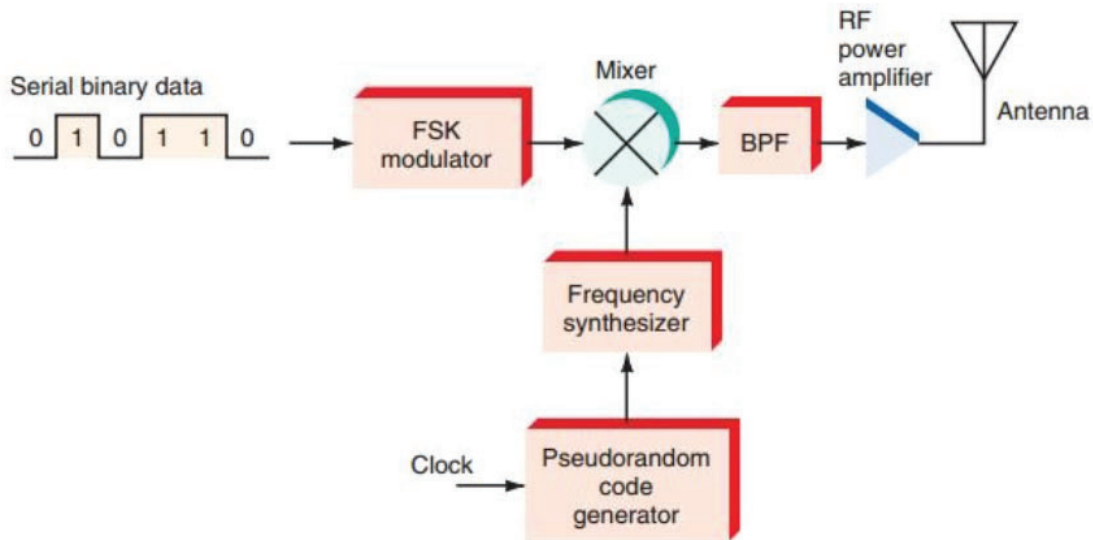


Figura 17- Transmisor FHSS en detalle.

En la figura anterior se ve un típico transmisor FHSS.

- 1) Fuente de datos: Es el origen de la información que deseas transmitir (puede ser voz, video, o datos).
- 2) Modulador: Convierte los datos de la fuente en una señal modulada, usando técnicas como FSK, PSK, QAM o en el caso de nuestro proyecto GMSK.
- 3) Generador de secuencia de saltos: Este bloque genera una secuencia pseudoaleatoria de frecuencias que define en qué frecuencia saltará el sistema. Este es el "patrón" de frecuencia que sigue el transmisor.
- 4) Oscilador controlado (Frequency synthesizer): Cambia la frecuencia de transmisión en función de la secuencia de saltos generada. Así, la señal modulada es transmitida en diferentes frecuencias.
- 5) Filtro pasa banda (BPF): Su función es permitir pasar sólo las frecuencias que están dentro del rango deseado para la transmisión y eliminar cualquier señal espuria o ruido.

- 6) Amplificador de Potencia de Radiofrecuencia: Se usa para aumentar la potencia de la señal antes de que sea enviada a la antena, lo que permite que la señal se transmita a mayores distancias.
- 7) Antena transmisora: Envía la señal modulada a través de las diferentes frecuencias según la secuencia de saltos.

Los bloques código PN, frequency synthesizer, y mixer, son partes clave del funcionamiento del transmisor FHSS. El código PN es una secuencia pseudoaleatoria de bits que parece ser ruido, pero en realidad sigue un patrón determinista conocido. Este código es lo que define la lista de frecuencias en la que el transmisor “salta”. El sintetizador de frecuencia cambia la frecuencia de transmisión de acuerdo con la secuencia definida por el código PN. Cada vez que el código indica un salto, el sintetizador cambia la frecuencia de la señal portadora, lo que permite transmitir en una nueva frecuencia (Como se explicó anteriormente, un VCO es una opción viable para cambiar de frecuencia). El mixer toma la señal de datos modulada con la frecuencia generada por el sintetizador y las mezcla. Esto da como resultado una señal modulada en la frecuencia que corresponde al salto actual.

Para recibir la información, es necesario tener el mismo código PN (del transmisor) en el receptor.

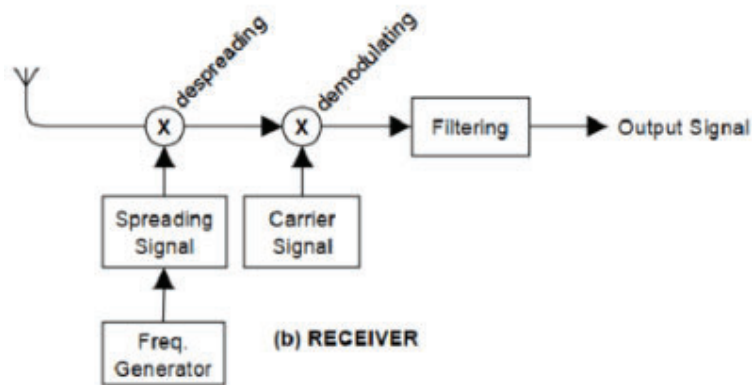


Figura 18- Receptor FHSS típico.

El funcionamiento es idéntico al transmisor. El código PN al ser el mismo, se sintoniza en la misma frecuencia al mismo momento. Una vez capturada la información, es demodulada para luego ser filtrada y obtener los datos.

Teorema de muestreo Nyquist-Shannon

El Teorema de Muestreo de Nyquist-Shannon establece que una señal analógica de banda limitada puede ser perfectamente reconstruida si se muestrea a una tasa al menos el doble de su frecuencia máxima. Matemáticamente, esto se expresa como:

$$F_s \geq 2F_{max}$$

Donde F_s es la frecuencia de muestreo y F_{max} es la frecuencia máxima presente en la señal. Este teorema asegura que no se pierde información crítica de la señal durante el proceso de muestreo, permitiendo una fiel reconstrucción de la señal original.

Para el proyecto, antes de transmitir el audio, se utiliza una tasa de muestreo de 48 kHz. Siguiendo la lógica, la frecuencia de Nyquist es de 24 kHz, lo que permite una captura efectiva de todas las frecuencias audibles (hasta 20 kHz en humanos) con un margen adicional para filtrado y procesamiento. Este margen contempla el filtrado anti-aliasing, mejorando la calidad de la señal.

Señales analógicas y digitales

En los sistemas de comunicación, las señales juegan un papel fundamental, ya que son el medio a través del cual se transmite la información. Las más conocidas y comúnmente utilizadas son las señales analógicas y digitales.

Las señales analógicas se caracterizan por tener una amplitud que puede tomar diferentes valores (infinitos) en un tiempo determinado. Es decir, señales de tiempo continuo.

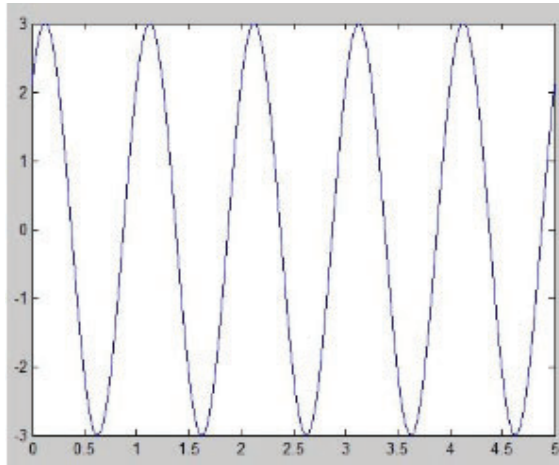


Figura 19- Señal analógica a modo ejemplo.

Por otro lado, las señales digitales se caracterizan por tener una amplitud que toma ciertos valores finitos en un tiempo determinado. Estas señales cambian de valor sólo en intervalos discretos (tiempo discreto), normalmente representados como números binarios. Esto simplifica el procesamiento de la información.

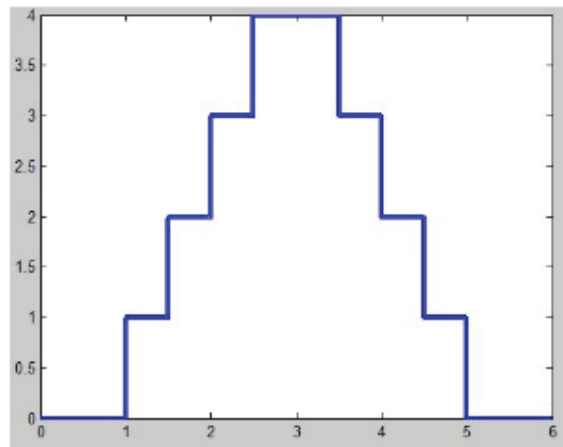


Figura 20- Señal digital a modo ejemplo.

En la figura 20, se observa una señal digital con su conjunto finito de valores en un intervalo de tiempo.

Las señales de tiempo discreto como las de tiempo continuo, clasifican la naturaleza de la señal a lo largo del tiempo. Por otro lado, las señales analógicas y digitales clasifican la amplitud de la señal.

Transformada rápida de Fourier (FFT)

La transformada rápida de Fourier es altamente usada en las telecomunicaciones. Mediante ella, se puede transformar una señal en el dominio del tiempo al dominio de la frecuencia.

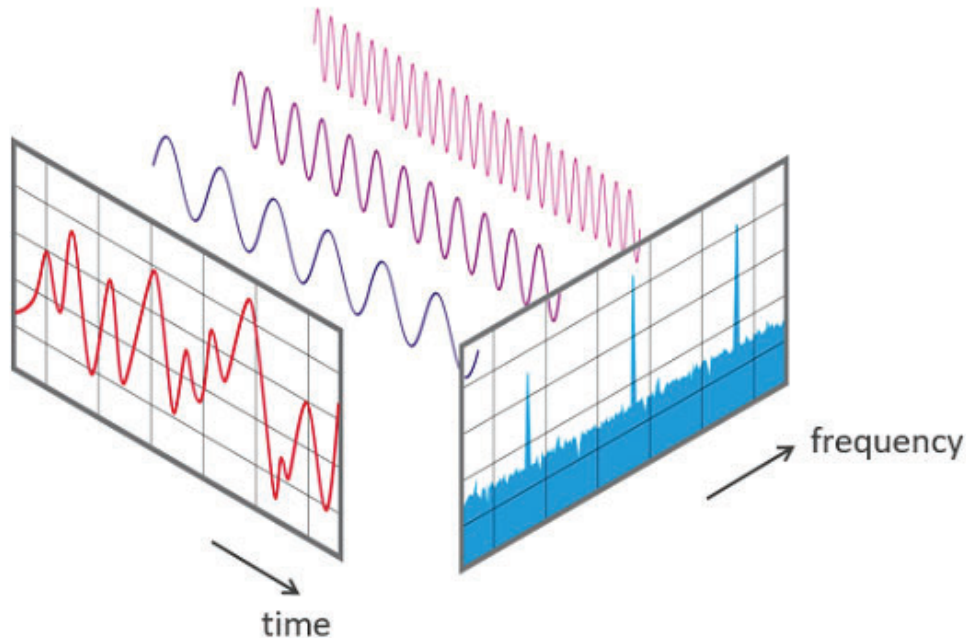


Figura 22- Pasaje del dominio del tiempo al dominio de la frecuencia.

Los valores en el dominio de la frecuencia son representados mediante picos. Se utiliza para obtener información de una señal que en el dominio del tiempo no es legible. Matemáticamente, para señales discretas, se representa de la siguiente manera.

$$X[k] = \sum_{n=0}^{N-1} x[n] * e^{-j * \frac{2\pi}{N} * kn}$$

Donde $X[k]$ es la k -ésima componente de frecuencia de la señal. $x[n]$ es la n -ésima muestra de la señal en el dominio del tiempo. N es el número total de muestras de la señal. k es el índice de frecuencia que va desde 0 hasta $N-1$. j es la unidad imaginaria.

La FFT es rápida y eficiente, lo que permite analizar señales en tiempo real, como señales de audio, video, etc. Este proceso permite entender los componentes de frecuencia de cualquier señal, tanto analógica como discreta.

Transmisión inalámbrica

Problemas comunes

La comunicación inalámbrica tiene sus problemáticas, provocadas mayormente por la dispersión y la divergencia natural. Las más comunes son:

- 1) Interferencias que degraden la señal.
- 2) Atenuación, por distancia u obstáculos.
- 3) Ruido, tanto electromagnético como ambiental.
- 4) Pérdida de paquetes.
- 5) Multi trayecto, se crean múltiples trayectos hacia el destino por el rebote de la señal (esta última entra en dispersión).

