

# **DISEÑO Y MONTAJE DE UNA ESTACIÓN DE SEGUIMIENTO DE SATÉLITE EN ÓRBITA BAJA**

**Laura Esteve Moreno**

**Tutor: Ana Vidal Pantaleoni**

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación.

Curso 2013-14

Valencia, 2 de Septiembre de 2014

## ***RESUMEN***

En este proyecto se ha llevado a cabo una solución de recepción para el primer satélite desarrollado por la Universidad Politécnica de Valencia, el Politech.1. El receptor consiste en un dispositivo de tipo SDR (Software Defined Radio) denominado NI-USRP 2920.

A lo largo del proyecto, se han efectuado diversas versiones de programas de demodulación FSK hasta llegar a la óptima para realizar la recepción de datos del satélite mediante el USRP. Se han realizado distintos escenarios con diferentes dispositivos de transmisión de señal para estudiar la tasa de error de bit, analizar las potencias y frecuencias así como verificar la sensibilidad máxima del USRP.

Finalmente, se ha comprobado que dada la baja potencia que proporciona el HiDAC, no es viable realizar esta solución de recepción debido a la baja sensibilidad del USRP. Por consiguiente, como línea futura de investigación restaría aumentar la potencia al HiDAC y estudiar otras soluciones de recepción para el Politech.1.

## ***RESUM***

En aquest projecte s'ha dut a terme una solució de recepció per al primer satèl·lit desenvolupat per la Universitat Politècnica de València, el Politech.1. El receptor consisteix en un dispositiu de tipus SDR (Software Defined Radio) denominat NI-USRP 2920.

Al llarg del projecte, s'han efectuat diverses versions de programes de demodulació FSK fins arribar a l'òptima per realitzar la recepció de dades del satèl·lit mitjançant el USRP. S'han realitzat diferents escenaris amb diferents dispositius de transmissió de senyal per estudiar la taxa d'error de bit, analitzar les potències i freqüències així com verificar la sensibilitat màxima del USRP.

Finalment, s'ha comprovat que donada la baixa potència que proporciona l'HiDAC, no és viable realitzar aquesta solució de recepció a causa de la baixa sensibilitat del USRP. Per tant, com a línia futura d'investigació restaria augmentar la potència al HiDAC i estudiar altres solucions de recepció per al Politech.1.

# ***ABSTRACT***

This project help us to find a receiving solution for the first satellite developed by the Technical University of Valencia (UPV, Universidad Politécnica de Valencia), the Politech.1. The receiver will use SDR (Software Defined Radio) device called NI-USRP 2920.

Throughout of the project, various versions of FSK demodulation programs have been carried out until reaching the optimum LabVIEW code for receiving satellite data using the USRP. Various scenarios with different signal transmission devices have been implemented to study the bit error rate, to analyze powers and frequencies as well as verify the maximum sensitivity of the USRP.

Finally, it has been checked that due to the low power provided by the HiDAC, the USRP device is not the best solution due to its low sensitivity. Therefore, as a future line of research, it would be interesting to increase power of HiDAC and to study alternative solutions for the data reception of the Politech.1.

# ÍNDICE

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>INTRODUCCIÓN.....</b>               | <b>6</b>  |
| 1.1      | MOTIVACIÓN.....                        | 6         |
| 1.2      | ESTADO DEL ARTE .....                  | 7         |
| 1.3      | OBJETIVOS .....                        | 8         |
| 1.4      | ESTRUCTURA DE LA MEMORIA.....          | 8         |
| <b>2</b> | <b>METODOLOGÍA DE TRABAJO .....</b>    | <b>10</b> |
| 2.1      | GESTIÓN DEL PROYECTO .....             | 10        |
| 2.2      | DISTRIBUCIÓN EN TAREAS.....            | 11        |
| 2.3      | DIAGRAMA TEMPORAL .....                | 11        |
| <b>3</b> | <b>ENTORNO DE DESARROLLO.....</b>      | <b>12</b> |
| 3.1      | POLITECH.1 .....                       | 12        |
| 3.1.1    | Plataforma NAOSat .....                | 12        |
| 3.1.2    | Cargas útiles.....                     | 13        |
| 3.1.2.1  | GEODEYE .....                          | 14        |
| 3.1.2.2  | SatFOSS.....                           | 14        |
| 3.1.2.3  | Antena en banda C (HiDAC).....         | 14        |
| 3.2      | NI USRP 2920 .....                     | 17        |
| 3.2.1    | Características .....                  | 17        |
| 3.2.2    | Hardware .....                         | 19        |
| 3.2.3    | Transmisor USRP .....                  | 20        |
| 3.2.4    | Receptor USRP.....                     | 20        |
| 3.3      | LABVIEW .....                          | 21        |
| 3.3.1    | Entorno de Labview .....               | 21        |
| 3.3.2    | Programación .....                     | 26        |
| 3.3.3    | Ejecución de un VI .....               | 26        |
| 3.4      | Modulación FSK.....                    | 28        |
| 3.4.1    | Demoduladores detectores de FM.....    | 30        |
| 3.4.2    | Demoduladores de filtro-tipo fsk ..... | 31        |
| 3.5      | TRANSCEIVER ML5805.....                | 34        |
| 3.5.1    | Modo transmisión del ML5805 .....      | 35        |
| 3.5.2    | Starter Kit ML5805.....                | 35        |

|          |   |           |
|----------|---|-----------|
| <b>4</b> | <b>DESARROLLO.....</b>                      | <b>37</b> |
| 4.1      | CONTROL CON EL USRP .....                   | 37        |
| 4.2      | DEMODULACIÓN FSK .....                      | 38        |
| 4.3      | DESARROLLO DEL DEMODULADOR EN LABVIEW ..... | 41        |
| 4.3.1    | Primera versión .....                       | 41        |
| 4.3.2    | Segunda versión.....                        | 44        |
| 4.3.3    | Tercera versión.....                        | 47        |
| 4.3.4    | Última versión.....                         | 51        |
| <b>5</b> | <b>RESULTADOS.....</b>                      | <b>54</b> |
| 5.1      | STARTER-KIT 03.....                         | 54        |
| 5.2      | HÍBRIDO .....                               | 58        |
| 5.3      | HIDAC.....                                  | 61        |
| <b>6</b> | <b>CONCLUSIONES .....</b>                   | <b>68</b> |
|          | Índice de figuras .....                     | 70        |
|          | Índice de tablas .....                      | 72        |
|          | Bibliografía.....                           | 73        |

# 1 INTRODUCCIÓN

## 1.1 MOTIVACIÓN

El presente proyecto estudia la viabilidad de utilizar un receptor determinado en la estación terrestre del satélite Politech.1. La estación se situará en el campus de la Universidad Politécnica de Valencia, concretamente en el RadioClub.

