



**UNIVERSIDAD INDUSTRIAL DE SANTANDER**

ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES  
Perfecta Combinación entre Energía e Intelecto

**DISEÑO DEL MANUAL DE PRÁCTICAS PARA UN LABORATORIO DE  
COMUNICACIONES DIGITALES BASADO EN LA TÉCNICA DE RADIO  
DEFINIDO POR SOFTWARE**

**JAVIER MAURICIO SUÁREZ MONSALVE**

**VIVIAN PAOLA TRIANA GALEANO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS  
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES  
BUCARAMANGA**

**2012**



**UNIVERSIDAD INDUSTRIAL DE SANTANDER**

ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES  
Perfecta Combinación entre Energía e Intelecto

**DISEÑO DEL MANUAL DE PRÁCTICAS PARA UN LABORATORIO DE  
COMUNICACIONES DIGITALES BASADO EN LA TÉCNICA DE RADIO  
DEFINIDO POR SOFTWARE**

**JAVIER MAURICIO SUÁREZ MONSALVE**

**VIVIAN PAOLA TRIANA GALEANO**

**Proyecto de investigación presentado para obtener el título de:  
Ingeniero(a) Electrónico(a)**

**Director**

**MsC. Jorge Hernando Ramón Suárez**

**Codirector**

**Ing. Diego Rafael Medina Pulido**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECAÑICAS  
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES  
BUCARAMANGA**

**2012**



*“Me pregunto –dijo- si las estrellas están escondidas a fin de que cada uno pueda encontrar la suya algún día.”*

*“..., en toda su vida, no había renunciado a una pregunta, una vez que la había formulado.”*

*Antoine de Saint-Exupéry*

*A esas personas que fueron, son y serán parte de mi vida...*

*Gracias.*

*(='.'=)*

*Vivian ::*

*En primer lugar quiero agradecer a Dios por mi madre, la mujer más maravillosa del mundo, trabajadora, luchadora, humilde, por la cual soy lo que soy y estoy ahora. Quien sin su esfuerzo y sacrificio no alcanzaría este gran logro en mi vida y que ahora celebra en el cielo este pequeño paso alcanzado por mi, todos mis triunfos presentes y futuros se los dedico a ella.*

*A mi hermana por darme el soporte y la motivación de siempre seguir adelante y acompañarme en todos los momentos de mi vida.*

*A mis amigos, Yuri, Martha, Erika, Astrid, Christian, Aquimin y Juan Pablo, quienes han sido un constante apoyo y me han sacado sonrisas en las situaciones difíciles y han estado a mi lado todo este tiempo de gran esfuerzo, siempre ocuparan un lugar muy especial en mi corazón.*

*A mi compañera de proyecto, con quien sorteamos tantas situaciones a lo largo de nuestra carrera y en particular con el desarrollo de este proyecto.*

*A todas y cada una de las personas que contribuyeron a lo largo de este camino a alcanzar este gran triunfo.*

*Javier M.*

**TABLA DE CONTENIDO**

INTRODUCCIÓN.....	14
1 PRELIMINARES.....	17
1.1 PLANTEAMIENTO DEL PROBLEMA .....	17
1.2 OBJETIVOS.....	19
1.2.1 Objetivo general.....	19
1.2.2 Objetivos específicos.....	19
1.3 JUSTIFICACIÓN.....	20
1.4 ALCANCE .....	22
1.5 ANTECEDENTES.....	23
2 MARCO TEÓRICO .....	25
2.1 Comunicaciones Digitales .....	25
2.1.1 Sistemas analógicos vs Sistemas digitales.....	27
2.2 Conceptos Básicos.....	28
2.3 Modulación.....	30
2.3.1 Modulación Digital De Amplitud .....	31
2.3.2 Modulación Por Desplazamiento De Frecuencia .....	33
2.3.3 Modulación Por Desplazamiento Mínimo Gaussiano.....	36
2.3.4 Modulación Por Desplazamiento De Fase .....	38
2.3.5 Modulación Por Desplazamiento Binario De Fase .....	40
2.3.6 Modulación Por Desplazamiento Cuaternario De Fase.....	42
2.3.7 Modulación PSK De 8 Fases.....	43
2.3.8 Modulación Por Corrimiento De Fase Diferencial.....	44
2.4 SDR.....	47



# UNIVERSIDAD INDUSTRIAL DE SANTANDER

ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES  
Perfecta Combinación entre Energía e Intelecto

2.4.1	Antecedentes .....	47
2.4.2	Concepto.....	49
2.4.3	Implementación .....	50
2.5	GNU Radio.....	51
3	ESTRUCTURA GENERAL Y METODOLOGIA DE DESARROLLO .....	54
3.1	Estructura General .....	54
3.1.1	USRP .....	55
3.2	Metodología De Desarrollo .....	63
4	PRÁCTICAS A IMPLEMENTAR .....	65
	CONCLUSIONES.....	68
	RECOMENDACIONES Y TRABAJOS FUTUROS.....	70
	BIBLIOGRAFÍA.....	72
	ANEXOS .....	76

## LISTA DE FIGURAS

Figura 1. Diagrama simplificado de bloques de un sistema de comunicaciones..	25
Figura 2. Esquema del principio de modulación ASK. ....	32
Figura 3. Esquema del principio de modulación FSK. ....	34
Figura 4. Espectro de frecuencias FSK. ....	35
Figura 5. Demodulador FSK no coherente. ....	36
Figura 6. Espectro de la señal MSK (línea punteada).....	38
Figura 7. Esquema del principio de modulación PSK. ....	39
Figura 8. Proyección de una señal en fase (coseno) y una señal en cuadratura (seno) en la constelación I-Q. ....	40
Figura 9. Relación de fase de salida en función del tiempo para un modulador BPSK. ....	41
Figura 10. Diagrama de constelación modulación BPSK.....	41
Figura 11. Fase de salida en función de tiempo para un modulador QPSK .....	42
Figura 12. Diagrama de constelación modulación QPSK .....	43
Figura 13. Diagrama de constelación modulación 8-PSK.....	44
Figura 14. Diagrama de bloques transmisor DPSK. ....	45
Figura 15. Diagrama de sincronización transmisor DPSK .....	45
Figura 16. Diagrama de bloques receptor DBPSK.....	46
Figura 17. Secuencia de sincronización receptor DBPSK .....	46
Figura 18. Componentes GNU Radio. ....	53
Figura 19. Montaje sistema de comunicaciones digitales SDR. ....	54
Figura 20. Bloques principales en un sistema de comunicaciones SDR. ....	55
Figura 21. Diagrama de bloques del Universal Software Radio Peripheral. ....	59
Figura 22. USRP con tarjeta hija RFX2400 utilizado en el proyecto. ....	59
Figura A23. Acceso al Terminal. ....	78
Figura A24. Comando de ejecución GNU Radio en la Terminal. ....	79
Figura A25. Interfaz GNU Radio Companion. ....	79

Figura A26. Esquemático Proyecto Introductorio.....	82
Figura A27. Error en la conexión entre puertos no compatibles. ....	83
Figura A28. Conexión correcta entre dos puertos.....	84
Figura A29. Ventana de ejecución Proyecto Introductorio. ....	85
Figura A30. Uso de USRP. Diagrama de bloques transmisión.....	86
Figura A31. Uso USRP. Diagrama de bloques recepción.....	87
Figura B32. Diagrama de bloques Modulador DPSK.....	91
Figura B33. Diagrama de bloques Demodulador DPSK. ....	93
Figura C34. Diagrama de bloques Modulador - Demodulador GMSK. ....	106
Figura G35. WX GUI Chooser. Drop Down. ....	169
Figura G36. WX GUI Chooser. Radio button horizontal.....	169
Figura G37. WX GUI Chooser. Radio button vertical.....	170
Figura G38. WX GUI Button.....	170



**LISTA DE ANEXOS**

Anexo A. Práctica No. 1 .....	76
Anexo B. Práctica No. 2 .....	88
Anexo C. Práctica No. 3 .....	100
Anexo D. Práctica No. 4 .....	111
Anexo E. Sistemas de comunicaciones basados en la técnica SDR .....	116
Anexo F. Instalación GNU Radio.....	128
Anexo G. Referencia bloques de procesamiento de señales GNU Radio.....	138



## **RESUMEN**

**TÍTULO: DISEÑO DEL MANUAL DE PRÁCTICAS PARA UN LABORATORIO DE COMUNICACIONES DIGITALES BASADO EN LA TÉCNICA DE RADIO DEFINIDO POR SOFTWARE\***

**AUTOR:** Javier Mauricio Suárez Monsalve, Vivian Paola Triana Galeano \*\*

**PALABRAS CLÁVES:** Radio Definido por Software (SDR), GNU Radio, Universal Software Radio Peripheral (USRP), Modulación Digital, GMSK, PSK.

En este proyecto se presenta el concepto de Radio Definido por Software (SDR) para implementación de sistemas de comunicaciones inalámbricas, que entre sus múltiples aplicaciones, se orienta hacia el ámbito académico, dando un enfoque hacia el estudio de las comunicaciones digitales, siendo una alternativa de análisis de diversas técnicas de modulación digital como GMSK y PSK, de manera que pueda ser integrado en el programa de la asignatura como una opción para afianzar los conceptos relacionados con este tema.

En este trabajo, se utiliza la plataforma de software libre GNU Radio, que junto con dispositivos USRP (*Universal Software Radio Peripheral*) integran un sistema de comunicaciones SDR, configuración que fue seleccionada debido a la eficiencia que representa el hardware, al realizar el procesamiento de señal mediante un dispositivo reprogramable de alta potencia, FPGA *Altera Cyclone Epic12*; y la flexibilidad que proporciona el software, GNU Radio, a través de su aplicación GNU Radio Companion, que permite realizar la configuración de los sistemas de comunicación a través de una interfaz gráfica basada en bloques que facilita la labor de diseño; siendo estos componentes de carácter libre, lo que implica que no es necesario adquirir ni comprar licencias para su uso, asumiendo el costo que tiene el dispositivo USRP, comercializado por la compañía Ettus Research.

Como resultado final, se propusieron guías de uso práctico que permiten al usuario realizar el montaje adecuado de un sistema SDR, analizar las herramientas de software y hardware implementadas, y a su vez enfocar su uso al estudio de las comunicaciones digitales. Además, se presenta una guía de estudio de las modulaciones digitales anteriormente mencionadas y una breve descripción de diferentes trabajos realizados a nivel mundial, en los cuales se evidencia el uso de la técnica SDR, en una diversidad de aplicaciones diferentes al ámbito académico.

---

\* Proyecto de Grado.

\*\* Facultad de Ingenierías Físico Mecánicas, Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: MsC. Jorge Hernando Ramón Suárez. Codirector: Ing. Diego Rafael Medina Pulido.



## **SUMMARY**

**TITLE: DESIGN OF A PRACTICAL HANDBOOK FOR DIGITAL COMMUNICATIONS LABORATORY BASED ON SOFTWARE DEFINED RADIO TECHNIQUE**

**AUTHORS:** Javier Mauricio Suárez, Vivian Paola Triana Galeano \*\*

**Keywords:** Software Defined Radio (SDR), GNU Radio, Universal Software Radio Peripheral (USRP), Digital Modulation, GMSK, PSK.

This project introduces the concept of Software Defined Radio (SDR) for implementation of wireless communication systems, which among its many applications, it is oriented to the academic context, focusing on the study of digital communications, as an analysis tool of different digital modulation techniques such as GMSK and PSK, so it can be integrated into the program of the subject as an option to consolidate the concepts related to this topics.

In this thesis work, it is used the free and open source software GNU Radio, which is connected with the USRP devices (Universal Software Radio Peripheral) to create a SDR communication system, this configuration was selected because it represents the efficiency of the hardware, performing processing signal through a high power reprogrammable device, Altera Cyclone Epic12 FPGA, and the flexibility provided by the software, GNU Radio, and its GNU Radio Companion application, which allows the configuration of communication systems by way of a block-based graphical interface that facilitates the design work, these components are open source, so it means that it is not needed to purchase or buy any licenses for using, assuming the cost of the USRP device, marketed by Ettus Research.

As a final result, it is proposed some useful guides that allow the user to perform the appropriate assembly of a SDR system, analyzing the hardware and software tools implemented, and getting focus on the study of digital communications. In addition, it is presented a study guide of the digital modulations and a brief description of different investigation projects performed worldwide, in which are presented the use of SDR technique in a variety of different applications beyond the academic context.

---

\* Thesis work.

\*\* School of Physicomechanical Engineering, Department of Electrical, Electronics and Telecommunications Engineering. Director: MsC Jorge Ramón Suárez. Co-director: Eng. Diego Medina Pulido.

## INTRODUCCIÓN

Los sistemas de comunicaciones digitales han avanzado desde sus inicios gracias a la evolución de dispositivos electrónicos como procesadores DSP<sup>1</sup> y hardware fácilmente configurable, FPGA.<sup>2</sup> Es así, como con el paso del tiempo se han implementado sistemas de comunicaciones cada vez más eficientes y de menor complejidad. Una de las técnicas para implementación de sistemas de comunicaciones digitales que ha ido ganando terreno a nivel mundial, especialmente en el ámbito académico y militar, es la técnica Radio Definido por Software, SDR (*Software Defined Radio*), debido a su gran flexibilidad, versatilidad y eficiencia. SDR se define como radios en los cuales algunas o todas las funciones de la capa física son definidas por medio de software e implementadas en hardware reconfigurable.

Una de las múltiples aplicaciones de la técnica SDR, consiste en utilizarla como una opción para el estudio de los diferentes tipos de modulación digital, debido a que su implementación no depende del tipo de modulación que se utilice, siendo esta característica una de las ventajas clave, permitiendo así que los conceptos teóricos sean verificados en práctica en un ambiente interactivo y realista.

Para la implementación de sistemas de comunicaciones SDR, existen diferentes arquitecturas, dentro de las cuales se tiene implementación directa en FPGA y DSP. En este trabajo, se utiliza la herramienta de software libre GNU Radio Companion, que junto con dispositivos USRP (*Universal Software Radio Peripheral*) integran un sistema de comunicaciones SDR. Esta configuración fue seleccionada debido a la eficiencia que representa el hardware, al trabajar con una

---

<sup>1</sup> DSP: Procesador Digital de Señales (*Digital Signal Processor*)

<sup>2</sup> FPGA: *Field Programmable Gate Array*.

FPGA *Altera Cyclone Epic12*; y la flexibilidad que proporciona el software, GNU Radio, a través de su aplicación GNU Radio Companion, que permite realizar la configuración de los sistemas de comunicación y el diseño del flujo de señal a través de una interfaz gráfica, intuitiva para el usuario. A su vez, estos dos componentes son de carácter completamente libre lo que implica que no es necesario adquirir ni comprar licencias para su uso, asumiendo el costo que tiene el dispositivo USRP, comercializado por la compañía Ettus Research<sup>3</sup>.

Este proyecto se presenta como una alternativa de estudio de las técnicas de modulación digital GMSK y PSK, utilizando tecnología SDR, de manera que pueda ser integrado en el programa de comunicaciones digitales, como una opción de laboratorio para afianzar los conceptos relacionados con la materia.

A lo largo del texto se muestran los antecedentes en cuanto a laboratorios de comunicaciones digitales dentro de la Universidad Industrial de Santander. Se presenta un marco teórico que fundamenta los principales conceptos relacionados con comunicaciones digitales, enfocados en técnicas de modulación digital. Se realiza una introducción al concepto de SDR, seguido de la presentación de la plataforma GNU Radio Companion. El montaje requerido para la implementación del laboratorio es descrito, acompañado de las prácticas de laboratorio propuestas. Finalmente, se realizan las conclusiones del trabajo y se proponen las respectivas recomendaciones y sugerencias para trabajos futuros.

Como complemento, se presentan documentos de apoyo relacionados con la instalación de las herramientas necesarias, la descripción de los parámetros de configuración de los bloques de procesamiento de señales utilizados para la implementación de las prácticas de laboratorio, y documentación sobre algunos trabajos realizados a nivel mundial relacionados con la técnica SDR.

---

<sup>3</sup> Disponible en: [www.ettus.com](http://www.ettus.com) [Consultado en julio 25 de 2012]

## 1 PRELIMINARES

### 1.1 PLANTEAMIENTO DEL PROBLEMA

El avance de las comunicaciones de forma inalámbrica marcó grandes cambios en los sistemas de comunicación, originándose tecnologías que posibilitan obtener mejor flexibilidad y eficiencia, al mismo tiempo que permiten dar más funcionalidad al software del sistema que al hardware.

Las comunicaciones digitales han adquirido gran importancia en la actualidad, principalmente por las ventajas que tienen las señales digitales con respecto a las señales analógicas. Dentro de estas ventajas se encuentran: inmunidad al ruido, si hay pérdida de información la señal puede ser reconstruida a través de sistemas de regeneración de señales, además, las tramas digitales pueden ser fácilmente almacenadas y procesadas, ocupando menor espacio de memoria que las señales analógicas.

La técnica de Radio Definido por Software, SDR (*Software Defined Radio*), surgió para dar la flexibilidad, eficiencia de costos y potencia a los sistemas de comunicaciones.<sup>4</sup> SDR es una combinación de tecnologías tanto de hardware como de software que permiten implementar, a través de software modificable, diferentes configuraciones en las funciones de operaciones de radio, sin los requerimientos de nuevo hardware.

Al realizar la respectiva investigación de los proyectos realizados en torno a las modalidades de prácticas y laboratorios de comunicaciones digitales implementados en la Universidad Industrial de Santander, se encontró que lo

---

<sup>4</sup> Disponible en: <http://www.wirelessinnovation.org/assets/documents/SoftwareDefinedRadio.pdf> [Consultado en julio 10 de 2012]

realizado hasta el momento ha sido, por un lado, un desarrollo de hardware limitado en cuanto a manipulación de parámetros y configuración del sistema, ya que son tarjetas electrónicas las que realizan las modulaciones y para cambiar alguna variable es necesario realizar un cambio directo de algún elemento del circuito físico, es decir no es fácilmente reconfigurable, aunque la ventaja de esta propuesta sería que existe un sistema físico para manipular. [1][18] Por otro lado, se encuentra el desarrollo de software que en gran medida permite describir el comportamiento de un sistema de comunicaciones, a través de programas avanzados como MatLab y LabView en los que se logran simulaciones verídicas y que permiten analizar el comportamiento de las señales.[4][8]

La técnica SDR se presenta como una alternativa viable para implementar un laboratorio de comunicaciones digitales, que combina tanto hardware como software, es sencillamente reconfigurable, práctico, de fácil manejo y costo razonable.

## **1.2 OBJETIVOS**

### **1.2.1 Objetivo general**

Diseñar y elaborar una guía de uso práctico de un sistema de comunicaciones digitales basado en el concepto de Radio Definido por Software, implementado utilizando la herramienta de desarrollo GNU Radio.

### **1.2.2 Objetivos específicos**

- Realizar el manual de usuario del sistema de comunicaciones digitales utilizando Radio Definido por Software.
- Elaborar una guía para el estudio de algunas modulaciones digitales implementadas en el sistema de comunicación.
- Recopilar la documentación sobre sistemas de comunicaciones digitales utilizando Radio Definido por Software.

### **1.3 JUSTIFICACIÓN**

El propósito fundamental de este trabajo es la elaboración de un manual de prácticas para un sistema de comunicaciones digitales basado en la tecnología SDR, Radio Definido por Software, implementado mediante la herramienta de software libre GNU Radio y dispositivos USRP; y la realización de una guía de estudio sobre modulación digital, teniendo en cuenta las técnicas de modulación: GMSK, BPSK, QPSK, 8PSK y DPSK, incluyendo una documentación relacionada con sistemas de comunicaciones basados en tecnología SDR.

Hasta la fecha, las propuestas realizadas para el desarrollo de un laboratorio de comunicaciones digitales en la Universidad Industrial de Santander, van desde prototipos físicos en circuitos impresos de sistemas de comunicaciones digitales, hasta diseños basados completamente en software; estos últimos permiten realizar simulaciones aproximadas a lo que se obtendría en un sistema de comunicaciones real.

Aunque estos proyectos [1][4][18][8] han servido de gran fundamento para el aprendizaje integral de la asignatura, son limitados en funcionalidad, en el sentido que no permiten una experiencia de estudio que reúna en un solo sistema una manipulación tanto física como de programación.

El concepto de SDR se presenta como una excelente alternativa para un laboratorio de comunicaciones digitales ya que permite contar con un sistema de radiocomunicaciones integrado por tecnologías de software, hardware y Radiofrecuencia (RF) que posibilita transmitir, procesar y recibir señales de radio, donde la mayor parte del tratamiento de la señal y los elementos que componen el sistema son implementados en software, todo esto sin perder la funcionalidad de la parte física del sistema.



## UNIVERSIDAD INDUSTRIAL DE SANTANDER

ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES  
Perfecta Combinación entre Energía e Intelecto

La integración de Hardware y Software en un solo entorno, como lo hace un sistema SDR, permite aprovechar las ventajas de ambas partes: tener un sistema físico, manipulable, práctico y a su vez, fácilmente reconfigurable para cualquier variación en los parámetros que intervienen en el sistema de comunicaciones, lo cual representa grandes ventajas con respecto a otras metodologías propuestas [1][4][18][8]. Además, teniendo en cuenta que esta tecnología es en su totalidad de carácter libre, no deben ser adquiridas licencias para su uso, estudio, modificación o redistribución de este software. [12]

Desarrollar una experiencia con un sistema de comunicaciones físico y reprogramable es una oportunidad para reafirmar los conceptos teóricos aprendidos con respecto a las comunicaciones digitales. Es así como la guía de laboratorio se presenta como una herramienta de gran aporte equivalente a un acompañamiento práctico de la teoría que en muchas ocasiones no es suficiente para un adecuado entendimiento de los conceptos, y a su vez, como una alternativa de reconocimiento de la tecnología SDR en el entorno local, aportando al desarrollo investigativo nacional en este concepto.



## **1.4 ALCANCE**

Realizar la elaboración de un manual de prácticas de un sistema de comunicaciones digitales basado en Radio Definido por Software, que permita la interacción del usuario con el software y hardware reconfigurable, acompañada de una documentación teórica de las modulaciones digitales implementadas en este sistema para una experiencia de estudio teórico-práctica completa. Este proyecto también permite evidenciar la experiencia de trabajo con un componente basado en SDR que podría servir como base e iniciativa para la investigación e implementación de diversos proyectos futuros en torno a esta tecnología a nivel local.

## **1.5 ANTECEDENTES**

En el año de 1999, Aguilar, Botello y Campo [1], desarrollaron un proyecto que tuvo como objetivo la creación e implementación del laboratorio de la asignatura de comunicaciones en la Universidad Industrial de Santander. Se plantearon prácticas construyendo 8 prototipos comprobados utilizando las herramientas para diseño de circuitos impresos, tres para las prácticas de modulación analógica (AM, FM y de fase); tres para las prácticas de modulación digital; un prototipo para práctica de ruido y un prototipo para práctica de modulación por codificación de pulsos.

En 2001, Ballesteros, Mojica y Mercado [4], desarrollaron un proyecto que consistió en la implementación de un software elaborado en lenguaje de programación gráfico LabView, el cual incluía la simulación de trece tipos de modulaciones diferentes, guías de laboratorio para cada una de ellas y a su vez un manual de usuario para un manejo general de la herramienta.

Más adelante, en el año de 2002, Lizcano [18], desarrolló un proyecto tomando como base la concepción del proyecto realizado por Aguilar y Botello en 1999. Este trabajo tuvo como objetivo diseñar e implementar tarjetas electrónicas para el desarrollo de las prácticas del laboratorio de comunicaciones; las cuales representaban las siguiente modulaciones; Modulación y desmodulación analógica: AM (Amplitud Modulada), FM (Frecuencia Modulada); PM (Fase Modulada); Modulación de pulsos analógicos: PAM (Modulación de Amplitud de pulso), PWM (Modulación de Ancho de pulso), PPM (Modulación de posición de Pulso), Modulación de pulsos digitales: PCM (Modulación de código de pulso), Modulación y Desmodulación digital: FSK (Conmutación de Frecuencia). Se soldaron setenta (70) tarjetas electrónicas que correspondían a diez copias de las siete tarjetas básicas de las modulaciones mencionadas anteriormente.

Pocos años después, en el año 2004 Duarte y Uribe [8], desarrollaron el proyecto que consistió en el diseño de una herramienta para el estudio y análisis de sistemas de modulación digital pasabanda empleando el software Matlab-Simulink. La herramienta estaba compuesta por un conjunto de guías, y tres modelos con sus respectivas interfaces. La interfaz tenía como función asignar los parámetros correspondientes a cada uno de los bloques, interconectados en Simulink, de manera rápida. Se abarcaron los tipos de modulación: modulación por variación de fase (PSK), modulación por variación de frecuencia (FSK) y modulación de amplitud en cuadratura (8QAM).

En cuanto a trabajos desarrollados a nivel nacional sobre Radios Definidos por Software, Ferney Amaya [2], de la Universidad Javeriana, presentó una introducción al desarrollo de sistemas de Radio inalámbricos sobre FPGA, implementando un modulador-demodulador BPSK, a través de la herramienta Simulink de MatLab. Adicionalmente, el profesor visitante de la Universidad Pontificia Bolivariana de Bucaramanga, Yair Linn [17], expuso la implementación de un laboratorio SDR de bajo costo utilizando los módulos de desarrollo Spartan 3A de la empresa Xilinx. El producto entregado por el profesor Linn es un laboratorio totalmente autónomo que permite al estudiante simular sistemas de modulación–desmodulación con diferentes técnicas de modulación digital, entre las cuales se tienen PSK (Phase Shift Keying) y QAM (Quadrature Amplitude Modulation). El sistema está configurado para que las señales de salida puedan ser analizadas utilizando osciloscopios sencillos, permitiendo observar Diagramas de constelaciones, análisis vectorial, etc.

Un compendio de trabajos realizados a nivel mundial basados en SDR, se encuentra el documento “Sistemas de comunicaciones basados en la técnica SDR” (Anexo E).

## 2 MARCO TEÓRICO

### 2.1 Comunicaciones Digitales

El propósito de un sistema de comunicación es transmitir señales portadoras de información desde una *fuentes*, localizada en un punto del espacio, a un *usuario de destino*, localizado en otro punto [14]. Todos los sistemas de comunicación involucran tres subsistemas principales: *transmisor*, *medio de transmisión* y *receptor* [7].

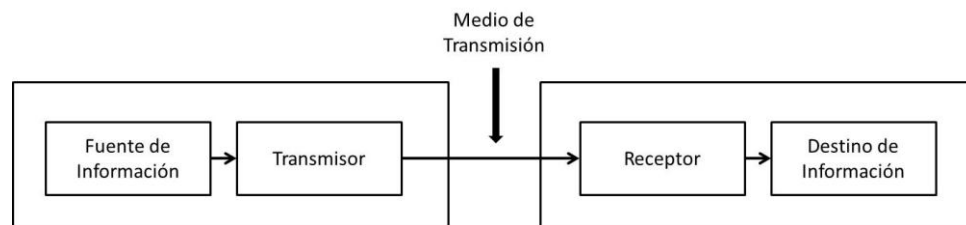


Figura 1. Diagrama simplificado de bloques de un sistema de comunicaciones. Fuente [25].

El *transmisor* es un conjunto de uno o más dispositivos o circuitos electrónicos que convierte la información de la fuente original en una señal adecuada para la transmisión sobre el medio de transmisión [25]. Esta transformación se logra por medio del proceso de modulación que consiste en plasmar la información fuente sobre una señal pasabanda con una frecuencia de portadora  $f_c$ , mediante la introducción de perturbaciones en amplitud, fase o ambas [7]. La señal resultante del transmisor es una señal pasabanda, ya que está diseñada para poseer frecuencias ubicadas en una banda alrededor de  $f_c$ .