La tecnología FHSS combinada con un protocolo de comunicación, permite solventar o mitigar los problemas mencionados. Una medida que ayuda a la detección de errores es el CRC (verificación de redundancia cíclica). Esta técnica, garantiza la integridad de los datos transmitidos. Funciona agregando una suma de verificación a la información, que luego se verifica en el receptor. En el caso particular de este proyecto se decidió utilizar un CRC de 4 Bytes, ya que se asumió enviar paquetes de 8 Bytes de datos.

Distancia según frecuencia

La frecuencia a la cual es transmitida la información tiene un impacto significativo en la comunicación inalámbrica, el más directo es la distancia que pueden situarse los equipos. Las frecuencias más altas, suelen propagarse peor en entornos con obstáculos como paredes, debido a una mayor atenuación y absorción de materiales. Aunque, suelen tener menos interferencia en áreas congestionadas, esto se debe a que los dispositivos que operan en esas frecuencias son más especializados. Por último, mientras más grande sea la frecuencia mayor será la capacidad de transmitir datos. En general, a medida que se desplaza hacia una frecuencia de transmisión más alta, suele haber más espacio disponible para asignar anchos de banda más amplios.

Por otro lado, las frecuencias más bajas proporcionan una mayor penetración de obstáculos. A su vez, los anchos de banda disponibles suelen ser menores porque es una banda compartida para muchas aplicaciones de baja potencia.

Algunas Consideraciones a tener en cuenta para el uso de frecuencias:

- 1) Distancia y entorno: Si se necesita cubrir largas distancias o trabajar en entornos con muchos obstáculos, es preferible usar frecuencias bajas.
- 2) Capacidad de datos: Si se requiere altas velocidades de transmisión de datos, las frecuencias más altas permiten manejar anchos de banda más amplios.
- 3) Disponibilidad del espectro: Regulaciones para saber qué bandas están disponibles.

Pérdida de Trayectoria en Espacio Libre

Los problemas anteriormente mencionados se rigen por una ecuación matemática denominada Pérdida de Trayectoria en Espacio Libre (FSPL). Es un concepto fundamental en la teoría de las telecomunicaciones y se refiere a la atenuación de la señal a medida que viaja a través del espacio libre sin obstáculos, como el aire o el vacío.

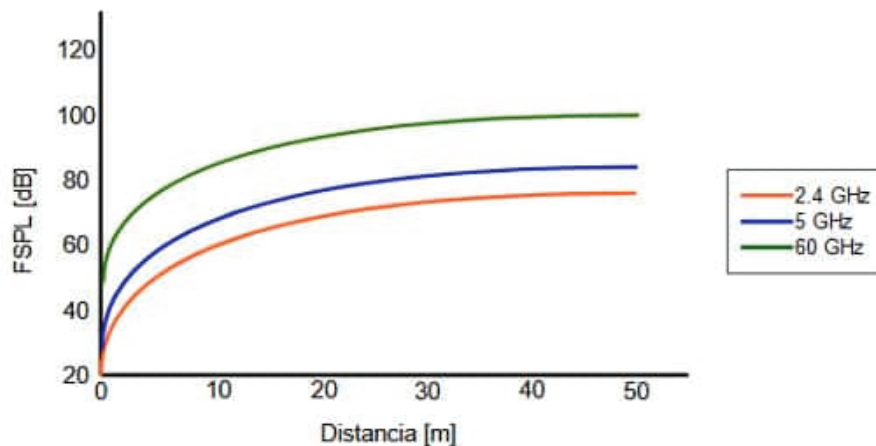


Figura 23- Comparación de distancia entre frecuencias.

Tal como se mencionó anteriormente, la dispersión y la divergencia natural influyen negativamente en la propagación de señales. Estos son los causantes de que la intensidad de la señal disminuya, en cuanto más lejos se encuentre del transmisor.

$$FSPL(dB) = 20 * \log_{10}(d) + 20 * \log_{10}(f) + 20 * \log_{10}\left(\frac{4\pi}{c}\right)$$

La ecuación contempla todo lo necesario:

- 1) Distancia (d): La distancia entre el transmisor y el receptor, medida en metros.
- 2) Frecuencia de la señal (f): La frecuencia de la señal que se está transmitiendo, medida en hertz (Hz).
- 3) Velocidad de la luz (C): La velocidad de la luz en el vacío, que es aproximadamente 3×10^8 metros por segundo.
- 4) Constante de espacio libre: $20 * \log_{10}(\frac{4\pi}{c})$ Es una constante para tener en cuenta las propiedades del espacio libre y la frecuencia de la señal

Para una mejor comunicación, el valor de FSPL debe ser lo más bajo posible. Debido a que representa la pérdida de la señal. Por lo que con la ecuación matemática podemos llegar a la conclusión de que a menor distancia y frecuencia, más baja será la pérdida de señal.

GNU Radio

GNU Radio es un conjunto de herramientas de desarrollo de software gratuito y de código abierto que proporciona bloques de procesamiento de señales para implementar radios definidas por software. Se utiliza ampliamente en entornos de investigaciones, industriales, académicos, etc. Con el fin de respaldar tanto la investigación de comunicaciones inalámbricas como los sistemas de radio del mundo real.



Figura 14- Logo GNU Radio.

Radio Definido por Software

Un SDR (Software Defined Radio), es un sistema de comunicación por radio en el que los componentes que convencionalmente se han implementado en hardware analógico se implementan mediante software en una computadora.

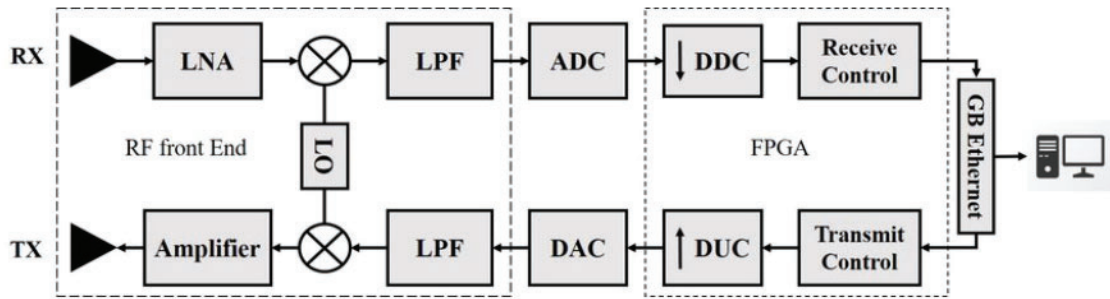


Figura 25- Diagrama de bloques de un SDR típico.

En la figura 25, se muestra el diagrama de bloques de un SDR para recibir y transmitir. Se puede dividir en dos partes:

- 1) RF front end (radio frequency front end): Se refiere a la parte del sistema que gestiona la recepción y transmisión de señales de RF (antenas, amplificadores, etc)
- 2) FPGA (Field-Programmable Gate Arrays): Es un dispositivo programable que permite la implementación flexible de circuitos digitales y el procesamiento eficiente de datos, utilizado en diversas aplicaciones, incluidas las comunicaciones y el procesamiento de señales en SDRs.

Los bloques del receptor son los siguientes:

- 1) Antena: Se conecta a una antena externa que captura las señales de RF. La elección de la antena es crucial para la recepción efectiva de las frecuencias deseadas.
- 2) LPF (Filtro pasa bajos): Elimina frecuencias no deseadas, reduce el ruido, previene el aliasing, etc.

- 3) Mixer y local oscillator (LO): El mezclador utiliza un oscilador local para convertir la señal RF de la antena a una frecuencia intermedia o de banda base. Este componente es esencial para la sintonización de las señales de interés.
- 4) ADC (Analog to digital converter): Se puede tener uno o varios convertidores que digitalizan las señales analógicas provenientes del mezclador. Esto permite que las señales sean procesadas por software.
- 5) DDC (Digital Down Converter): Reduce la frecuencia y tasa de muestreo de las señales digitales, permitiendo que el software se enfoque en la señal de interés. Este proceso ayuda a manejar menos datos y a mejorar la eficiencia del procesamiento.
- 6) Receive Control (Hardware): Este componente gestiona la configuración del receptor como la ganancia, frecuencia y otros parámetros operativos.
- 7) GB Ethernet (Gigabit Ethernet): Son puertos de conexión (como Gigabit Ethernet) que permiten la transferencia de datos digitales a una computadora u otro sistema para su procesamiento. Esta conexión es crucial para la comunicación entre el hardware del SDR y el software que realiza el procesamiento.

Por otro lado, el receptor tiene bloques similares (se lee de derecha a izquierda), los bloques nuevos son:

- 1) Transmit control: Gestiona y controla todo el proceso de transmisión. Esto incluye la configuración de parámetros como la frecuencia de transmisión, la potencia de salida, y otros ajustes.
- 2) DUC (Digital Up Converter): Aumenta la frecuencia de la señal digital a la frecuencia de transmisión deseada, permitiendo que se alineen ambas frecuencias. Esto se puede hacer mediante una técnica de modulación digital.
- 3) DAC (Digital to analog converter): Convierte la señal digital a una señal analógica para que pueda ser transmitida por la antena.

- 4) Amplifier (Amplificador): Aumenta la potencia de la señal generada antes de ser transmitida.

La capacidad de reconfigurar el hardware (como un FPGA) a través del software, es lo que permite que los sistemas SDR sean tan versátiles. Por ejemplo, se puede cambiar la modulación de AM a FM o a PSK simplemente modificando el software que controla el FPGA.

Algunas ventajas de esta tecnología son:

- Flexibilidad: Permite reconfigurar el hardware para soportar múltiples estándares y protocolos mediante cambios en el software.
- Actualización y Mantenimiento: Las actualizaciones de software pueden mejorar las capacidades de la radio sin necesidad de cambios en el hardware.
- Investigación y Desarrollo: Facilita la experimentación y el desarrollo de nuevas tecnologías de comunicación.

SDRSharp

SDRSharp, o mejor conocido como SDR#, es un software de recepción y visualización diseñado específicamente para trabajar con receptores de radio definidos por software. Es muy utilizado por radioaficionados para recibir, analizar y demodular señales de radio en diferentes bandas de frecuencia.

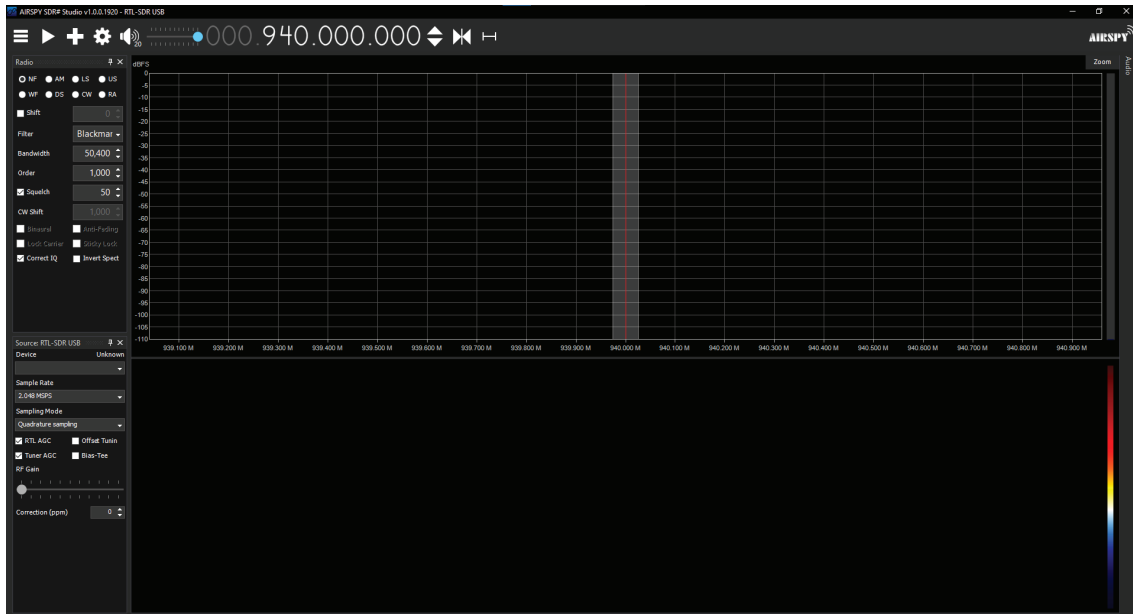


Figura 26- SDR#.

Algunas características son:

- 1) Interfaz intuitiva: Ofrece una interfaz gráfica fácil de usar para controlar y visualizar la recepción de señales.
- 2) Compatibilidad con múltiples dispositivos: Funciona con una variedad de hardware SDR, incluyendo RTL-SDR, HackRF, Aircspy.
- 3) Demodulación de señales: Soporta demodulación de diferentes tipos de señales como AM, FM, entre otras.
- 4) Visualización espectro y waterfall: Permite visualizar la energía de las señales en el espectro y a lo largo del tiempo con gráficos de cascada.