El satélite Politech.1 es un proyecto desarrollado por la Universidad Politécnica de Valencia en colaboración con la empresa Emxys que cuenta además con el soporte de la Agencia Espacial Europea (ESA), el Consorcio Espacial Valenciano (VSC) y la spin-off Aurora SAT.

La Universidad Politécnica de Valencia inició diversas actividades bajo el lema “UPV Technology in Space”. Uno de los retos que se plantean es el desarrollo, la puesta en marcha y operación de un picosatélite (acogido al estándar Cubesat) llamado Politech.1.

El Politech.1 se convertirá así en el primer satélite lanzado por una universidad valenciana, la Universidad Politécnica de Valencia. Con este proyecto se pretende llevar a cabo diversos experimentos científicos que están siendo desarrollados por parte de distintos grupos de investigación de la UPV.

El picosatélite Politech.1 es un satélite de pequeñas dimensiones, concretamente 10 x 10 x 30 cm cuyo peso será de 3 kg aproximadamente. En el satélite irá embarcada una cámara telescópica geodésica para observar la Tierra mediante la toma de imágenes, un transmisor en banda C con antenas desplegadas y un experimento de sensores de fibra para monitorizar temperaturas.

En el RadioClub de la UPV se instalará la estación terrestre donde se descargarán todos los datos e imágenes que recoja el satélite, una vez puesto en órbita. El objetivo es que el Politech.1 se mantenga en órbita durante dos años y su misión consiste en promover actividades tecnológicas y científicas que sirvan de oportunidad educativa para los estudiantes de la universidad.

En el presente proyecto se estudiará una solución para la recepción de los datos procedentes del satélite consistente en un SDR (Software Defined Radio). Se explicarán las distintas ventajas que presenta este tipo de dispositivos así como su viabilidad para la recogida de información proveniente del satélite.

A lo largo del proyecto se desarrollarán distintos escenarios para realizar la recepción de datos del satélite minimizando los errores. También se llevarán a cabo distintos experimentos con el objetivo de comparar el escenario final presentado con distintos transmisores, además del embarcado en el satélite. En definitiva, se llevarán a cabo distintas soluciones de recepción y finalmente se elegirá la óptima para el propósito que nos ocupa.

## 1.2 ESTADO DEL ARTE

Desde el primer satélite mandado al espacio hasta la actualidad ha variado mucho el concepto y la finalidad que se le ha dado a los satélites. Durante la Guerra Fría, el Sputnik 1 marcó el inicio de la carrera espacial con un objetivo únicamente militar [1]. No obstante, a lo largo de estos años los satélites han sido utilizados para cubrir distintas y diversas finalidades. Hoy en día, los satélites dan servicios de telecomunicaciones, meteorológicos, de navegación, de propósitos experimentales...

En lo que respecta a estos últimos, en los años 80, científicos universitarios empezaron a desarrollar proyectos espaciales. Sin embargo, no fue hasta 1999 cuando se establecieron unos requisitos fijados por la Universidad Politécnica del Estado de California y la Universidad de Standford para que todas las universidades fueran capaces de desarrollar investigaciones espaciales de bajo coste. De esta forma se avanza en este concepto de pequeños satélites desde mini, micro, nano hasta los picosatélites de la actualidad.

Así, el *CubeSat Program* aporta grandes ventajas para los estudiantes ya que les brinda la posibilidad de introducirse en investigaciones espaciales con un bajo coste de elaboración, diseño, lanzamiento y operación. En contraposición, el inconveniente que presentan es el poco tiempo de desarrollo, las fuentes de financiación, la corta vida útil y la potencia entregada. [3]

A pesar de los inconvenientes citados, los sistemas de comunicaciones basados en pequeños satélites de órbita baja (LEO) han gozado de una cálida acogida en los últimos años. Este tipo de satélites se utilizan especialmente en misiones de corta duración y proporcionan soluciones efectivas a los problemas presupuestarios. Al disminuir el tamaño de los satélites se consigue una reducción del coste de lanzamiento lo que resulta muy atractivo para las entidades que financian este tipo de proyectos.

La Universidad Politécnica del Estado de California y la Universidad de Standford propusieron un estándar para el desarrollo de picosatélites con el objetivo de poder realizar una gran cantidad de lanzamientos bajo una plataforma de separación que pudiera albergar hasta tres picosatélites como carga secundaria de un cohete, con lo que se reducirían los costes de los lanzamientos. Este estándar es conocido como Cubesat y existen diferentes tipos aunque el que se utilizará en el presente proyecto será la configuración 3U. La U hace referencia al número de unidades que se colocan consecutivamente, por lo tanto, en este caso estaría formado por la combinación de tres cubos, con unas dimensiones de 10 x 10 x 30 cm con un máximo de 3 Kg. [4]

## 1.3 OBJETIVOS

El objetivo final de este proyecto es diseñar un sistema de comunicaciones viable cuyo receptor es lo que se conoce como un SDR (Software Defined Radio), es decir, un sistema de radiocomunicaciones donde los componentes están implementados en software utilizando un ordenador. Mediante el análisis de este receptor se pretende estudiar las prestaciones que este dispositivo es capaz de ofrecer. En consecuencia, se probarán y simularán distintos escenarios para comprobar cuáles son las ventajas y desventajas que puede dar esta solución de recepción de los datos del satélite.