La información en el mensaje puede ser analógica o digital, dependiendo del sistema en particular, y puede representar información de audio, de video o de algún otro tipo. Los espectros (o frecuencias) del mensaje de la fuente están

concentrados alrededor de  $f=0$ ; por consecuencia, estas se consideran como señales de *banda base* [7].

Una *señal analógica* es una señal en la cual tanto las *amplitudes* como el *tiempo* varían *continuamente* sobre sus respectivos intervalos. En una *señal digital*, tanto la *amplitud* como el *tiempo* toman *valores discretos* [15].

En un sistema de *comunicación digital* la señal de información es procesada de manera que puede ser representada por una *secuencia de mensajes discretos* [15].

Las señales analógicas se convierten en señales digitales por medio del proceso secuencial de muestreo, cuantificación y codificación. El muestreo es la operación por la cual se toman valores muestreados de la señal analógica en instantes de tiempo discretos espaciados uniformemente. Por su parte la cuantificación, toma los valores muestreados de la señal aproximándolos a un nivel cercano en un conjunto finito de niveles discretos; por último la codificación, donde los niveles seleccionados se representan por un símbolo que consiste en un número prescrito de elementos de código [15].

El *medio de transmisión* o *canal* transporta las señales desde el transmisor hasta el receptor [25]. El medio de transmisión puede clasificarse en dos categorías: alámbricos e inalámbricos. Algunos ejemplos de canales alámbricos son las líneas telefónicas de par trenzado, los cables coaxiales, guías de onda y fibra óptica. Algunos canales inalámbricos típicos son el aire, el vacío y el agua de mar [7]. En la propagación a través del canal, la señal transmitida es distorsionada debido a no linealidades y/o imperfecciones en la respuesta en frecuencia del canal. Otras fuentes de degradación son el ruido, y la interferencia captados por la señal durante su proceso de transmisión a través del medio de transmisión, los cuales constituyen dos problemas básicos en el diseño de sistemas de comunicaciones

[14]; donde el ruido se refiere a cualquier energía indeseable que queda entre la banda de paso de la señal de información, y la interferencia, que son señales de información de una fuente que se producen por frecuencias que caen fuera de su ancho de banda asignado y se interponen con otras señales de otra fuente.

El *receptor* es un conjunto de dispositivos y circuitos electrónicos que toma la señal distorsionada a la salida del canal y la convierte a una señal de banda base que el procesador de banda base en el receptor puede manipular, el cual “limpia” esta señal y envía una estimación de la información de la fuente a la salida del sistema de comunicación, proceso conocido como *Desmodulación*, que es el nombre de la operación inversa a la operación de modulación usada en el transmisor.

El objetivo es el desarrollo de sistemas de comunicaciones que transmitan información de una fuente a un receptor con el mínimo deterioro posible, al tiempo que se satisfacen las restricciones del diseño como la energía permitida, el ancho de banda permitido para cada señal y el costo en los sistemas digitales. La medición del deterioro a menudo se toma como la probabilidad de error en el bit ( $P(e)$ ) o como la tasa de error en el bit (BER), de los datos entregados <sup>5</sup>[7].

### **2.1.1 Sistemas analógicos vs Sistemas digitales**

Las comunicaciones digitales –las que tienen moduladoras digitales-, tienen como ventaja cierto grado de *inmunidad* a las perturbaciones de la transmisión, que no comparten con las comunicaciones analógicas: en esta ventaja radica la principal razón de que se hayan impuesto [10]. Las señales digitales presentan una facilidad, comparadas con las señales analógicas, para ser regeneradas. Las

---

<sup>5</sup>Referirse a [6], [12], u otra referencia sobre tratamiento de señales en comunicaciones.

formas de onda son afectadas por la no linealidad de los canales de transmisión y por el ruido eléctrico u otras interferencias que llegan a distorsionar el pulso ideal. Para darle una alternativa de solución a este inconveniente, se tienen los *repetidores regenerativos*, que al detectar la señal que viaja durante la transmisión, mientras se pueda hacer de forma confiable su detección, amplifica el pulso por medio de un amplificador digital [22].

Otra ventaja, radica en que los circuitos digitales son más confiables y pueden ser producidos a menor costo que los circuitos analógicos. También, el hardware digital permite una implementación más flexible que el hardware analógico [22].

Una de las mayores desventajas de la transmisión digital es que esta típicamente requiere un mayor ancho de banda para comunicar la misma información en forma digital, comparado con la forma analógica. Otro costo de la transmisión digital es que la detección digital requiere sistemas de sincronización, mientras que generalmente las señales analógicas no tienen este requerimiento [22].

## 2.2 Conceptos Básicos

**Digito binario (bit):** unidad fundamental de información para todos los sistemas digitales [22].

**Símbolo (mensaje digital):** grupo de  $k$  bits considerados como una unidad o carácter  $m_i$ , de un conjunto finito de símbolos o alfabeto. El tamaño del alfabeto,  $M$ , es  $M = 2^k$  ( $k$  es el número de bits por símbolo) [22].

**Forma de onda digital:** una forma de onda de voltaje o corriente (un pulso para transmisión banda base o una senoide para transmisión pasabanda) que

representa un símbolo digital. Las características de la forma de onda (amplitud, ancho, posición de impulsos, o amplitud, frecuencia y fase para las sinusoides) permiten su identificación como uno de los símbolos en el alfabeto de símbolos finito [22].

**Frecuencia de bits:** En la modulación digital, la rapidez de cambio de entrada del modulador se llama *frecuencia de bits*, o rapidez de transferencia de bits, la cual tiene unidades de bits por segundo (bps) [25]. Esta dada por  $R=k/T=(1/T)\log_2M$  bits/s, donde k bits identifica a un símbolo de un  $M=2^k$  alfabeto de símbolo, y T es la duración del k-bit símbolo [22].

**Baudio:** o *símbolos por segundo*, es la *rapidez de cambio* de la salida del modulador [25]. La *velocidad de modulación* se define como la tasa a la cual los niveles de señal son cambiados, dependiendo de la naturaleza del formato usado para representar los datos digitales [15].

**Ancho de banda:** El ancho de banda es una de las limitaciones más importantes en el funcionamiento de un sistema de comunicaciones [25]. Se define como el intervalo de frecuencias *positivas* en el que la magnitud se mantiene dentro de determinado factor numérico <sup>6</sup>[23].

**Densidad espectral:** la densidad espectral de una señal caracteriza la distribución de la energía de la señal o la potencia en el dominio de la frecuencia. Este concepto es importante cuando se considera la filtración en sistemas de comunicaciones. La densidad espectral de energía (ESD) o la densidad espectral de potencia (PSD) son usadas en su evaluación <sup>7</sup>[22].

---

<sup>6</sup>Existen diferentes definiciones para este concepto, algunas de las cuales pueden ser consultadas en [7] páginas 101 a 109.

<sup>7</sup> Consultar [6] páginas 14 a 16 para el desarrollo matemático.

**Constelación:** Una constelación es un gráfico N-dimensional de los probables vectores de señal correspondientes a las posibles señales digitales [7]. Se parece a un diagrama fasorial, pero no se dibuja todo el fasor, sólo se muestran las posiciones relativas de los máximos de los fasores [25].

**BER:** Tasa o frecuencia de errores de bits (BER, por *bit error rate*) es un registro empírico (histórico) del funcionamiento real del sistema en cuanto a errores [25]. La *probabilidad de error*,  $P(e)$ , es la expectativa teórica o matemática, de que determinado sistema tenga una tasa de errores [25]. Para evaluar el desempeño en un sistema de comunicación digital, como una figura de mérito, se mide la tasa de error de bit y se compara con la probabilidad esperada de error.

### 2.3 Modulación

Es una operación básica de procesamiento de señales en un sistema de comunicaciones. En el proceso se introducen cambios de amplitud, frecuencia y/o fase de una onda portadora sinusoidal con respecto a una onda modulante. La modulación se realiza con el propósito de proveer una transmisión eficiente de la señal sobre el canal [15].

Modulación digital es el proceso por el cual símbolos digitales son transformados en formas de onda que son compatibles con las características del canal. En el caso de modulación bandabase estas formas de onda son pulsos, pero en el caso de modulación pasabanda la señal de información deseada modula una senoide llamada *onda portadora*. Para radio transmisión la portadora es convertida a un campo electromagnético para propagación al destino deseado [22].

El objeto fundamental de la modulación es reducir la distorsión que el medio produce en la información, además de conseguir mejores condiciones para la propagación, eligiendo adecuadamente las frecuencias de portadoras [10].

Teniendo en cuenta las diferentes características que pueden variarse de la señal modulada, se tiene como resultado del proceso tres grandes grupos de modulaciones, a saber, modulación digital de amplitud (ASK, *amplitude-shift keying*), modulación por desplazamiento de frecuencia (FSK, *frequency-shift keying*) y modulación por desplazamiento de fase (PSK, *phase-shift keying*).

### 2.3.1 Modulación Digital De Amplitud

Modulación digital por desplazamiento de amplitud (ASK, *amplitude-shift keying*), es una modulación donde la señal moduladora (datos digitales) se representa como una variación de amplitud de la onda portadora, manteniendo la frecuencia y la fase constante. Los dos valores binarios se representan con dos amplitudes diferentes. En la señal modulada el valor lógico 1 es representado mediante la presencia de la portadora a amplitud constante, mientras que el valor lógico 0 es representado por la ausencia de una portadora. A este tipo de modulación digital de amplitud se le suele llamar modulación por manipulación *encendido-apagado*, o *todo o nada* (OOK, de *on-off keying*) [25].

La ecuación que describe la modulación digital de amplitud mediante una señal binaria es:

$$v_{am}(t) = [1 + v_m(t)] \left[ \frac{A}{2} \cos(w_c t) \right]$$

Donde,  $v_{am}(t)$  = voltaje de la onda de amplitud modulada.

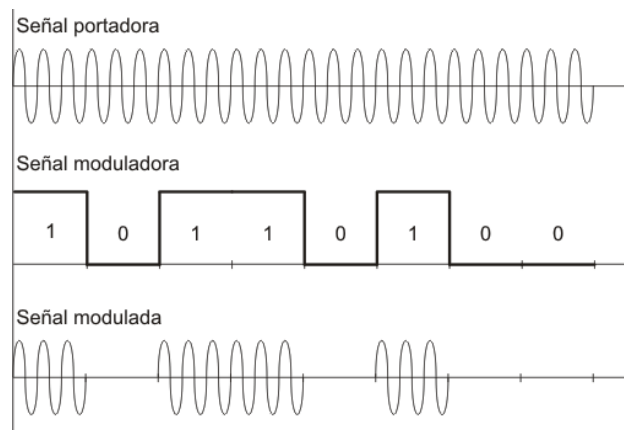
$A$  = amplitud de la portadora no modulada (volts)

$v_m(t)$  = señal binaria moduladora (volts)

$w_c$  = frecuencia de la portadora en radianes (radianes por segundo)

La señal moduladora  $v_m(t)$  es una forma de onda binaria normalizada, en la que +1 V = 1 lógico, y -1 V = 0 lógico [25].

En la siguiente figura se observan la señal portadora, la señal moduladora y la señal modulada para una modulación digital ASK.



**Figura 2. Esquema del principio de modulación ASK<sup>8</sup>.**

Este tipo de modulación digital es relativamente de baja calidad y bajo costo, por lo que rara vez se utiliza en sistemas de comunicación de gran capacidad y alta eficiencia [25].

<sup>8</sup> Fuente: <http://www.textoscientificos.com/redes/modulacion/ask> [Consultado en mayo 5 de 2012]

### 2.3.2 Modulación Por Desplazamiento De Frecuencia

La modulación por desplazamiento de frecuencia (FSK, *frequency-shift keying*), es una forma de modulación binaria de ángulo, de amplitud constante donde la señal moduladora es una señal binaria que varía entre dos valores discretos de voltaje [25]. Consiste en el corrimiento de la frecuencia de una portadora sinusoidal desde una *frecuencia de marca* (correspondiente, por ejemplo, al envío de un 1 binario) a una *frecuencia de espacio* (correspondiente al envío de un 0 binario), de acuerdo con la señal digital de banda base [7].

La ecuación general de la FSK binaria es:

$$v_{fsk}(t) = V_C \cos[2\pi[f_c + v_m(t)\Delta f]t]$$

En donde  $v_{fsk}(t)$  = forma de onda binaria FSK

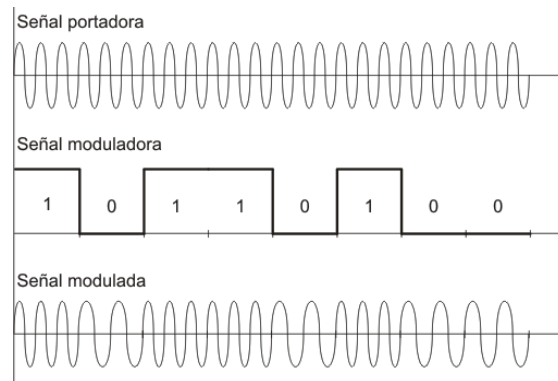
$V_C$  = amplitud de la portadora (volts)

$f_c$  = frecuencia central de la portadora (Hertz)

$\Delta f$  = desviación máxima de frecuencia (Hertz)

$v_m(t)$  = señal moduladora de entrada binaria ( $\pm 1$ )

En la siguiente figura se observan la señal portadora, la señal moduladora y la señal modulada para una modulación digital FSK.



**Figura 3. Esquema del principio de modulación FSK<sup>9</sup>.**

La señal moduladora es una forma de onda binaria normalizada, en la que el 1 lógico = 1, y el 0 lógico = -1. Cuando la señal binaria de entrada cambia de un 0 lógico a un 1 lógico y viceversa, la frecuencia de salida se desplaza entre las dos frecuencias: *frecuencia de marca*, *frecuencia de trabajo* o frecuencia de 1 lógico ( $f_m$ ), y la *frecuencia de espacio* o de 0 lógico ( $f_s$ ). La frecuencia de marca y de espacio están separadas de la frecuencia de portadora por la desviación máxima de frecuencia, es decir, por  $f_c \pm \Delta f$  [25].

El ancho de banda mínimo apropiado para una modulación FSK es:

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

En donde B= ancho de banda mínimo (Hertz)

$\Delta f$  = desviación mínima de frecuencias máximas (Hertz)

$f_m$  = frecuencia de marca (Hertz)

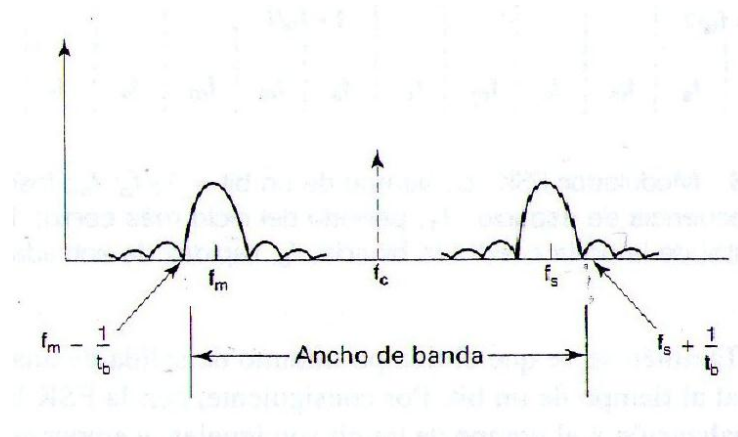
$f_s$  = frecuencia de espacio (Hertz)

$f_b$  = rapidez de bits de entrada (bps)

<sup>9</sup> Fuente: <http://www.textoscientificos.com/redes/modulacion/FSK> [Consultado en mayo 5 de 2012]

Para garantizar la continuidad de fase en FSK las portadoras asociadas a cada pulso deben ser de la forma  $w_i = 2\pi N_i/T$ ,  $i = 0, \dots, M-1$  para algún número entero  $N_i$  y además, esta asignación de frecuencias conduce a la ortogonalidad de los pulsos empleados [3].

En Figura 4 se observa el espectro de frecuencias FSK<sup>10</sup>.

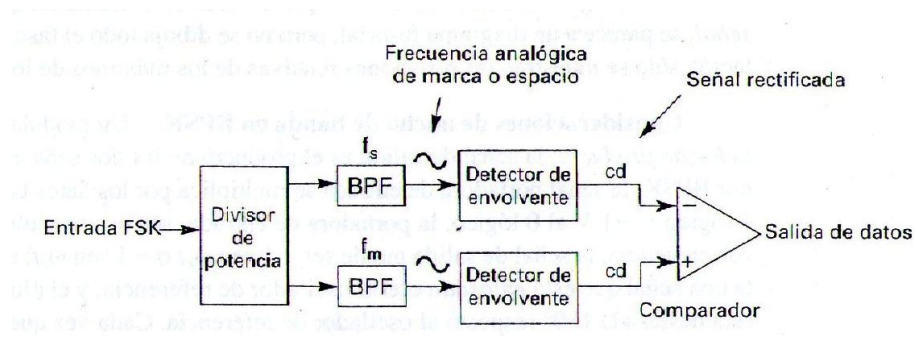


**Figura 4. Espectro de frecuencias FSK. Fuente [25].**

El espectro de la señal FSK aumenta con la separación frecuencial entre portadoras [3].

La detección de una señal FSK más práctica se hace a través de un *detector no coherente*, como el mostrado en Figura 5.

<sup>10</sup> Para mayor información sobre el espectro de frecuencias FSK referirse [6], página 474 y [10], página 401.



**Figura 5. Demodulador FSK no coherente. Fuente [25].**

La señal de entrada se aplica simultáneamente a las entradas de los dos filtros pasabanda (BPF) a través de un divisor de potencia. El filtro respectivo sólo pasa la frecuencia de marca o sólo la de espacio a su respectivo detector de envolvente. Los detectores de envolvente indican la potencia total en cada banda pasante, y el comparador responde a la mayor de las potencias<sup>11</sup> [25].

Este tipo de modulación no se utiliza en sistemas digitales de radio de alto rendimiento, por su menor eficiencia comparado con PSK y QAM [25].

### 2.3.3 Modulación Por Desplazamiento Mínimo Gaussiano

Un caso especial de la FSK llamada modulación por desplazamiento mínimo gaussiano (GMSK, Gaussian Minimum Shift Keying) se utiliza en radiotelefonía celular GMSK [5]. Los datos (pulsos en forma rectangular) se pasan a través de un filtro con una característica de respuesta en frecuencia en forma gaussiana

<sup>11</sup> Referirse a [6] página 476, [5] página 283 u otra fuente de consulta sobre receptores en comunicaciones digitales.

antes de que se modulen en frecuencia dentro de la portadora. El filtro reduce los lóbulos laterales espectrales en la señal MSK [7]. En este tipo de modulación, se obtiene una fase continua en la señal analógica de salida [16].

La palabra gaussiano se refiere a la forma de un filtro que se utiliza antes del modulador para reducir el ancho de banda transmitido de la señal. En este tipo de sistema, las frecuencias de marca y espacio están separadas por la mitad de la tasa de bits, es decir [5]:

$$f_m - f_s = 0.5f_b$$

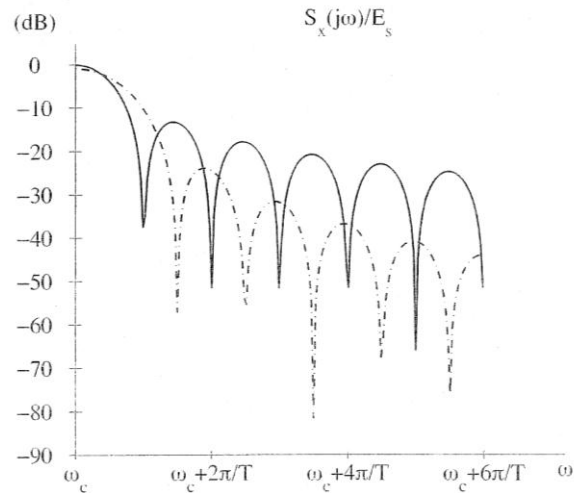
Donde  $f_m$  = frecuencia transmitida para la marca (1 binario)

$f_s$  = frecuencia transmitida para el espacio (0 binario)

$f_b$  = tasa de bits

GMSK utiliza menos ancho de banda que la FSK básica, debido a que el filtro hace que la frecuencia transmitida se mueva poco a poco entre las frecuencias de marca y de espacio [5].

El espectro de la señal MSK se puede observar en Figura 6 en la cual se detallan los lóbulos laterales que al aplicar el filtro gaussiano se eliminan permitiendo utilizar menos ancho de banda.

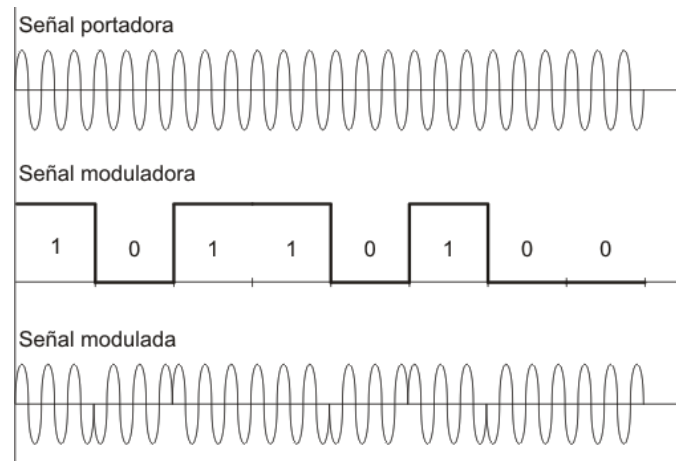


**Figura 6. Espectro de la señal MSK (línea punteada). Fuente [3].**

### 2.3.4 Modulación Por Desplazamiento De Fase

La modulación por desplazamiento de fase (PSK, *phase-shift keying*) es una forma de modulación en ángulo de envolvente constante que consiste en hacer variar la fase de la portadora entre una cantidad limitada de posibles valores.

En la siguiente figura se observan la señal portadora, la señal moduladora y la señal modulada para una modulación digital PSK.



**Figura 7. Esquema del principio de modulación PSK.**<sup>12</sup>

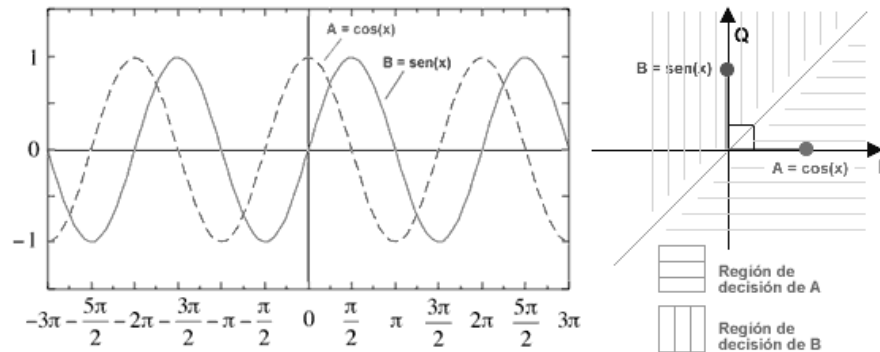
La modulación de fase es ventajosa cuando el canal incluye efectos no lineales sobre la amplitud, como es el caso de los amplificadores de radiofrecuencia empleados en comunicaciones móviles y vía satélite [3].

#### **2.3.4.1 Modulación I-Q**

La amplitud y la fase de una señal pueden modularse simultáneamente o por separado, aunque esto resulta difícil de generar y principalmente de detectar. En sistemas prácticos la señal es separada en dos componentes: componente en fase (I) cuando el ángulo de fase de la señal es de  $0^\circ$ , y componente en cuadratura (Q) cuando la señal se encuentra desfasada  $90^\circ$  con respecto a la señal en fase. Estas dos componentes tienen una representación bidimensional llamada constelación en la cual la componente en fase se proyecta en el eje de las abscisas y la componente en cuadratura en el eje de las ordenadas, de manera

<sup>12</sup> Fuente: <http://www.textoscientificos.com/redes/modulacion/psk> [Consultado en mayo 5 de 2012]

que cada señal que se mapea tiene asociada una posición precisa (I,Q) y una región de decisión como se puede observar en Figura 8.



**Figura 8. Proyección de una señal en fase (coseno) y una señal en cuadratura (seno) en la constelación I-Q<sup>13</sup>.**

### 2.3.5 Modulación Por Desplazamiento Binario De Fase

En la modulación por desplazamiento de fase (BPSK, *Binary phase shift keying*) son posibles dos fases de salida para una sola frecuencia portadora. Una fase de salida representa un 1 lógico, y la otra un 0 lógico. Cuando la señal de entrada digital cambia de estado, la fase de la portadora de salida varía entre dos ángulos que están desfasados  $180^\circ$  [25].

<sup>13</sup> Fuente: <http://www.cinit.org.mx/articulo.php?idArticulo=10> [Consultado en mayo 5 de 2012]

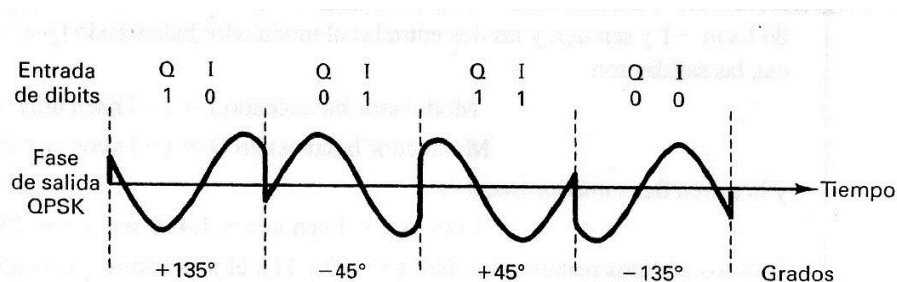


Siendo  $f_b$  rapidez de entrada de bits.<sup>14</sup>

### 2.3.6 Modulación Por Desplazamiento Cuaternario De Fase

La modulación por desplazamiento cuaternario de fase (QPSK, *Quaternary Phase Shift Keying*), es otra forma de modulación digital angular de amplitud constante. En muchas aplicaciones, un sistema de transmisión resulta más económico si, en un ancho de banda determinado, pueden transmitirse más bits por segundo. Esto conduce a la aplicación de métodos de modulación M-aria en los que se transmite una de M posibles señales durante cada intervalo de señalización [23]. M-aria se refiere a la codificación de los símbolos a un nivel más alto que un bit. Cuando se tiene  $M = 4$ , cada símbolo es representado por dos bits, con lo cual se obtienen cuatro posibles combinaciones, que se representan con cuatro posibles fases de salida.

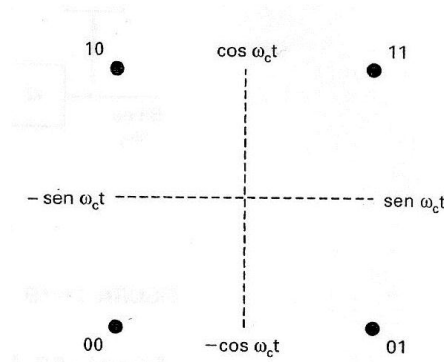
En este tipo de modulación, por cada dos bits introducidos al modulador ocurre un solo cambio a la salida. En consecuencia, la tasa de cambio en la salida (baud) es la mitad de la tasa de bits de entrada [16].



**Figura 11. Fase de salida en función de tiempo para un modulador QPSK. Fuente [25].**

<sup>14</sup> Referirse a [6] páginas 479 a 481 para información sobre consideraciones de ancho de banda en BPSK.