- 5) Plugins y extensibilidad: Admite plugins para añadir nuevas funciones y características, fomentando la personalización.
- 6) Grabación y reproducción: Posibilita la grabación de señales recibidas para posteriores análisis.

Hardware SDR

Uno de los dispositivos usados, es la USRP B210 de la marca Ettus. Proporciona una plataforma de software periférico de radio universal de placa única totalmente integrada. Algunas características generales son:

- Dispositivo USRP de dos canales totalmente integrado con cobertura de radiofrecuencia (RF) continua de 70 MHz a 6 GHz
- Funcionamiento full duplex, MIMO (2 Tx y 2 Rx) con hasta 56 MHz de ancho de banda en tiempo real (cuadratura de 61,44 MS/s).
- Conectividad USB 3.0 SuperSpeed.
- Compatibilidad con GNU Radio.



Figura 27- Dispositivo USRP B210.

Por otro lado, se cuenta con la tecnología del HackRf One. Consta de:

- Rango de 1MHz a 6GHz.
- Funcionamiento Half-Duplex con una capacidad de ancho de banda en tiempo real de 20MHz.
- Conectividad USB 2.0.
- Compatibilidad con GNU Radio.



Figura 28- Dispositivo HackRF One.

El último dispositivo utilizado en el proyecto es el Receptor Dvb-t, modelo 820T2.



Figura 29- Dispositivo Dongle RTL.

Este fue utilizado para captar las frecuencias portadoras al momento de enviar información con los dispositivos.

Algunas características son:

- Rango de sintonización de 25 MHz-1700 MHz.
- Canal táctil de escaneo de ancho de banda 6/7/8 MHz.
- Compatibilidad con GNU Radio.

A lo largo del capítulo, se definió qué tipo de comunicación se va a establecer entre los dos dispositivos. Además, se explican las modulaciones, con sus respectivas características, utilizadas en la transmisión y recepción de información. Se realiza una explicación teórica del funcionamiento de FHSS y de ciertos aspectos a tener en cuenta al realizar la transmisión de datos de forma inalámbrica. Recorriendo todo el hardware y software necesario.

Próximamente, se hará énfasis en la parte práctica del proyecto y cómo se dividió el mismo.

Metodología

Habiendo desarrollado los conceptos teóricos necesarios para interpretar los resultados finales y parciales que se obtuvieron. Pasamos a explicar cómo se decidió llevar adelante el proyecto.

Las tareas a realizar se dividen en 5 grandes grupos:

- 1) Preparación del laboratorio de trabajo.
- 2) Armado del transmisor simulado.
- 3) Armado de receptor simulado.
- 4) Implementación en placas.
- 5) Separación del transmisor y receptor.

Estos pasos se detallan a continuación.

Simulación en GNU Radio

Preparación del laboratorio de trabajo

Antes de comenzar a diagramar el transmisor, se preparó el ambiente de trabajo. Para el desarrollo, se eligió crear una máquina virtual desde el software Virtualbox. A la hora de montar un sistema operativo, fue escogido Ubuntu 20.04. Se procede a utilizar dicha versión por su estabilidad y abundante documentación. Finalmente, se instala el software GNU Radio y todas las librerías necesarias para su correcto funcionamiento. Además, se instalan librerías necesarias para la compatibilidad del dispositivo USRP B210.

Una vez instalado lo mencionado anteriormente, se conecta el dispositivo ettus a la notebook de trabajo.

realizar el filtrado de la señal en el dominio de la frecuencia para eliminar componentes no deseados o interferencias.

- 5) Python Block: Se trata de un bloque que ejecuta un script de python utilizado para controlar la frecuencia de desvío aplicada al filtro FIR.
- 6) Mutiply Const: multiplica cada muestra de la señal de entrada por una constante específica.
- 7) QT GUI Waterfall Sink: Analizador de espectro en tiempo real.
- 8) Function Probe: Llama constantemente a la función “change_channel”, del bloque python, para tomar los diferentes canales al momento de transmitir la señal.
- 9) QT GUI Check Box: Proporciona la opción de activar o desactivar variables en un entorno gráfico.

Luego se le añade ruido a la señal modulada, correspondiente a los bloques de la parte superior:

- 1) Noise Source: Generador de ruido. En este caso Gaussiano.
- 2) Throttle: Limita la velocidad a la que los datos fluyen a través de un flujo de procesamiento de señales digitales.
- 3) FFT Low Pass Filter: Implementa un filtro pasa bajos, utilizando la transformada rápida de Fourier (FFT).
- 4) Add: Suma dos señales, en este caso la modulada y el ruido.
- 5) Virtual Sink: Para este caso, simula la antena transmisora.

Al comenzar la simulación, se muestra como los paquetes viajan en diferentes frecuencias

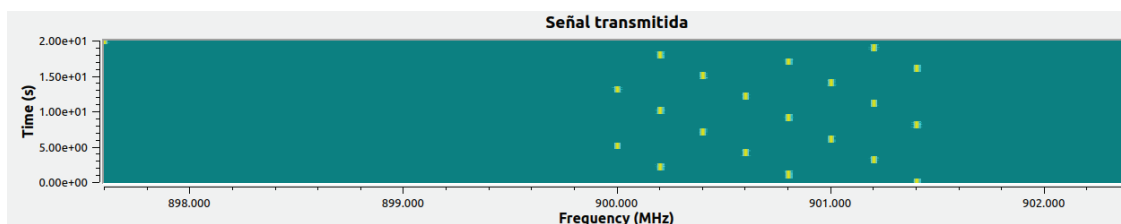


Figura 32- Paquetes viajando en diferentes frecuencias.

Para poder obtener la figura anterior, los bloques Throttle (valor en 4.8M) y Mutiply const (valor en 0.001) son esenciales. Estos permiten visualizar los paquetes de forma agradable a la vista.

Los saltos de frecuencias los genera el bloque que programamos en Python.

```
def changeChannel(self, activateTxFhss):
    if not activateTxFhss:
        #Si no esta activado Se publica corrimiento 0
        PortFreqCommand = pmt.cons(pmt.string_to_symbol("modFreq"), pmt.from_double(0))
        self.message_port_pub(pmt.intern("modFreq"), PortFreqCommand)
        return
    freq = (self.channel + 3)%8
    freq = freq * -200e3
    freq1 = freq * -1
    PortFreqCommand = pmt.cons(pmt.string_to_symbol("Freq"), pmt.from_double(freq))
    PortFreqCommand1 = pmt.cons(pmt.string_to_symbol("freq1"), pmt.from_double(freq1))
    self.message_port_pub(pmt.intern("modFreq"), PortFreqCommand)
    self.message_port_pub(pmt.intern("modFreq1"), PortFreqCommand1)
    return 1
```

Figura 33- Código generador de saltos en frecuencia.

La figura 33 muestra la función encargada del salto en frecuencia. El método `change_channel` calcula la nueva frecuencia que se envía al filtro, esta frecuencia se obtiene calculando un entero a partir de una función matemática y multiplicando por 200 KHz . Si FHSS no está activado, la información se transmite por la frecuencia central predefinida. Si FHSS está activado, calcula la nueva frecuencia de desplazamiento a partir del último desplazamiento obtenido. Además, existe un bloque que llama a la función que cambia de canal. Este tiene un poll rate de 1 Hz, es decir que el canal cambia cada un segundo.

Receptor

Para el receptor, se utilizó el siguiente diagrama de bloques.

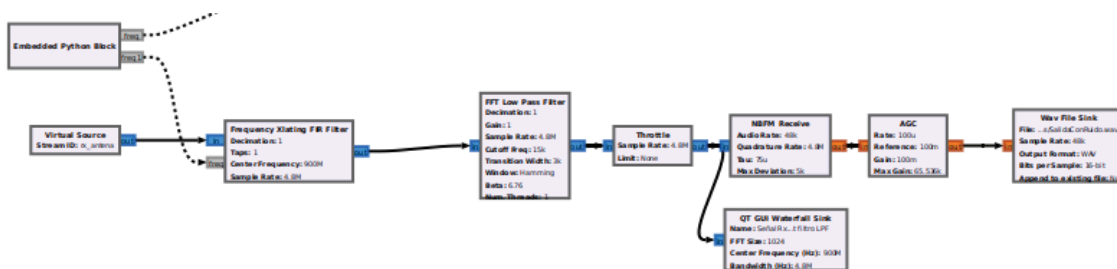


Figura 34- Receptor FHSS simulado.

- 1) Virtual Source: Receptor del bloque “virtual sink”, es como conectar un cable entre ambos equipos.
- 2) NBFM Receive: Toma un único flujo de entrada de banda base complejo y produce un único flujo de salida flotante de muestra de audio

- 3) AGC: Ajusta automáticamente la ganancia de una señal recibida para mantener un nivel de amplitud constante.
- 4) Wav File Sink: Genera un archivo “.wav” para guardar el audio recibido.

Al enviar un archivo WAV, este es guardado en uno nuevo. Se puede escuchar el audio con un nivel de ruido. Los audios están incluidos en un drive, específicamente en la carpeta “Audios Simulación”. El acceso al Drive se encuentra en el anexo, al final del informe. Hasta este momento verificamos la correcta transmisión y recepción de la simulación en GNU Radio.

Implementación en hardware (audio)

Pruebas de funcionamiento

Para la primera prueba de implementación, trabajamos con el Receptor Dvb-t y la USRP B210. Además, modificamos los bloques de la simulación para que la transmisión la realice este último.

Luego preparamos el receptor para visualizar la portadora, se utiliza SDRSharp. Provee una interfaz gráfica para configurar diferentes parámetros del dispositivo.

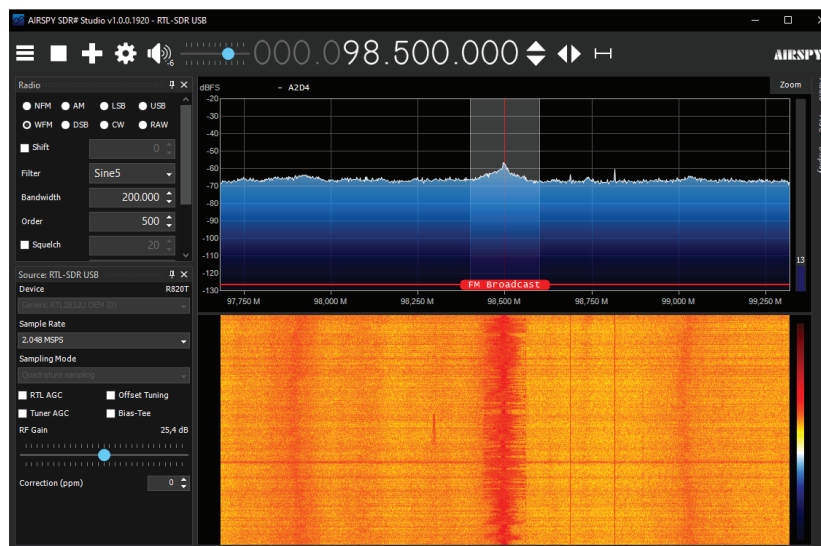


Figura 35- Software SDRSharp.

Antes de transmitir la información mediante FHSS, se transmite un tono de 1GHz a través del dispositivo USRP.

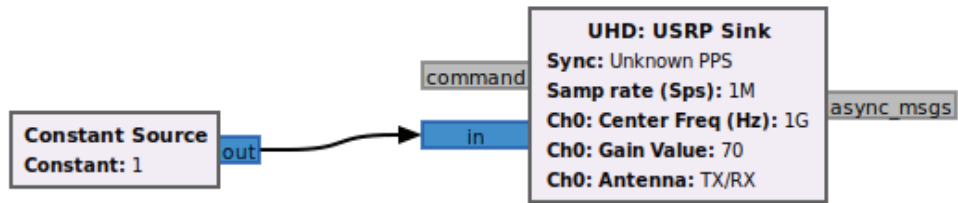


Figura 36- Transmisión de un tono.

Para ello, se preparan los bloques en GNU Radio, y en el software SDRSharp se visualiza el tono.

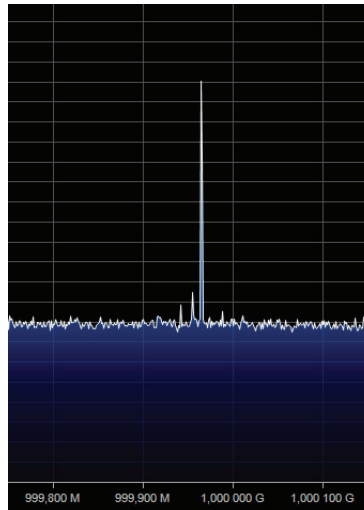


Figura 37- Recepción de un tono.