Así pues, los objetivos técnicos serán:

- Analizar y estudiar las prestaciones de la utilización de un receptor SDR, concretamente el NI-USRP 2920 fabricado por la empresa National Instruments.
- Estudiar los métodos de programación y el entorno de desarrollo del software LabVIEW, que utiliza el controlador hardware NI-USRP.
- Diseñar un demodulador FSK en LabVIEW.
- Crear distintos escenarios, con distintos dispositivos de transmisión de la señal para estudiar la tasa de error de bit dada cada sistema de comunicación.
- Analizar las distintas potencias y frecuencias recibidas en cada escenario en función de las diferentes fuentes de señal.
- Verificar cuál será la sensibilidad máxima del USRP en función de los distintos escenarios construidos con sus diversos dispositivos de transmisión de la señal.
- Concluir con los resultados de si resultaría viable mantener un sistema de comunicación cuya recepción esté basada en el USRP.

## 1.4 ESTRUCTURA DE LA MEMORIA

La memoria que se expone en el presente proyecto ofrece la siguiente estructura. En primer lugar se realiza una explicación de los aspectos generales que son esenciales para ser capaces posteriormente de asimilar los conceptos más concretos del trabajo que se ha ido elaborando exponiendo en cada capítulo los objetivos y al final de ellos la conclusión a la que se ha llegado. De esta manera, en el **capítulo 1** se realiza una introducción donde se explica en líneas generales en qué consiste el proyecto del Politech.1, cómo se dividen los grupos de trabajo que lo componen, cuáles son las expectativas y cuáles son los aspectos que se tratarán de solucionar en el presente proyecto. Además, en el Estado del Arte se realiza una breve explicación de dónde surge la idea de los pequeños satélites de bajo coste, su estándar Cubesat y sus ventajas e inconvenientes. Por otro lado, en este capítulo también se esclarecen los objetivos que se pretenden satisfacer en el presente proyecto.



El **capítulo 2** aborda aspectos relativos a la planificación y la estructura de la memoria. Se incluye aquí la gestión del proyecto, la distribución en tareas y el diagrama temporal.

En el **capítulo 3** se introduce el entorno en el que se va a desarrollar el proyecto. En primer lugar, se realiza una introducción al Politech.1, para después dar paso a la explicación de la plataforma NAOSat. A continuación, se describen brevemente las cargas útiles: Geodeye, SatFOSS y la antena en banda C. Ésta última será la que más importancia tenga dado que el proyecto se centra en diseñar uno de los receptores en banda C del satélite, ubicados en la estación terrestre, que se explicará en este apartado.

En este capítulo se explicarán las características del dispositivo de recepción, es decir, el NI-USRP. Se recogerán en este apartado las ventajas de utilizar este aparato, así como sus características hardware. Este dispositivo es un transceptor, por lo tanto es capaz tanto de transmitir como de recibir señales, por ello, se describirán las funciones particulares del modo transmisión y recepción. Por otra parte, el NI-USRP es un SDR (Software Defined Radio), es decir, un sistema de radiocomunicaciones donde los elementos están implementados en software. Es por ello, que el USRP se sirve del software LabVIEW para ofrecer sistemas potentes para trabajar. De esta manera, en este capítulo se explica también en qué consiste este software, cuál es su entorno y cuál es la forma de programar y de ejecutar un programa. Se hace especial hincapié en estos dos aspectos (USRP y LabVIEW) dado que serán especialmente relevantes para llevar a cabo el presente proyecto.

En este capítulo también se abordará la modulación FSK (Frequency Shift Keying) dado que la transmisión y recepción de la información del satélite se realizará en dicha modulación. De esta manera, se explican dos tipos de demodulación de la señal: los demoduladores detectores de FM y los de filtro-tipo FSK. Para terminar este capítulo, se describirá el transceptor ML5805 que será el que irá embarcado en el satélite. Se explicarán las características de este dispositivo así como las funciones específicas del modo de transmisión y sus características, así como las de la placa de evaluación Starter Kit.

El **capítulo 4** se centra en el objetivo principal del proyecto, el diseño de un demodulador en LabVIEW. Para ello, se realiza primero una introducción dónde se especifican las distintas funciones para realizar el control del USRP así como las utilizadas para realizar una demodulación FSK.

En este capítulo se entra en aspectos concretos del proyecto, concretamente en la programación en LabVIEW de un demodulador en FSK. Por ello, se hace un repaso cronológico por las distintas versiones llevadas a cabo y cuáles fueron las soluciones propuestas a los problemas que fueron surgiendo. Para terminar, se explica la versión final a la que se ha llegado para realizar la correcta demodulación de la señal FSK.

Por último, en el **capítulo 5** se presenta el test del código implementado en LabVIEW para el demodulador FSK mediante la evaluación de diversos esquemas de sistemas de comunicación. Es decir, se utilizan distintos transmisores con sus distintas características para un receptor común, el USRP con el programa de demodulación presentado. A partir de estos escenarios, se plantean pruebas para hallar la tasa de error de bit (BER) en cada uno, así como hallar la sensibilidad máxima soportable por el USRP en cada uno de los escenarios diseñados. También se llevan a cabo distintas pruebas con el analizador de espectros para hallar las frecuencias y potencias que proporcionan estos montajes.

## 2 METODOLOGÍA DE TRABAJO

En este capítulo se explica la estructura del trabajo desde su inicio hasta su fin incluyendo los cambios que se han producido a lo largo del desarrollo del mismo. En el primer apartado, *Gestión del Proyecto* se abordarán las distintas tareas del proyecto de manera genérica. En el apartado *Distribución en Tareas* se explicará cómo han sido distribuidas las tareas para realizar el proyecto. Finalmente, en el apartado *Diagrama Temporal* se incluirá un diagrama de Gantt para observar la distribución de las tareas del presente proyecto.

### 2.1 GESTIÓN DEL PROYECTO

En este apartado se explicará el planteamiento inicial así como la organización del proyecto. Inicialmente se produjo una reunión con la tutora del proyecto el día 5 de febrero como primera toma de contacto para explicar los objetivos del trabajo así como entregar la documentación necesaria para entender el proyecto. No obstante, debido a motivos académicos no se pudo comenzar hasta el mes de mayo. El término del proyecto fue en el mes de septiembre para su presentación como fecha límite impuesta por la UPV el día 5 del mismo mes.