Para esta modulación se tienen cuatro posibles fasores de salida, que tienen exactamente la misma amplitud. La separación angular entre dos fasores adyacentes es de  $90^\circ$ . En consecuencia, una señal QPSK puede sufrir cuando mucho un desplazamiento de fase de  $+45^\circ$  o  $-45^\circ$  durante la transmisión y retener en forma correcta la información codificada, cuando se desmodule en el receptor <sup>15</sup>[25].



**Figura 12. Diagrama de constelación modulación QPSK. Fuente [25].**

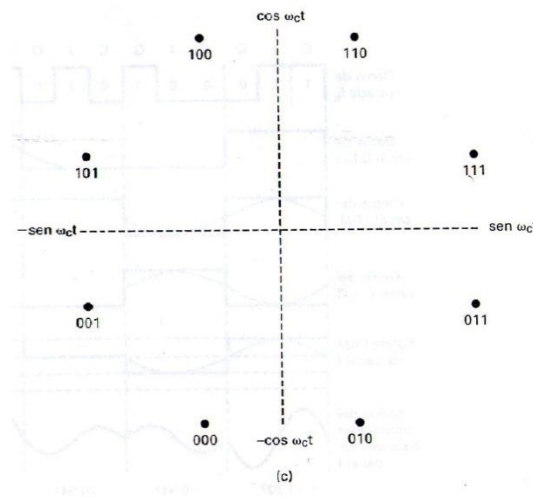
### 2.3.7 Modulación PSK De 8 Fases

La modulación PSK de 8 fases es una técnica de codificación M-aria en la cual cada símbolo es representado por 3 bits, de esta manera se obtienen ocho posibles fases de salida.

En este tipo de modulación la separación angular entre dos fasores adyacentes cualesquiera es  $45^\circ$ , por lo cual, una señal 8-PSK puede tener cuando mucho un desplazamiento de fase de  $\pm 22.5^\circ$  durante la transmisión, para seguir reteniendo su integridad [25].

<sup>15</sup> Referirse a [6] páginas 486, 487 para el desarrollo matemático.

Para reducir la cantidad de errores de transmisión, se utiliza *Código Gray*, también llamado *código de distancia mínima* o *código binario reflejado*, en el cual el código de tribits (tres bits) entre dos fases adyacentes cualesquiera sólo se cambia en un bit. Si una señal fuera a sufrir un desplazamiento de fase durante su transmisión, lo más probable es que se desplazara hacia un fador adyacente. Al utilizar código de Gray, sólo habría error de recepción en un solo bit [25].



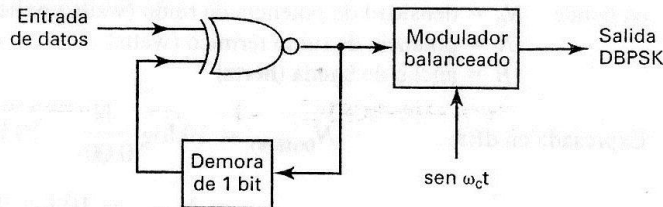
**Figura 13. Diagrama de constelación modulación 8-PSK. Fuente [25].**

Con un modulador 8-PSK hay un cambio de fase en la salida, por cada tres bits de datos de entrada. En consecuencia, los baudios en el 8-PSK equivalen a un tercio la rapidez de entrada ( $\text{baud} = f_b/3$ ), al igual que el ancho de banda mínimo [25].

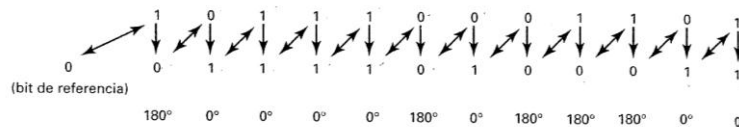
### 2.3.8 Modulación Por Corrimiento De Fase Diferencial

Las señales moduladas por corrimiento de fase no pueden ser detectadas en forma no coherente. Sin embargo, se puede utilizar una técnica parcialmente coherente denominada *codificación diferencial* [7]. En la modulación por

desplazamiento diferencial de fase (DPSK, *Differential Phase Shift Keying*) la información binaria de la entrada es contenida en la diferencia entre las fases de dos elementos sucesivos de señalización, y no en la fase absoluta. Al utilizar DPSK no se hace necesario recuperar una portadora de fase coherente. En lugar de ella, un elemento de señalización recibido se demora por un intervalo de frecuencia del elemento de señalización, y a continuación se compara con el siguiente elemento de señalización recibido. La diferencia de fase de los dos elementos de señalización determina la condición lógica de los datos [25].



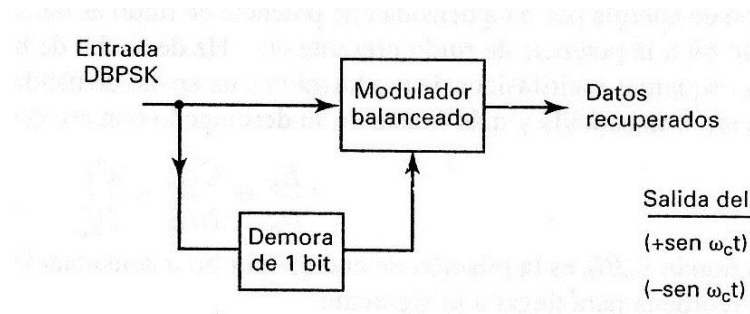
**Figura 14. Diagrama de bloques transmisor DPSK. Fuente [25].**



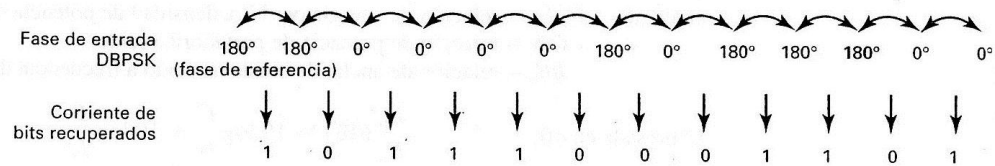
**Figura 15. Diagrama de sincronización transmisor DPSK. Fuente [25].**

En el transmisor DBPSK, un bit que entra se reconoce XNOR<sup>16</sup> con el bit anterior antes de llegar al modulador BPSK. Se supone un bit inicial de referencia para iniciar la comparación.

<sup>16</sup> XNOR: compuerta lógica que proporciona un 1 lógico, sólo si las dos entradas son iguales. Sólo es verdadera su salida, si ambos componentes de entrada tienen el mismo valor lógico.



**Figura 16. Diagrama de bloques receptor DBPSK. Fuente [25].**



**Figura 17. Secuencia de sincronización receptor DBPSK. Fuente [25].**

En el receptor DPSK, la señal recibida se retrasa un tiempo de bit y a continuación se compara con el siguiente elemento de fase de referencia en el demodulador balanceado. Si son iguales se genera un 1 lógico, si son distintas, se genera un 0 lógico. Si se supone la fase en forma incorrecta sólo el primer bit desmodulado es erróneo [25].

## **2.4 SDR**

Radio Definido por Software (sus siglas son provenientes de su definición en inglés, *Software Defined Radio*).

SDR representa un concepto, que surgió hace aproximadamente dos décadas, para dar mayor control al software de un sistema de comunicación que a su contraparte de hardware. Un sistema SDR reúne tecnologías tanto de Software, como de Hardware y de Radiofrecuencia (RF), donde la integración de estas tres permite que puedan ser recibidas, procesadas y transmitidas señales de radio con mayor flexibilidad y eficiencia, comparado con enfocar el sistema a sólo parte de hardware. El principal objetivo es que la mayor parte de los componentes del sistema puedan ser implementados en software para que así, mediante un dispositivo reprogramable, estos componentes puedan ser reconfigurados fácilmente cuando así se requiera, y los problemas que anteriormente eran relacionados con el hardware (elementos físicos), sean ahora problemas de software (de código o programación) [20].

### **2.4.1 Antecedentes**

A finales del siglo XX el auge de las comunicaciones inalámbricas trajo consigo un gran cambio en la arquitectura de los sistemas de radiocomunicación.

Los primeros sistemas de comunicación basados en su totalidad en hardware eran adecuados para la época en la medida que permitían el intercambio de la información entre puntos dispersos geográficamente de forma relativamente buena, aunque las máquinas tendían a tener un corto periodo de vida ya que eran diseñadas para ser descartadas y remplazadas [21]; dado que la implementación

totalmente en hardware resultaba bastante costosa debido a que en el momento de presentarse alguna falla, para cambiar una característica o el propósito particular del sistema, se hacía necesario modificar uno o más elementos y por lo tanto toda la configuración del sistema debía ser modificada.

Fue así, como se comenzó una carrera por parte de desarrolladores de hardware por crear circuitos más elaborados con mejoras y avances en los dispositivos físicos para solucionar estos inconvenientes, surgiendo tecnologías cada vez más innovadoras y avanzadas que pudieran, con eficiencia, responder a los requerimientos cada vez más exigentes del mercado para optimizar y facilitar el intercambio de información de una fuente a un receptor.

Los dispositivos de Radio Definidos por Software surgieron como una excelente alternativa para la solución de estos inconvenientes. Permiten ofrecer mayor flexibilidad al sistema de radiocomunicación ya que con unos cuantos cambios en código y programación, se pueden implementar diferentes funciones en una misma pieza física, sin necesidad de realizar alguna modificación al hardware del sistema.

El origen exacto de esta técnica se remonta a requerimientos de tipo militar en el año de 1984, en un programa del gobierno estadounidense denominado *SpeakEasy*, que buscaba configurar más de 10 tipos de tecnologías inalámbricas en un equipo programable [21]. Luego, en 1990 Joseph Mitola, considerado como el padre de esta tecnología, definió los primeros estándares y regulaciones que permitieron darle funcionalidad a esta técnica y conocerlo como Radio Definido por Software. Desde ese tiempo en adelante, han surgido importantes y diversas aplicaciones tanto comerciales como civiles que le han dado el posicionamiento actual y la importancia en la industria de este concepto.

## 2.4.2 Concepto

SDR es una tecnología que define un conjunto de técnicas y procedimientos para el procesamiento de señales de radio por medio de un dispositivo de propósito general. [11]

La técnica SDR, Radio Definido por Software, surgió para ofrecer una mayor flexibilidad y, potencialmente, una vida productiva más larga a los sistemas de radiocomunicación; [11] definiendo un radio como cualquier dispositivo que transmite en forma inalámbrica o recibe señales en la parte de radiofrecuencia del espectro electromagnético, la cual esta situada entre unos 3 kHz y unos 300 GHz.

Como se mencionó anteriormente, la idea era convertir los problemas de hardware en problemas de software, debido a que si algún parámetro o característica del sistema de comunicación debía cambiarse, todo el hardware debía ser reconfigurado; por lo cual eran frecuentes problemas tanto de costo, por parte del usuario, como de funcionalidad, por parte del dispositivo. De este modo un mismo dispositivo de hardware, podía realizar diferentes funciones, en distintos instantes de tiempo, por medio de cambios en su configuración mediante software, lo que generó gran flexibilidad a la hora de realizar un sistema de comunicación. De aquí radica la mayor ventaja de un Radio Definido por Software sobre un radio digital, ya que aunque en este último, la información sea digitalizada en algún punto entre el procesado de la señal y la antena, si no es reconfigurable y/o reprogramable no puede ser considerado como un sistema SDR.

SDR usa dispositivos digitales reprogramables para mejorar el procesamiento de señal necesario para transmitir y recibir información en banda base a radiofrecuencia. Dispositivos como los procesadores digitales de señales (DSP's), y los arreglos de compuertas de campo programables (FPGA's) usan software para proveerlos de la funcionalidad de procesamiento de señales requerido [21].

### **2.4.3 Implementación**

Un sistema SDR está integrado por tres secciones que se comunican entre sí:

1. Sección de Radio Frecuencia: Parte de hardware que se encarga de recibir y transmitir las señales de radio.
2. Sección de Frecuencia Intermedia: Parte de hardware que se encarga de convertir la señal digital en una señal analógica en la transmisión y de digitalizarla en recepción, realizando el cambio de frecuencia al rango adecuado.
3. Sección Banda base: Herramienta de software para el procesamiento de la señal en banda base (filtrado, modulación, etc).

## 2.5 GNU Radio

GNU Radio es una herramienta de software libre que combinada con una parte de hardware (USRP) permite construir e implementar sistemas de Radios Definidos por Software en el cual las formas de onda transmitidas y recibidas son especificadas por medio de software.<sup>17</sup> [24]

GNU Radio contiene archivos y aplicaciones agrupadas en librerías donde se encuentran una gran variedad de bloques de procesamiento de señal como: filtros, moduladores (GMSK, PSK, QAM, OFDM), demoduladores, fuentes de señal, osciloscopio, analizador de espectros, entre otros elementos fundamentales de comunicaciones, que permiten crear un flujo de señal para el envío y recepción de los datos.

Una de las mayores ventajas que presenta esta herramienta es que al ser de carácter libre, cualquier desarrollador o usuario de este software puede realizar nuevas funciones y modelos, aportando a la mejora de la herramienta y ofreciendo alternativas de aplicaciones software a los demás usuarios. Por otro lado, este software puede ser instalado en sistemas operativos GNU-Linux como Ubuntu, aunque también puede ser instalado en Mac y Windows.

Hay dos lenguajes de programación utilizados en GNU Radio, C++ y Python, los cuales tienen diferentes usos en el sistema. Todos los bloques de procesamiento de señal están escritos en C++, mientras que Python es usado para unir estos bloques y crear el flujo completo de gráfico para el sistema de comunicaciones, lo cual implica que este último es un lenguaje de más alto nivel, un lenguaje interpretado, es decir, que no se compila sino que el sistema operativo lo ejecuta directamente. [20]

---

<sup>17</sup> Una guía de instalación de GNU Radio Companion se encuentra en Anexo F.

Como se mencionó anteriormente, GNU Radio cuenta con una gran variedad de bloques, por lo tanto, para la mayoría de aplicaciones básicas no es necesario entrar en contacto con el lenguaje C++, solamente utilizar lenguaje Python para integrar los bloques y llevar a cabo el flujo adecuado del sistema. Sin embargo para un trabajo más sofisticado y para una aplicación en particular se puede utilizar C++ para crear un bloque con características y funciones especiales<sup>18</sup>.

Para un uso más flexible de la herramienta, y no necesariamente utilizar el lenguaje Python para la integración de los diferentes bloques, existe la aplicación GNU Radio Companion (GRC, *GUI Radio Design system*), que por medio de una interfaz gráfica permite al usuario conectar los bloques creando el gráfico de flujo de señal y generando el código fuente del flujo gráfico implementado. [12]

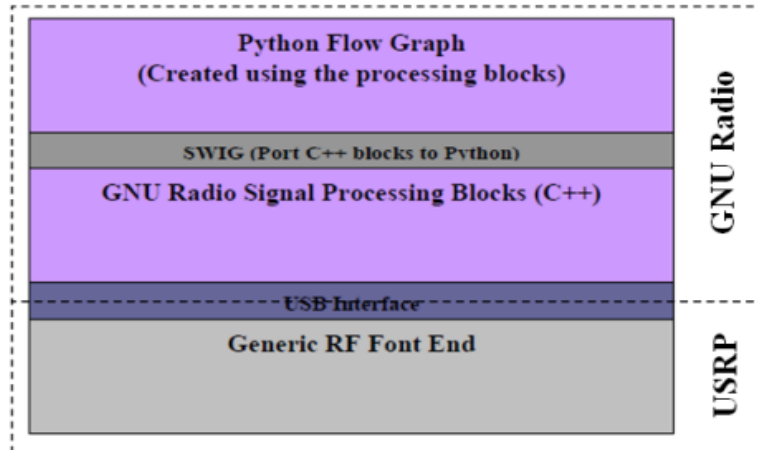
Un bloque en GNU Radio procesa secuencias de datos de forma continua, los cuales fluyen de los puertos de entrada a los puertos de salida de cada bloque. Uno de los parámetros más importantes que se debe determinar en un bloque es el tipo de dato que procesa, entre ellos se encuentran: byte, entero corto, entero, flotante y complejo, el cual dependiendo de la función que realice el bloque, procesa dos o más de estos tipos de datos. Algunos bloques cuentan con solo puertos de entrada o puertos de salida, estos sirven como sumideros y fuentes respectivamente en el esquemático. [13]

Para que los bloques implementados en C++ puedan ser reconocidos y portados a Python se utiliza el SWIG (*Simplified Wrapper and Interface Generator*), que es usado como interfaz de compilación permitiendo la integración de los bloques en C++ con el diseño del sistema en lenguaje Python. SWIG genera el correspondiente código Python (.py) y la librería (.so). [6]

---

<sup>18</sup> Referirse al Anexo D, "Práctica No. 4. Creación de bloques".

Estas comunicaciones entre los diferentes componentes de GNU Radio (software) y con el dispositivo USRP (Hardware), que conforman en su totalidad el sistema SDR, se presentan en la siguiente figura:



**Figura 18. Componentes GNU Radio.**<sup>19</sup>

<sup>19</sup> Fuente: Experimental Study of OFDM Implementation Utilizing GNU Radio and USRP – SDR. Proceedings of the 2009 IEEE 9th Malaysia International Conference on Communications. Kuala Lumpur Malaysia. 15 -17 Diciembre, 2009.

### 3 ESTRUCTURA GENERAL Y METODOLOGIA DE DESARROLLO

#### 3.1 Estructura General

El objetivo general de las prácticas de laboratorio es realizar una introducción al estudio de comunicaciones digitales usando la técnica de Radio Definido por Software (SDR, *Software Defined Radio*), por medio de la herramienta GNU Radio y dispositivos USRP (*Universal Software Radio Peripheral*) para su implementación.

En Figura 19 se muestra un esquema general del montaje utilizado para implementar el sistema de comunicaciones digitales inalámbricas en el laboratorio:

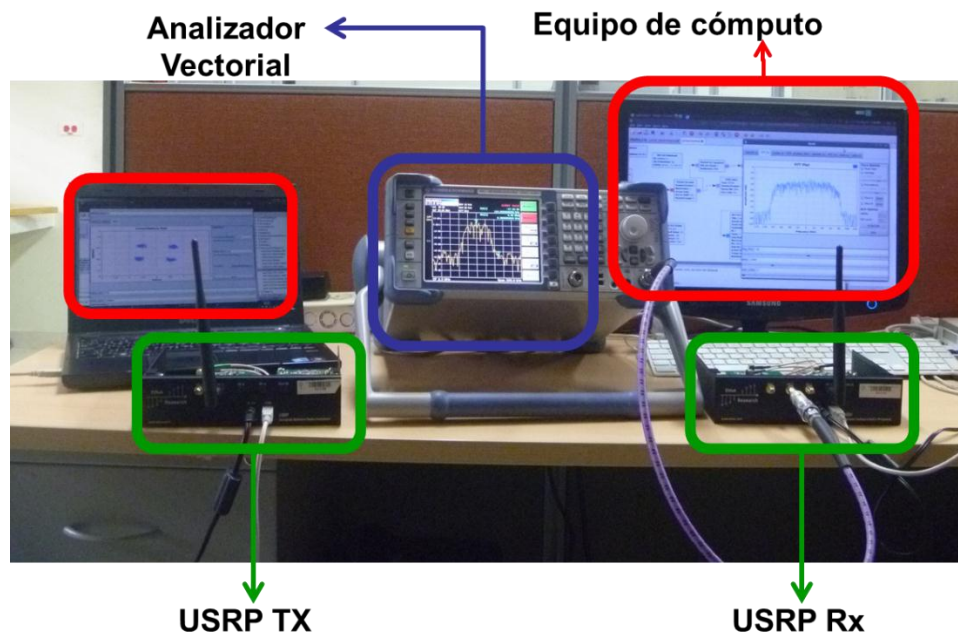
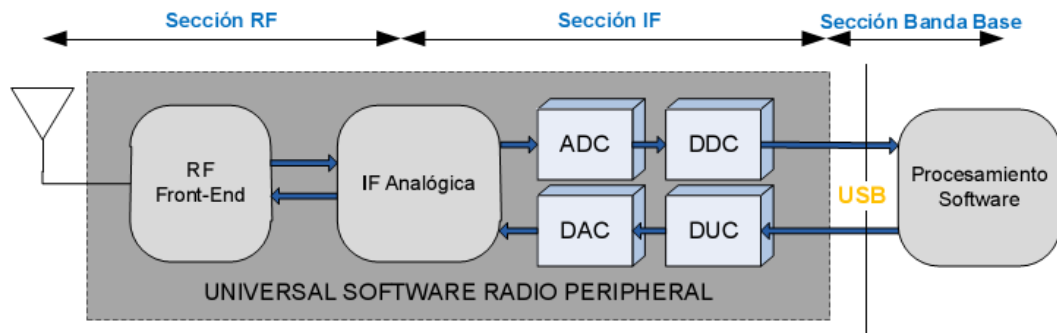


Figura 19. Montaje sistema de comunicaciones digitales SDR.

### 3.1.1 USRP

*Universal Software Radio Peripheral*, es una plataforma flexible de bajo costo para sistemas SDR (*Software Defined Radio*) desarrollada por Matt Ettus<sup>20</sup> [19], diseñada para trabajar con un procesador externo (computador) que permite implementar de forma rápida, sistemas flexibles y potentes de Radio Definido por Software.

A continuación se ilustra la configuración completa de un sistema SDR.



**Figura 20. Bloques principales en un sistema de comunicaciones SDR. Fuente [20].**

Las secciones RF y de IF se llevan a cabo en el dispositivo USRP. En la sección RF se adecuan las señales que se reciben del medio y las convierte en

<sup>20</sup> **Matt Ettus:** Fundador y presidente de Ettus Research LLC. Obtuvo dos títulos de grado en Ingeniería Eléctrica y Ciencias de la Computación en 1996 de la Universidad de Washington, St Louis. Luego, un título de maestría en Ingeniería Eléctrica y Computación de la Universidad de Carnegie, Mellon en 1997.

Matt Ettus es un importante contribuyente al proyecto GNU Radio, una herramienta de software libre para SDR, y es el creador del Universal Software Radio Peripheral (USRP). Su compañía, Ettus Research, esta localizada en Mountain View, California y vende la línea de sistemas de Radio Definido por Software denominada Universal Software Radio Peripheral (USRP).

Disponible en: <http://engineering.wustl.edu/alumniawards.aspx?year=2011> [Consultado en julio 19 de 2012].

frecuencias intermedias para el caso de la recepción. En la transmisión, las señales son amplificadas y moduladas adecuándolas para la transmisión en el aire. En la sección IF se lleva a cabo la digitalización y paso a banda base de la señal en el caso de la recepción, para la transmisión se cambia la señal de banda base a frecuencia intermedia y luego se convierte a forma analógica. [11]

Integrando el USRP junto con la herramienta de software libre GNU Radio, que permite realizar el procesamiento y tratamiento de la señal en banda base, se obtiene un sistema completo de radiocomunicación que permite reconfigurar las características del sistema con modificaciones sólo de software sin tener que incurrir en cambios de hardware. Una vez instalado GNU Radio y luego de configurar y conectar el USRP al computador a través de la interfaz USB, el sistema SDR se encuentra listo para transmitir y recibir gran variedad de señales, en diversas configuraciones.

## **Ventajas del USRP**

Las principales ventajas del uso de dispositivos USRP para la implementación de sistemas SDR son:

- Bajo costo.<sup>21</sup>
- Plataforma flexible.

---

<sup>21</sup> El costo total del dispositivo USRP utilizado en este proyecto es de USD \$1250. Disponible en <https://www.ettus.com/product/details/USRP-PKG> [Fecha de consulta: julio 20 de 2012].

Siendo significativamente más económico, para el alcance de este proyecto, comparado con la plataforma National Instruments NI PXIe-5641R, que cuenta con el soporte y la capacidad de cómputo de LabView, cuyo precio es de USD \$7699. Disponible en <http://sine.ni.com/nips/cds/view/p/lang/en/nid/207050> [Fecha de consulta: agosto 8 de 2012].

Otra plataforma para desarrollo de Sistemas SDR es FLEX-6500 Signature Series SDR Transceiver de la empresa FlexRadio Systems – Software Defined Radios, cuyo precio es de USD \$4299. Disponible en [http://cart.flexradio.com/FLEX-6500-Signature-Series-SDR-Transceiver\\_p\\_912.html](http://cart.flexradio.com/FLEX-6500-Signature-Series-SDR-Transceiver_p_912.html) [Fecha de consulta: julio 22 de 2012].

- Software libre.
- Cuenta con una gran comunidad de desarrolladores que han contribuido con una gran variedad de aplicaciones ofreciendo mejoras y actualizaciones del software. [9]

En la familia de productos de Ettus Research<sup>22</sup>, existen dos versiones de USRP, a saber, USRP1 y USRP2. En este proyecto se trabaja con la versión USRP1, tarjeta hija (*daughter board* RFX2400).

### **Características del USRP**

El USRP1<sup>23</sup> consiste en dos niveles de tarjetas principales, tarjeta madre (*motherboard*) y las tarjetas hijas<sup>24</sup> (*daughter boards*).

La tarjeta madre contiene:

- Cuatro (4) ADCs (*Analog to Digital Converters*) de 12 bit con una tasa de muestreo de 64 MS/s (*Million samples per second*).
- Cuatro (4) DACs (*Digital to Analog Converters*) de 14 bit con una tasa de muestreo de 128 MS/s (*Million samples per second*).

---

<sup>22</sup> Disponible en: <http://www.ettus.com/> [Consultado en julio 20 de 2012].

<sup>23</sup> *Datasheet* Disponible en [https://www.ettus.com/content/files/06983\\_Ettus\\_USRP1\\_DS\\_Flyer\\_HR.pdf](https://www.ettus.com/content/files/06983_Ettus_USRP1_DS_Flyer_HR.pdf) [Consultado en julio 21 de 2012].

<sup>24</sup> *Daughter board* RFX2400. Disponible en <https://www.ettus.com/product/details/RFX2400> [Consultado en julio 21 de 2012].

- Dos (2) DUC (*Digital Up Converters*) que lleva a cabo el proceso de aumento o expansión de la frecuencia de muestreo (interpolación).
- Interfaz de alta velocidad USB 2.0.
- Conector de alimentación de DC.
- Altera Cyclone EP1C12 FPGA (*Field Programmable Array*): Esta incluye cuatro (4) DDC (*Digital Down Converter*) para disminuir (diezmar) la frecuencia de muestreo. [9]

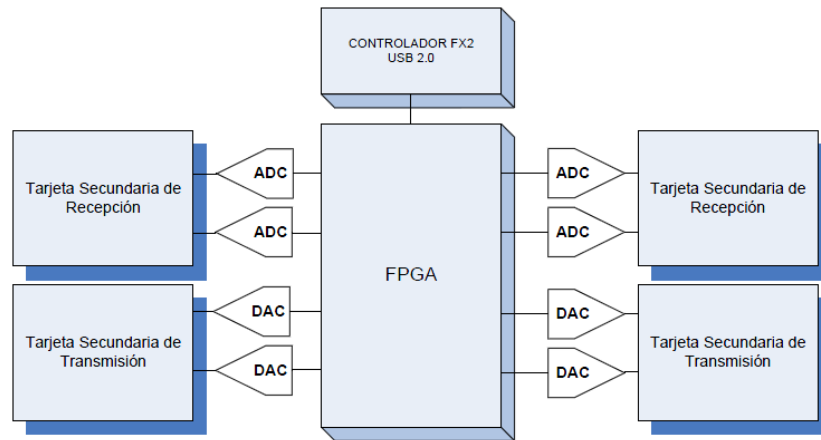
La tarjeta hija (*daughter board*) actúa como la sección de Radiofrecuencia del SDR. Es la encargada de recibir-transmitir las señales de radiofrecuencia, además de adecuarlas. En el caso de la recepción convierte las señales a frecuencia intermedia (IF) y en el caso de la transmisión amplifica y modula las señales de IF.