Una vez entendido cómo se trabaja con la señal, se transmite un audio para sintonizarlo y escucharlo a través del RTL con el software SDRSharp.

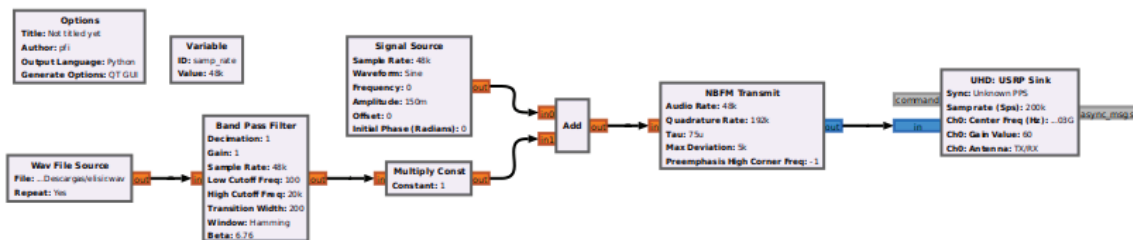


Figura 38- Transmisión de audio con hardware.

Los nuevos bloques son:

- 1) Signal Source: Genera una variedad de formas de onda. En este caso, senoidal.

- 2) NBFM Transmit: Produce una única salida de banda base compleja modulada en FM, de banda angosta.
- 3) USRP sink: Bloque utilizado para transmitir muestras desde un dispositivo USRP.

Al utilizar hardware, no hace falta el bloque "Throttle". En un entorno de simulación pura, los bloques de procesamiento pueden ejecutarse tan rápido como la CPU lo permita. Esto puede causar que GNU Radio consuma todos los recursos de la CPU, lo que puede ralentizar o bloquear el sistema. El dispositivo USRP ya controla la tasa de muestreo real. Opera a una tasa de muestreo específica y proporciona o consume datos a esa tasa.

Luego de varias pruebas, se llega a la conclusión de que en el bloque USRP Sink es necesario tener una frecuencia de muestreo de 200 KHz para recibir el audio de forma correcta. Además, configurar una ganancia en el rango de 50 hasta 70 dB.

Es importante mencionar que la frecuencia central a la que se configura el bloque USRP Sink, es de 1 GHz sumándole 34 KHz. Ya que, si se coloca en 1 GHz la portadora se encuentra desfasada 34 KHz de forma negativa. Luego, al sintonizar 1 GHz en el SDRSharp, se logra escuchar el audio.

En conclusión, luego de mandar un tono con la frecuencia corregida y poder transmitir un audio modulado en NBFM, se implementan ambas funciones para modificar lo simulado en GNU Radio.

Transmisor en Hardware

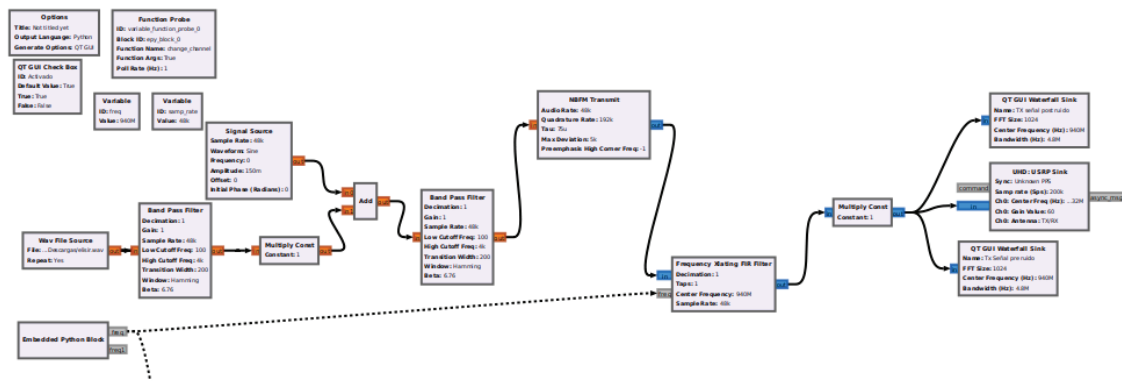


Figura 39- Transmisión FHSS con hardware.

Utilizamos los conocimientos adquiridos de las simulaciones para implementar un diagrama de transmisor mejorado. Se pueden ver los saltos de frecuencias desde el SDRSharp. El video se encuentra en la carpeta “Video SDRSharp FHSS” del drive, especificado en el anexo. Es importante destacar que la frecuencia central es de 940 MHz.

En un principio, el software SDR Sharp detecta la señal. Tanto con el FHSS activado y desactivado. Al momento de armar el receptor, se hicieron ciertas modificaciones al transmisor. En este caso se simplificó lo más posible para optimizarlo.

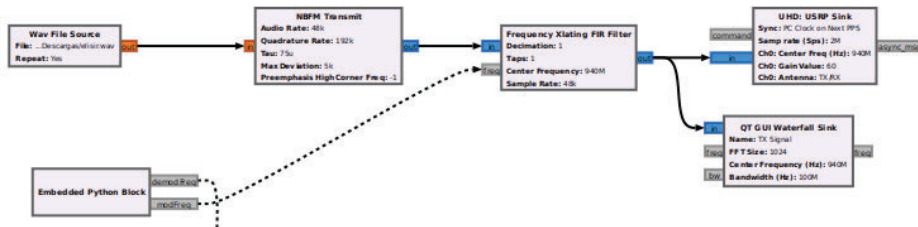


Figura 40- Transmisor FHSS optimizado.

Solo consta de 5 bloques, ya explicados anteriormente. El único cambio, es que se transmite a 2 millones de muestras por segundo. Esto último, ayuda a mejorar la calidad y la precisión de la señal reduciendo el aliasing.

Posteriormente se procede al armado del receptor. Para ello se usa una segunda antena en el USRP B210. El cambio de dispositivo de receptor se debe a que se necesita muestrear a 4 millones de muestras por segundo, el dongle RTL no llega a cubrir dicho muestreo. Se utiliza una tasa de muestreo alta en el receptor, debido a que proporciona una mayor resolución temporal. Lo que significa, que más detalles de la señal se capturan en cada ciclo de muestreo. Permitiendo una sincronización correcta de los saltos en frecuencias.



Figura 41- USRP B210 Tx y Rx.

Receptor en Hardware

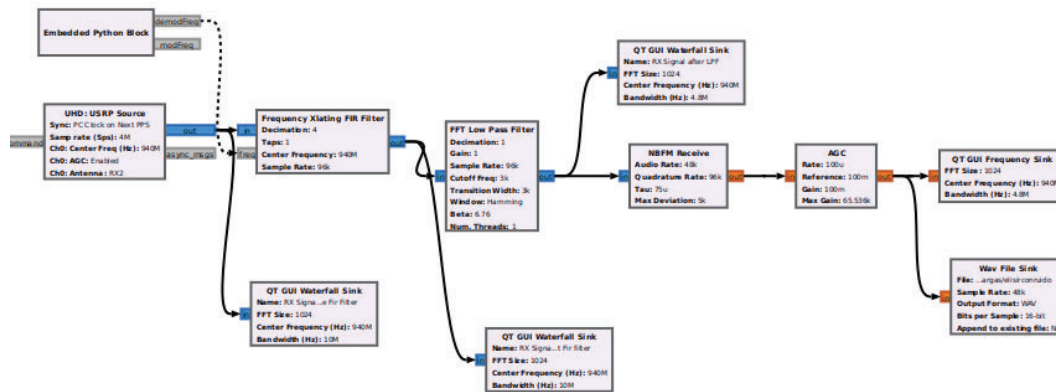


Figura 42 - Receptor FHSS con hardware.

En la figura 42, se observa el armado del receptor. En donde se rearma la señal en el FIR Filter para luego eliminar el ruido con un filtro pasa bajos.

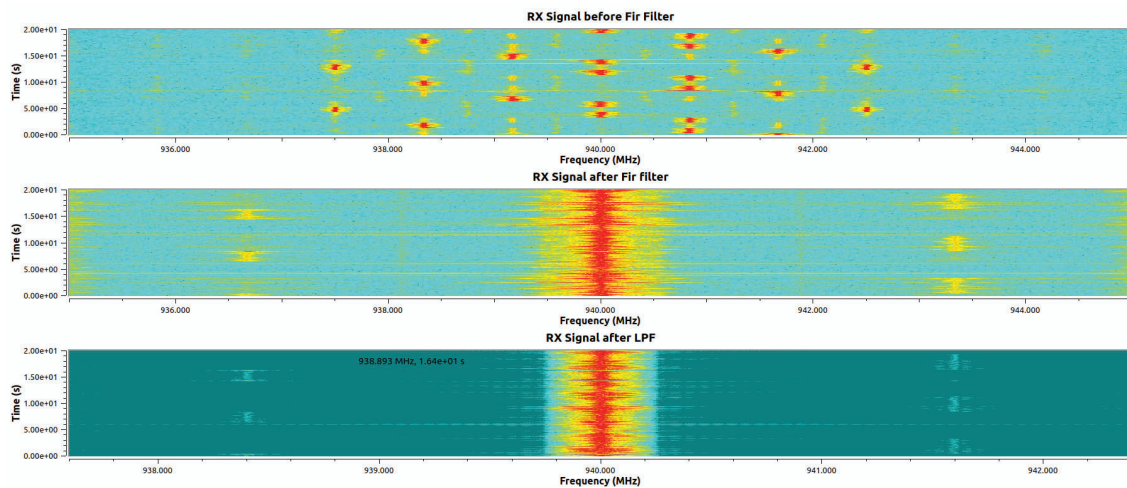


Figura 43- Proceso de recepción de la señal.

En la figura anterior se observa que antes del FIR Filter se reciben los paquetes en diferentes frecuencias. Luego, el bloque reconstruye la señal en la frecuencia central de 940 MHz y elimina parte del ruido. Finalmente, el filtro pasa bajos elimina el ruido restante para obtener mayor calidad de señal.

El audio transmitido se encuentra en el drive adjunto en el anexo, precisamente en la carpeta “Audios Placas” (solo se envía una vez el audio, para que no se repita infinitamente).

Hasta el momento, la transmisión y recepción funciona dentro de la misma máquina virtual. Para recrear un entorno aún más realista, el siguiente paso es tener el receptor y transmisor en diferentes dispositivos.

Separación del transmisor y receptor en Hardware

Para lograr una correcta sincronización entre ambos dispositivos se hicieron algunas mejoras al diagrama de bloques.

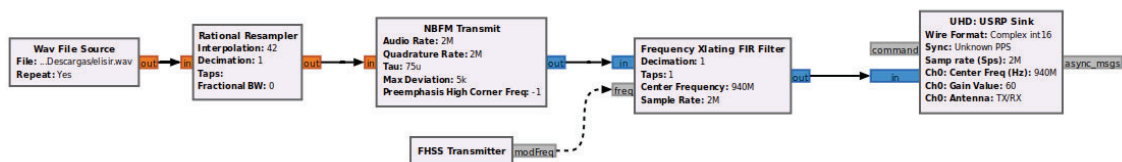


Figura 44- Transmisor optimizado.

En la imagen anterior, se muestra la versión final del transmisor. Se lleva a cabo un resamplio del audio, llevándolo a una tasa de muestreo de 2 millones de muestras.

El cambio más significativo es la optimización del código "FHSS Transmitter", en este caso, se modificó la lista de desplazamiento de frecuencias. La lista contempla desde 0 Hz a 700 kHz, variando de a 100 kHz. Es decir, 8 desplazamientos de frecuencia respecto de la central (en este caso, 940 MHz). Este método simplifica la recepción de la información.

```
def changeChannel(self, activateTxFhss):
    possibleChannels = [0, -100e3, -200e3, -300e3, -400e3, -500e3, -600e3, -700e3]
    if not activateTxFhss:
        #Si no esta activado Se publica corrimiento 0
        PortFreqCommand = pmt.cons(pmt.string_to_symbol("modFreq"), pmt.from_double(0))
        self.message_port_pub(pmt.intern("modFreq"), PortFreqCommand)
        return
    self.chIndex = (self.chIndex + 3)%8 #Generador de Indice
    centerFreq = possibleChannels[self.chIndex]
    PortFreqCommand = pmt.cons(pmt.string_to_symbol("modFreq"), pmt.from_double(centerFreq))
    self.message_port_pub(pmt.intern("modFreq"), PortFreqCommand)
    return 1
```

Figura 45- Código optimizado.