El planteamiento inicial considerado fue el siguiente:

- Febrero: lectura de la documentación necesaria para la realización del proyecto.
- Marzo: estudio y manejo del programa LabVIEW y del dispositivo NI-USRP 2920.
- Abril: desarrollo del demodulador FSK óptimo para el proyecto.
- Mayo: pruebas del BER, medidas y redacción del proyecto.
- Junio: redacción y revisión del proyecto.

Sin embargo, como se ha indicado anteriormente, se tuvo que posponer el inicio hasta el mes de mayo cuando se finalizó el periodo lectivo. Por consiguiente, el planteamiento que se realizó fue el siguiente:

- Mayo: estudio y manejo del programa LabVIEW y del dispositivo NI-USRP 2920.
- Junio: desarrollo del demodulador FSK óptimo para el proyecto.
- Julio: pruebas del BER, medidas, redacción del proyecto y revisión.

Éste pretendía ser el planteamiento de trabajo definitivo; no obstante, se produjeron problemas con una de las placas transmisoras, concretamente el HiDAC por lo que tuvo que procederse a su reparación. De esta manera, el planteamiento definitivo que se realizó fue:

- Mayo: estudio y manejo del programa LabVIEW y del dispositivo NI-USRP 2920.
- Junio: desarrollo del demodulador FSK óptimo para el proyecto.
- Julio: desarrollo del demodulador FSK óptimo, pruebas del BER y medidas.
- Agosto: redacción del proyecto y revisión del mismo.

Mediante este último planteamiento se logró completar satisfactoriamente todas las tareas necesarias para la correcta realización del proyecto.

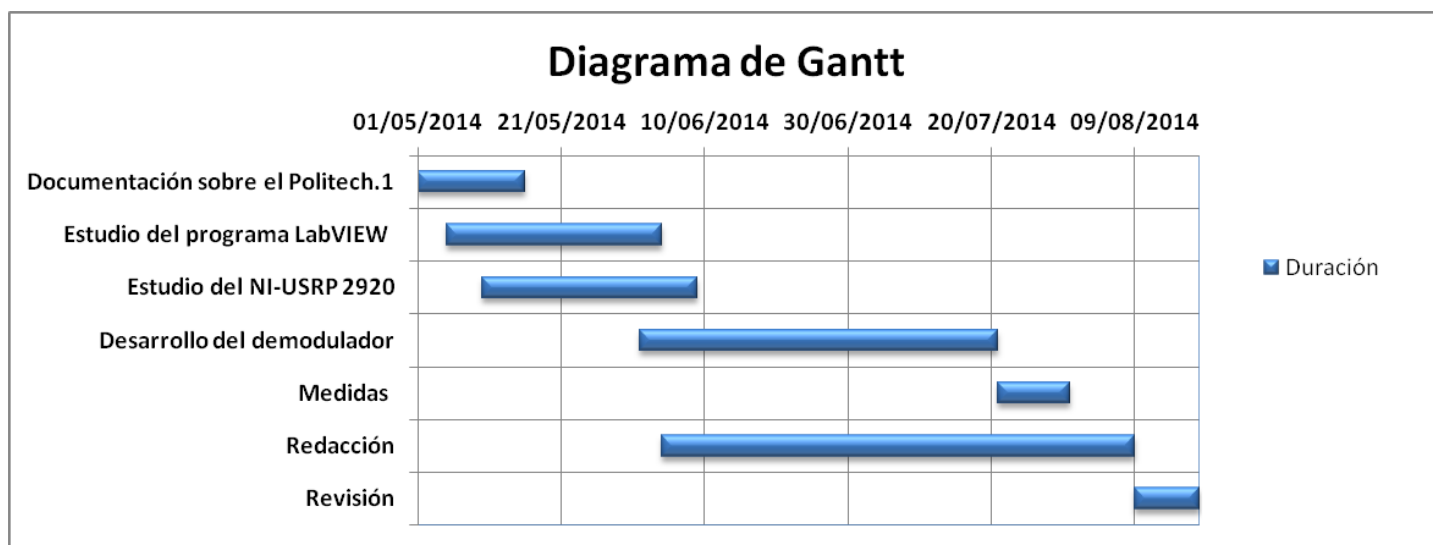
## 2.2 DISTRIBUCIÓN EN TAREAS

En este apartado se enumerarán las tareas específicas realizadas para llevar a cabo el proyecto. Las tareas no se explicarán de detalle ya que serán desarrolladas a lo largo de los capítulos siguientes de la memoria.

1. Lectura y comprensión del proyecto Politech.1.
2. Búsqueda de información y estudio del programa LabVIEW.
3. Desarrollo y creación de distintos programas de ejemplos para la correcta asimilación de la programación en LabVIEW.
4. Búsqueda de información y estudio del dispositivo NI-USRP 2920.
5. Desarrollo de ejercicios para la correcta asimilación de la forma de trabajar del dispositivo NI-USRP 2920.
6. Desarrollo de diversas versiones de demodulación FSK.
7. Testeo de las distintas versiones de demodulador.
8. Desarrollo de un programa de demodulación FSK óptimo para la recepción mediante el NI-USRP 2920.
9. Creación de distintos escenarios con distintas fuentes de señal.
10. Medida de las frecuencias obtenidas en cada escenario.
11. Medida de las potencias dadas por cada uno de las distintas fuentes de señal.
12. Medida del BER (Bit Error Rate) en cada uno de los escenarios creados.
13. Análisis de la sensibilidad máxima admisible por el NI-USRP 2920 en cada uno de los montajes realizados.
14. Análisis de los resultados obtenidos en cada uno de los escenarios.
15. Conclusiones de los resultados obtenidos.

## 2.3 DIAGRAMA TEMPORAL

En este apartado se realiza un diagrama de Gantt para observar más fácilmente la distribución de las tareas desarrolladas en función del tiempo.