Hay cuatro ranuras en la tarjeta madre lo que implica que pueden ser conectadas hasta dos tarjetas hijas (cada una de ellas tiene un transmisor y un transceptor), dos de ellas etiquetadas como TXA y TXB que sirven para transmisión y dos de ellas RX/TX A y RX/TX B que son transceptores (sirven tanto para transmisión como para recepción).

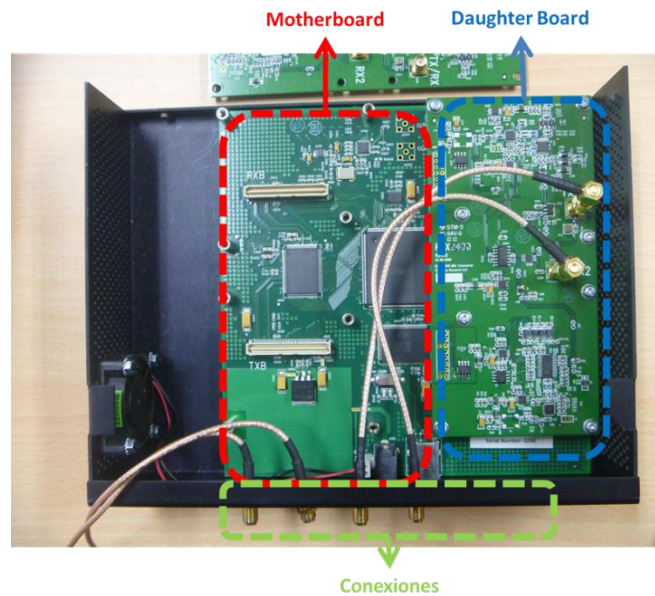
Cada ranura de la tarjeta hija tiene acceso a 2 de los 4 convertidores analógicos de alta velocidad de señal (DAC de salida para tarjeta de transmisión, ADC de entrada para tarjeta de recepción).

Una amplia variedad de tarjetas hijas están disponibles para manejar diferentes rangos de frecuencia y así utilizarse en una extensa gama de aplicaciones [6].

A continuación se ilustra la forma como se dispone la configuración de los componentes en el USRP especificados anteriormente.



**Figura 21. Diagrama de bloques del Universal Software Radio Peripheral. Fuente [20].**



**Figura 22. USRP con tarjeta hija RFX2400 utilizado en el proyecto.**

## Configuración del USRP

Para utilizar el USRP es necesario conectarlo a un procesador externo, en este caso el computador. Para verificar que el dispositivo es reconocido, se procede a abrir la terminal de Ubuntu y se escribe la siguiente línea de código:

```
uhd_usrp_probe
```

Este comando imprime en pantalla todas las propiedades y características del dispositivo conectado al computador, que en este caso es el USRP1, muestra el número de serial (único para cada dispositivo) y los rangos de frecuencia de la tarjeta hija, rangos de ganancia ,entre otras características.

Para los dos dispositivos USRP utilizados en este proyecto se obtuvieron las siguientes características<sup>25</sup>:

```
/
|   Device: USRP1 Device
|   /
|   |   Mboard: USRP1 (Classic)
|   |   serial: 4cc49b63
|   |   Time sources: none
|   |   Clock sources: internal
|   |   Sensors:
|   |   /
|   |   |   RX DSP: 0
|   |   |   Freq range: -32.000 to 32.000 Mhz
|   |   /
|   |   |   RX DSP: 1
|   |   |   Freq range: -32.000 to 32.000 Mhz
|   |   /
|   |   |   RX Dboard: A
|   |   |   ID: RFX2200 (0x002c)
```

<sup>25</sup> Para el otro dispositivo utilizado, las características son las mismas, exceptuando el número de serial, que es 4d05be08.

---

RX Subdev: 0  
Name: RFX RX  
Antennas: TX/RX, RX2, CAL  
Sensors: lo\_locked, rssi  
Freq range: 2000.000 to 2400.000 Mhz  
Gain range PGA0: 0.0 to 70.0 step 0.0 dB  
Connection Type: QI  
Uses LO offset: No

---

RX Codec: A  
Name: ad9522  
Gain range pga: 0.0 to 20.0 step 1.0 dB

---

RX Dboard: B  
ID: RFX2200 (0x002c)

---

RX Subdev: 0  
Name: RFX RX  
Antennas: TX/RX, RX2, CAL  
Sensors: lo\_locked, rssi  
Freq range: 2000.000 to 2400.000 Mhz  
Gain range PGA0: 0.0 to 70.0 step 0.0 dB  
Connection Type: QI  
Uses LO offset: No

---

RX Codec: B  
Name: ad9522  
Gain range pga: 0.0 to 20.0 step 1.0 dB

---

TX DSP: 0  
Freq range: -44.000 to 44.000 Mhz

---

TX DSP: 1  
Freq range: -44.000 to 44.000 Mhz

---

TX Dboard: A  
ID: RFX2200 (0x002d)

---

TX Subdev: 0  
Name: RFX TX  
Antennas: TX/RX, CAL  
Sensors: lo\_locked  
Freq range: 2000.000 to 2400.000 Mhz  
Gain Elements: None

```

| | | | Connection Type: IQ
| | | | Uses LO offset: Yes
| | | | /
| | | | TX Codec: A
| | | | Name: ad9522
| | | | Gain range pga: -20.0 to 0.0 step 0.1 dB
| | | | /
| | | | TX Dboard: B
| | | | ID: RFX2200 (0x002d)
| | | | /
| | | | TX Subdev: 0
| | | | Name: RFX TX
| | | | Antennas: TX/RX, CAL
| | | | Sensors: lo_locked
| | | | Freq range: 2000.000 to 2400.000 Mhz
| | | | Gain Elements: None
| | | | Connection Type: IQ
| | | | Uses LO offset: Yes
| | | | /
| | | | TX Codec: B
| | | | Name: ad9522
| | | | Gain range pga: -20.0 to 0.0 step 0.1 dB

```

## Aplicaciones

El dispositivo USRP es usado en una gran variedad de aplicaciones. Algunos de los campos más importantes donde este dispositivo es utilizado<sup>26</sup>:

- Aplicaciones comerciales.
- Defensa y seguridad de la nación.
- Investigación inalámbrica.
- Educación.

<sup>26</sup> Referirse al documento “Sistemas de comunicaciones basados en la técnica Radios Definidos por Software” (Anexo E), para encontrar aplicaciones específicas.

### **3.2 Metodología De Desarrollo**

La metodología utilizada para el desarrollo de cada una de las prácticas en este proyecto, con el objetivo de ser usadas como guías para un laboratorio de comunicaciones digitales utilizando la técnica Radio Definido por Software (*Software Defined Radio*, SDR), se presenta de forma estructurada y secuencial, de tal forma que el usuario se familiarice con la técnica y así desarrolle las bases para la realización de trabajos de mayor complejidad en el campo.

A continuación se presenta la estructura de las prácticas:

**1. Título de la práctica:**

Tema a desarrollar durante la guía de trabajo.

**2. Objetivo general:**

Precisa la finalidad de la práctica, en cuanto a sus expectativas más amplias.

**3. Objetivos específicos:**

Determinan la precisión y cumplimiento de los requerimientos o propósito de la práctica.

**4. Conceptos previos:**

Plantea un compilado de los conceptos fundamentales, que el usuario debe poseer a la hora del desarrollo de la práctica, para un óptimo aprendizaje.

**5. Tiempo estimado:**

Muestra una aproximación del tiempo estimado para el desarrollo de las actividades planteadas en la práctica.



Universidad  
Industrial de  
Santander

## UNIVERSIDAD INDUSTRIAL DE SANTANDER

ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES  
Perfecta Combinación entre Energía e Intelecto

### 6. **Procedimiento:**

Presenta la actividad dividida en ítems que indican los pasos requeridos para implementar los objetivos planteados.

### 7. **Referencias:**

Cita libros y páginas web de consulta para la adquisición de los conocimientos que facilitan la comprensión del enfoque general de la guía, en cuanto a teoría y conocimientos fundamentales requeridos para el desarrollo de la guía.

## **4 PRÁCTICAS A IMPLEMENTAR**

El propósito de estas guías de laboratorio, junto con los documentos anexos, es introducir al lector en el desarrollo de sistemas de comunicaciones digitales utilizando la técnica de Radio Definido por Software, SDR, teniendo GNU Radio Companion como plataforma de desarrollo y realizando la implementación a través de dispositivos USRP.

Cada una de las guías está diseñada para que el usuario interactúe con los componentes básicos de este tipo de sistemas de comunicaciones, de tal manera que pueda iniciarse en la implementación de esta tecnología, creando las bases para futuros desarrollos y propuestas en este campo de las comunicaciones, a la vez que estudia los conceptos relacionados con las técnicas de modulación digital GMSK y PSK, permitiendo poner en práctica la teoría adquirida en el aula de clases sobre comunicaciones digitales.

A continuación se presenta una descripción de cada una de las prácticas a implementar:

### **1. Introducción a GNU Radio.**

GNU Radio es la herramienta de software libre utilizada para el desarrollo de radios definidos por software en este proyecto, motivo por el cual, el primer objetivo del usuario es conocer e identificar la plataforma, a través del desarrollo de un proyecto introductorio explicado de forma detallada.

Con el desarrollo de esta práctica, el usuario estará en la capacidad de:

- Desarrollar diferentes tipos de proyectos en GNU Radio, utilizando las librerías incluidas por defecto.
- Realizar simulaciones de sistemas de comunicaciones digitales en el entorno proporcionado por GNU Radio.

## **2. Modulación PSK.**

PSK es una de las técnicas de modulación utilizada en el desarrollo de este proyecto. En GNU Radio y con dispositivos USRP se implementa el sistema de comunicaciones utilizando esta técnica de modulación.

Con el desarrollo de esta práctica, el usuario estará en la capacidad de:

- Reconocer las características de las señales transmitidas mediante modulación PSK y sus distintas configuraciones, a saber, BPSK, QPSK, 8PSK, DPSK.
- Identificar los aspectos principales relacionados con la modulación PSK y sus diferentes configuraciones.

## **3. Modulación GMSK.**

Junto con la plataforma de desarrollo GNU Radio y utilizando dispositivos USRP es posible implementar un sistema de comunicación inalámbrico, donde la

información es modulada por medio de la técnica de modulación digital GMSK, que es un tipo especial de modulación por desplazamiento de frecuencia de fase continua. De esta manera, se introduce al usuario en la implementación de sistemas de Radio Definidos por Software.

Con el desarrollo de esta práctica, el usuario estará en la capacidad de:

- Desarrollar sistemas de comunicaciones basados en la técnica de Radios Definidos por Software.
- Reconocer las características de las señales transmitidas mediante modulación GMSK.
- Identificar los aspectos principales relacionados con la modulación GMSK.

#### **4. Creación de bloques GRC.**

GNU Radio, en su entorno gráfico, utiliza bloques de procesamiento de señal, los cuales son programados en C++ y Python. Existen diferentes formas para desarrollar un bloque de procesamiento de señal, una de las cuales es utilizando una plantilla predeterminada, la cual se modifica de acuerdo a los requerimientos del usuario.

Con el desarrollo de esta práctica, el usuario estará en la capacidad de:

- Desarrollar bloques de procesamiento de señales para el entorno gráfico de GNU Radio.

## **CONCLUSIONES**

En el desarrollo del presente proyecto se logró documentar la experiencia de trabajo con un sistema de radiocomunicación definido por software, SDR, para el estudio de la asignatura de comunicaciones digitales, presentándose a su vez, como una alternativa de soporte para su respectivo laboratorio. En la Universidad Industrial de Santander no se había trabajado ni investigado esta tecnología, por lo cual este proyecto presenta un punto de partida para el estudio del concepto a nivel local, que aunque ya lleva varios años siendo implementado a nivel mundial, no ha sido investigado ampliamente en el país, teniendo en cuenta el impacto positivo que generan sus aplicaciones en el campo de las comunicaciones inalámbricas.

Las guías de laboratorio que se proponen en este trabajo, representan el esfuerzo por parte de los autores de plantear una base práctica para la asignatura estudiando los conceptos de comunicaciones digitales más importantes. Con la eficiencia del hardware y la flexibilidad del software integrado en un sistema SDR, se pretende dar un punto de inicio innovador para un laboratorio de comunicaciones digitales que permita generar expectativa e interés por parte de los estudiantes de trabajar con una tecnología actual y de investigación continua a nivel mundial.

La experiencia adquirida con la herramienta y el trabajo personal llevado a cabo por los autores de este proyecto, permite asegurar que el sistema SDR implementado, tanto la parte de hardware (USRP) como la parte de software (GNU Radio) son bastantes intuitivas de utilizar. El usuario debe tener algunos conceptos básicos de Linux, si es el caso que no haya manejado un sistema operativo con un núcleo de este tipo, pero la instalación del software y la sincronización con el USRP son tareas relativamente sencillas junto con el diseño

del flujo de señal mediante la interfaz gráfica GNU Radio Companion, siendo esta última de gran utilidad al no tener que profundizar en lenguajes de programación como Python para la integración de los bloques de procesamiento de señal. Todos estos factores hacen del trabajo con un sistema SDR, una experiencia interesante, práctica y a la vez innovadora, como se evidencia en la documentación sobre sistemas SDR a nivel mundial presentado en este proyecto.

La radio cognitiva es una tecnología de gran investigación en el campo de las comunicaciones inalámbricas, que teóricamente permitiría proveer la flexibilidad e inteligencia en un sistema de comunicaciones cambiando sus características de comunicación automáticamente dependiendo de los requerimientos de los usuarios, lo cual sería un punto de partida para una nueva era de transmisión de la información. Sin embargo esto sólo se puede lograr con un buen análisis de los sistemas de Radios Definidos por Software los cuales representan la plataforma para esta innovadora tecnología. Por lo tanto abordar un estudio profundo, interpretando el concepto de SDR, la integración y comunicación de sus componentes, la forma como se implementa, tal como se presenta en este documento, es de gran importancia para los avances e investigaciones futuras en el área de las comunicaciones.

El uso de un sistema operativo como Ubuntu, el cual utiliza un núcleo Linux, y la herramienta de software libre GNU Radio, permiten hacer de SDR una tecnología de carácter completamente libre por lo cual, no deben ser adquiridas ni compradas ningún tipo de software o licencias. De igual forma, debido a esta característica, existe una gran comunidad de desarrolladores de software de esta tecnología que contribuyen al mejoramiento y nuevos avances en esta área, así como están dispuestos a ofrecer soporte en foros a las personas que requieran ayuda con inquietudes de algún tema en particular ya sea del dispositivo USRP o del software GNU Radio.

## **RECOMENDACIONES Y TRABAJOS FUTUROS**

Se recomienda realizar la elaboración de nuevos bloques de procesamiento de señal en GNU Radio, a partir de la guía de creación de bloques sugerida en este documento, que permitan ofrecer nuevas herramientas y componentes a los sistema de radio comunicación digital que puedan desarrollarse en un futuro, además, de implementar otras técnicas de modulación digital diferentes a las presentadas en este trabajo (QAM, OFDM) que puedan permitir una mayor variedad de aplicaciones de GNU Radio y USRP, poniendo en práctica los conocimientos de comunicaciones digitales aprendidos.

Con las múltiples alternativas que permiten USRP y GNU Radio, se sugiere hacer uso de estas herramientas de tal forma que se implementen en otras importantes aplicaciones, diferentes a la enseñanza de modulaciones digitales, como Open BTS (*Base Transceiver System*) para servicios de Volp, lo que permitiría realizar mayor investigación y posicionamiento local de esta tecnología. Adicionalmente, sería interesante adquirir otro tipo de *daughterboards* para aplicaciones en bandas de frecuencia diferentes, permitiendo obtener otro tipo de sistemas de comunicaciones como por ejemplo, sistemas FM.

En el desarrollo del proyecto se utilizó la aplicación GNU Radio Companion (GRC) de la plataforma GNU Radio, el cual por medio de una interfaz gráfica maneja el diseño de bloques del sistema de radiocomunicación implementado que representa el flujo gráfico de la señal. Este proceso de conexión de bloques puede realizarse también por medio de la manipulación del lenguaje de programación de alto nivel Python. Se sugiere realizar algunos programas básicos en GNU Radio aplicando la filosofía de este lenguaje que en algunos casos puede resultar más



## **UNIVERSIDAD INDUSTRIAL DE SANTANDER**

**ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES**  
**Perfecta Combinación entre Energía e Intelecto**

práctico y funcional que utilizar la respectiva interfaz gráfica de la herramienta de software.

## **BIBLIOGRAFÍA**

- [1] AGUILAR CÁRDENAS, Eulises. BOTELLO CASTELLANOS, Luis Hernando. CAMPO TOVAR, Roberto Carlos. Concepción, diseño e implementación del laboratorio de comunicaciones para la Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones, Universidad Industrial de Santander, Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Bucaramanga 1999.
- [2] AMAYA, Ferney. Aplicaciones para telecomunicaciones empleando FPGAs: una aproximación a Radio Software. Revista Colombiana de Tecnologías de Avanzada, Volumen 1, Número, 7 año 2006.
- [3] ARTÉS RODRIGUEZ, Antonio. Comunicaciones digitales, Madrid: Pearson Educación, Prentice-Hall, 2007.  
ISBN: 84-8322-348-2
- [4] BALLESTEROS LARROTTA, Dora María. MOJICA RODRIGUEZ, Néstor Yezid. VERA MERCADO, Erick José. Laboratorio virtual de comunicaciones usando LABVIEW. Director Jaime Barrero Pérez, codirectores Faver Amorocho, Jesús David Acero. Universidad Industrial de Santander. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Bucaramanga 2001.
- [5] BLAKE, Roy. Sistemas electrónicos de comunicaciones, México: Thomson Learning, 2004.  
ISBN: 970-686-365-6



- [6] KE-YU, Chen. ZHI-FENG, Chen. GNU Radio. Dept. of Electrical Computer Engineering. University of Florida, Gainesville, Florida.
- [7] COUCH, Leon W. Sistemas de comunicación digitales y analógicos, México: Pearson Educación, 2008, p. 8, 9, 339, 345, 353.  
ISBN: 970-26-1216-2
- [8] DUARTE DUARTE, Dili. URIBE, Oscar Daniel. Modelado y simulación de sistema de comunicaciones. Director Oscar Mauricio Reyes. Universidad Industrial de Santander. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Bucaramanga 2004.
- [9] Ettus Research- Universal Software Radio Peripheral. The Foundation for Complete Software Radio Systems. Datasheet USRP1.
- [10] FIGUEIRAS R., Aníbal. Una panorámica de las telecomunicaciones, España: Prentice Educación, 2002, p. 10, 12.  
ISBN: 84-205-3100-6
- [11] GALVIS QUINTERO, Alexander. CEBALLOS BETANCOUR, Christian A. DE SANCTIS GIL, Lukas. SDR: La alternativa para la evolución inalámbrica a nivel físico. Semillero de Tecnologías Inalámbricas (STI) – GIDATI. Universidad Pontificia Bolivariana, Medellín.
- [12] GNU Radio: GNU Radio Companion [en línea]. [Fecha de consulta: febrero 12 de 2012].  
Disponible en:  
<http://gnuradio.org/redmine/projects/gnuradio/wiki/GNURadioCompanion>

- [13] GNU Radio Hughes Systique HSC [en línea]. [Fecha de consulta: febrero 10 de 2012].  
Disponible en:  
[www.hsc.com](http://www.hsc.com)
- [14] HAYKIN, Simon. Communication systems, New York: John Wiley, 2001, p. 7, 8.  
ISBN: 0-471-17869-1
- [15] HAYKIN, Simon S. Digital communications, New York: John Wiley, 1988, p. 238.  
ISBN: 0-471-63775-0
- [16] HERRERA PÉREZ, Enrique. Comunicaciones II: Comunicación digital y ruido. México: Limusa, 1999, p. 126, 134.  
ISBN: 968-18-5719-4
- [17] LINN, Yair. An Ultra Low Cost Software Defined Radio Laboratory for Education and Research. SDR'09 Technical Conference and Product Exposition.
- [18] LIZCANO MARTINEZ, Juan Pablo. Implementación del laboratorio de comunicaciones analógicas y digitales. Director Homero Ortega Boada. Universidad Industrial de Santander. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Bucaramanga 2002.
- [19] MARWANTO, Arief. SARIJARI, Mohd Adib. FISAL, Norsheila. SYED YUSOF, Sharifah Kamilah. RASHID, Rozeha A. "Experimental Study of OFDM Implementation Utilizing GNU Radio and USRP-SDR". Proceedings of the 2009



IEEE 9th Malaysia International Conference on Communications 15 -17  
December 2009 Kuala Lumpur Malaysia.

- [20] PINAR DOMINGUEZ, Iván. Software Defined Radio: USRP y GNU Radio. España.
- [21] PROAÑO R., Enrique. RAMÍREZ, Erick. Transmisores De radio basados en software. Ingeniería Electrónica y Telecomunicaciones Comunicaciones Inalámbricas. Escuela Politécnica Nacional.
- [22] SKLAR, Bernard. Digital communications: Fundamentals and applications, Englewood Cliffs: Prentice-Hall, 1988, p. 3, 9, 10, 11, 14, 118.  
ISBN: 0-13-211939-0
- [23] STREMLER, Ferrel G. Sistemas de comunicación, Bogotá: Alfaomega: Fondo educativo interamericano, 1999, p. 116, 586.  
ISBN: 958-682-049-1
- [24] SUNDQUIST, Thomas. Waveform Development using Software Defined Radio. Linköpings Universitet, Sweden, 2006.
- [25] TOMASI, Wayne. Sistemas de comunicaciones electrónicas, México: Pearson Educación, 2003, p. 2, 8, 479, 485, 507, 509.  
ISBN: 970-26-0316-1



## **ANEXOS**

### **ANEXO A**

#### **PRÁCTICA No. 1**

##### **INTRODUCCIÓN A GNU RADIO**

### **INTRODUCCIÓN**

GNU Radio es una herramienta de desarrollo de software libre que provee bloques de procesamiento de señales para la implementación de Radios Definidos por Software (SDR, *Software Defined Radio*). Un diagrama de flujo de GNU Radio es un grupo de bloques de procesamiento de señal que son conectados para obtener sistemas de comunicaciones.

### **OBJETIVOS**

#### **OBJETIVO GENERAL**

Describir los principales pasos para el manejo de la herramienta GNU Radio, mediante el uso de diagramas de flujo.

## OBJETIVOS ESPECÍFICOS

1. Instalar el software GNU Radio con las respectivas librerías requeridas para las aplicaciones en este trabajo.
2. Implementar un proyecto sencillo que permita reconocer el manejo de la herramienta de software GNU Radio.
3. Construir un diagrama de flujo que permita obtener una señal *dial tone*.<sup>27</sup>

## TRABAJO PREVIO

Para realizar esta práctica se recomienda leer y estudiar la documentación relacionada con el proceso de instalación de GNU Radio que se encuentra en el documento “Instalación GNU Radio” (Anexo F), así como la información sobre GNU Radio que se encuentra en [1], además de revisar la documentación sobre USRP proporcionada.

Conocimiento básico en tratamiento de señales, relacionado con la teoría de muestreo.

---

<sup>27</sup> Dial tone: es una señal de telefonía utilizada para indicar que la central telefónica está trabajando, se ha reconocido un descolgado y está lista para aceptar una llamada. El tono se detiene cuando el primer numeral es marcado.

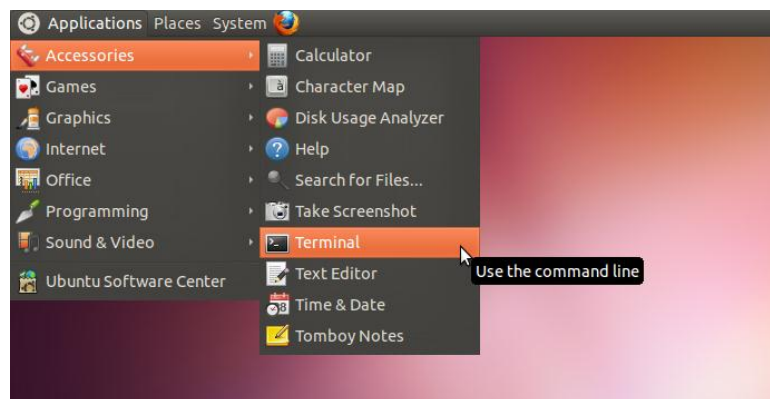
## TIEMPO ESTIMADO

El tiempo estimado para la realización de esta guía es de dos sesiones de dos horas cada una.

## PROCEDIMIENTO

### Primeros pasos

1. Para acceder al software GNU Radio-Companion es necesario ejecutar el comando ***gnuradio-companion*** en la *Terminal*<sup>28</sup> de Ubuntu, que se encuentra en el menú *Aplicaciones*, submenú *Accesorios*, como se muestra en las siguientes figuras.



**Figura A23. Acceso al Terminal.**

---

<sup>28</sup> Terminal: herramienta en forma de ventana, que permite controlar todo el sistema operativo de una manera rápida y eficaz. Es un interprete de comandos que traduce la entrada del usuario en instrucciones de máquina.

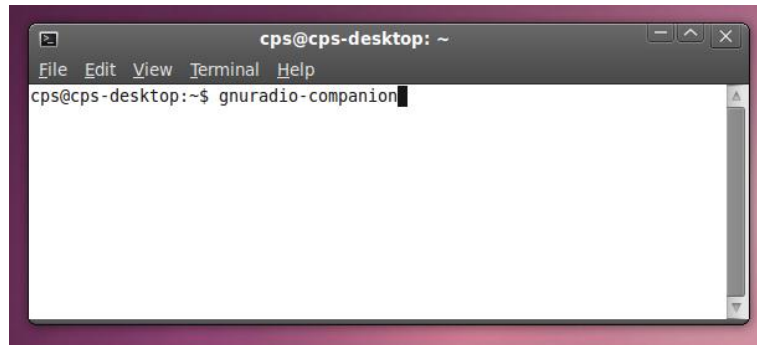


Figura A24. Comando de ejecución GNU Radio en la Terminal.

2. Al iniciar GNU Radio Companion se muestra la ventana principal de la interfaz, como se puede observar en la siguiente figura:



Figura A25. Interfaz GNU Radio Companion.

La ventana principal de la interfaz GNU Radio Companion está dividida en tres partes importantes:

- **Espacio de trabajo (Interfaz Gráfica de Usuario, GUI):** proporciona un entorno visual sencillo que posibilita la interacción del usuario con el sistema.
  - **Terminal o consola:** Muestra toda la información referente al proyecto, *Warnings* o advertencias, sugerencias generadas y los errores cometidos durante el proceso.
  - **Librerías de bloques:** Colección de filtros, moduladores, demoduladores, fuentes de señal y otras herramientas que pueden ser usadas en el procesamiento de señales, además de herramientas como analizador de espectros y osciloscopio.
3. Para crear un nuevo proyecto se selecciona el menú *Archivo (File)* opción *Nuevo (New)*, o con el acceso rápido desde el teclado pulsando las teclas *Control+N*. De esta manera se obtiene una interfaz en blanco con el nombre *Untitled*, el cual puede ser cambiado a conveniencia utilizando la opción *Guardar (Save)* del menú *Archivo* o pulsando desde el teclado las teclas *Control+S*. Para guardar una copia de un proyecto ya existente con otro nombre se utiliza la opción *Guardar como (Save as)*, desde el teclado *Shift+Control+S*.