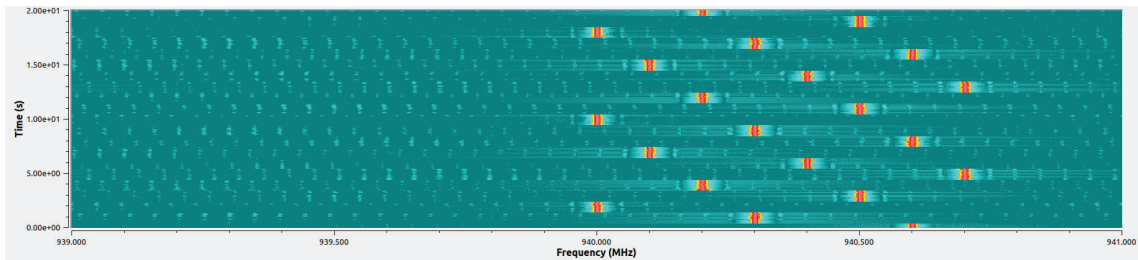


Figura 46- Paquetes saltando en frecuencia.

En la imagen 46, se detallan los paquetes con su corrimiento en frecuencia.

El receptor también fue modificado para adaptarse al cambio de frecuencias en el transmisor.

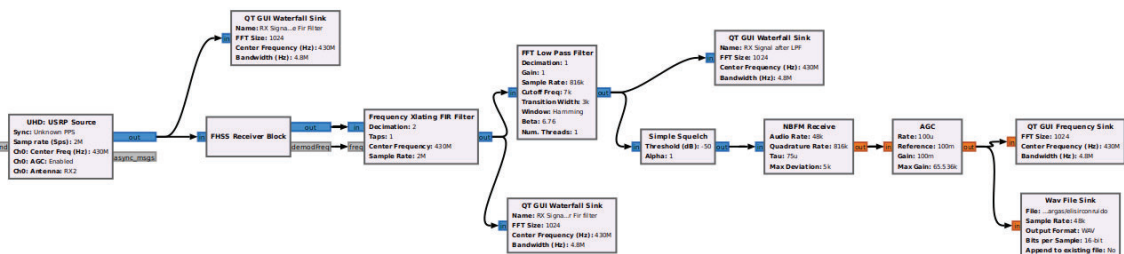


Figura 47- Receptor optimizado.

El “FHSS Receiver” recibe como entrada los datos del receptor, estos son procesados mediante una transformada rápida de Fourier para poder encontrar la frecuencia de transmisión en la lista de posibles frecuencias. La lista es precargada en el receptor. El código de detección y procesamiento envía como parámetro la frecuencia de desplazamiento al filtro FIR. También son pasados al siguiente bloque los datos sin procesar, esto se hace con el objetivo de mantener la sincronización de datos y de frecuencia de pico que se envía al filtro.

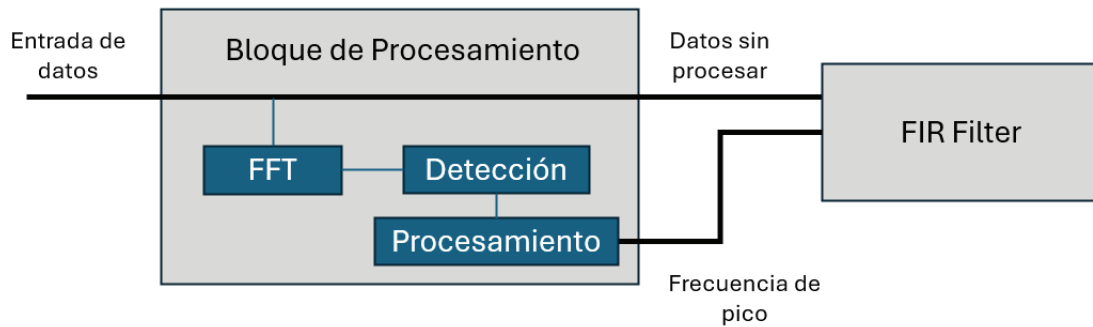


Figura 48- Representación gráfica del bloque FHSS Receiver.

La imagen anterior es una demostración gráfica de lo explicado anteriormente. Para detectar el pico se realizan los siguientes pasos:

- 1) Transformada rápida de Fourier (FFT) para captar las frecuencias.
- 2) Se guardan las frecuencias detectadas, denominadas “frecuencias relativas”.
- 3) Se realiza la diferencia entre la frecuencia relativa recibida con la lista de corrimientos.
- 4) Se transmite al FIR Filter la frecuencia de corrimiento más cercana a la frecuencia central.

Como se indicó anteriormente, un atributo fundamental del funcionamiento de este modelo es que el receptor conoce el listado de frecuencias en las que debe efectuar el corrimiento.

```

def detectPeak(self, samples):
    possibleChannels = [0,100e3,200e3,300e3,400e3,500e3,600e3,700e3]

    # Se calcula la FFT de la señal recibida
    fftResult = np.fft.fft(samples)
    # Se calculan los Bins de la FFT
    bins = np.fft.fftfreq(len(samples), 1 / self.samplingRate)
    # Busca el valor máximo de la señal
    peak = np.max(fftResult.real)
    # Calculo la frecuencia a la que se encuentra el maximo
    relativePeakPosition = np.argmax(fftResult.real)
    PeakPosition = bins[relativePeakPosition]
    # Ajusto la frecuencia del maximo al despazamiento conocido mas cercano
    nearest_freq = self.findNearest(possibleChannels,PeakPosition)

    if peak > self.threshold: #Solo continua si el pico hayado es significativo
        P_freq_cmd1 = pmt.cons(pmt.string_to_symbol("demodFreq"), pmt.from_double(nearest_freq))
        self.message_port_pub(pmt.intern("demodFreq"), P_freq_cmd1)
        return samples

    return 0

```

Figura 49- Código detector de picos de frecuencia.

A modo aclaración, se denomina `threshold` a el umbral de ruido. Si la señal recibida no supera ese `threshold`, se lo toma como ruido por lo que no es tomada en cuenta.

Análisis de la señal recibida (FFT)

Mientras se realizaban las pruebas de recepción fue fundamental comprender cómo se detectarían los picos de frecuencia de la señal recibida. Para esto se tomaron como muestra los datos en un instante del receptor y se analizaron con un código en python implementado por fuera de GNU Radio, que permitiría graficar los datos recibidos y la transformada de Fourier de los mismos. Los datos son extraídos del receptor sintonizado en 940 MHz, por lo que todas las mediciones son en torno a dicha frecuencia central.

A continuación se muestran dos muestras de la señal. La primera es una muestra cuando la señal de audio se transmite en una de las posibles frecuencias desplazadas.

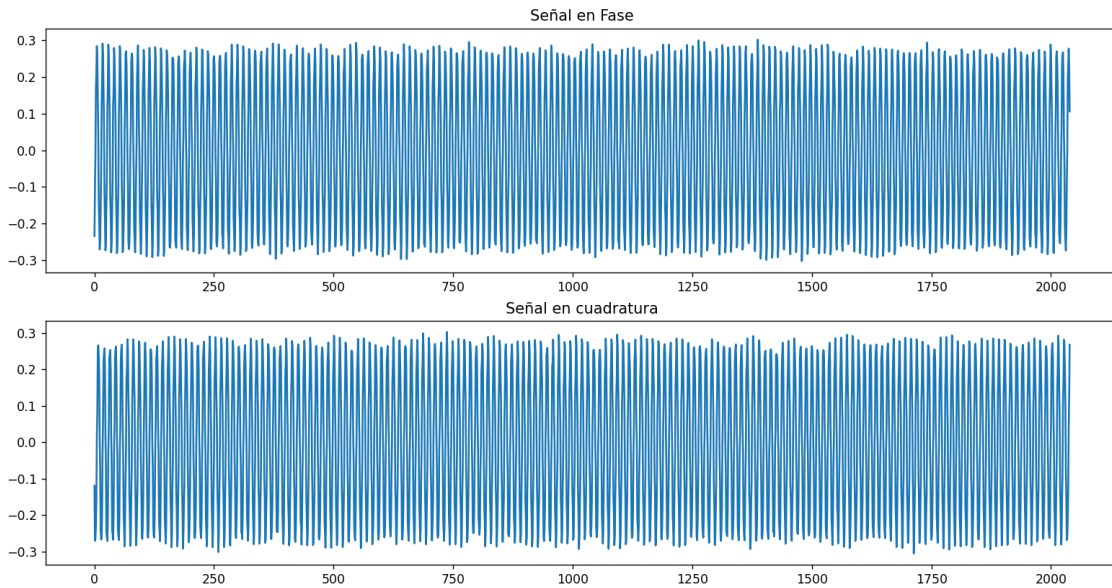


Figura 50- Muestra de la señal en un instante de tiempo cualquiera

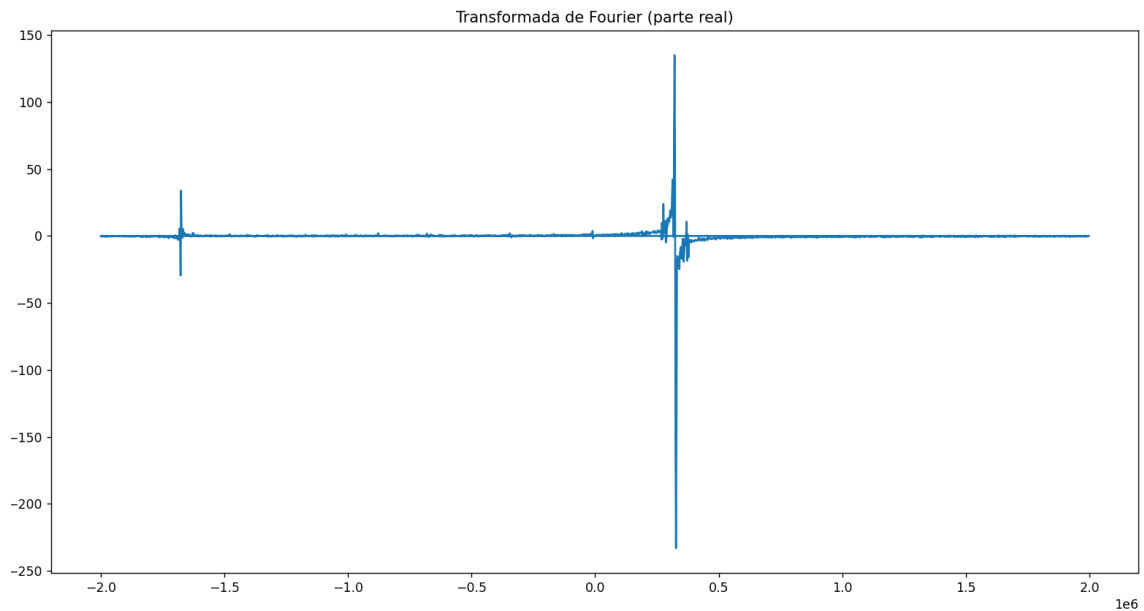


Figura 51- Transformada de Fourier de la señal en un instante de tiempo cualquiera.

En la imagen anterior se puede observar que el pico más alto de la transformada de Fourier aparece en torno a los 400 KHz.

La segunda muestra tomada se buscó para que coincidiera con la señal siendo transmitida en la frecuencia central, es decir, sin desplazamiento.