## 3 ENTORNO DE DESARROLLO

En este capítulo se abordarán aspectos generales necesarios para la definición del proyecto. En primer lugar se explicarán aspectos relativos al satélite Politech.1. A continuación se explicará el dispositivo de recepción que se analizará en este proyecto, el NI-USRP 2920. Seguidamente, se expondrá el entorno de programación del software LabVIEW y el tipo de modulación que se utiliza en este proyecto, la FSK. Finalmente, se realizará una breve descripción del transceptor utilizado para transmitir la señal en este trabajo, el ML5805.

### 3.1 POLITECH.1

La Universidad Politécnica de Valencia persigue el ambicioso proyecto de llevar a cabo su primera misión espacial en colaboración con la empresa Emxys, ubicada en el Parque Científico y Empresarial de la Universidad Miguel Hernández de Elche. El objetivo será poner en órbita el satélite llamado Politech.1 con los diversos experimentos científicos llevados a cabo por distintos grupos de investigación. [3]

#### 3.1.1 PLATAFORMA NAOSAT

El satélite Politech.1 se basa en la plataforma NAOSat de la empresa Emxys. Se trata de una arquitectura modular de satélites que está destinada a experimentos científicos de demostración tecnológica.

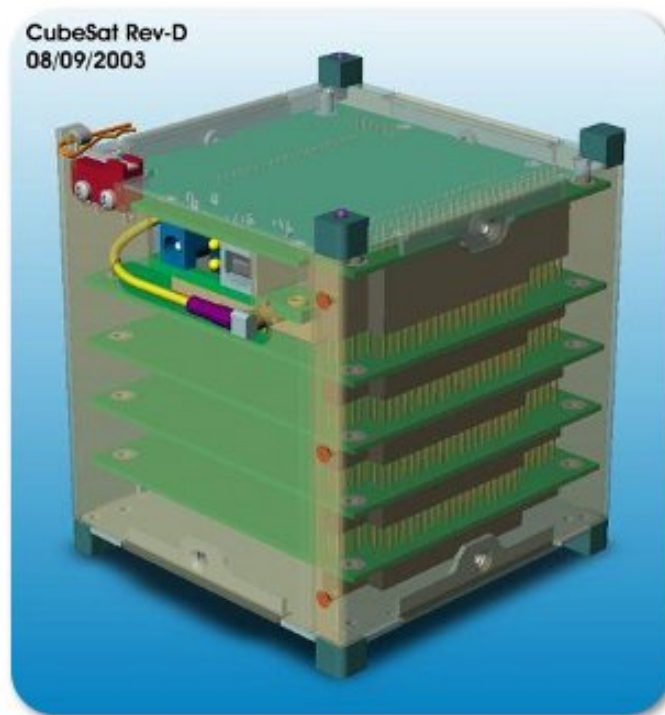
NAOSat proporciona los siguientes subsistemas de plataformas comunes:

- Ordenador de a bordo de alta fiabilidad.
  - Procesador Cortex de 32 bits.
  - Supervisor OBC redundante.
  - Reloj en tiempo real.
  - Memoria flash de almacenamiento masivo.
- Distribución de energía y control.
  - Doble o triple unión de células solares.
  - MPPT y etapas de protección de los paneles solares.
  - Múltiples salidas aisladas de convertidor DC/DC.
- Telemetría ascendente y descendente.
  - Transceptor FSK Half-Duplex.
  - 400 MHz - 900 MHz.
  - 1200 bps enlace descendente > 1W de potencia de salida.
  - Motor a bordo de codificación/decodificación.
- ADCS
  - IMU a bordo.
  - 2 magnetorquer.
  - Micro ruedas de reacción.

- Control de la temperatura.
  - Controladores térmicos pasivos en cada ranura.
  - Etapas TEC y TEC bajo demanda para partes sensibles.

Además, la plataforma NAOSat incorpora un bastidor de carga útil de bus compacto con su respectiva placa del controlador para así acomodar un número de elementos de carga útil en sus respectivas ranuras de carga, que consiste en una ranura de conexión en el bastidor y una tarjeta de alojamiento de carga útil conectado al bus principal. Cada una de estas ranuras proporciona:

- Una interfaz mecánica robusta.
- Interfaz eléctrica con energía regulada.
- Control térmico (pasivo).
- Interfaz lógica de bus para comandos y datos (no es preciso implementar un protocolo de comunicación para los diseños de carga útil). [2]



**Figura 3.1: Plataforma NAOSat**

### **3.1.2 CARGAS ÚTILES**

El Cubesat de 10 x 10 x 30 cm, en formato 3U, tendrá un peso de 3,5 Kg y las características serán:

- Potencia de 3 W en promedio dedicado a la carga útil.
- Volumen a partir de 80 x 80 x 15 mm por ranura. 1U para subsistemas de la plataforma y 2U para la carga útil.
- La masa de la carga útil será de 2,5 Kg.

Las diferentes cargas útiles del Politech.1 están siendo llevadas a cabo por distintas Escuelas e Institutos de Investigación de la Universidad Politécnica de Valencia.

Politech.1 llevará a bordo una cámara telescópica para aplicaciones de observación de la Tierra (GEODEYE). Esta carga útil será desarrollada por las Escuelas de Ingeniería Geodésica, Cartográfica y Topográfica y de Ingeniería de Telecomunicaciones, que además proporcionará datos de Teledetección con fines académicos.

Otra carga útil es el sistema de comunicaciones en banda C unidireccional (HiDAC) que se encargará de la descarga de datos. El diseño y desarrollo de este enlace lo están realizando los Grupos de Radiación y Aplicaciones de las Microondas del Instituto de Telecomunicaciones y Aplicaciones Multimedia (iTEAM) junto con la Escuela de Ingeniería de Telecomunicaciones, que también se encargará del diseño de la estación terrestre ubicada en el campus de la UPV.

Además de estas cargas, Politech.1 también realizará un experimento de fibra óptica (SatFOSS) para la monitorización de la temperatura que será llevado a cabo por Grupo de Comunicaciones Ópticas del Instituto iTEAM. [4]

### **3.1.2.1 GEODEYE**

Una de las cargas útiles del satélite será una cámara telescópica para la observación terrestre. El objetivo es realizar experimentos de investigación para comprobar si resultaría viable implementar sistemas de observación geodésica de la Tierra con el fin de obtener información geodésica exacta.