4. Para abrir o modificar un proyecto ya existente se selecciona del menú Archivo (*File*) la opción Abrir (*Open*) o desde el teclado presionando las teclas *Control+O*.

## Implementación del proyecto

5. En un nuevo proyecto realizar el montaje mostrado en Figura A26,<sup>29</sup> teniendo en cuenta que cada uno de los bloques se encuentra en las librerías predeterminadas de GNU Radio y para utilizarlos tan solo se selecciona el bloque deseado y se arrastra al área de trabajo, o se da doble clic en su respectivo nombre. Otra posibilidad para agregar un bloque es seleccionando su nombre en la librería y presionando el botón *Add*, que se encuentra en la parte inferior derecha de la ventana principal de GNU Radio.
  - En el esquemático se tienen dos generadores de señal, los cuales están configurados para producir una forma de onda tipo cosenoidal, uno de los cuales tiene frecuencia determinada en 350 Hz, el otro tiene frecuencia variable que puede ser modificada en tiempo de ejecución por medio del control deslizante (*slider*)<sup>29</sup> de nombre **slider\_freq** y oscila entre 350 Hz y 500 Hz, con un valor por defecto de 440 Hz.

---

<sup>29</sup> Referirse al documento “Referencia Bloques de Procesamiento de Señales GNU Radio Companion” (Anexo G).

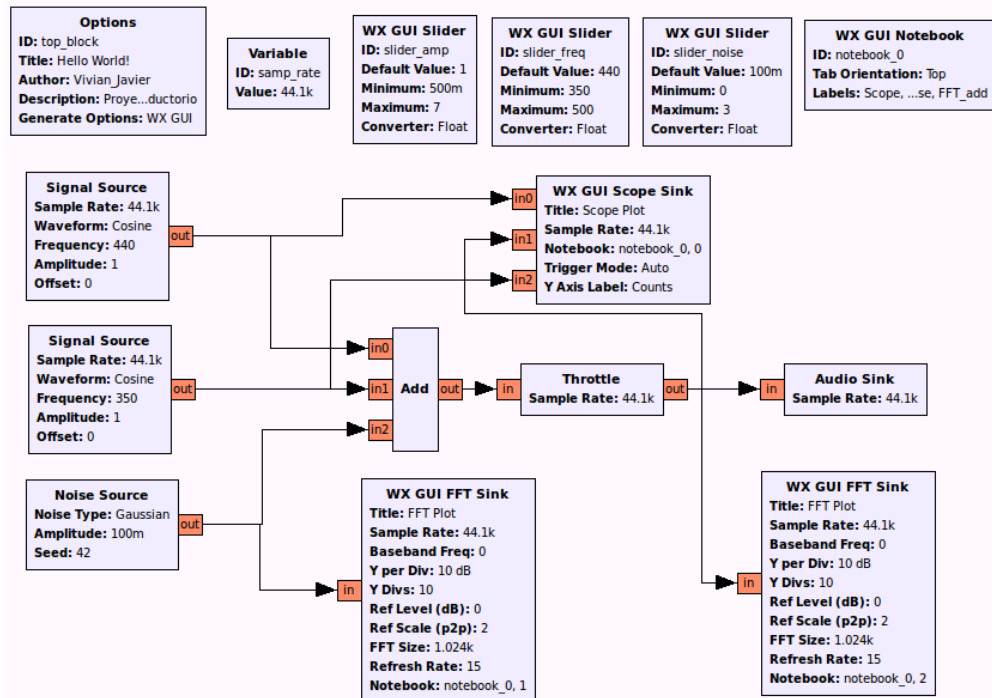
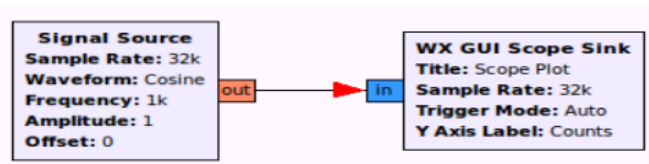


Figura A26. Esquemático Proyecto Introductorio.

- Un generador de ruido, configurado para generar ruido tipo Gaussiano, de amplitud variable, modificable en tiempo de ejecución usando el control deslizante (*slider*) de ID **slider\_noise**, con un valor por defecto de 0.1, que puede oscilar entre 0 y 3.
- Las señales cosenoidales y el ruido Gaussiano son sumados y su resultado se lleva a un sumidero de audio (*audio sink*) que permite escuchar el tono de marcación (*dial tone*).

- Se tiene un osciloscopio<sup>30</sup> con tres entradas, que permite observar la señal de los dos generadores de señal y la señal resultante luego de la sumarlas con la señal de ruido Gaussiano.
- La variable **samp\_rate** determina la frecuencia de muestreo a la cual cada uno de los bloques procesa los datos de entrada a los mismos.
- Para realizar la conexión entre cada uno de los bloques, primero se debe verificar que manejan el mismo tipo de datos de entrada y salida. Cuando se conectan dos puertos con diferente tipo de datos se presenta un error y la punta del cable conector cambia a color rojo, como se observa en Figura A27. Sólo es posible conectar un puerto de entrada con uno de salida. Al intentar conectar por ejemplo dos entradas entre sí, se genera un error en el terminal indicando que la conexión no puede ser realizada. Para conectar los dos puertos se da clic en el pin del primer bloque y luego en el pin del segundo, con lo cual se traza el cable por el que fluirán los datos. Si la conexión es correcta se obtiene una gráfica como la mostrada en Figura A28.



**Figura A27. Error en la conexión entre puertos no compatibles.**

<sup>30</sup> Referirse al documento “Referencia Bloques de Procesamiento de Señales GNU Radio Companion” (Anexo G).

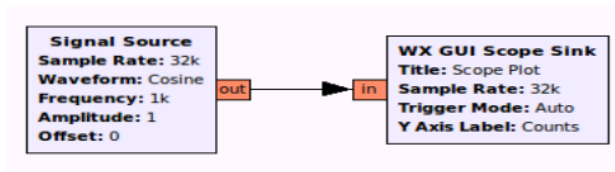
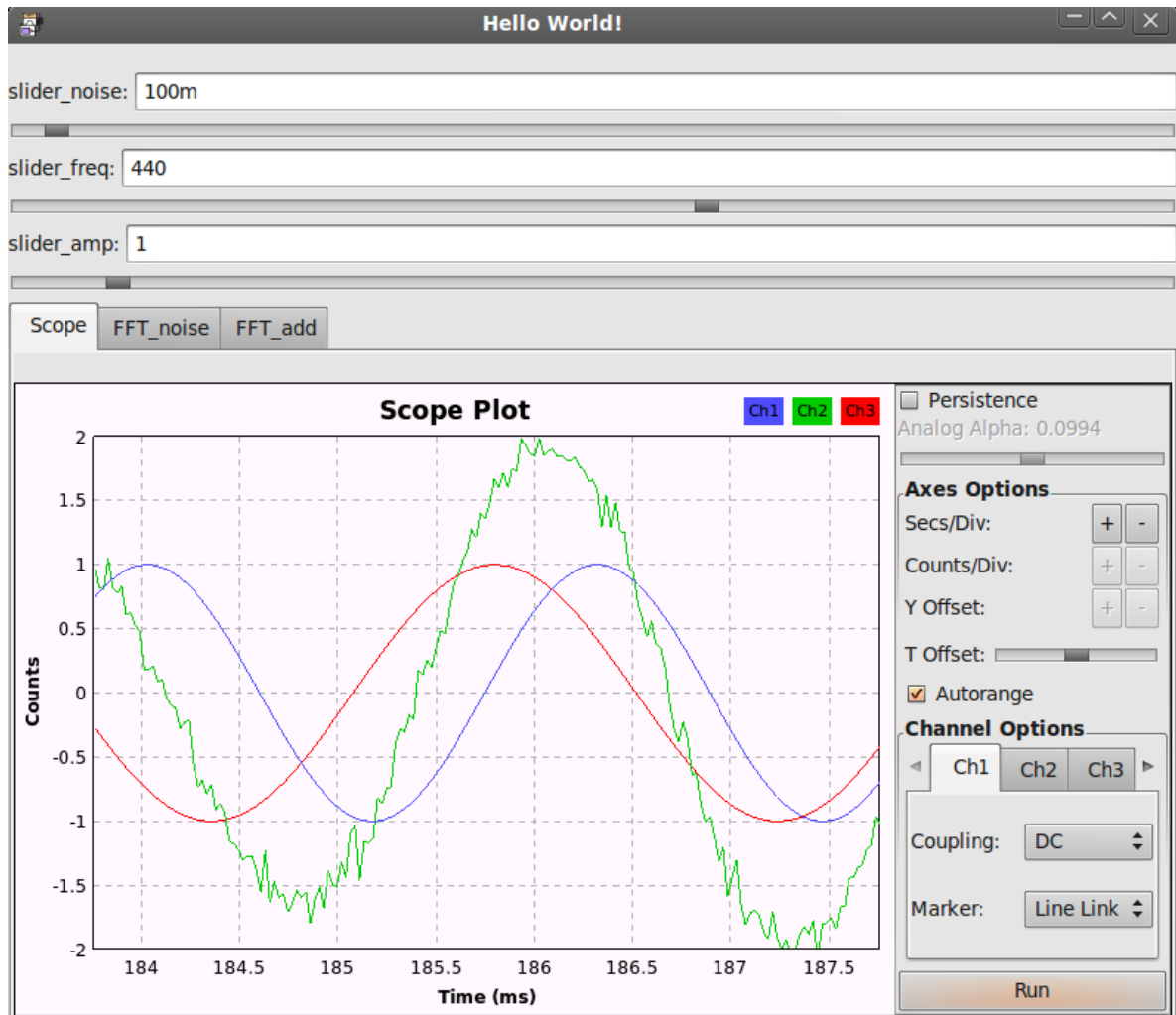


Figura A28. Conexión correcta entre dos puertos.

6. Luego de tener el esquema montado correctamente, con sus respectivos bloques, parámetros y conexiones, se procede Generar el archivo Python y a ejecutar el proyecto, lo cual se puede hacer desde el menú Construir (*Build*), opción Generar (*Generate*), o presionando la tecla F5 y Ejecutar (*Execute*), o con la tecla F6. De esta manera GNU Radio presenta una ventana de ejecución como se muestra en Figura A29, y utilizando auriculares se puede escuchar el tono de marcación.
7. Variar la frecuencia y las amplitudes observando en el osciloscopio los cambios generados en las señales.
8. Implementar otras configuraciones utilizando los bloques de procesamiento de señal encontrados en las librerías básicas, de tal forma que se reconozca la plataforma.



**Figura A29. Ventana de ejecución Proyecto Introductorio.**

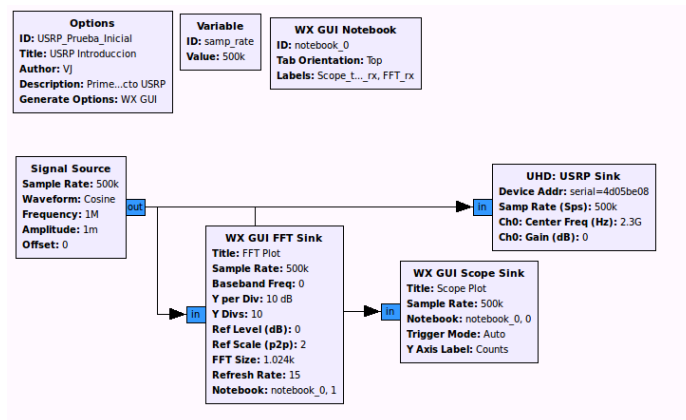
## Reconocimiento Dispositivos USRP

9. En un nuevo proyecto, se implementa un esquema que permita transmitir una señal utilizando el dispositivo USRP<sup>31</sup>, para ello se hace uso del bloque *UHD: USRP Sink*<sup>32</sup>

<sup>31</sup> Referirse al datasheet de USRP1, daughterboard RFX2400.

<sup>32</sup> Referirse al documento "Referencia Bloques de Procesamiento de Señales GNU Radio Companion" (Anexo G).

- a. Verificar que el computador reconoce el dispositivo USRP y obtener el número de serial que se coloca en el parámetro *Device Addr* del bloque.
- 33
- b. Ubicar la frecuencia central en 2.3 GHz. Los demás parámetros no se modifican.



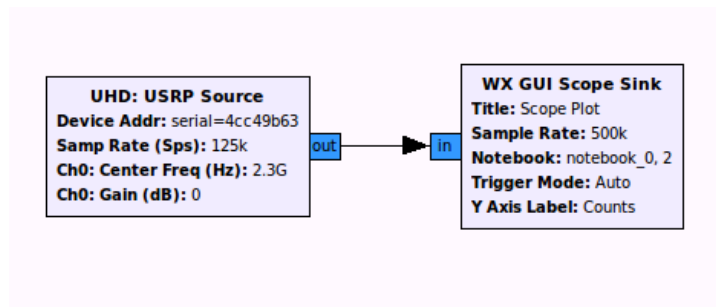
**Figura A30. Uso de USRP. Diagrama de bloques transmisión.**

10. Para la recepción del mensaje utilizar el bloque UHD: USRP Source.

- a. Ubicar la frecuencia central en 2.3 GHz. Realizar un ajuste fino para este valor, por medio de un control deslizante que permita sumarle a esta frecuencia central una variación para sintonizar las dos tarjetas. Esto se debe a que ningún oscilador de cristal es perfecto, por lo tanto, siempre habrá alguna descompensación en la frecuencia. El ajuste puede oscilar entre -10 kHz y 10 kHz.

<sup>33</sup> Referirse al datasheet de USRP1, daughterboard RFX2400.

- b. Verificar que el computador reconoce el dispositivo USRP y obtener el número de serial que se coloca en el parámetro *Device Addr* del bloque.
- c. Utilizar las herramientas de visualización necesarias par observar el comportamiento de la señal a través del proceso.



**Figura A31. Uso USRP. Diagrama de bloques recepción.**

- d. Generar el archivo Python y ejecutar.
- e. Observar que suceden si se varían parámetros como frecuencia de muestreo, frecuencia central del dispositivo.

## REFERENCIAS

[1] GNU Radio [en línea]. [Fecha de Consulta: julio 15 de 2012].

Disponible en:

<http://gnuradio.org/redmine/projects/gnuradio/wiki>

[2] OPPENHEIM, Alan S. WILLSKY, S. Hamid Nawab. Señales y sistemas. México: Pearson Education, Segunda edición, 1998.

## **ANEXO B**

### **PRÁCTICA No. 2**

#### **MODULACIÓN PSK**

##### **INTRODUCCIÓN**

En el campo de las comunicaciones digitales la modulación por desplazamiento de fase (PSK) representa una de las modulaciones básicas y fundamentales ya que representa la base para entender sistemas de modulación de fase multinivel implementados en diversas aplicaciones comerciales y civiles. Una alternativa para este tipo aplicaciones de es la modulación por desplazamiento diferencial de fase (DPSK), donde su gran importancia se fundamenta en que la información no esta contenida en la fase absoluta, como en la modulación PSK convencional sino en las transiciones entre fases, lo cual permite codificar la fase de forma diferencial y en el momento de la desmodulación no se hace necesario recuperar la señal portadora simplificando en cierta forma el diseño del sistema. GNU Radio contiene bloques de procesamiento de señal que permiten implementar un sistema de comunicaciones utilizando este tipo de modulación, proporcionando una alternativa práctica para comprender los conceptos relacionados con esta técnica.



## **OBJETIVOS**

### **OBJETIVO GENERAL**

Utilizar GNU Radio y USRP para implementar un sistema de comunicaciones digitales mediante modulación PSK.

### **OBJETIVOS ESPECÍFICOS**

1. Implementar un diagrama de flujo que permita transmitir y recibir una señal cosenoidal modulada mediante PSK y sus derivaciones.
2. Transmitir y recibir una imagen y archivos tipo wav utilizando modulación PSK en sus diferentes tipos.
3. Observar el comportamiento de las señales en cada etapa del proceso de modulación.
4. Estudiar los conceptos relacionados con modulación PSK en sus diferentes tipos.

### **TRABAJO PREVIO**

Para realizar esta práctica y obtener resultados satisfactorios, es necesario haber realizado la práctica “Introducción a GNU Radio”, de tal forma que se hayan

analizado los conceptos relacionados y comprendido el funcionamiento de la herramienta. Por otra parte, se hace necesario revisar la documentación sobre USRP y estudiar el documento “Referencia bloques de procesamiento de señales GNU Radio Companion” (Anexo G).

Estudiar teoría de comunicaciones digitales; codificación, muestreo, modulación, técnicas de modulación, profundizando en modulación PSK y sus derivados.

### **TIEMPO ESTIMADO**

El tiempo estimado para la realización de esta guía es de dos sesiones de dos horas cada una.

### **PROCEDIMIENTO**

11. En un nuevo esquemático, configurar el bloque de opciones para personalizar el diseño que se va a implementar, con el título, nombre del autor y descripción. Recordar cambiar el ID para que no se sobrescriba el archivo de Python.
  
12. Especificar la tasa de muestreo que se va a utilizar en todo el esquemático, por medio del bloque *Variable* que viene por defecto al crear un nuevo proyecto. Es recomendable utilizar frecuencias de muestreo entre 250 kHz y 1 MHz, para que el desempeño del computador sea óptimo. Tener en

cuenta el teorema de Nyquist para establecer este valor. Iniciar con un valor de 250 kHz.

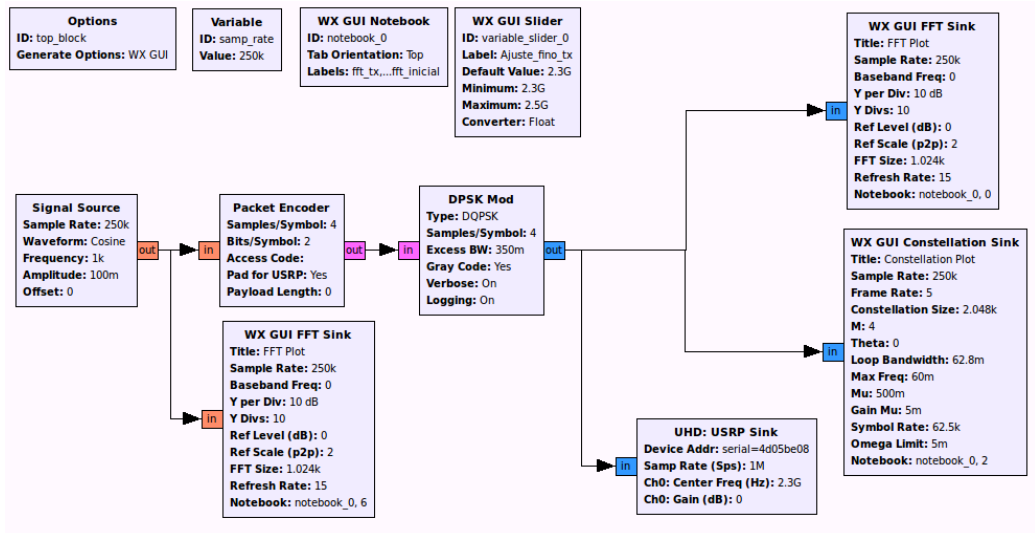


Figura B32. Diagrama de bloques Modulador DPSK.

13. Transmitir una señal coseno con amplitud de 0.1 a una frecuencia de 1 kHz y sin nivel de DC. Crear un control deslizante para variar en tiempo real la frecuencia, la amplitud y el nivel de DC de la señal. Se propone variar la amplitud de 0.01 a 1, la frecuencia de 1 k Hz a 50 kHz y el offset de 0 a 1. Dejar por defecto los valores iniciales.
  
14. Utilizar el bloque *Packet Encoder* para realizar el empaquetamiento de los bits resultantes del muestreo de esta señal. Dependiendo de la modulación a realizar (DBPSK o DQPSK) se modifica el parámetro de bits/symb. Para comenzar se sugiere implementar la modulación DBPSK utilizando 2

muestras/símbolo, valor que debe mantenerse en todos los bloques que requieran de este parámetro. Los demás parámetros no se modifican.<sup>34</sup>

15. Adjunto al flujo normal de los bloques para realizar la modulación y posterior transmisión del mensaje, conectar a este bloque, uno de tipo *Packed to Unpacked* y seguidamente un bloque de cambio de tipo de dato de *Char to Float*, y observar esta salida en el osciloscopio (WX GUI Scope<sup>34</sup>) en modo normal. ¿Cómo es la salida? ¿Qué representa? ¿Qué operación digital esta realizando?
  
16. Configurar el bloque *DPSK Mod* ajustando el número de muestras por símbolo que se está manejando y la modulación que se esta implementando (DBPSK o DQPSK). Los demás parámetros no se modifican.
  
17. Antes de transmitir la señal por medio de los dispositivos USRP, realizar la simulación de la modulación, conectando el bloque *DPSK Mod* directamente al bloque *DPSK Demod*.
  
18. Configurar el bloque *DPSK Demod* ajustando el número de muestras por símbolo y la modulación que se esta implementando. Los demás parámetros no se modifican.<sup>34</sup>

---

<sup>34</sup> Referirse al documento “Referencia Bloques de Procesamiento de Señales GNU Radio Companion” (Anexo G).

19. Utilizar el bloque *Packet Decoder*<sup>35</sup> para desempaquetar de los bits resultantes de la desmodulación.

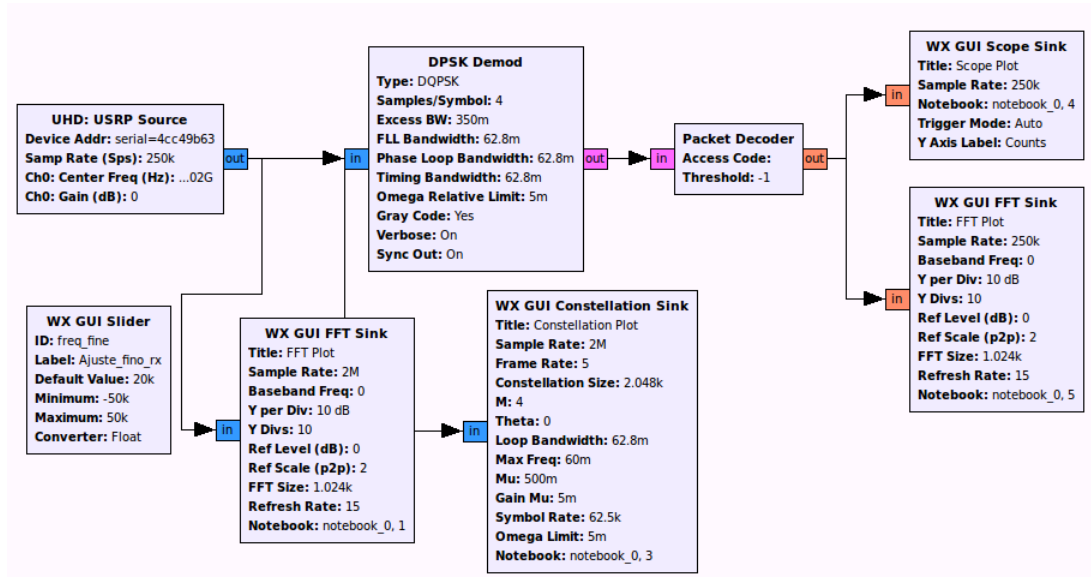


Figura B33. Diagrama de bloques Demodulador DPSK.

20. Para observar la señal recibida luego de la desmodulación y desempaquetado utilizar un sumidero tipo osciloscopio (*WX GUI Scope Sink*) y uno tipo analizador de espectros (*WX GUI FFT Sink*).<sup>35</sup>

21. El comportamiento de la señal en cada una de las etapas puede visualizarse utilizando sumideros tipo osciloscopio (*WX GUI Scope Sink*) o tipo Analizador de espectros (*WX GUI FFT Sink*). GNU Radio Companion incluye un bloque de análisis de señal que permite observar la constelación de una señal modulada (*WX GUI Constellation Sink*<sup>35</sup>). Se recomienda

<sup>35</sup> Referirse al documento "Referencia Bloques de Procesamiento de Señales GNU Radio Companion" (Anexo G).

implementar un bloque *Notebook*<sup>36</sup> para observar estas salidas de forma ordenada en pestañas en una sola ventana gráfica.

22. Rectificar el esquema, tratando de minimizar errores de configuración, tales como coherencia entre tipos de datos, conexiones, parámetros asociados a la modulación, entre otros.

23. Generar el archivo de Python y ejecutar el sistema. Referirse a la Práctica No. 1 (Anexo A).

24. Verificar que se obtienen los resultados esperados, de lo contrario regresar al paso 12.

25. Para transmitir el mensaje utilizando el dispositivo USRP, se procede de la siguiente manera:

26. Después del bloque *DPSK Mod*, implementar el bloque *Multiply Const*<sup>36</sup> para disminuir el nivel de potencia de la señal. Este actúa como un atenuador.

27. Utilizar, luego del atenuador, el bloque *UHD: USRP Sink*.<sup>36</sup>

---

<sup>36</sup> Referirse al documento "Referencia Bloques de Procesamiento de Señales GNU Radio Companion" (Anexo G).

28. Verificar que el computador reconoce el dispositivo USRP y obtener el número de serial que se coloca en el parámetro *Device Addr* del bloque.<sup>37</sup>

29. Ubicar la frecuencia central en 2.3 GHz. Los demás parámetros no se modifican.

30. Para la recepción del mensaje utilizar el bloque *UHD: USRP Source*.<sup>38</sup>

f. Ubicar la frecuencia central en 2.3 GHz. Realizar un ajuste fino para este valor, por medio de un control deslizante que permita sumarle a esta frecuencia central una variación para sintonizar las dos tarjetas. Esto se debe a que ningún oscilador de cristal es perfecto, por lo tanto, siempre habrá alguna descompensación en la frecuencia. El ajuste puede oscilar entre -10 kHz y 10 kHz.

g. Verificar que el computador reconoce el dispositivo USRP y obtener el número de serial que se coloca en el parámetro *Device Addr* del bloque.

h. Después del bloque *UHD: USRP Source*, conectar el bloque *DPSK Demod* y seguir los pasos 8 al 14.

31. Proceder a realizar cambios al esquemático diseñado con los controles deslizantes creados. Cada vez que se realice uno de estos cambios,

---

<sup>37</sup> Referirse al datasheet de USRP1, daughterboard RFX2400.

<sup>38</sup> Referirse al documento "Referencia Bloques de Procesamiento de Señales GNU Radio Companion" (Anexo G).

analizar los resultados y regresar al diseño original propuesto en los anteriores pasos.

- a. Variar la frecuencia del coseno paulatinamente en el rango establecido, sin cambiar ningún otro parámetro, y observar el comportamiento de la señal en el tiempo y en frecuencia en cada uno de las etapas. ¿Se distorsiona? ¿Se obtiene en cada una de las etapas el espectro y forma esperadas?
- b. Con una señal cosenoidal de frecuencia 40 kHz, aumentar la frecuencia de muestreo en el rango especificado en el respectivo control deslizante. ¿Cómo varia la señal a medida que se aumenta o cuando se disminuye esta frecuencia?
- c. Variar la amplitud del coseno en el intervalo especificado y observar los cambios que se presentan en la señal a través del proceso, tanto en tiempo como en frecuencia. ¿Qué se esperaría observar?
- d. Variar el nivel de DC de la señal en el intervalo especificado y observar los cambios que se presentan en la señal a través del proceso, tanto en tiempo como en frecuencia. ¿Los resultados obtenidos son los esperados?
- e. Variar las muestras por símbolo (*Samples/Symbol*), ejecutando el esquema cada vez que modifique su valor. Se propone hacer la variación en 2, 4 y 8 muestras/símbolo. No se puede utilizar control deslizante, ya que los cambios no serán aceptados en tiempo de ejecución. ¿Qué se esperaría encontrar? Observar el comportamiento de la señal. ¿Es acorde con la teoría?