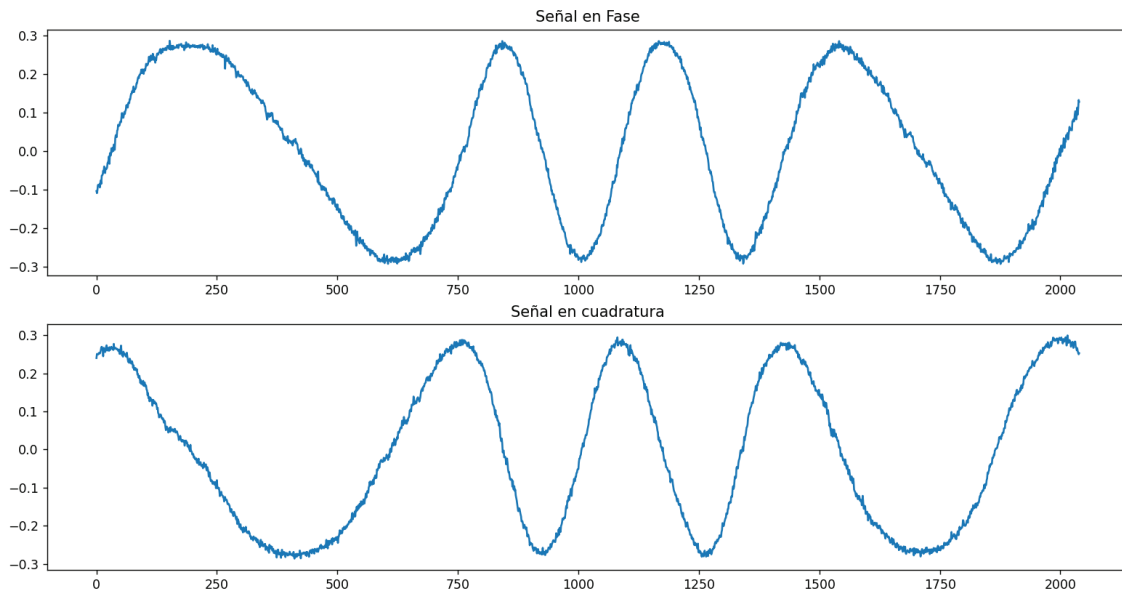


Figura 52- Muestra de la señal cuando se transmite en frecuencia central

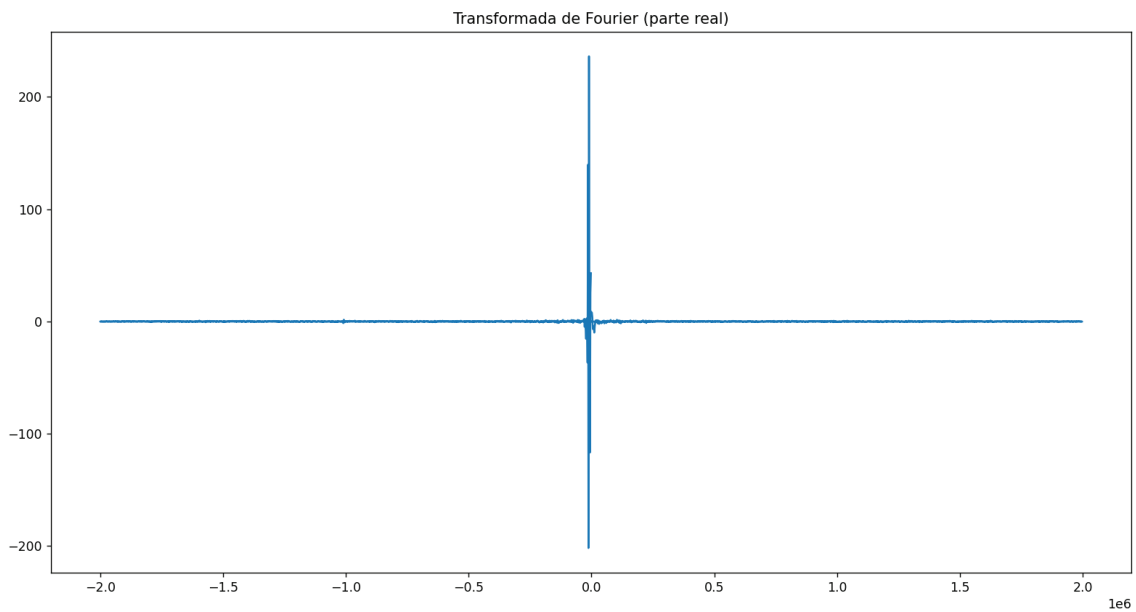


Figura 53- Transformada de Fourier de la señal cuando transmite en frecuencia central.

En este caso, se observa que en el dominio del tiempo, la señal recibida varía su frecuencia en función del tiempo, recordemos que se está transmitiendo con una modulación FM. Además, el pico de la señal transformada se encuentra en torno a los 0 Hz.

La detección y el procesamiento del Bloque “FHSS Receiver” se encargan de tomar los datos, procesarlos de esta manera y hallar la posible frecuencia de desplazamiento más cercana al pico.

Ejemplo de funcionamiento de receptor

Con el objetivo de repasar los procedimientos que se realizan en el receptor, se detalla el siguiente ejemplo.

- 1) La frecuencia central es 940 MHz
- 2) La lista de corrimiento es: [0 Hz,100 kHz,200 kHz,300 kHz,400 kHz,500 kHz,600 kHz,700 kHz]
- 3) La frecuencia relativa recibida es: 940.120 kHz (940,120 MHz)

Se realiza la diferencia entre la frecuencia recibida y todas las frecuencias de la lista de corrimiento. Al hacer eso, la frecuencia más cercana a 940 MHz (frecuencia central) es el corrimiento de 100 kHz. Entonces, la frecuencia que se transmite al filtro FIR (centrado en 940 MHz) es esta última mencionada. De esta manera, el receptor se sintoniza en 940.100 kHz (940,1 MHz).

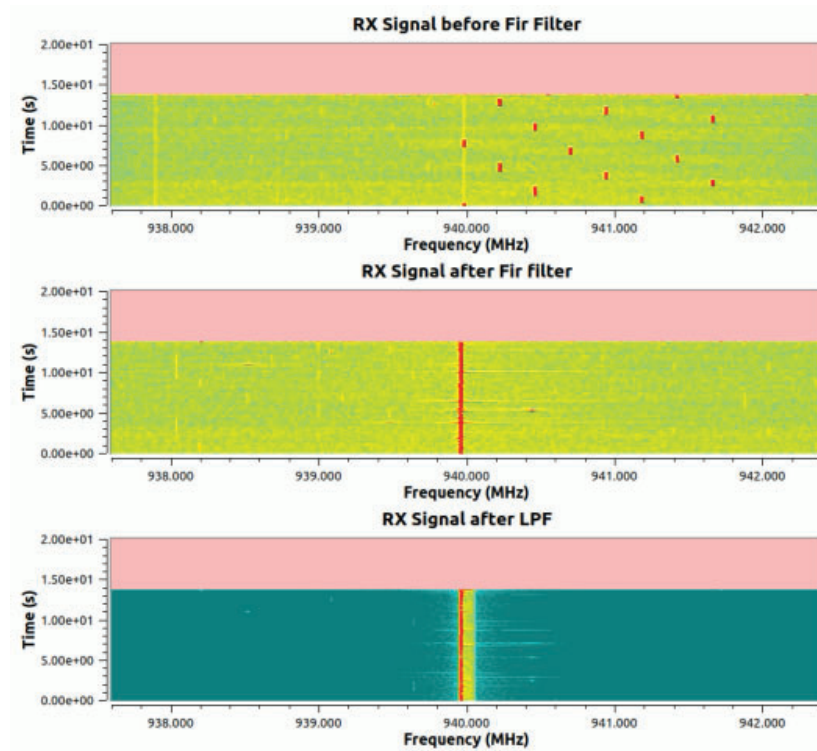


Figura 54- Recepción de paquetes.

Implementación en hardware (texto e imagen)

Para la transmisión de texto e imágenes es necesario codificar lo que se desea transmitir. En este paso aparecen dos nuevos bloques, denominados encoder y GMSK Mod.

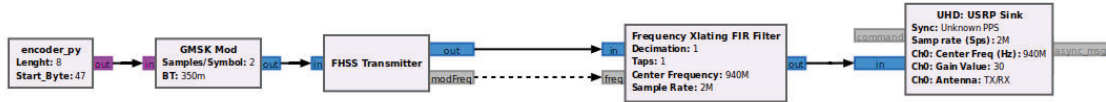


Figura 55- Transmisor adaptado.

El bloque encoder, además de leer un archivo de texto, se encarga de dividir el mensaje en paquetes más pequeños. Adicionalmente, se le agrega un byte de inicio, un CRC (verificación por redundancia cíclica) y un número de orden al paquete. El byte de inicio permite identificar paquetes, el CRC ayuda a descartar errores, mientras que el número de orden, valga la redundancia, permite al mensaje llegar ordenado. La lógica del salto en frecuencia es idéntica a la transmisión del audio.

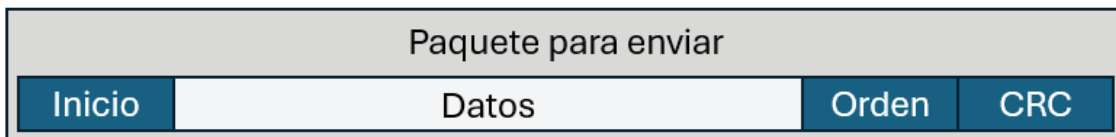


Figura 56- Paquete transmitido.

Por otro lado, el GMSK mod se encarga de modular la información por desplazamiento mínimo gaussiano.

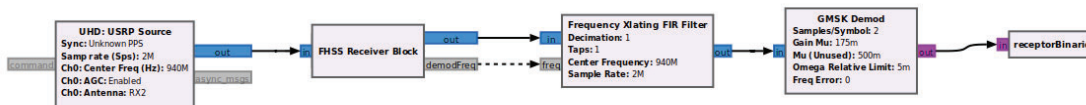


Figura 57- Receptor adaptado.

En la figura 57, se aprecia el receptor. Una vez sintonizados los paquetes y demodulados, son almacenados en un archivo binario. Esta tarea la realiza el bloque denominado “receptor binario”. Una vez generado el archivo, es decodificado por un script por fuera del GNU Radio. El motivo de realizar la decodificación por separado, se debe al procesamiento de la

máquina virtual. Decodificar la información le exige cierto nivel de procesamiento, lo que provoca que la máquina deje de responder.

En un principio, se transmitió un archivo de texto sin CRC y luego con CRC. Esto se realizó con el objetivo de comprobar el funcionamiento de esta técnica de detección de errores. El resultado concluyó que el archivo decodificado sin CRC, tiene un tamaño de aproximadamente 202 KB. Mientras que el que cuenta con la técnica de detección de errores tiene un tamaño de 75 KB. Este último sólo contiene el mensaje puro, es decir no tiene ningún carácter con error, entendiéndose a la perfección.

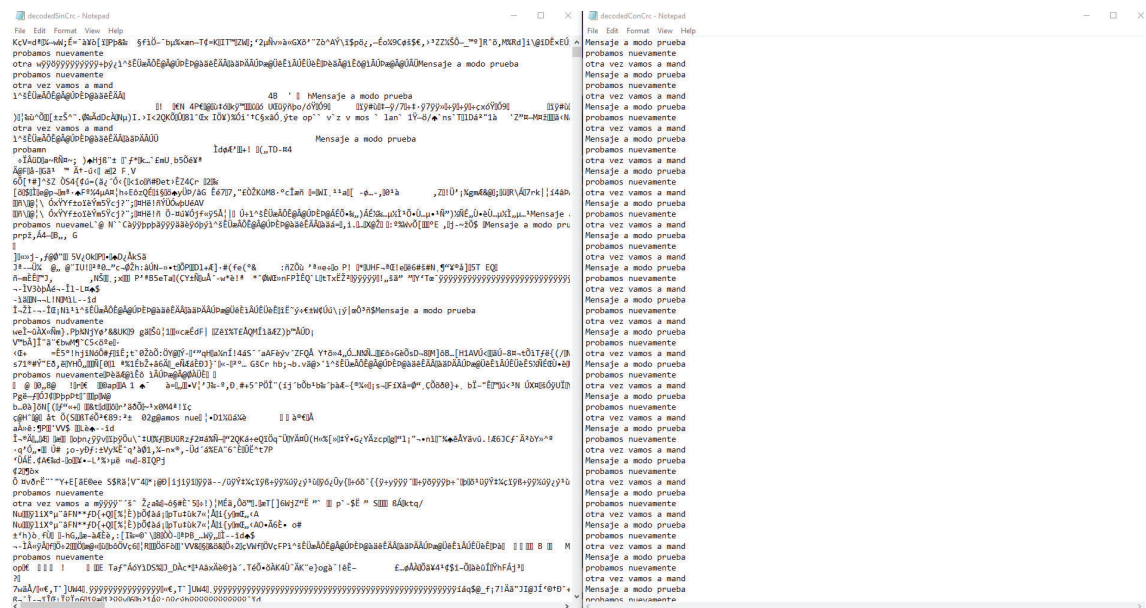


Figura 58- Texto sin CRC vs con CRC.

Aclaración importante, el texto transmitido es:

“Mensaje a modo prueba
problemas nuevamente
otra vez vamos a mand”

Una vez comprobando el funcionamiento del CRC, se dispuso escribir solamente una vez el mensaje. Con el fin de mejorar la decodificación.

Retomando, el decodificador se encarga de analizar todo el archivo, realizando los siguientes pasos:

- 1) Lectura del archivo binario.
- 2) Búsqueda del byte de inicio.
- 3) Almacena desde el byte de inicio hasta el byte de orden.
- 4) Ordena los paquetes recibidos, quedando con uno de cada orden (si llegan dos paquetes con número de orden 2, se queda con solo uno).
- 5) Comprueba que el CRC sea correcto.
- 6) Escribe en un archivo de texto los paquetes.