Para descargar las imágenes de la Tierra se utiliza el enlace en banda C y el receptor GPS a bordo permite obtener información de ubicación con el fin de utilizar dichas imágenes para la geodesia. Es decir, el receptor GPS ASTERIX captará la posición real de las imágenes adquiridas por el sensor CMOS para enviarlas a través del enlace en banda C al ordenador.

### **3.1.2.2 SatFOSS**

En esta carga útil se pretende probar un nuevo concepto sobre la fibra óptica basado en la temperatura y sensores de tensión. La carga consiste en un conjunto óptico implementado en un PCB 10 cm x 10 cm que intenta probar la idoneidad de un filtro implementado en FO para discriminar la longitud de onda de Bragg (FBG) del sensor situado en el mismo circuito impreso a una temperatura conocida. SatFOSS tiene el objetivo de demostrar el funcionamiento de un sistema de lectura sencilla basada en un diseño de filtro óptico desarrollado en el Instituto iTEAM de la UPV.

### **3.1.2.3 Antena en banda C (HiDAC)**

La carga útil del satélite HiDAC ha sido desarrollada por el Grupo de Radiación Electromagnética (GRE) en colaboración con el Grupo de Aplicaciones de Microondas (GAM) de la Universidad Politécnica de Valencia. Consiste en una antena parche trabajando en banda C y diseñada para operar en diversas plataformas, lo que le permite realizar un enlace de comunicaciones de alta velocidad de datos.

Los principales objetivos de esta carga son, por un lado demostrar el rendimiento del enlace en banda C a 5.8 GHz, ya que normalmente para las comunicaciones picosatélites se

trabaja en la banda de 2.4 GHz. Por otro lado, obtener un enlace con una elevada capacidad, de hasta 100 kbits / s con un mínimo consumo a bordo.

Se analizan dos casos de lanzamiento, uno corresponde a una órbita a una altura de 418 km y el otro a la altura de 500 km. El objetivo es que sea una órbita casi circular y casi helio-síncrona con un ángulo de elevación de 20°. [5]

En lo que se refiere al hardware utilizado tanto para la estación terrestre como para la carga útil del picosatélite todos ellos son componentes COTS (Commercial Off-The-Self) lo que significa que ofrecen un buen rendimiento a bajo precio. Además estos componentes tienen diversas ventajas entre las que se encuentran su coste reducido, fiabilidad, fácil mantenimiento, mayor calidad... etc.

En lo que respecta al satélite, el esquema de diseño se puede observar en la Figura 3.2 donde podemos apreciar los siguientes segmentos:

- Antena parche diseñada para trabajar en banda C.
- Amplificador de potencia.
- El transmisor utiliza el transceiver ML5805 de RFMD.