32. Utilizar el analizador vectorial<sup>39</sup>, conectarlo a la salida del USRP transmisor y verificar que el espectro corresponde al observado en el analizador que tiene GNU Radio en la Interfaz gráfica.
33. El bloque Generador de señales (*Signal Source*) permite obtener señales de diferente tipo, además de cosenoidal. Realizar el proceso anterior para cada una de las opciones de señal restantes, a saber, diente de sierra, sinusoidal, cuadrada y triangular.<sup>40</sup>
34. GNU Radio permite generar diferentes tipos de mensaje, entre los cuales se encuentra archivos wav e imágenes.
- a. Para enviar una imagen se reemplaza el bloque *Signal Source* por el bloque *File Source*, en el cual se especifica la dirección en la que se encuentra guardada la imagen. Al final del esquema de comunicación agregar un bloque *File Sink*, que permite guardar la imagen recibida.<sup>40</sup>
    - Enviar la imagen sin repetición, de tal forma que cuando en la FFT gráfica de GNU Radio no se observe espectro alguno en la recepción, se pueda cerciorar que la imagen ha sido recibida en su totalidad. Tener en cuenta que se debe esperar un tiempo considerable.

---

<sup>39</sup> Guía Rápida R&S@ZVL Analizador de Redes Vectoriales. Disponible en: [http://www2.rohde-schwarz.com/file\\_11716/ZVL\\_QuickStart\\_ES.pdf](http://www2.rohde-schwarz.com/file_11716/ZVL_QuickStart_ES.pdf) [Consultado en julio 3 de 2012].

<sup>40</sup> Referirse al documento “Referencia Bloques de Procesamiento de Señales GNU Radio Companion” (Anexo G).

- Verificar la coherencia en el esquema y ejecutar la configuración. Observar el comportamiento del mensaje en las diferentes etapas del proceso y verificar la correcta recepción abriendo el archivo correspondiente.
- b. Para enviar un archivo tipo wav, reemplazar el bloque *Signal Source* por el bloque *Wav File Source*, en el cual se especifica la dirección en la que se encuentra guardado el archivo. Al final del esquema de comunicación agregar un bloque *Wav File Sink* que permite guardar el archivo.<sup>41</sup>
- Verificar la coherencia en el esquema y ejecutar la configuración. Observar el comportamiento del mensaje en las diferentes etapas del proceso y verificar la correcta recepción abriendo el archivo correspondiente.
35. Realizar el procedimiento anterior, tomando los pasos que se consideren necesarios para un adecuado estudio del sistema de comunicaciones utilizando modulación DQPSK. Analizar y documentar los resultados obtenidos.
36. Opcional. Se puede cambiar el bloque *DPSK Mod* y *DPSK Demod* por los respectivos *PSK Mod* y *PSK Demod*. Configurar los parámetros dependiendo de la modulación que se efectúe (BPSK, QPSK, 8PSK).

---

<sup>41</sup> Referirse al documento “Referencia Bloques de Procesamiento de Señales GNU Radio Companion” (Anexo G).



Realizar lo procedimiento anterior, tomando los pasos que se consideren adecuados para un estudio adecuado de este tipo de modulación.

## REFERENCIAS

- [1] COUCH, Leon W. Sistemas de comunicación digitales y analógicos, México: Pearson Educación, 2008.  
ISBN: 970-26-1216-2
  
- [2] BLAKE, Roy. Sistemas electrónicos de comunicaciones, Mexico: Thomson Learning, 2004.  
ISBN: 970-686-365-6
  
- [3] ARTÉS RODRIGUEZ, Antonio. Comunicaciones digitales, Madrid: Pearson Educación, Prentice-Hall, 2007.  
ISBN: 84-8322-348-2
  
- [4] HERRERA PÉREZ, Enrique. Comunicaciones II: Comunicación digital y ruido. México: Limusa, 1999.  
ISBN: 968-18-5719-4
  
- [5] HAYKIN, Simon S. Digital communications, New York: John Wiley, 1988.  
ISBN: 0-471-63775-0



## **ANEXO C**

### **PRÁCTICA No. 3**

#### **MODULACIÓN GMSK**

##### **INTRODUCCIÓN**

GMSK es el acrónimo de *Gaussian Minimum Shift Keying*, una técnica de modulación continua en fase, que permite reducir los requerimientos de ancho de banda con respecto a otras técnicas de modulación. GNU Radio contiene bloques de procesamiento de señal que permiten implementar un sistema de comunicaciones utilizando este tipo de modulación, proporcionando una alternativa práctica para comprender los conceptos relacionados con esta técnica.

##### **OBJETIVOS**

###### **OBJETIVO GENERAL**

Utilizar GNU Radio y USRP para estudiar e implementar un sistema de comunicaciones digitales mediante modulación GMSK.

## **OBJETIVOS ESPECÍFICOS**

5. Implementar un diagrama de flujo que permita transmitir y recibir una señal cosenoidal modulada mediante GMSK.
6. Transmitir y recibir una imagen y archivos tipo wav utilizando modulación GMSK.
7. Observar el comportamiento de las señales en cada etapa del proceso de modulación.
8. Estudiar los conceptos relacionados con modulación GMSK.

## **TRABAJO PREVIO**

Para realizar esta práctica y obtener resultados satisfactorios, es necesario haber realizado la práctica “Introducción a GNU Radio”, de tal forma que se hayan analizado los conceptos relacionados y comprendido el funcionamiento de la herramienta. Por otra parte, se hace necesario revisar la documentación sobre USRP y estudiar el documento “Referencia bloques de procesamiento de señales GNU Radio Companion” (Anexo G).

Estudiar teoría de comunicaciones digitales; codificación, muestreo, modulación, técnicas de modulación, profundizando en modulación GMSK.

## TIEMPO ESTIMADO

El tiempo estimado para la realización de esta guía es de una sesión de dos horas.

## PROCEDIMIENTO

1. En un nuevo esquemático, configurar el bloque de opciones para personalizar el diseño que se va a implementar, con el título, nombre del autor y descripción. Recordar cambiar el ID para no sobrescribir el archivo de Python.
2. Especificar la tasa de muestreo que se va a utilizar en todo el esquemático, por medio del bloque *Variable* que viene por defecto al crear un nuevo proyecto. Es recomendable utilizar frecuencias de muestreo entre 250 kHz y 1 MHz, para que el desempeño del computador sea óptimo. Tener en cuenta el teorema de Nyquist para establecer este valor. Iniciar con un valor de 250 kHz.
3. Transmitir una señal coseno de amplitud de 100 m, frecuencia de 1 kHz y nivel de DC cero. Crear un control deslizante para variar en tiempo real la frecuencia, uno para la amplitud y otro para el nivel de DC (*Offset*). Se propone variar la amplitud de 10 m a 1, la frecuencia de 1 kHz a 100 kHz y el nivel de DC de 0 a. Dejar por defecto los valores iniciales.

4. Utilizar el bloque *Packet Encoder* para realizar el empaquetamiento de los bits resultantes del muestreo de esta señal. Para esta modulación el parámetro bits/symb debe ser 1. Los demás parámetros no se modifican.<sup>42</sup> Para observar la operación que realiza este bloque se utiliza el bloque *Packed to Unpacked*, seguido de un bloque de cambio de tipo de dato *Chart to Float* y observar esta salida en un osciloscopio. ¿Cómo es la salida? ¿Qué representa? ¿Qué operación digital esta realizando?
  
5. Configurar el bloque *GMSK Mod* ajustando el número de muestras por símbolo que se está manejando. Se propone utilizar 2 muestras/símbolo, valor que debe mantenerse en todos los bloques que requieran de este parámetro. Los demás parámetros no se modifican.<sup>42</sup>
  
6. Antes de transmitir la señal por medio de los dispositivos USRP, realizar la simulación de la modulación, conectando el bloque *GMSK Mod* directamente al bloque *GMSK Demod*.
  
7. Configurar el bloque *GMSK Demod* ajustando el número de muestras por símbolo que se está manejando. Los demás parámetros no se modifican.<sup>42</sup>
  
8. Utilizar el bloque *Packet Decoder* para desempaquetar los bits resultantes de la desmodulación.

---

<sup>42</sup> Referirse al documento "Referencia Bloques de Procesamiento de Señales GNU Radio Companion" (Anexo G).

9. Para observar la señal recibida luego de la desmodulación y desempaquetado utilizar un sumidero tipo osciloscopio (*WX GUI Scope Sink*) y uno tipo analizador de espectros (*WX GUI FFT Sink*)<sup>43</sup>
  
10. El comportamiento de la señal en cada una de las etapas puede visualizarse utilizando sumideros tipo osciloscopio (*WX GUI Scope Sink*) o tipo Analizador de espectros (*WX GUI FFT Sink*). GNU Radio Companion incluye un bloque de análisis de señal que permite observar la constelación de la señal modulada (*WX GUI Constellation Sink*)<sup>43</sup>. Se recomienda implementar un bloque *Notebook* para observar estas salidas de forma ordenada por medio de pestañas en una sola ventana gráfica.
  
11. Rectificar el esquema, tratando de minimizar errores de configuración, tales como coherencia entre tipos de datos, conexiones, parámetros asociados a la modulación, entre otros.
  
12. Generar el archivo de Python y ejecutar el sistema. Referirse a la Práctica No. 1 (Anexo A).
  
13. Verificar que se obtienen los resultados esperados, de lo contrario regresar al paso 11.
  
14. Para transmitir el mensaje utilizando el dispositivo USRP, se procede de la

---

<sup>43</sup> Referirse al documento “Referencia Bloques de Procesamiento de Señales GNU Radio Companion” (Anexo G).

siguiente manera:

- a. Utilizar, luego del bloque de modulación, el bloque *UHD: USRP Sink*.<sup>44</sup>
- b. Verificar que el computador reconoce el dispositivo USRP Transmisor y obtener el número de serial, el cual se coloca en el parámetro *Device Addr* del bloque.<sup>45</sup>
- c. Ubicar la frecuencia central en 2.3 GHz. Los demás parámetros no se modifican.

15. Para la recepción del mensaje utilizar el bloque *UHD: USRP Source*.<sup>44</sup>

- a. Ubicar la frecuencia central en 2.3 GHz. Realizar un ajuste fino para este valor, por medio de un control deslizante que permita sumarle una variación de frecuencia para sintonizar las dos tarjetas. Esto se debe a que ningún oscilador de cristal es perfecto, por lo tanto, siempre habrá alguna descompensación en la frecuencia. El ajuste puede oscilar entre -10 kHz y 10 kHz.
- b. Verificar que el computador reconoce el dispositivo USRP Receptor y obtener el número de serial que se coloca en el parámetro *Device Addr* del bloque.<sup>44</sup>

---

<sup>44</sup> Referirse al documento “Referencia Bloques de Procesamiento de Señales GNU Radio Companion” (Anexo G).

<sup>45</sup> Referirse al datasheet de USRP1, daughterboard RFX2400.

- c. Después del bloque *UHD: USRP Source*, conectar el bloque *GMSK Demod* y seguir los pasos 8 al 13.

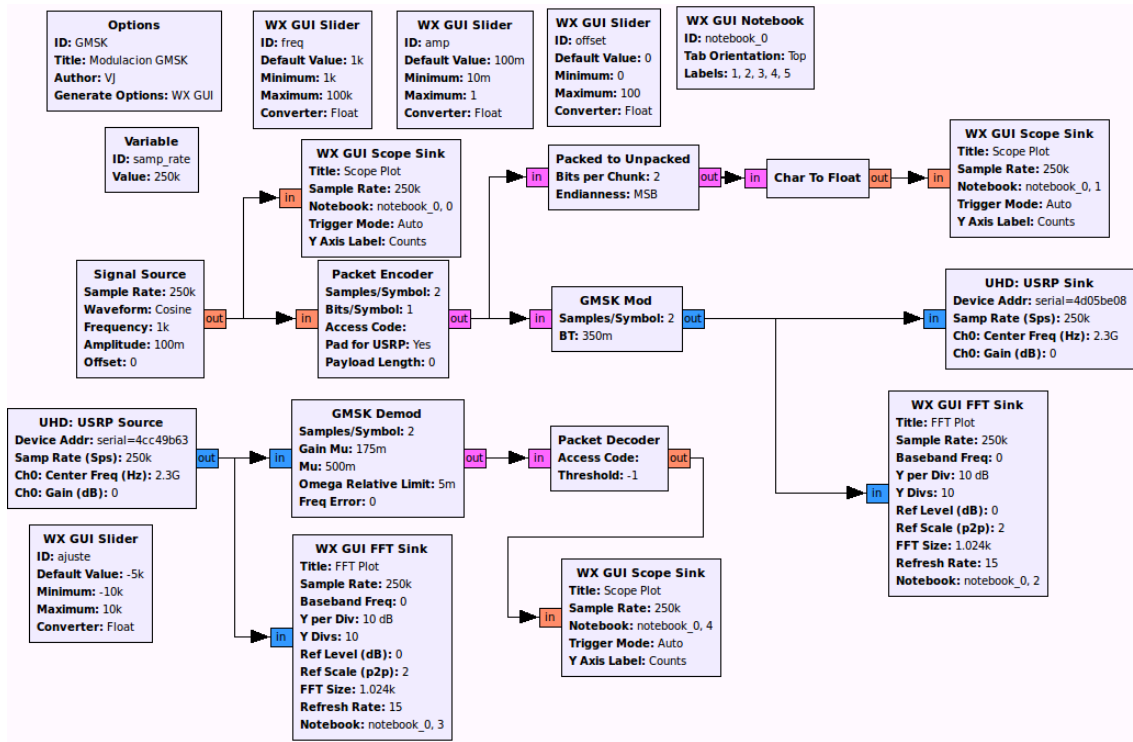


Figura C34. Diagrama de bloques Modulador - Demodulador GMSK.

16. Proceder a realizar cambios al esquemático diseñado con los controles deslizantes creados. Cada vez que se realice uno de estos cambios, analizar los resultados y regresar al diseño original propuesto en los anteriores pasos.

- a. Variar la frecuencia del coseno paulatinamente en el rango establecido, sin cambiar ningún otro parámetro, y observar el comportamiento de la señal en el tiempo y en frecuencia en cada

uno de las etapas. ¿Se distorsiona? ¿Se obtiene en cada una de las etapas el espectro y forma esperadas?

- b. Con una señal cosenoidal de frecuencia 40 kHz, aumentar la frecuencia de muestreo en el rango especificado en el respectivo control deslizante. ¿Cómo varia la señal a medida que se aumenta o cuando se disminuye esta frecuencia?
  
- c. Variar la amplitud del coseno en el intervalo especificado y observar los cambios que se presentan en la señal a través del proceso, tanto en tiempo como en frecuencia. ¿Qué se esperaría observar?
  
- d. Variar el nivel de DC de la señal en el intervalo especificado y observar los cambios que se presentan en la señal a través del proceso, tanto en tiempo como en frecuencia. ¿Los resultados obtenidos son los esperados?
  
- e. Variar las muestras por símbolo (*Samples/Symbol*), ejecutando el esquema cada vez que modifique su valor. Se propone hacer la variación en 2, 4 y 8 muestras/símbolo. No se puede utilizar control deslizante, ya que los cambios no serán aceptados en tiempo de ejecución. ¿Qué se esperaría encontrar? Observar el comportamiento de la señal. ¿Es acorde con la teoría?

17. Utilizar el analizador vectorial<sup>46</sup>, conectarlo a la salida del USRP transmisor y verificar que el espectro corresponde al observado en el analizador que tiene GNU Radio en la Interfaz gráfica.
18. El bloque Generador de señales (*Signal Source*) permite obtener señales de diferente tipo, además de cosenoidal. Realizar el proceso anterior para cada una de las opciones de señal restantes, a saber, diente de sierra, sinusoidal, cuadrada y triangular.<sup>47</sup>
19. GNU Radio permite generar diferentes tipos de mensaje, entre los cuales se encuentra archivos wav e imágenes.
- a. Para enviar una imagen se reemplaza el bloque *Signal Source* por el bloque *File Source*, en el cual se especifica la dirección en la que se encuentra guardada la imagen. Al final del esquema de comunicación agregar un bloque *File Sink*, que permite guardar la imagen recibida.<sup>47</sup>
    - Enviar la imagen sin repetición, de tal forma que cuando en la FFT gráfica de GNU Radio no se observe espectro alguno en la recepción, se pueda cerciorar que la imagen ha sido recibida en su totalidad. Tener en cuenta que se debe esperar un tiempo considerable.

<sup>46</sup> Guía Rápida R&S@ZVL Analizador de Redes Vectoriales. Disponible en: [http://www2.rohde-schwarz.com/file\\_11716/ZVL\\_QuickStart\\_ES.pdf](http://www2.rohde-schwarz.com/file_11716/ZVL_QuickStart_ES.pdf) [Consultado en julio 3 de 2012].

<sup>47</sup> Referirse al documento “Referencia Bloques de Procesamiento de Señales GNU Radio Companion” (Anexo G).

- Verificar la coherencia en el esquema y ejecutar la configuración. Observar el comportamiento del mensaje en las diferentes etapas del proceso y verificar la correcta recepción abriendo el archivo correspondiente.
- b. Para enviar un archivo tipo wav, reemplazar el bloque *Signal Source* por el bloque *Wav File Source*, en el cual se especifica la dirección en la que se encuentra guardado el archivo. Al final del esquema de comunicación agregar un bloque *Wav File Sink* que permite guardar el archivo.<sup>48</sup>
- Verificar la coherencia en el esquema y ejecutar la configuración. Observar el comportamiento del mensaje en las diferentes etapas del proceso y verificar la correcta recepción abriendo el archivo correspondiente.

## REFERENCIAS

- [1] COUCH, Leon W. Sistemas de comunicación digitales y analógicos, México: Pearson Educación, 2008.  
ISBN: 970-26-1216-2
- [2] BLAKE, Roy. Sistemas electrónicos de comunicaciones, Mexico: Thomson Learning, 2004.  
ISBN: 970-686-365-6

---

<sup>48</sup> Referirse al documento “Referencia Bloques de Procesamiento de Señales GNU Radio Companion” (Anexo G).



- [3] ARTÉS RODRIGUEZ, Antonio. Comunicaciones digitales, Madrid: Pearson Educación, Prentice-Hall, 2007.  
ISBN: 84-8322-348-2
- [4] HERRERA PÉREZ, Enrique. Comunicaciones II: Comunicación digital y ruido. México: Limusa, 1999.  
ISBN: 968-18-5719-4
- [5] HAYKIN, Simon S. Digital communications, New York: John Wiley, 1988.  
ISBN: 0-471-63775-0



## **ANEXO D**

### **PRÁCTICA No. 4**

#### **CREACIÓN DE BLOQUES GRC**

##### **INTRODUCCIÓN**

GNU Radio permite la creación y modificación de bloques de procesamiento de señales. Para la creación existen varias alternativas de realización, de las cuales se presenta una sencilla a partir de una plantilla básica, donde se encuentran los archivos organizados en sus respectivas carpetas, algunos de los cuales son modificados para los propósitos deseados.

##### **OBJETIVOS**

##### **OBJETIVO GENERAL**

Realizar la introducción a la creación de bloques GRC para procesamiento de señales en GNU Radio utilizando una plantilla.

## **OBJETIVOS ESPECÍFICOS**

1. Ejecutar los pasos para la creación de un bloque GRC utilizando la plantilla suministrada.
2. Crear un nuevo bloque GRC a partir de la plantilla.
3. Conocer una de las formas que existen para crear bloques en GNU Radio.

## **TRABAJO PREVIO**

Para la realización de esta práctica se requiere conocimiento previo en programación C y C++.

## **TIEMPO ESTIMADO**

El tiempo estimado para la realización de esta guía es de dos sesiones de dos horas cada una.

## **PROCEDIMIENTO**

1. Organizar una carpeta nueva con el nombre del bloque que se desea crear, la cual contenga las subcarpetas y archivos correspondientes que se exponen en Tabla No.1. Para ello, utilizar la plantilla suministrada.

**Tabla No. 1.** Carpeta de archivos instalación bloque GRC.

Subcarpeta	Archivo	Descripción	Modificable
<b>apps</b>	Makefile.am	Aplicaciones de prueba.	No
<b>config</b>	Diferentes archivos.	Configuración.	No
<b>grc</b>	Gr_my_sum_ff.xml	Descripción del bloque en la interfaz gráfica.	Si
	Makefile.am	Makefile. <sup>49</sup>	Si
<b>lib</b>	Gr_my_sum_ff.cc	Descripción de la función del bloque.	Si
	Gr_my_sum_ff.h	Descripción de la función del bloque.	Si
	Makefile.am	Makefile. <sup>49</sup>	Si
<b>python</b>	_init_.py	Secuencia de comandos de python.	No
	Makefile.am	Makefile. <sup>49</sup>	No
	Run_tests		No
	Tun_tests.in		No
<b>swig</b>	Gr_my.i		Si
	Gr_my_sum_ff.i		Si
	Makefile.am	Makefile. <sup>49</sup>	Si
	Makefile.swig.gen		No
<b>Archivos de configuración</b>			No

<sup>49</sup> Archivos Makefile: *Makefile* en conjunto con el comando *make*, permite, entre otras cosas, establecer una serie de acciones que se deben seguir para crear una versión ejecutable de un programa o sistema que consiste de varios archivos separados. A través de éste se le indica al sistema qué archivos dependen de otros, y se especifica cuáles.

2. Luego de tener la carpeta con los archivos correspondientes, se procede a ejecutar la instalación. Para ello, en una terminal se ejecutan los siguientes comandos:

**sudo ./bootstrap** <sup>50</sup>(Instala un conjunto de utilidades de construcción requeridas para construir la mayoría de módulos)

**sudo ./configure** (Configura las fuentes para ser compiladas, busca los programas necesarios (y advierte si no están) para crear el *Makefile*)

**cd swig** (Cambia al directorio *Swig*)

**sudo make generate-makefile-swig** (Regenera *Makefile.swig.gen* en la carpeta *Swig*).

**cd ..** (Sube un nivel de directorios)

**sudo make** (Herramienta que controla la creación de ejecutables y otros archivos de un programa a partir de los archivos fuente)

**sudo make install** (Se encarga de copiar los binarios obtenidos a donde deban estar)

**sudo ldconfig** (Solamente la primera vez que el modulo es instalado. Actualiza los enlaces necesarios para los enlaces en tiempo de ejecución)

---

<sup>50</sup> **sudo** (abreviatura del inglés *superuser do* o *substitute user do*) es una utilidad de los sistemas operativos tipo Unix, como GNU/Linux, BSD, o Mac OS X, que permite a los usuarios ejecutar programas con los privilegios de seguridad de otro usuario (normalmente el usuario root) de manera segura.

3. Se reinicia el sistema y se ejecuta GNU Radio Companion. En la sección Librerías de bloques de GNU Radio, aparece el nombre de la nueva categoría y en ella el bloque creado.
  
4. Realizar un montaje que permita comprobar el correcto funcionamiento del bloque.
  
5. Si se desea, haciendo uso de la plantilla, modificar los archivos necesarios para crear un bloque que realice una nueva función. Para personas con conocimientos avanzados en programación C++ y Python, se propone realizar un bloque que permita modular una señal mediante la técnica ASK.

## REFERENCIAS

- [1] DEITEL, Harvey M., DEITEL, Paul J. Cómo programar en C/C++ y Java. México: Pearson Educación, cuarta edición, 2004.

ISBN: 970-26-0531-8

- [2] Discuss-gnuradio - GNU Radio, a free software defined radio [en línea] [Fecha de consulta: Julio 20 de 2012]  
Disponible en:

<https://lists.gnu.org/mailman/listinfo/discuss-gnuradio>

## **ANEXO E**

### **SISTEMAS DE COMUNICACIONES BASADOS EN LA TÉCNICA RADIOS DEFINIDOS POR SOFTWARE.**

A continuación se presenta una breve descripción de diferentes trabajos realizados a nivel mundial, en los cuales se evidencia el uso de la técnica Radios Definidos por Software, SDR.

#### **Referencia 1.**

MITOLA, J. "Software Radios: Survey, Critical Evaluation and Future Directions", IEEE AES Systems Magazine. Abril, 1993.

#### **Descripción.**

El autor fue la persona quien acuñó el término *Software Radio* en 1992, para esta técnica en los sistemas de comunicación.<sup>51</sup>

En el artículo se explica el concepto de *Software Radio*, incluyendo las ventajas de la técnica en cuanto a flexibilidad, costo y eficiencia, además, la posibilidad de tener un sistema de comunicaciones en el cual se pueda reconfigurar

---

<sup>51</sup> Disponible en: <http://transition.fcc.gov/pshs/techttopics/techttopics4.html> [Fecha de consulta: julio 3 de 2012]

instantáneamente el formato apropiado de señal, presentándola como una excelente opción para uso en comunicaciones militares.

Presenta un estudio de la viabilidad de las aplicaciones de *Software Radio* teniendo en cuenta las características y capacidad de los convertidores A/D de la época, resaltando que el desempeño del sistema se puede mejorar con algoritmos avanzados y mejorando las características de los DSP. Muestra, además, un estudio de la tendencia en tecnología de comunicaciones, concluyendo que la industria de las comunicaciones esta en una transición de hardware a software, siendo la arquitectura de software libre la que proporciona la mayoría de la ventaja para esta tendencia.

## **Referencia 2.**

OZA, Jignesh. PATEL, Yogesh. TRIVEDI, Pratik. RANPURA, Nilesh. PATEL, Zuber. DALAL, Upena. JANI, Rachna. S.R., Vijay. “Optimized configurable architecture of modulation techniques for SDR applications”, IEEE. 2010.

## **Descripción.**

Presenta el diseño completo, tanto transmisor como receptor, de un sistema de comunicaciones utilizando la técnica SDR y haciendo uso de una arquitectura reconfigurable de técnicas de modulación FSK/QPSK, combinación de técnicas de modulación configurable seleccionada debido a la compensación que se obtiene en cuanto eficiencia en ancho de banda de QPSK y desempeño BER de FSK, que permiten optimizar la arquitectura SDR, en cuanto a ancho de banda analógico en el extremo RF, incrementado el desempeño del sistema.

**Referencia 3.**

SHATON III, John L. “A Software Defined Radio transformation”. Thales Communications, Inc., 22605 Gateway Center Drive. Clarksburg, MD, USA. Mayo 17, 2009.

**Descripción.**

En el artículo se presenta las adaptaciones de hardware realizadas, que permitieron utilizar un Radio de mano VHF/UHF para que funcionara como radio central en la banda HF mediante simples mejoras de software.

**Referencia 4.**

CHEN, Ke-Yu. CHEN, Zhi-Feng. “GNU Radio”. Gainesville Florida. University of Florida. Dept. of Electrical Computer Engineering.

**Descripción.**

En el artículo se presenta una introducción al concepto de Radio Definido por Software y su implementación con GNU Radio y USRP. Utilizando conexión TCP/IP, se simula un canal inalámbrico para transmitir una imagen utilizando modulación DBPSK y DQPSK. Se plantea el ejemplo como una introducción a los sistemas de comunicaciones utilizando SDR.

### **Referencia 5.**

YEH, Hen-Geul. INGERSON, Paul. “Software Defined Radio for OFDM transceivers”. IEEE. 2010.