Transmisión de una imagen

La transmisión de una imagen a través de este sistema puede parecer un proceso diferente a priori, sin embargo se logró implementar utilizando los mismos componentes que para transmitir un texto luego de realizar un pre-procesado a la imagen. Para lograr la transmisión fue necesario definir algunas consideraciones:

- 1) La imagen enviada es de un tamaño de 30x30 Píxeles (900 Bytes).
- 2) Al realizar pruebas se encontró que enviar paquetes constituidos únicamente por secuencias de 1's resultaba en errores y pérdidas de paquetes. Por lo tanto se evitó el uso de imágenes con fondo blanco (El color blanco se representa en forma hexadecimal como #FFFFFF).

Teniendo en cuenta lo anteriormente mencionado, la imagen es traducida a un archivo de texto que contiene los valores de luminancia de la imagen. Esto se realiza con con el siguiente script de Python implementado por fuera de GNU-Radio.

```
def read_image():  
    #convierte la imagen en un texto que puede ser interpretado por el transmisor  
    img = im.open('autoGris.png')  
    numpydata = np.asarray(img)  
    numpydata = np.reshape(numpydata, (900))  
    print(numpydata)  
    f = open("msjToBeSent.txt", "ab")  
    f.write(numpydata)  
    f.close()
```

Figura 59- Pre-procesado de la imagen.

Habiendo generado un archivo de texto con los valores de luminancia de la imagen (números enteros), se pasa el archivo al transmisor. Se realiza una transmisión en paquetes de 10 Bytes. Luego, el proceso es idéntico al de un texto normal. Una vez recibido y decodificado, se hace el proceso inverso para recuperar los valores de luminancia del archivo binario y se acomodan en una matriz de 30 x 30 para poder guardar la imagen.

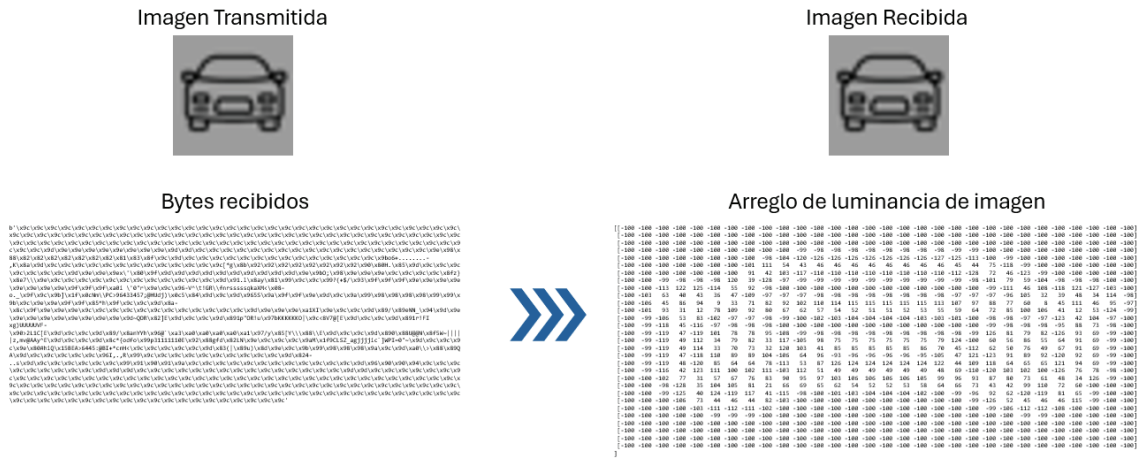


Figura 60- reconstrucción de la imagen.

Todos los scripts utilizados para procesar el archivo binario y recuperar la imagen se ejecutan por fuera del entorno de GNU-Radio y se encuentran adjuntos en los archivos de Drive.

De esta forma podemos ver que el proceso de transmisión y protocolo desarrollado para recuperar la información transmitida es compatible con cualquier tipo de archivo si este es pre-procesado para adaptarlo al encoder.

Una vez establecida la comunicación de forma correcta, en el próximo capítulo procederemos a evidenciar las pruebas de distancia. Para poder llegar a conclusiones más detalladas.

Pruebas de distancia

A modo laboratorio se realizaron pruebas de transmisión y recepción. Primero se utilizó el dispositivo Hack Rf como transmisor. La recepción la efectuó la USRP. En cada prueba se modificó la distancia entre los dispositivos como así también se le agregaron obstáculos. Luego se intercambiaron los dispositivos y se realizaron las mismas pruebas. Se transmitió la información alrededor de los 940 MHz y 430 MHz.

Audio

Las distancias se establecieron en: 75 cm, 1.5 metros, 3 metros, 5 metros, 5 NLOS (sin línea de vista), 10 metros NLOS. La última prueba no se pudo realizar con línea de vista, debido al espacio físico del laboratorio. Todos los videos mostrando el detalle de la comunicación con sus respectivos audios, se encuentran subidos al drive adjunto en el anexo. Precisamente en la carpeta "Transmisor y Receptor separados".

Tx Hack Rf - Rx USRP

Inicialmente en las primeras pruebas, el audio llega sin ningún problema. A partir de los 5 metros, se puede notar la degradación de la señal. Sin embargo, el audio se recibe y es claro al reproducirlo. Por más que se agreguen obstáculos (dos puertas de madera), la comunicación sigue siendo efectiva. A partir de los 10 metros ya la señal no puede ser captada por el receptor, debe atravesar vidrios y la distancia mencionada.

Hasta ese momento, la frecuencia central era de 940 MHz. Para poder lograr que llegue la información, se configuró el transmisor y receptor con una frecuencia central de 430 MHz. Es decir, los paquetes se transmiten en una frecuencia más baja. Proporcionando una mayor resistencia a la distancia y obstáculos. Si bien el audio no se escucha de la mejor forma, se logra entenderlo. Si la antena tuviese más potencia, la calidad del audio sería superior.

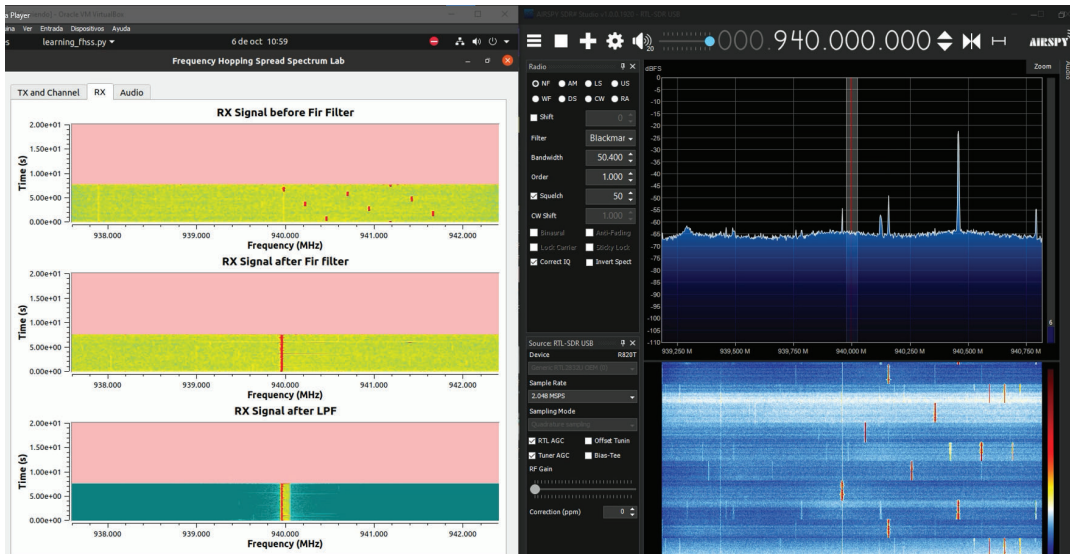


Figura 61- pruebas con una distancia de 1.5 metros.

Tx USRP- Rx Hack Rf

Como en el caso anterior, en las pruebas iniciales no se tuvo complicaciones. Es importante remarcar que los audios recibidos son de mejor calidad. Esto es porque el dispositivo USRP tiene mayor potencia para transmitir. En la prueba de los 10 metros se aprecia parte del audio, pero en algunos momentos no es entendible. Nuevamente se configura en 430 MHz la frecuencia central, esto nos permite recuperar la señal y calidad del audio. Para agregarle una mayor complejidad, se aisló el receptor. En esta prueba, la señal tendría que atravesar un obstáculo de metal. Por más que se tenga como frecuencia central 430 MHz, no se puede reconstruir el audio.

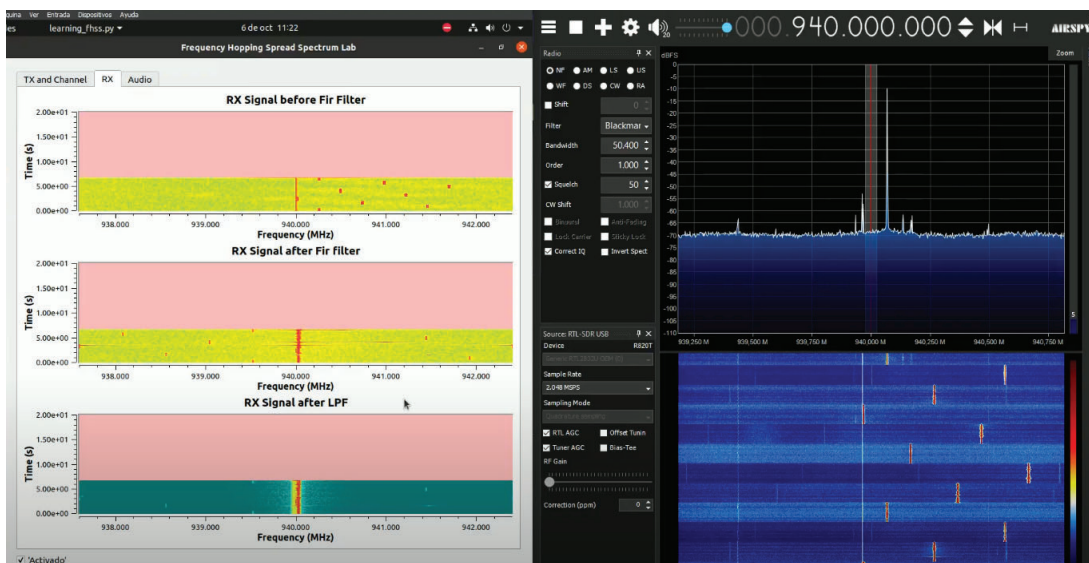


Figura 62- pruebas con una distancia 1.5 metros

Como breve conclusión se puede destacar que el dispositivo USRP funciona mejor como transmisor, debido a que tiene más potencia. A su vez, bajar la frecuencia central permite una mayor resistencia a la distancia y obstáculos entre dispositivos.

Texto e imagen

Las pruebas efectuadas para la transmisión de texto e imagen tuvieron un alcance de no más de 1 metro. Comparado con la modulación FM, GMSK es más eficiente en términos de ancho de banda, pero es más susceptible a la atenuación. La modulación digital utilizada, es muy dependiente de la potencia de transmisión y la sensibilidad del receptor.

En resumen, FM tiende a ser más robusto para transmisiones a mayor distancia. Por otro lado, GMSK es más eficiente en términos de espectro, comprometiendo el alcance si las condiciones no son óptimas o si el receptor no es lo suficientemente sensible.

En el video adjunto en el anexo, apenas se pueden identificar que se está transmitiendo información. Comparado con los videos de transmisión y recepción del audio, se concluye que la modulación FM utiliza mejor la potencia provista por los dispositivos.

A modo resumen de las pruebas realizadas, se comprobó el correcto funcionamiento de la comunicación entre dispositivos llegando al límite de recepción de la señal. Los resultados son coherentes, ya que respetan la teoría de la pérdida de trayectoria en espacio libre.

A continuación se plantean posibles mejoras al proyecto.

Trabajo futuro

En este capítulo se analizarán puntos de mejora o posibles adicionales que fortalezcan la comunicación entre los dispositivos.

Encriptación

Una mejora significativa para este proyecto sería la implementación de encriptación de datos durante la transmisión. La encriptación garantiza que la información transmitida sea accesible únicamente para los dispositivos autorizados, aumentando así la seguridad y confidencialidad más allá de la que es brindada por la utilización de Frequency hopping Spread Spectrum.

Esta mejora no sólo fortalecería la integridad del sistema, sino que también lo haría más adecuado para aplicaciones críticas en entornos industriales, militares y de IoT, donde la seguridad de los datos es primordial.

Sin embargo se tiene que considerar que las ventajas de encriptar los mensajes transmitidos están atadas a un costo elevado de procesamiento de datos y pueden inducir demoras en los tiempos de respuesta de un sistema que aplique esta tecnología.