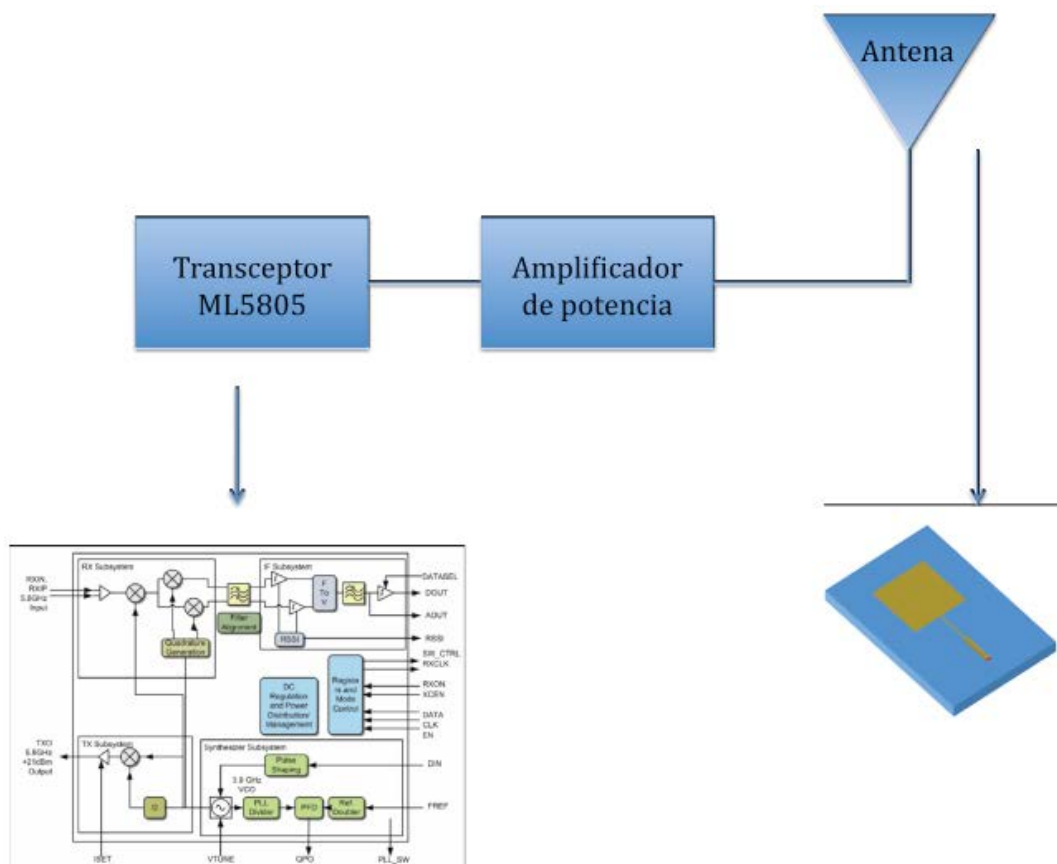


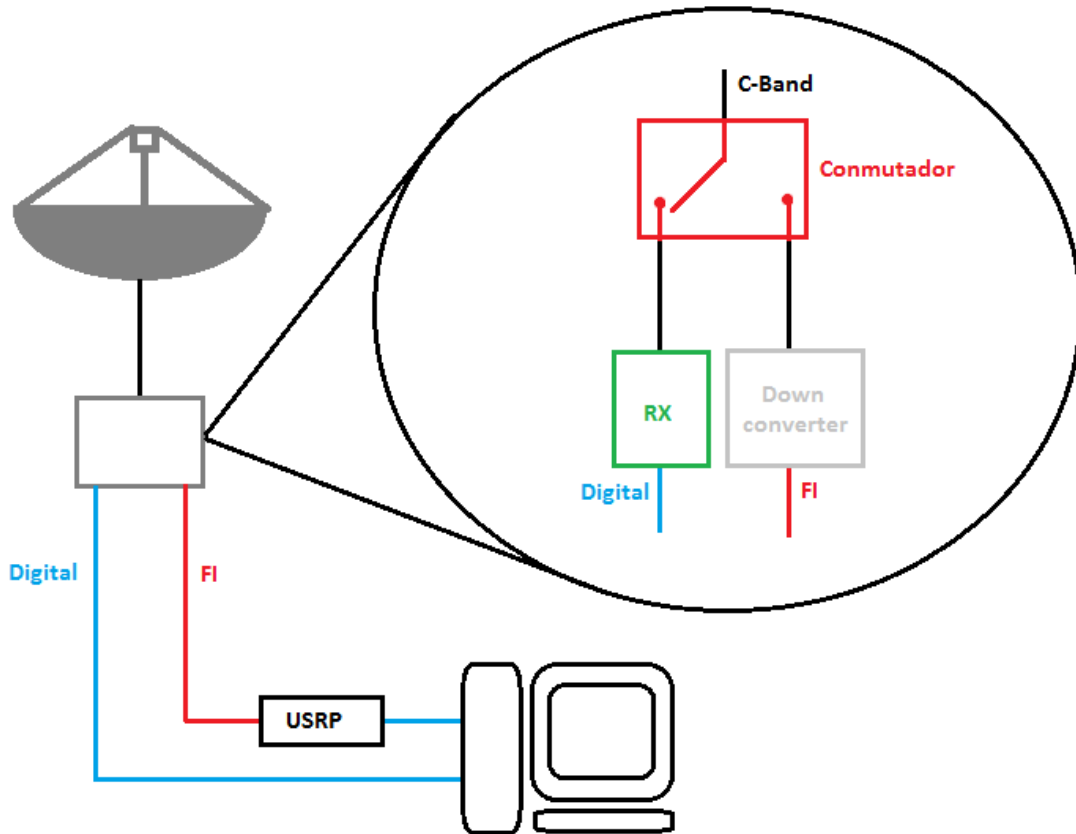
Figura 3.2: Esquema de diseño en el satélite.

Por otra parte, en la estación terrestre el esquema presenta dos bajantes, una en la que se encuentra el receptor que es el mismo dispositivo que el transmisor, un ML5805 de RFMD, con un amplificador de bajo ruido (LNA) de la marca KUHNE. En la otra bajante se halla el

convertidor de bajo ruido (LNC) de la marca KUHNE también para bajar la señal a FI y luego pasar por el USRP 2920 de National Instruments. [4]

En cuanto a la antena, se trata de un reflector comercial, que trabaja en banda C, sobre un rotor de seguimiento y las características son las siguientes:

- Frecuencia a 5.7 GHz.
- Polarización circular a izquierdas.
- Ganancia: 31.9 dB.
- Pérdidas de retorno > -28 dB.



**Figura 3.3: Esquema de diseño de la estación terrestre**

El LNC será el encargado de convertir la señal a FI. En concreto, el modelo utilizado para el diseño convierte la banda C a la banda de UHF de 400 a 700 MHz. En cuanto al USRP utilizado será descrito con detalle en un punto posterior de la memoria.

La potencia recibida dependerá de la órbita escogida, así para el caso de 418 km de altura, la potencia será -105.496 dBm. En comparación, para una altura de 500 km, la potencia recibida será -106.84 dBm.

La modulación empleada en la transmisión es 2FSK a una frecuencia central de 5831.5 MHz. La desviación en frecuencia tendrá una variación de  $\pm 680$  KHz. Por otra parte, la tasa binaria bruta no se sabe aún con certeza aunque no superará los 2.048 Mbps, probablemente sea la mitad. [3]



## 3.2 NI USRP 2920

NI USRP (Universal Software Radio Peripheral) es lo que se conoce como SDR (Software Defined Radio), es decir, un sistema de comunicación vía radio definido por software. Esto es, un sistema de radiocomunicaciones donde se utiliza el software para implementar los algoritmos necesarios para establecer una comunicación digital.

La empresa National Instruments utiliza el hardware NI USRP y el software LabVIEW para ofrecer transceptores rentables de SDR para poder realizar experimentos de comunicaciones en múltiples bandas de frecuencia. De esta forma, se ofrece una plataforma flexible y potente que es capaz de generar sistemas inalámbricos de radiofrecuencia y comunicación.

### 3.2.1 CARACTERÍSTICAS

NI USRP 2920 junto con otros tipos de transceptores de la categoría NI USRP-29xx están diseñados principalmente para la investigación así como para la enseñanza de las comunicaciones inalámbricas. USRP es una plataforma a un precio asequible y fácil de usar para diversas aplicaciones de RF.

Las características básicas de este dispositivo son las siguientes:

- Transceptor sintonizable RF.
- Frecuencia central ajustable desde 50 MHz hasta 2.2 GHz cubriendo así las bandas ISM, radio FM, GPS, GSM y radar.
- Capacidad de transmitir y recibir señales de RF de hasta 40 MHz de ancho de banda en tiempo real.
- ADC de alta velocidad y DAC para la transmisión de señales de banda base I/Q al ordenador a través de un enlace Gigabit Ethernet.
- Programable con el controlador NI-USRP en LabVIEW.