### **Descripción.**

Se presenta un sistema de comunicaciones de Radio Definido por Software utilizando transceptor OFDM (*Orthogonal Frequency Division Multiplexing*), además de una discusión sobre la adaptabilidad e interoperabilidad en modo de operación BPSK y QPSK del sistema OFDM. A partir de la implementación de las dos técnicas de modulación, se realiza un análisis comparativo y se presentan los resultados de desempeño del sistema para un DSP específico.

### **Referencia 6.**

GU, Boncheol. JUNG, Jinman. KIM, Kyongdong. HEO, Junyoung. PARK, Namhoon. JEON, Gwangil. CHO, Yookun. “SWICOM: An SDR-Based wireless communication gateway for vehicles”. IEEE Transactions on vehicular technology. Vol. 59, No.4. Mayo, 2010.

### **Descripción.**

Los autores presentan una puerta de comunicaciones inalámbricas basada en SDR para uso en comunicaciones vehiculares inalámbricas, como alternativa para remover la dependencia de hardware de la mayoría de las tecnologías utilizadas para tal fin. Se muestra una visión general de las arquitecturas de comunicaciones vehiculares y algunos de los problemas que se encuentran. Se

explica el prototipo desarrollado y los análisis de viabilidad, eficiencia, así como resultados experimentales de su aplicación, llegando a la conclusión de que es eficiente y viable para su aplicación práctica en control de dispositivos inalámbricos implementados por software.

### **Referencia 7.**

An Overview of the U.S. and Japanese Approaches to Cognitive Radio and SDR.  
James Miller. IEICI TRANS. COMMUN., VOL. E89-B, No 12 December 2006

### **Descripción.**

Este documento tiene como propósito principal examinar el proceso regulatorio que han seguido las tecnologías de radio definido por software (SDR) y radio cognitivo en Estados Unidos y Japón los cuales han tomado enfoques diferentes en ambos países.

El principal enfoque de regulación de SDR es preservar la integridad de los dispositivos para prevenir interferencia manteniendo los principios y características sobre el cual operan los dispositivos, mientras que para la radio cognitiva su principal enfoque es mejorar el proceso de gestión del espectro, asignación, servicio y concesión de licencias.

En Estados Unidos la FCC (Federal Communications Commission) tiene a su cargo la regulación del concepto SDR. Esta comisión tomó grandes iniciativas para aprobar la tecnología SDR, en el 2001 se conoció la primera definición legal de SDR aprobada por la FCC adoptando a su vez nuevos procedimientos de

certificación para la aprobación de dispositivos SDR, facilitando cada vez nuevos beneficios a esta tecnología.

La definición fue:

“SDR es un radio que incluye un transmisor en el cual los parámetros operacionales: rango de frecuencia, tipo de modulación o máxima potencia de salida (ya sea radiada o conducida) o las circunstancias bajo las cuales el transmisor opera en conformidad con las normas de la comisión, pueden ser alterados haciendo cambios en el software sin hacer ningún cambio en los componentes de hardware que afecten las emisiones de frecuencia de radio”

La FCC ha demostrado su compromiso de garantizar un equilibrio entre la flexibilidad para desarrollar e implementar dispositivos SDR, mientras evitan que los radios sean mal utilizados, previniendo grandes riesgos de interferencia antes que estos dispositivos sean vendidos, comercializados o importados.

En cuanto a los radios cognitivos la FCC consideró la capacidad de estos radios para alcanzar mayor eficiencia, flexibilidad buscando la forma de regular esta tecnología para lograr un uso exhaustivo del espectro mientras se reduce el riesgo de interferencias perjudiciales.

En cuanto a Japón, las comunidades de investigación han demostrado gran interés en SDR y las tecnologías relacionadas por más de diez años. Sin embargo, a excepción de un software para controlar la selección de la frecuencia para una clase de LAN inalámbrica en la banda de 5 GHz, SDR no está permitido en Japón. La postura para solo permitir el uso de SDR en este caso es debido a la falta que tiene de una autoridad como la FCC en ese país, que regule y sancione los malos o indebidos usos de dispositivos SDR.

MIC (Ministerio de Comunicaciones) ha invertido en proyectos de radio cognitivo

dirigidos a comunicaciones móviles y se espera que esta tecnología sea de importante investigación para aplicaciones comerciales en los próximos años.

### **Referencia 8.**

Experimental Study of OFDM Implementation Utilizing GNU Radio and USRP-SDR. Arief Marwanto, Mohd Adib Sarijari, Norsheila Fisal, Sharifah Kamilah Syed Yusof, Rozeha A.Rashid. Telematic Research Group, Fakulti Kejuruteraan Elektrik. Universiti Teknologi Malaysia, December 2009.

### **Descripción.**

Este documento presenta el uso del software libre GNU RADIO y el dispositivo USRP (Universal Software Radio peripheral) bajo el concepto de Software defined radio transmitir señales de radio por OFDM (*Orthogonal Frequency Division Multiplexing*) con modulación QPSK y BPSK. Además es analizada la calidad del servicio en función de la razón de paquetes recibidos (PRR).

Para realizar los experimentos se utilizaron dos periféricos USRP, un computador personal de escritorio y un computador portátil, las tarjetas hijas utilizados eran de referencia RFX2400 las cuales cubren un rango de frecuencias de 2.3GHz a 2.9GHz. La distancia entre cada USRP fue de 660 mm. La frecuencia central del USRP fue sintonizada a 2.5 GHz. El objetivo de estos experimentos fue determinar la relación entre la razón de paquetes recibidos (PRR) con la potencia transmitida utilizada por la multiplexación por división de frecuencias ortogonales usando USRP. El PRR es medido para dos diferentes modulaciones (QPSK y BPSK) variando a su vez la longitud de la FFT.

Los resultados muestran que la OFDM con modulación BPSK tiene una mejor eficiencia de comunicación comparada con la OFDM con modulación QPSK en términos de PRR. Igualmente el rendimiento de PRR en con respecto a la longitud de la FFT cumple los resultados esperados en la teoría, entre mas grande la longitud, menor el error y por lo tanto se obtendrá un mejor PRR.

Es así como los parámetros de un sistema de radiocomunicación como modulación, potencia y longitud de la FFT deben ser cuidadosamente escogidos para alcanzar un óptimo rendimiento de transmisión de la información.

### **Referencia 9.**

Implementation of a Low Cost Synthetic Aperture Radar using Software Defined Radio. Deepthi Maheswari Chinnam, Madhusudhan J., Nandhini C., Prathyusha S.N., Sowmiya Sw.,Ramanathan R\*, Dr. Soman K.P. Amrita Vishwa Vidyapeetham, Coimbatore, India. 2010 Second International conference on Computing, Communication and Networking Technologies

### **Descripción.**

En este articulo se presenta una aplicación particular de un sistema de radio definido por software para implementar un sistema de radar básico y a su vez un SAR (Synthetic Aperture Radar) utilizando el dispositivo USRP (Universal Software Radio Pheripheral) y la herramienta de software libre GNU Radio, que permiten implementar SDRs sofisticados y de bajo costo.

Un radar es un sistema ampliamente usado que usa ondas electromagnéticas para la detección de objetos, para el levantamiento de planos de terreno y previsiones meteorológicas.

SAR es un tipo de radar de imágenes que tiene un amplio rango de aplicaciones para obtener modelos de elevación digital de la superficie de la tierra y en sensores remotos. La principal ventaja de SAR es la alta resolución a pesar de la pequeña antena, la cual es obtenida usando la propiedad de desplazamiento de Doppler.

Los bloques principales de un sistema SAR consisten en la sección de RF y el procesador de señal.

El propósito del proyecto no es diseñar un nuevo sensor radar sino realizar una investigación de la utilidad del hardware y software existente para aplicaciones de radar definidas por software.

Bajo el concepto de SDR la mayoría de los bloques físicos de un radar, como el modulador, el generador de forma de onda, demodulador y el procesador de señal, pueden ser implementados en su equivalente digital. El desarrollo de este radio definido por software se lleva a cabo en GNU Radio representando el hardware del radar como un flujo gráfico de bloques.

El procedimiento que se lleva a cabo es el siguiente: La señal de radar es recibida por la antena la cual es recibida por la sección RF (implementada en el USRP) la información digital es enviada al computador donde es procesada para obtener la imagen del radar, este proceso se lleva a cabo en GNU Radio donde se realiza la integración de los elementos que integran la parte física del SAR.

### **Referencia 10.**

Design of Software-Defined Radio Channel Simulator for Wireless

Communications: Case Study With DSRC and UWB Channels. Jeich Mar, *Senior Member, IEEE*, Chi-Cheng Kuo, You-Rong Lin, and Ti-Han Lung. August 2009.

### **Descripción.**

En este artículo los autores presentan el diseño un simulador de canal de radio definido por software para probar la transmisión y recepción en banda base de varios sistemas de comunicaciones inalámbricas.

Las características de reconfiguración de un SDR son aplicadas para cambiar las condiciones del canal y reconfigurar el hardware de los módulos de procesamiento en el simulador de canal, siendo este ultimo capaz de simular hasta 6 canales multitrayecto en varias especificaciones o estándares de interfaz de aire de acuerdo a las exigencias de los usuarios.

Las especificaciones de la interfaz del aire de los sistemas de comunicaciones inalámbricas DSRC (*Dedicated Short-Range communications*) y UWB (*Ultrawideband*) fueron escogidos como ejemplo para describir la estructura de reconfiguración y los algoritmos del simulador de canal SDR

Los resultados obtenidos con este simulador son comparados con experimentos empíricos realizados a estos sistemas de comunicaciones inalámbricas, teniendo en cuenta las especificaciones de interfaz de aire, para validar su exactitud.

El simulador de canal SDR propuesto puede ser extendido a otras especificaciones de interfaz de aire de sistemas de comunicaciones inalámbricas sin necesidad de realizar ningún cambio en el hardware.

## **Referencia 11.**

The practical experience of implementing a GSM BTS through Open Software/ Hardware. \_E. Natalizio, V. Loscrí, G. Aloí. DEIS. University of Calabria, Italy. N. Paolí, N. Barbaro, Orangee srl. Via Pedro Alvarez Cabrai Rende – Italy

## **Descripción.**

Este documento tiene como propósito describir la experiencia que tuvieron los autores al implementar una estación base de telefonía de GSM BTS (*Base Transceiver System*) en software bajo el concepto de SDR apoyándose en Asterisk para servicio VoIP.

Asterisk representa un programa de software libre que proporciona funcionalidades de una central telefónica (PBX) el cual se conecta a un proveedor de VoIP.

Para permitir a GSM hacer llamadas y recibir mensajes de texto, fue necesario realizar algunas modificaciones de hardware en la tarjeta madre para configurar un servidor de protocolo inicial.

Se utilizó un computador, una tarjeta madre multipropósito para SDR en el cual se instalaron dos tarjetas hijas RX/TX (transceptoras, trabajando a frecuencias de GSM900) en un proyecto de software basado en GNU Radio, donde en este último se modeló la estación base y su comunicación con Asterisk para realizar llamadas.

### **Referencia 12.**

Next Generation Communication towards Open Wireless Standard and Software Defined Radio. Mr Y.B. Thakare. Mr S.S Musale. Mr P.G Shete.

### **Descripción.**

En este documento los autores presentan una arquitectura denominada ODWCS (*Open Distributed Wireless Communication System*) orientada para sistemas de acceso inalámbricos con antenas distribuidas, procesadores distribuidos y control distribuido a través de radio definido por software; el cual permite obtener tasas de transmisión en sistemas de próxima generación 4G de 20 a 100 Mbps,

El principal objetivo del documento es contribuir a la importancia de una plataforma de comunicación abierta con la habilidad de poder ser reconfigurado bajo el concepto de SDR.

### **Referencia 13.**

GNU Radio 802.15.4 En and Decoding. Thomas Schmid. NESL Department of Electrical Engineering. University of California, Los Angeles

### **Descripción.**

Los autores muestran la implementación de un bloque de decodificación en- and para el estándar inalámbrico IEEE 802.15.4 el cual es ampliamente utilizado en redes de sensado, automatización del hogar y otros sistemas conectados en red.



## **UNIVERSIDAD INDUSTRIAL DE SANTANDER**

ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES  
**Perfecta Combinación entre Energía e Intelecto**

El estándar implica una cuantización y codificación de los datos de entrada y una modulación O –QPSK, funciones que se realizan en GNU Radio, software libre basado en el concepto SDR. Así mismo se utiliza el periférico USRP para la transmisión de la señal RF.

## **ANEXO F**

### **INSTALACIÓN GNU RADIO**

Para instalar y compilar GNU Radio utilizamos el código fuente, la versión original escrita de este software libre, que se encuentra en la página web oficial:

[www.gnuradio.org](http://www.gnuradio.org)

Esta es la forma más adecuada ya que el código fuente genera paquetes compilados listos para instalar. Aunque puede tomar un poco más de tiempo para instalar y compilar que su contraparte por paquete binario, la instalación por código fuente representa la manera más versátil, poderosa y permanente de cualquier software, y su mayor ventaja es que permite que esté disponible la versión más reciente del programa.

Para realizar la instalación y compilación primero hay que descargar el archivo fuente y luego compilarlo antes de instalarlo.

En general, para poder utilizar GNU Radio se deben seguir los siguientes pasos:

1. Instalar los prerrequisitos.
2. Descargar el código fuente.
3. Configurar.
4. Compilar.
5. Instalar.

- 1) **Instalar los prerequisites:** Se debe instalar ciertos paquetes que son necesarios para compilar varias partes de GnuRadio en Ubuntu. Estos son:

Herramientas de desarrollo necesarias para compilación:

- g++
- git
- make
- autoconf, automake, libtool
- sdcc (from "universe")
- guile
- ccache

Librerías (necesarias para el tiempo de ejecución y compilación):

- python-dev
- SWIG
- FFTW 3.X (libfftw3-dev)
- cppunit (libcppunit-dev)
- Boost 1.35
- GSL GNU Scientific Library (libgsl0-dev)
- libusb and libusb-dev
- ALSA (alsa-base, libasound2 y libasound2-dev)

GNU Radio Companion

- Para GNU Radio Companion (GRC) es necesario instalar python-numpy, python cheetah y python-lxml.

## WX GUI

- Para los componentes de interfaz gráfica de usuario, WX GUI, se necesita instalar python-wxgtk2.8 y python-numpy.

## QT GUI

- Para los componente QT GUI se necesita instalar PyQT4, PyQwt5 for Qt4, QT-OpenGL, Fontconfig, Xrender y Xinput (python-qt4, python-qwt5-qt4, libqt4-opengl-dev, libqwt5-qt4-dev, libfontconfig1-dev, libxrender-dev, libxi-dev).

## Video-SDL

- Para Video-SDL se necesita instalar el Simple DirectMedia Layer development libraries (libsdl1.2-dev).

## Polyphase Filter Bank examples

- Para trabajar los ejemplos en gnradio-examples/python/pfb se necesita instalar python-scipy, python-matplotlib, y python-tk.

## Otros paquetes útiles

- doxygen (Para crear documentación del código fuente).
- octave.

Para cada versión de Ubuntu se utilizan comandos diferentes para instalar los requisitos necesarios de GNU Radio. Para el caso de la versión 11.04 se usan las siguientes líneas de código que deben ser copiadas en el terminal:

```
sudo apt-get -y install git-core autoconf automake libtool g++ python-  
dev swig \  
pkg-config libboost-all-dev libfftw3-dev libcppunit-dev libgsl0-dev \  
libusb-dev sdcc libsdl1.2-dev python-wxgtk2.8 python-numpy \  
python-cheetah python-lxml doxygen python-qt4 python-qwt5-qt4 libxi-dev \  
libqt4-opengl-dev libqwt5-qt4-dev libfontconfig1-dev libxrender-dev
```

Estos comandos realizan las siguientes funciones:

**Sudo:** Esta orden da el permiso de ejecución como administrador, es decir permite que el programa modifique el sistema de archivos protegidos.

**apt-get install nombre\_paquete:** Localizado el nombre del paquete que se quiere descargar e instalar (en este caso el *git core*), este comando se encargará del proceso. Buscará en el índice de dónde tiene que descargarse el paquete, lo hace, y posteriormente lo instala.

**Git-core:** Es un administrador de contenido de directorios diseñado para manejar de pequeños a grandes proyectos con eficiencia y rapidez.

**Autoconf:** Es una herramienta informática para crear *Shell Scripts* que configuren automáticamente el código fuente de un software para poder compilarlo. Dichos *scripts* creados por *Autoconf* son independientes de él cuando se ejecutan en el sistema en el que se quiere usar.

**Shell script:** Es el intérprete de comandos del sistema. Es una interfaz de texto de altas prestaciones, que sirve fundamentalmente para tres cosas: administrar el sistema operativo, lanzar aplicaciones (e interactuar con ellas) y como entorno de programación.

Los *scripts* son un conjunto de instrucciones generalmente almacenadas en un archivo de texto que deben ser interpretados línea a línea en tiempo real para su ejecución, se distinguen de los programas, pues deben ser convertidos a un archivo binario ejecutable para correrlos.

**Automake:** Es una herramienta de programación que produce programas *makefiles* portables, para el uso de la operación *make* en el proceso de compilación de software. Estos *makefile* contienen reglas de dependencia para determinar las acciones a seguir, es decir, comúnmente establecen que programas deben compilarse para obtener la aplicación, y será necesario por cada uno de los directorios en que se necesite realizar alguna tarea.

**Libtool:** Es una herramienta de programación usada para crear bibliotecas de software portables. *GNU Libtool* simplifica el trabajo mediante el encapsulado de las dependencias específicas de la plataforma, y la interfaz del usuario en un único script.

A partir de estos comandos y herramientas de programación se configuran las opciones para poder instalar todas las dependencias anteriormente especificadas.

Si el usuario usa otra versión de Ubuntu debe dirigirse a la página:

<http://gnuradio.org/redmine/projects/gnuradio/wiki/ubuntuinstall#Install-Dependencies>

En ella se encontrarán los comandos requeridos para instalar los prerequisites para la instalación de GNU Radio, en las versiones 10.10 y 10.04 de Ubuntu.<sup>52</sup>

- 2) **Descargar el código fuente:** Para realizar la instalación de GNU radio es necesario obtener el código fuente escribiendo en la Terminal:

```
git clone git://gnuradio.org/gnuradio.git
```

Esto crea además un directorio con el nombre GNU Radio.

Ahora se debe entrar al directorio recién creado:

```
cd gnuradio
```

Este será el directorio de inicio de GNU Radio.

- 3) **Configuración:** En este punto, luego de entrar al directorio de inicio de GNU Radio, es momento de configurarlo. Se ejecuta el script con estos comandos:

```
./bootstrap
```

```
./configure
```

Cuando se ejecuta el script de configuración, en realidad no se compila nada todavía. Este comando simplemente comprueba el sistema y asigna valores a las variables dependientes del sistema. Estos valores se utilizan para generar un *Makefile*. El *Makefile* a su vez se utiliza para generar el paquete binario real.

---

<sup>52</sup> Esta página fue consultada en mayo 24 de 2012.

**4) Compilación:** En esta parte se crea realmente el paquete binario, el programa ejecutable, del código fuente. Esto se realiza mediante la ejecución del comando *make* en la terminal:

```
Make
```

Este paso toma un poco de tiempo, dependiendo de qué tan grande es el programa y que tan rápido es el computador en donde se está compilando.

**5) Instalación:** Después de crear exitosamente el programa ejecutable, es aconsejable ejecutar el software de auto chequeo para GNU Radio

```
make check
```

Ahora finalmente se instala el programa, GNU Radio, colocando el siguiente comando en la terminal:

```
sudo make install
```

Se debe tener privilegio de administrador para realizar este paso (sudo).

Ahora GNU Radio esta listo para ser utilizado.

Para ejecutar el programa y disfrutar de sus beneficios se escribe en la terminal:

```
gnuradio-companion
```

En algunas ocasiones aparece un mensaje de error, referente a *python*, al ejecutar este comando. Esto se debe a que las librerías y los archivos de *python* entre otros, quedaron ubicados en un sitio diferente al adecuado. Para esto es necesario configurar el archivo *bashrc*, que es el lugar donde se hace esta manipulación.

El archivo *.bashrc* se encuentra dentro de la carpeta principal del usuario. Este archivo tiene la configuración de la consola, terminal o *bash*, que es donde se puede interactuar por medio de texto con la computadora, en una interfaz no gráfica. Es un archivo de comandos desarrollado al momento de ejecutar un nuevo *shell (bash)*. Permite personalizar el ambiente de cada *shell*. Aquí se definen, entre otros, alias, *prompt* y cualquier otra definición.

Para abrir este archivo en el editor de texto preferido, en este caso, en el escritor de texto de propósito general *gedit*, así:

```
gedit .bashrc
```

Con este comando se abre el archivo *bashrc* y se añade al final del archivo:

```
# GNU Radio installation
export PATH=$PATH:/usr/local/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
export PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig
export PYTHONPATH=$PYTHONPATH:/usr/local/lib/python2.7/site-packages
```

Ahora se reinicia el sistema operativo, se abre la terminal y se procede a escribir:

```
gnuradio-companion
```

Y se disfruta de todas las ventajas de este software.

## INSTALACIÓN UHD

UHD: Universal Software Radio Peripheral Hardware Driver

Este es el complemento que necesita GNU Radio para poder trabajar con el USRP (Universal Software Radio Peripheral).

La forma más fácil para compilar e instalar UHD+ GNU Radio de la fuente es usar el *build-gnuradio script*, el cual se encuentra en la siguiente página web:

<http://www.sbrac.org/files/build-gnuradio>

Se crea un archivo en el editor de texto preferido con extensión *sh* (que es la extensión que corresponde a un *script*) y se copia todo el texto contenido en la pagina, es decir, si se desea llamar al archivo como *nombre\_archivo*, se debe guardar de la forma *nombre\_archivo.sh*.

Realizado este proceso se escribe en el terminal el siguiente comando:

```
Chmod +x nombre_archivo.sh
```

Esto asigna permisos de ejecución al fichero. Seguidamente se ejecuta como sigue:

```
./nombre_archivo.sh
```

Y se espera a la instalación.

De esta forma queda configurado GNU Radio para trabajar con este dispositivo USRP.

## ANEXO G

### REFERENCIA BLOQUES DE PROCESAMIENTO DE SEÑALES GNU RADIO COMPANION

#### Tipos de datos

Los bloques de señal en GNU Radio se comunican con otros a través de flujos de datos. Un flujo está hecho de elementos individuales, donde todos los elementos tienen un tipo particular de dato. GNU Radio tiene una estricta verificación de tipo de datos, es decir: tipos de datos de entrada y salida deben coincidir exactamente o un error se producirá cuando se ejecute el programa.

Los tipos de datos pueden ser bytes, enteros cortos (*shorts*), enteros (*ints*), flotantes (*floats*) y complejos (*complex*). Además, un tipo de dato puede ser un vector de tipo byte, entero corto, entero, flotante o complejo. En la mayoría de casos, un flujo habitual de datos es solo un flujo de vectores de longitud uno.<sup>53</sup>

- Byte - 1 byte de datos (8 bits por elemento)
- Short - 2 byte entero
- Int - 4 byte entero
- Float - 4 byte punto flotante

---

<sup>53</sup> Introductory Tour of the GNU Radio Project. Josh Blum. Ettus Research [en línea]. [Fecha de Consulta: mayo 29 de 2012]

Disponible en:

<http://www.joshknows.com/gnuradio> .

- Complex - 8 bytes

## 1. Generadores de señal

### 1.1 File Source

Esta herramienta permite leer un archivo desde alguna ubicación en el sistema. Se encuentra en la librería *Source*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- File: permite seleccionar el archivo que se desea leer, seleccionando su ubicación.
- Output Type: tipo de datos de salida del bloque. *Complex, Float, Int, Short, Byte*.
- Repeat: Permite seleccionar si se desea repetir o no la lectura del archivo.
- Vec Length: Longitud del vector que procesa el bloque. Depende de la longitud de los datos que llegan a este bloque. A menos que el flujo de datos sea un arreglo de complejos, float, int o short, la longitud del vector debería ser uno.

## 1.2 Signal Source

Este bloque representa una fuente de señal, permite enviar una señal especificando sus características. Se encuentra en la librería *Source*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Output type: Determina el tipo de dato con el cual se va a enviar esta señal. Puede ser de tipo: Complejo (*complex*), Flotante (*Float*), Entero corto (*Short*) o Entero (*Int*).
- Sample Rate: número de muestras por segundo que se toman de la señal continua para producir una señal discreta (cuantizada en tiempo).
- Waveform: Esta opción permite seleccionar la forma de onda que se desea enviar. Entre sus opciones se encuentra señal constante, senoidal, cosenoidal, cuadrada, triangular y diente de sierra.
- Frequency: Velocidad de ocurrencia de la señal periódica a enviar.<sup>54</sup>
- Amplitude: Valor máximo que adquiere la señal periódica.<sup>54</sup>
- Offset: Nivel de DC de la señal.

---

<sup>54</sup> Diccionario de la Lengua Española - Vigésima segunda edición [en línea]. [Fecha de Consulta: julio 15 de 2012].

Disponible en:

<http://lema.rae.es/drae/>

### 1.3 Noise Source

Este bloque permite enviar una fuente de señal de ruido. Se puede seleccionar entre varios modelos de ruido. Se encuentra en la librería *Source*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Output type: Determina el tipo de dato con el cual se va a enviar esta fuente de ruido. Puede ser de tipo: Complejo (*complex*), Flotante (*Float*), Entero corto (*Short*) o Entero (*Int*).
- Noise type: Modelo de ruido que se va a introducir en el sistema de comunicación. Puede ser Gaussiano, ruido uniforme, Laplaciano o impulsivo.
- Amplitude: variación máxima de la señal de ruido.
- Seed: valor numérico al azar que se utiliza como entrada para inicializar el generador de secuencias pseudoaleatorias. Dos valores de “seed” idénticos producirán la misma secuencia aleatoria, así que debe cambiarse este valor cada vez que se desee que la secuencia pseudoaleatoria sea diferente.<sup>55</sup>

### 1.4 Random Source

Crea una secuencia de números aleatorios, especificando dos parámetros importantes: un intervalo en el cual se van a tomar estos números de la forma

---

<sup>55</sup> Disponible en: [http://www.borisfx.com/download\\_files/Spanish/RGBPixelNoise.pdf](http://www.borisfx.com/download_files/Spanish/RGBPixelNoise.pdf) [Fecha de Consulta: julio 16 de 2012]

[minimum, maximum), y el número de muestras que se toman de este intervalo. Se encuentra en la librería *Source*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Output type: Determina el tipo de dato con el cual se va a enviar esta fuente de señal aleatoria. Entero corto (*Short*), Entero (*Int*) o byte.
- Minimum: Especifica el valor mínimo que puede tomar la secuencia aleatoria.
- Maximum: Especifica el valor máximo que puede tomar la secuencia aleatoria. Este valor no es incluido en el intervalo, solo sirve como una referencia del extremo superior del intervalo. Por ejemplo, si se trabaja con enteros, el valor máximo que podría tomar en algún momento la secuencia sería *máximo-1*.
- Num samples: Determina cuantas muestras de números se toman dentro del intervalo especificado anteriormente.
- Repeat: Permite dar la opción si se desea que la secuencia aleatoria se repita indefinidamente (hasta que la ejecución del esquemático se detenga) o se envíe una sola vez.

### **1.5 WAV File Source**

Lee los datos a partir de archivos tipo wav con salida tipo *float*. Se encuentra en la librería *WX GUI Widgets*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- File: selecciona la ubicación donde se encuentra el archivo wav.
- Repeat: establece si el archivo se reproduce solo una vez o varias veces.
- N Channels: permite establecer varios canales de salida.