Protocolo de comunicación

Como se explicó anteriormente, en este proyecto se implementó un paquete de transmisión que contenía un byte de inicio, los n Bytes de datos, un byte utilizado para reordenar el mensaje en el receptor y 4 Bytes de CRC. Sería una mejora interesante la implementación de un protocolo de transmisión de datos fiable que permita la corrección de errores y la mejora en los tiempos de transmisión.

Transmitir y recibir archivos de mayor tamaño

Como se discutió anteriormente, para la realización de este proyecto, se utilizaron entornos de trabajo implementados en máquinas virtuales. Limitando de esta forma la capacidad de procesamiento disponible, por lo tanto resultó difícil hacer pruebas de transmisión de archivos de mayor tamaño.

Disponiendo de equipos dedicados y mejorando la eficiencia de los scripts de python desarrollados sería posible lograr la transmisión de archivos de mayor tamaño sin que se requiera de tanto tiempo de transmisión.

Etapa de potencia

Una mejora importante sería el agregado de amplificadores a la salida del transmisor o implementar antenas direccionales. De esta manera se mejoraría la potencia de señal en el receptor, incrementando el alcance de la transmisión, mejorando la calidad y reduciendo la tasa de error, y por último preparando los equipos para la implementación en entornos reales.

En este capítulo finaliza el proyecto, lo que procedemos a destacar las conclusiones.

Conclusiones

En un principio, se ha implementado una comunicación inalámbrica utilizando una placa USRP B210, con dos antenas dedicadas para transmisión y recepción. Empleando la técnica de espectro ensanchado por salto de frecuencia. El desarrollo y pruebas del sistema se realizaron en GNU-Radio, comenzando con una simulación virtual antes de la implementación real con las antenas.

En un primero momento se implementaron los bloques “virtual sink” y “virtual source” en GNU Radio, simulando una conexión directa entre el transmisor y el receptor. Esta fase inicial permitió verificar el correcto funcionamiento de la tecnología y ajustar parámetros sin los desafíos adicionales de la transmisión inalámbrica real. Este enfoque iterativo, comenzando con simulaciones y luego avanzando hacia pruebas con hardware real, ha sido crucial para el desarrollo del sistema.

Luego se utilizaron el dispositivo USRP y el Dongle RTL. Ante las limitaciones del último, se procedió a utilizar una segunda antena conectada al USRP. Esto proporcionó una mayor tasa de muestreo en el receptor.

Una vez que se consiguió el dispositivo Hack Rf One. Con este, se pudo empezar a realizar las pruebas en distintas computadoras. En este momento es donde surge el problema de sincronización entre ambos dispositivos (Hack Rf y la USRP). Luego de investigación exhaustiva, se toma la decisión de implementar la transformada de Fourier en el receptor para poder captar la información en el dominio del tiempo y pasarlo al dominio de la frecuencia. Lo que permitió tomar las frecuencias, para que el receptor sepa en qué momento y donde sintonizarse. Por temas de procesamiento y simplicidad, el receptor conoce la lista de las frecuencias en donde tiene que sintonizarse. Esta etapa, nos permitió administrar los recursos disponibles lo más eficientemente posible y entender la importancia de la sincronización entre dispositivos.

Luego surge el desafío de enviar un archivo de texto y una imagen. En este momento nace el codificador, con el objetivo de enviar paquetes más pequeños y no saturar la máquina virtual. Además de fraccionar la información, se le agrega un byte de inicio, un número de orden y un CRC. Estos últimos, permiten garantizar la integridad de la información. Este desarrollo de un “mini protocolo” nos ayudó a entender cómo viajan los paquetes y asentar

la teoría vista a lo largo de toda la carrera. El decodificador, se encarga de analizar el archivo recibido buscando el byte de inicio y corroborando que el CRC sea correcto.

En cuanto a las pruebas de laboratorio, la transmisión y recepción de audio es robusta para los recursos disponibles. Si bien se nota cierto ruido en el audio recibido, se puede entender perfectamente hasta una determinada distancia. Además, se probó que si se utilizan frecuencias más bajas, la señal es más resistente a obstáculos y distancias. Esto es consistente con la teoría de FSPL.

Por otro lado, la transmisión y recepción de texto e imagen sufre más la atenuación. Esto se debe a que GMSK es más dependiente de la potencia de transmisión y la sensibilidad del receptor que FM. Incluso con esa limitación, se puede recuperar un texto y una imagen a una distancia no mayor a 1 metro. Si se aumenta la potencia, la información podría recorrer mayores distancias.

El uso de FHSS ha demostrado ser una elección efectiva para mejorar la resistencia a interferencias y aumentar la seguridad de la comunicación. Esta técnica permite que la frecuencia de transmisión salte de una frecuencia a otra dentro de un rango específico, en este caso alrededor de una frecuencia central de 940 MHz o 430 MHz, siguiendo un código predefinido programado en Python. Esto provoca una mejora en la calidad general de la comunicación.

La transmisión de los archivos con sus diferentes modulaciones fue recibida de forma correcta, demostrando que el sistema puede manejar datos de audio y binarios con eficacia. La tasa de muestreo jugó un papel crucial en la reconstrucción precisa de la señal, asegurando que, por ejemplo, el audio se mantuviera claro y sin distorsiones. La elección de una tasa de muestreo adecuada es vital para cualquier aplicación de comunicación, ya que afecta directamente la fidelidad de la señal recibida.

El proyecto tiene un mayor potencial si se hace énfasis en aumentar la potencia de transmisión e implementar un protocolo de comunicación con corrección de errores. Por supuesto que esto obliga a aumentar los recursos de los dispositivos donde se ejecuta tanto el transmisor como el receptor.

El proyecto queda abierto a futuras mejoras, para aplicaciones más robustas y versátiles en escenarios de comunicaciones inalámbricas complejas.

Agradecimientos

La realización del proyecto final integrador, no hubiera sido posible sin la ayuda de la institución. Por lo que se agradece profundamente a la Universidad Nacional de San Martín.

A el ingeniero Rodolfo Salvatore por la buena predisposición a lo largo de toda la cursada de la carrera ingeniería en telecomunicaciones.

Mención especial al ingeniero Leandro Carmona, quien además de ser profesor, es nuestro tutor de tesis. Estamos agradecidos por el acompañamiento en esta última etapa de la carrera.

A todos los docentes que nos acompañaron a lo largo de la carrera y formaron parte de este proceso tan importante en nuestras vidas. Destacando al ingeniero Marcelo Romeo y a nuestros amigos del último tramo, Mateo e Ignacio.

Emmanuel:

Gracias a mis padres, Laura y Agustin. A mis hermanos Ricky, Juani, Eze, Gio y Vicen. Y a toda mi familia, parte fundamental de este trayecto.

Gracias a mi pareja, por soportar mi aislamiento facultativo y tener mucha paciencia.

Gracias amigos de toda la vida y los que se hicieron en el camino:

Ivan, Juampi, Gordo, Naza, Tincho, Marian, Lucas, Mati, Marshall, Santi, Juli, Sol, Juani, Fusa y Fran.

Gracias a todo el equipo de fútbol 11 de la UNSAM.

Gracias a Xelere, por la flexibilidad entre mi carrera profesional y universitaria.

Juan:

Gracias a mis Padres, Gloria y Dario, a mi hermano Juan Ignacio y a toda mi familia por acompañarme en este proceso desde el día que entré a la universidad por primera vez.

Quiero agradecer a mi pareja, Iari, por bancarme siempre y no dejar que baje los brazos.

Cursar, sobre todo al principio, no hubiera sido lo mismo si no lo compartía con quienes hoy son mis amigos, Maite, Nico Quintana, Nico Passarelli y Camila. Gracias!

A todas las personas que de una u otra manera contribuyeron a la culminación de este trabajo, nuestro más sincero agradecimiento.

Referencias

- [1] Simplex vs. duplex fiber patch cable. (n.d.). BlackBox. Retrieved from <https://www.blackbox.com.mx/mx-mx/page/25065/Recursos/Technical/black-box-explic/a/Fibre-Optic-Cable/Simplex-vs-duplex-fiber-patch-cable>
- [2] FHSS. (2022). Medium. Retrieved from <https://medium.com/networks-security/fhss-frequency-hopping-spread-spectrum-860e49c186e8>
- [3] About GNU Radio. (n.d.). GNU Radio. Retrieved from <https://www.gnuradio.org/about/>
- [4] USRP B210. (n.d.). Ettus. Retrieved from <https://www.ettus.com/all-products/ub210-kit/>
- [5] HackRF One. (n.d.). Great Scott Gadgets. Retrieved from <https://greatscottgadgets.com/hackrf/one/>
- [6] GNU Radio Wiki. (n.d.). GNU Radio. Retrieved from https://wiki.gnuradio.org/index.php/Main_Page
- [7] Electronics Notes. (n.d.). What is GMSK: Gaussian Minimum Shift Keying. Retrieved from https://www.electronics-notes.com/articles/radio/modulation/what-is-gmsk-gaussian-minimum-shift-keying.php#google_vignette
- [8] Sabando, M. V. (2014). Notas de Aplicación FVC: Filtros Variables con Componentes Sintonizables. Retrieved from <http://lcr.uns.edu.ar/fvc/NotasDeAplicacion/FVC-SabandoMariaVirginia.pdf>
- [9] Panama Hitek. (2015). Señales continuas, analógicas, discretas y digitales. Retrieved from <https://panamahitek.com/senales-continuas-analogicas-discretas-y-digitales/>
- [10] DSP Stack Exchange. (2021). Difference between MSK and GMSK. Retrieved from

<https://dsp.stackexchange.com/questions/76167/difference-between-msk-and-gmsk>

[11] Scribd Juan Riofrío. (n.d.). *MSK*. Retrieved from <https://es.scribd.com/document/144197589/msk>

[12] Facet UNT. (2017). Moduladores y transmisores de FM. Retrieved from <https://catedras.facet.unt.edu.ar/ft/wp-content/uploads/sites/123/2017/03/08-Moduladores-y-Transmisores-de-FM.pdf>.

[13] UTN FRC. (2010). Transmisores de FM. Retrieved from <https://www.profesores.frc.utn.edu.ar/electronica/electronicaaplicadaiii/aplicada/cap07fmtransmisores.pdf>.

[14] UPCT. (2010). Tema 3: Modulación y demodulación de frecuencia. Retrieved from https://ocw.bib.upct.es/pluginfile.php/5165/mod_resource/content/1/tema3_mod_y_de_mod_de_frecuencia.pdf.

[15] ResearchGate. (n.d.). FHSS: Frequency hopping spread spectrum transmitter block diagram. Retrieved from https://www.researchgate.net/figure/FHSS-Frequency-hopping-spread-spectrum-transmitter-block-diagram40_fig4_334307035.

[16] ResearchGate. (n.d.). Block diagram of the a transmitter and b receiver of the FHSS system. Retrieved from https://www.researchgate.net/figure/Block-diagram-of-the-a-transmitter-and-b-receiver-of-the-FHSS-system_fig3_329286286.

[17] Download Scientific Diagram. (n.d.). A block diagram of frequency-shift keying (FSK) modulation. Retrieved from

https://www.researchgate.net/figure/FHSS-Frequency-hopping-spread-spectrum-transmitter-block-diagram40_fig4_334307035.

[18] ScienceDirect Topics. (2012). Frequency Shift Keying - an overview. Retrieved from <https://www.sciencedirect.com/topics/engineering/frequency-shift-keying>.

[19] ABC Xperts. (2023). Importancia de la pérdida de trayectoria en espacio libre en el diseño y planificación de radioenlaces. Retrieved from <https://abcxperts.com/importancia-de-la-perdida-de-trayectoria-en-espacio-libre-en-el-dise-no-y-planificacion-de-radioenlaces/>.

[20] ResearchGate. (n.d.). Block diagram of SDR platform. Retrieved from https://www.researchgate.net/figure/Block-diagram-of-SDR-platform_fig2_349414166.

[21] Electronics Notes. (n.d.). What is GMSK Modulation: Gaussian Minimum Shift Keying. Retrieved from <https://www.electronics-notes.com/articles/radio/modulation/what-is-gmsk-gaussian-minimum-shift-keying.php>.

[22] Electronics Notes. (n.d.). What is MSK: Minimum Shift Keying Modulation. Retrieved from <https://www.electronics-notes.com/articles/radio/modulation/what-is-msk-minimum-shift-keying.php>.

Anexos

[1] Documentación de pruebas. (2024). Retrieved from

<https://drive.google.com/drive/u/1/folders/1WSOyguD9lQNUOr2OOIxGLjsKMdhzVGwj>

[2] Códigos. (2024). Retrieved from

https://drive.google.com/drive/folders/1_vgQiD81kqn9cUHQfDFiAR3TP0wYoifc?usp=drive_link