## 2. Adecuación de señal

### 2.1 Packet Encoder

Esta herramienta es un codificador, el cual transforma las muestras de tipo *float* en bytes. Se utiliza con bloques de modulación GMSK, DPSK y QAM. Se encuentra en la librería *Synchronizers*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Input Type: especifica el tipo de datos de entrada. *Complex, Float, Int, Short, Byte*.

- Samples/Symbol: especifica el número de muestras que se transmiten por cada tiempo de símbolo.
- Bits/Symbol: especifica la cantidad de bits con la cual se representa un símbolo. Este parámetro debe ser especificado como sigue, dependiendo del tipo de modulación.
  1. GMSK→ 1
  2. DBPSK→1
  3. DQPSK→2
  4. D8PSK→3
  5. QAM8→3
  6. QAM16→4
  7. QAM64→6
  8. QAM256→8
- Access Code: es una contraseña de acceso de seguridad de la información. Se deja en blanco cuando no se desea.
- Pad for USRP: Plataforma para USRP. Yes/No.
- Payload Length: dejar en cero para automático.

## 2.2 Packet Decoder

Herramienta que permite deshacer la codificación para que la información original pueda ser recuperada. Se encuentra en la librería *Synchronizers*. Este bloque se utiliza con bloques de desmodulación GMSK, QAM y PSK.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Output Type: especifica el tipo de datos de salida. *Complex, Float, Int, Short, Byte*.
- Threshold: número de bits que pueden estar erróneos en el vector de sincronización.
- Access Code: cadena de 1s y 0s. Dejar en blanco para automático.

### 2.3 Multiply Const

Este bloque representa un multiplicador, un amplificador en software. Sirve para aumentar la potencia de la señal transmitida. Se encuentra en la librería *Operators*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- IO Type: Especifica el tipo de datos que procesa el bloque. Puede ser de tipo: Complejo (*Complex*), Flotante (*Float*), Entero corto (*Short*) o Entero (*Int*).
- Constant: Valor por el cual se va a amplificar la señal.
- Vec Length: Longitud del vector que procesa el bloque. Depende de la longitud de los datos que llegan a este bloque. A menos que el flujo de

datos sea un arreglo de complejos, flotantes, enteros o enteros cortos, la longitud del vector debería ser uno.

## 2.4 Rational Resampler

Bloque que representa un filtro que permite convertir la frecuencia de muestreo de la señal a otra frecuencia de muestreo que difiere de la original por un factor racional. Se encuentra en la librería *Filters*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Type: Determina el tipo de dato que procesa el bloque y el tipo de coeficientes del filtro (*taps*). Complejo a Complejo (*Complex Taps*), Complejo a Complejo (*Real Taps*), Float a Float (*Real Taps*).
- Decimation: Diezmado. Factor por el cual se desea comprimir (disminuir) la frecuencia de muestreo.
- Interpolation: Interpolación. Factor por el cual se desea expandir (aumentar) la frecuencia de muestreo.
- Taps: Número de coeficientes del filtro. Es una indicación de 1) la cantidad de memoria requerida para implementar el filtro, 2) el número de cálculos requeridos, 3) la cantidad de filtrado que el filtro puede hacer. Mayor número de coeficientes (*taps*) implican una mejor banda de atenuación,

menos rizado, filtros más estrechos. Lo adecuado es dejar vacía esta casilla para un ajuste automático por parte de la herramienta.<sup>56</sup>

- Fractional BW: Es una medida de la relación entre el ancho de banda absoluto  $B$ , con respecto a su frecuencia central  $f_c$ ,  $B/f_c$ . Este parámetro varía entre 0 y 2 y es comúnmente definido en porcentaje (entre 0 y 200 %). Entre más grande es el porcentaje, más amplio es el ancho de banda. Lo adecuado es dejar en cero este valor para un ajuste automático por parte de la herramienta<sup>57</sup>.

## 2.5 Packet to Unpacked

Este bloque realiza un desempaquetado de los bytes que recibe. Se encuentra en la librería *Misc Conversions*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquemático.
- Type: Especifica el tipo de datos que procesa el bloque. Entero (*int*), entero corto (*short*) y byte.
- Bits per chunk: bits por símbolo.
- Endianness: Especifican si se empiezan a escribir los bits resultantes del desempaquetado en la posición MSB (*Most Significant Bit*) o LSB (*Less Significant Bit*) del byte.

---

<sup>56</sup> Disponible en: <http://www.dspguru.com/dsp/faqs/fir/basics> [Fecha de Consulta: julio 16 de 2012]

<sup>57</sup> Disponible en: <http://www.antenna-theory.com/definitions/fractionalBW.php> [Fecha de Consulta: julio 16 de 2012]

## 2.6 Throttle

Este bloque solamente permite que una cierta cantidad de bits atraviesen el bloque (esto no es exactamente una tasa, pero la tasa promedio de bits que dejan el bloque estará dada por la tasa de muestreo del mismo). Si se omite este bloque, se corre el riesgo de que la CPU procese el diagrama de flujo a toda velocidad y consuma toda la capacidad de procesamiento del computador.<sup>58</sup> Se encuentra en la librería *Misc*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Type: Especifica el tipo de datos que procesa el bloque. *Complex, Float, Int, Short, Byte*.
- Sample Rate: Tasa a la cual el bloque procesa los datos.
- Vec Length: Longitud del vector que procesa el bloque. Depende de la longitud de los datos que llegan a este bloque. A menos que el flujo de datos sea un arreglo de complejos, float, int o short, la longitud del vector debería ser uno.

---

<sup>58</sup> Disponible en: <http://gnuradio.org/redmine/projects/gnuradio/wiki/Simulations> [Fecha de Consulta: Julio 16 de 2012].

### 3. Moduladores

#### 3.1. GMSK Mod

Herramienta de procesamiento de señales que ejecuta la modulación digital GMSK. Se encuentra en la librería *Digital Modulators*. La entrada al bloque son símbolos del alfabeto M-ary, en el que M generalmente es 2. La salida es la señal modulada en fase continua.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Samples/Symbol: especifica el número de muestras que se transmiten por cada tiempo de símbolo. Se debe manejar el mismo número de muestras por símbolo en los bloques que involucren este parámetro en todo el proceso de modulación.
- BT: parámetro asociado con el ancho de banda a 3 dB, tiene un valor por defecto de 0.35.
- Verbose: On/Off
- Logging: On/Off

#### 3.2. GMSK Demod

Bloque que permite la desmodulación GMSK. Se encuentra en la librería *Digital Modulators*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Samples/Symbol: especifica el número de muestras que se transmiten por cada tiempo de símbolo. Se debe manejar el mismo número de muestras por símbolo en los bloques que involucren este parámetro en todo el proceso de modulación.
- Gain Mu: parámetro de ganancia del error en el algoritmo de sincronización Mueller & Müller para ajustar Mu.
- Mu: parámetro inicial del interpolador.
- Omega Relative Limit: Valor máximo de omega (valor inicial para el número de muestra por símbolos).
- Freq Error: error permitido en la sincronización en frecuencia. Es cero por defecto.
- Verbose: On/Off
- Logging: On/Off

### 3.3. DPSK Mod

Este bloque realiza una modulación digital DPSK (*Differential Phase shift keying*, modulación por desplazamiento diferencial de fase). Se encuentra en la librería digital *Modulators*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Type: Especifica el tipo de modulación digital que se va a realizar. Entre las opciones posibles se encuentra: DBPSK, DQPSK, D8PSK.
- Samples/symbol: especifica el número de muestras que se transmiten por cada tiempo de símbolo. Se debe manejar el mismo número de muestras por símbolo en los bloques que involucren este parámetro en todo el proceso de modulación.
- Excess BW: Exceso de ancho de banda o también definido como factor de roll off del filtro raíz de coseno alzado, incorporado en este bloque de modulación para reducir la interferencia intersímbolo (ISI). Este factor permite medir la suavidad de la transición del filtro.
- Gray Code: Permite seleccionar si se envía la información binaria con codificación Gray.
- Verbose: On/ Off
- Logging: On/Off

### 3.4. DPSK Demod

Este bloque realiza una demodulación digital DPSK. Se encuentra en la librería *Digital Modulators*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Type: Especifica el tipo de desmodulación digital que se va a realizar. Entre las opciones posibles se encuentra: DBPSK, DQPSK.
- Samples/symbol: Especifica el número de muestras que se reciben por cada tiempo de símbolo.
- Excess BW: Exceso de ancho de banda o también definido como factor de roll off del filtro raíz de coseno alzado, incorporado en este bloque de modulación para reducir la interferencia intersímbolo (ISI). Este factor permite medir la suavidad de la transición del filtro. Debe coincidir con el mismo factor de roll off que se transmite.
- FLL Bandwidth: *Frequency lock loop* (Frecuencia de cierre de bucle). Parámetro para la sincronización en frecuencia de la señal. Se recomienda usar el valor por defecto de este parámetro.
- Phase Loop Bandwidth: Parámetro de sincronización en fase para la recuperación de la señal en el receptor. Se recomienda usar el valor por defecto de este parámetro.

- Timing Bandwidth: Parámetro de sincronización en tiempo para la recuperación de la señal en el receptor. Se recomienda usar el valor por defecto de este parámetro.
- Omega Relative Limit: Valor máximo de omega (valor inicial para el número de muestra por símbolos).
- Gray Code: Permite elegir si se recibe la información binaria con codificación Gray.
- Verbose: On/Off
- Logging: On/Off

### **3.5.PSK Mod**

Este bloque realiza una modulación digital PSK (*Phase shift keying*, modulación por desplazamiento de fase). Se encuentra en la librería *Digital Modulators*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Number of constellations: Orden de la constelación. El valor válido de este parámetro claramente debe corresponder a una potencia de 2, dependiendo del tipo de modulación que se va a realizar (BPSK, QPSK, 8PSK...)

- Gray code: Permite seleccionar si se envía la información binaria con codificación Gray.
- Differential encoding: Permite especificar si se realiza codificación diferencial de la señal en la modulación para su transmisión.
- Samples/Symbol: Especifica el número de muestras que se transmiten por cada tiempo de símbolo.
- Excess BW: Exceso de ancho de banda o también definido como factor de roll off del filtro raíz de coseno alzado, incorporado en este bloque de modulación para reducir la interferencia intersímbolo (ISI). Este factor permite medir la suavidad de la transición del filtro.
- Verbose: On/Off.
- Logging: On/Off.

### **3.6. PSK Demod**

Este bloque realiza una desmodulación digital PSK. Se encuentra en la librería Digital Modulators.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Number of constellations: Orden de la constelación. Este parámetro determina el tipo de modulación que se desea realizar. Este valor debe coincidir con el que se configura en el bloque PSK Mod.

- Differential encoding: Permite especificar si se envió la señal con codificación diferencial para su adecuada recepción.
- Samples/Symbol: Especifica el número de muestras que se transmiten por cada tiempo de símbolo.
- Excess BW: Exceso de ancho de banda o también definido como factor de roll off del filtro raíz de coseno alzado, incorporado en este bloque de modulación para reducir la interferencia intersímbolo (ISI). Este factor permite medir la suavidad de la transición del filtro.
- Frequency BW: Medida de que tan rápido se desea adquirir sincronización en frecuencia en el receptor para recuperar de forma adecuada la señal. Se recomienda usar el valor por defecto de este parámetro.
- Timing BW: Medida de que tan rápido se desea adquirir sincronización en tiempo en el receptor para recuperar de forma adecuada la señal. Se recomienda usar el valor por defecto de este parámetro.
- Phase BW: Medida de que tan rápido se desea adquirir sincronización en fase en el receptor para recuperar de forma adecuada la señal. Se recomienda usar el valor por defecto de este parámetro.
- Gray Code: Permite seleccionar si se recibe la información binaria con codificación Gray.
- Verbose: On/Off.
- Logging: On/Off.

## 4. Sincronización USRP

### 4.1. USRP Source

Herramienta de GNU Radio que recibe y procesa la información desde el dispositivo USRP-Receptor. Se encuentra en la librería *UHD*.

- Output type: controla el tipo de datos del flujo en GNU Radio. *Complex float32*, *Complex int 16*, *VITA word32*.
- Wire format: controla la forma de los datos sobre el bus/red. Bytes complejos pueden ser usados para sacrificar precisión por ancho de banda. No todos los formatos son soportados en todos los dispositivos USRP. *Automatic*, *Complex int16*, *Complex int8*.
- Stream args: es una lista de claves/valores pares; el uso es determinado por la implementación. *Scalar=1024*: afecta la escala entre 16 y 8 bit enteros en formato de clave *sc8*.
- Device addr: es una cadena delimitada usada para localizar dispositivos UHD en el sistema. Si se deja en blanco, el primer dispositivo UHD encontrado será usado. Se utiliza para especificar la dirección un dispositivo específico o lista de dispositivos. USRP1 Ejemplo: *serial=4d05be08*.
- Sync:
  1. Unknown PPS:
  2. Don't sync:
- Clock rate (Hz): Especifica la frecuencia del reloj del dispositivo. Default, valor por defecto.

- Num Mboards: selecciona el número tarjetas madre del USRP (*motherboards*) en el dispositivo a configurar.
- Mb0: clock source: necesario si el bloque de fuente y sumidero están referenciados al mismo dispositivo. Esto solo es necesario para fijar la referencia de la fuente en uno de los bloques.
- Mb0: time source: necesario si el bloque de fuente y sumidero están referenciados al mismo dispositivo. Esto solo es necesario para fijar la referencia de la fuente en uno de los bloques.
- Mb0: subdev Spec: subdevice specification: Especificación del subdispositivo de la tarjeta madre en uso. Dejar en blanco para que el sistema seleccione uno por defecto.
- Num channels: Selecciona el número total de canales en la configuración de la USRP.
- Samp rate (sps): es el número de muestras por segundo de entrada a este bloque. El controlador del dispositivo UHD probará la mayor frecuencia que se adapta a los requerimientos. Si la tasa requerida no es posible, el bloque UHD imprimirá un error en tiempo de ejecución.
- Ch0: center freq (Hz): es la frecuencia global de la cadena de RF. Para un mayor control de como el UHD sintoniza los elementos de la cadena RF, se coloca un rango de frecuencias y no una frecuencia puntual.
- Ch0: gain (dB): ganancia aplicada a las señales que entran en el canal 0. Por defecto se deja cero.



- Ch0: Antenna:
- Ch0: Bandwidth (Hz):

#### **4.2. USRP Sink**

Los parámetros de configuración son iguales a los explicados en la sección *USRP Source*.

### **5. Sumideros de señal**

#### **5.1. WX GUI Number Sink**

Esta herramienta se utiliza para visualizar números. Se encuentra en la librería *WX GUI Widgets*. Los parámetros de configuración son los siguientes:

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Type: Especifica el tipo de datos que procesa el bloque. *Complex, Float*.

- Title: corresponde al título asignado a la ventana de visualización de los números.
- Sample rate: Tasa a la cual el bloque procesa los datos.
- Min value: define el límite inferior de los números a visualizar.
- Max value: define el límite superior de los números a visualizar.
- Factor: los números que entran son multiplicados por el factor asignado.
- Decimal places: se utiliza para especificar la cantidad de cifras decimales que pueden ser visualizadas.
- Reference level: los números entrantes, luego de ser multiplicados, se les adiciona un nivel de referencia.
- Peak hold: tiene la opción on u off.
- Average: tiene opción on u off. Presenta un promedio de los números observados a intervalos de tiempo.
- Show gauge: se selecciona entre *show* (mostrar) y *hide* (ocultar). Permite observar y ocultar la indicadora de números.
- Window size: dejar en blanco si se desea utilizar el tamaño por defecto, de lo contrario se especifica los pixeles de ancho y alto.

- Grid position: Coordenadas que permiten especificar la posición y el tamaño del elemento en la interfaz gráfica. (Fila, columna, tamaño de fila, espacio de columna).
- Notebook: Tener en cuenta este parámetro solo si se ha creado con anterioridad un bloque de notebook para visualizar varias pestañas dentro de una misma ventana. Si se desea que este elemento se visualice en la interfaz gráfica en la pestaña índice\_de\_página de una ventana, del bloque Notebook\_ID (Notebook\_ID, índice de página).

## 5.2. File Sink

Herramienta que permite escribir un archivo en el sistema. Se encuentra en la librería *Sink*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- File: Especifica la ubicación en el sistema, el nombre del archivo y la extensión.
- Input Type: tipo de datos de entrada al bloque. *Complex, Float, Int, Short, Byte*.
- Vec Length: Longitud del vector que procesa el bloque. Depende de la longitud de los datos que llegan a este bloque. A menos que el flujo de datos sea un arreglo de complejos, float, int o short, la longitud del vector debería ser uno.

- Unbuffered: Off/On.

### 5.3. Wav File Sink

Bloque de procesamiento de señales que se encuentra en la librería *Sink* y que permite almacenar archivos tipo wav en el sistema.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- File: Especifica la ubicación en el sistema, el nombre del archivo y la extensión.
- N Channels: Especifica en número de entradas al bloque.
- Sample Rate: Frecuencia de muestreo con la cual se graba el archivo, esta frecuencia generalmente es de 16, 22.05, 24, 32, 44.1 y 48 kHz.
- Bits per Sample: número de bits que representan cada símbolo.

### 5.4. WX GUI FFT Sink

Sumidero gráfico que permite visualizar la FFT (Transformada rápida de Fourier, del inglés *Fast Fourier Transform*) de la señal en un punto. Se encuentra en la librería *WX GUI Widgets*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Type: Especifica el tipo de datos que procesa el bloque. Puede ser de tipo: Complejo (*Complex*) o Flotante (*Float*).
- Title: Título o nombre que tendrá este elemento en la interfaz gráfica.
- Sample Rate: Tasa a la cual el bloque procesa los datos.
- Baseband frequency: Permite especificar la frecuencia a la cual estará centrada la visualización.
- Y per Div: Permite cambiar la escala en amplitud. Especificar cuantos decibeles (dB) habrán por cada división.
- Y divs: Define cuantas divisiones en amplitud van a existir.
- Ref level (dB): Es el nivel mostrado por el analizador en lo alto del diagrama de medida. Es la referencia a la cual se especifica que las medidas esperadas no superaran este valor.
- FFT size: Número de puntos de la FFT. Especifica la resolución de la pantalla (display). A medida que se aumenta este valor, la resolución incrementa. Un valor adecuado es el mostrado por defecto en el bloque, 1024.
- Refresh rate: Tasa de refresco. Que tan rápido se actualizan los cuadros en la pantalla para dar forma y optima visualización al espectro de la señal. Se mide en Hz.

- Peak Hold: Enciende o no el detector de picos.
- Average: Especificar si se desea o no activar la opción del promedio. Promediar muchos espectros juntos mejora la precisión y la repetibilidad de las mediciones.<sup>59</sup>
- Window: Determinar el tipo de ventana con el cual va a trabajar la FFT. Puede ser de tipo: *automatic*, *hanning*, *hanning*, *rectangular*, *flattop*. El criterio operativo de una ventana es el de tener un lóbulo principal lo más estrecho posible para conseguir una buena resolución espectral y ausencia de lóbulos laterales para evitar la dispersión espectral.
- Window size: Determinar el tamaño de la ventana, debe introducirse de la forma (ancho, alto) en pixeles. Si se deja en blanco será el tamaño por defecto de la ventana.<sup>60</sup>
- Grid position: Coordenadas que permiten especificar la posición y el tamaño del elemento en la interfaz gráfica. (Fila, columna, tamaño de fila, espacio de columna).
- Notebook: Tener en cuenta este parámetro solo si se ha creado con anterioridad un bloque de notebook para visualizar varias pestañas dentro de una misma ventana. Si se desea que este elemento se visualice en la interfaz gráfica en la pestaña índice\_de\_página de una ventana, del bloque Notebook\_ID (Notebook\_ID, índice de página).

---

<sup>59</sup> Disponible en: [www.thinkSRS.com](http://www.thinkSRS.com) [Fecha de Consulta: julio 16 de 2012]

<sup>60</sup> Disponible en: <http://www.ehu.es/Procesadodesenales/tema8/tz2y.html> [Fecha de Consulta: julio 16 de 2012]

### 5.5. WX GUI Scope Sink

Osciloscopio. Sumidero gráfico que permite visualizar una señal en el tiempo. Se encuentra en la librería *WX GUI Widgets*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Type: Especifica el tipo de datos que procesa el bloque. Complejo (*Complex*) o Flotante (*Float*).
- Title: Título o nombre que tendrá este elemento en la interfaz gráfica.
- Sample Rate: Tasa a la cual el bloque procesa los datos.
- V scale: Escala de amplitud en volts. Especificar cuantos Volts habrán por cada división en el eje de amplitud- eje Y. Dejar en cero si se desea una escala automática.
- V offset: Nivel de DC de la señal.
- T scale: Escala de tiempo en segundos. Especificar cuantos segundos habrán por cada división en el eje de tiempo-eje X. Dejar en cero si se desea que la herramienta realice una configuración automática.
- AC Couple: Activar o no el acoplo AC.
- XY Mode: Especificar si se desea observar la señal en Modo XY.
- Num inputs: Especificar el número de entradas al osciloscopio.

- Window size: Determinar el tamaño de la ventana, debe introducirse de la forma (ancho, alto) en pixeles. Si se deja en blanco, será el tamaño por defecto de la ventana.
- Grid position: Coordenadas que permiten especificar la posición y el tamaño del elemento en la interfaz gráfica. (fila, columna, tamaño de fila, espacio de columna).
- Notebook: Tener en cuenta este parámetro solo si se ha creado con anterioridad un bloque de notebook para visualizar varias pestañas dentro de una misma ventana. Si se desea que este elemento se visualice en la interfaz gráfica en la pestaña índice\_de\_página de una ventana, del bloque Notebook\_ID (Notebook\_ID, índice de página).
- Trigger Mode: Modo de disparo. Determina si el osciloscopio dibuja o no una forma de onda al no detectar un *trigger* (nivel que permite estabilizar formas de onda repetitivas en la pantalla del osciloscopio). Las opciones posibles para este parámetro son: auto, normal, *freerun*, *stripchart*.
- Y Axis label: Etiqueta o nombre que tendrá el eje de las ordenadas- eje Y.

## 6. Ajuste de variables

### 6.1. WX GUI Slider

Este bloque tiene como propósito crear una variable para permitir que tome diferentes valores dentro de un rango específico y observar como varia en tiempo real la respuesta del esquema mientras ocurren estos cambios. Esto se logra a través de una manipulación de esta variable desde la interfaz gráfica que proporciona la opción de mover un control deslizante (*Slider*) desde el nivel mínimo hasta el nivel máximo, tomando valores según el número de pasos especificados. Se encuentra en la librería *WX GUI Widgets*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Label: Es el nombre o etiqueta que va a aparecer en pantalla al momento de ejecutar el esquema y servirá de referencia para distinguirlo de otros controles deslizantes (*sliders*) en la interfaz gráfica. Se deja esta casilla en blanco si se usa la variable ID como etiqueta.
- Default Value: Es el valor por defecto que va a tener la variable. Si no se realiza ningún cambio de la variable en la interfaz gráfica por parte del usuario, el valor con el cual va a trabajar el proceso será este. Debe ser un valor real y estar entre los límites mínimo y máximo definido en las siguientes opciones.
- Minimum: valor mínimo que puede tener la variable creada.
- Maximum: valor máximo que puede tener la variable creada.

- Num Steps: Número de divisiones. Debe estar entre 0 y 1000.
- Style: Da la opción si se desea que la barra, slider, sea de forma horizontal o vertical.
- Converter: Permite convertir los datos a flotantes o complejos.
- Grid position: Coordenadas que permiten especificar la posición y el tamaño del elemento en la interfaz gráfica. (fila, columna, tamaño de fila, espacio de columna).
- Notebook: Tener en cuenta este parámetro solo si se ha creado con anterioridad un bloque de notebook para visualizar varias pestañas dentro de una misma ventana. Si se desea que este elemento se visualice en la interfaz gráfica en la pestaña índice\_de\_página de una ventana, del bloque Notebook\_ID (Notebook\_ID, índice de página).

## 6.2. Variable

Este bloque determina un único valor a una variable en todo el esquemático. Se encuentra en la librería *Variable*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Value: Magnitud que toma la variable.

*Nota:* Cuando se desee utilizar esta variable en algún bloque del esquema, simplemente se escribe el ID (nombre de la variable) en la

casilla de interés del bloque donde se desee usar este valor. Esto evita en el esquemático, donde se necesite un mismo valor en repetidos bloques, no copiar exactamente el valor numérico, sino el ID de la variable creada.

### 6.3. Parameter

Este bloque representa un parámetro para el diagrama de flujo. Un parámetro puede ser usado para establecer un valor fijo de una variable, con un tipo de dato definido. Se encuentra en la librería *Variables*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Label: etiqueta para designar el parámetro. Puede dejarse en blanco si se desea usar el ID como etiqueta.
- Value: valor que se le asigna al parámetro.
- Type: tipo de dato para definir el parámetro. *None, Complex, Float, Int, Long, String*.

### 6.4. WX GUI Chooser

Este bloque crea una variable con opciones de selección en forma desplegable, botones de radio y botones. Se encuentra en la librería *WX GUI Widgets*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.

- Label: Etiqueta para designar el bloque. Dejar en blanco si se desea utilizar el ID como etiqueta.
- Default Value: define la elección por defecto cuando el diagrama de flujo se ejecuta.
- Choices: lista de los posibles valores.
- Labels: lista de etiquetas de los posibles valores. Se deja en blanco si se desea que los valores posibles sean las etiquetas.
- Type: especifica la forma de presentación para las opciones de selección.

1. *Drop Down*:



**Figura G35. WX GUI Chooser. Drop Down.**

2. *Radio button*: Horizontal o vertical.

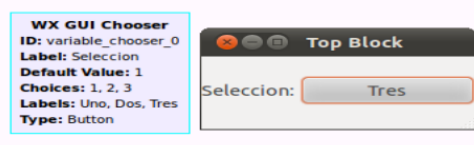


**Figura G36. WX GUI Chooser. Radio button horizontal.**



**Figura G37. WX GUI Chooser. Radio button vertical.**

### 3. *Button*:



**Figura G38. WX GUI Button.**

- Grid position: Coordenadas que permiten especificar la posición y el tamaño del elemento en la interfaz gráfica. (fila, columna, tamaño de fila, espacio de columna).
- Notebook: Tener en cuenta este parámetro solo si se ha creado con anterioridad un bloque de notebook para visualizar varias pestañas dentro de una misma ventana. Si se desea que este elemento se visualice en la interfaz gráfica en la pestaña índice\_de\_página de una ventana, del bloque Notebook\_ID (Notebook\_ID, índice de página).

## 6.5. WX GUI Notebook

Con esta herramienta se pueden crear varias pestañas dentro de una misma ventana para seleccionar cada una de las visualizaciones adecuadamente. Se encuentra en la librería *WX GUI Widgets*.

- ID: Nombre asignado al bloque para identificarlo dentro del esquema grc específico.
- Tab orientation: *Top* (arriba), *Right* (derecha), *Bottom* (abajo) y *Left* (izquierda). Indica la posición en la cual se ubican las pestañas seleccionables.
- Labels: se le asignan nombres a cada una de las pestañas, teniendo en cuenta el formato siguiente: ['tag1', 'tag2', 'tag3']
- Grid position: Coordenadas que permiten especificar la posición y el tamaño del elemento en la interfaz gráfica. (fila, columna, tamaño de fila, espacio de columna).
- Notebook: Tener en cuenta este parámetro solo si se ha creado con anterioridad un bloque de notebook para visualizar varias pestañas dentro de una misma ventana. Si se desea que este elemento se visualice en la interfaz gráfica en la pestaña índice\_de\_página de una ventana, del bloque Notebook\_ID (Notebook\_ID, índice de página).