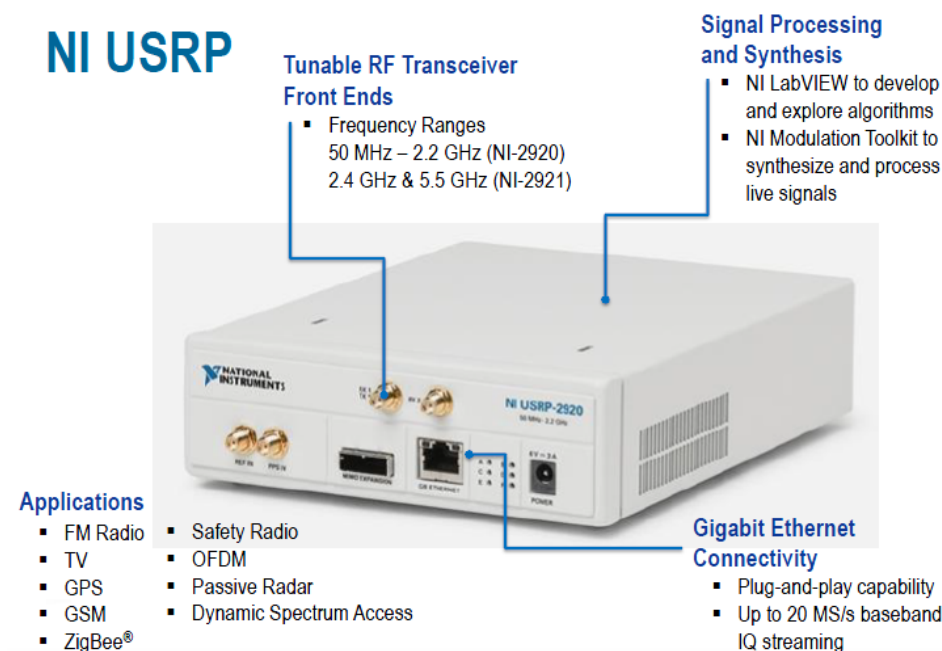
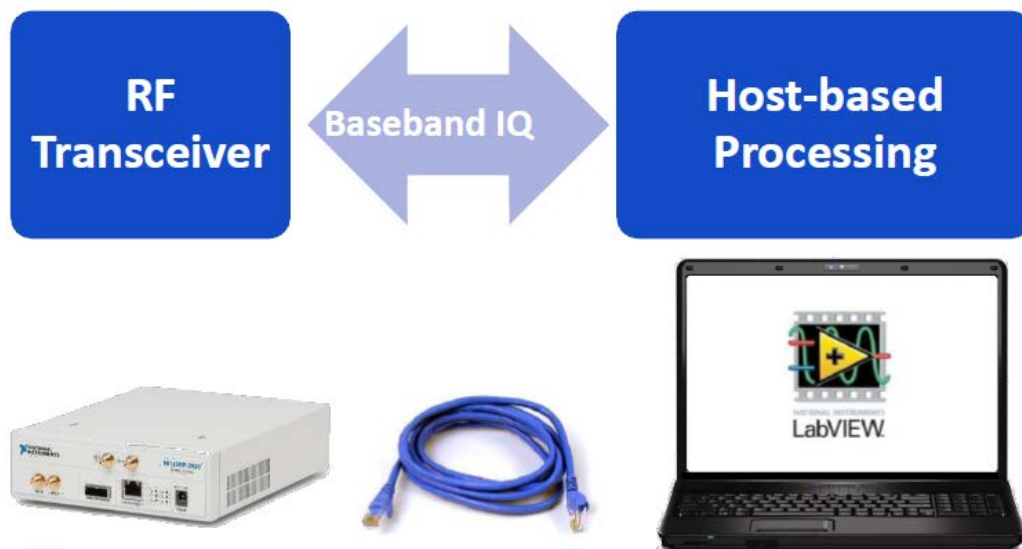


Figura 3.4: NI USRP

NI-USRP se sirve del software LabVIEW, así pues una vez se haya instalado éste en un ordenador lo único necesario es el kit hardware de NI USRP-2920 que contiene:

- NI USRP Software Suite DVD.
  - Controlador software NI-USRP.
  - LabVIEW Modulation Toolkit.
  - LabVIEW MathScript RT Module.
  - LabVIEW Digital Filter Design Toolkit.
- Transceptor radio NI USRP-2920.
- Adaptador de corriente.
- Cable Ethernet blindado.
- Cable SMA-SMA.
- Atenuador de 30 dB.

Una vez instalado el LabVIEW se realiza la conexión entre el USRP y el ordenador mediante el cable gigabit Ethernet para que el PC pueda controlar el USRP.



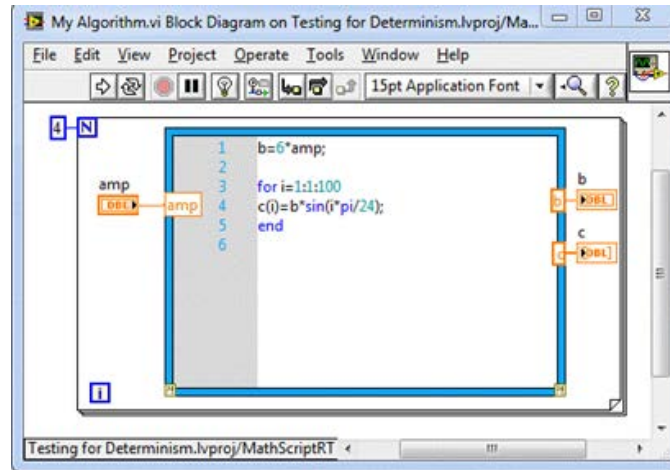
**Figura 3.5: Conexión NI-USRP con PC**

Mediante el software LabVIEW se puede interactuar con el hardware NI-USRP para el desarrollo de algoritmos de comunicación que procesan las señales recibidas así como sintetizan las señales para la transmisión. El controlador software NI-USRP proporciona funciones de LabVIEW para la configuración hardware/software con las herramientas para apertura/cierre de sesiones y operaciones de lectura/escritura.

LabVIEW Modulation Toolkit proporciona un rápido desarrollo de los sistemas de comunicaciones para la simulación o para operar en las señales asociadas al hardware NI-USRP. En lo que se refiere a la transmisión, se incluyen proyectos, llamados VIs en LabVIEW, que proporcionan funcionalidad para la generación de secuencias, la codificación de canal y modulación de banda base. Las funcionalidades para el receptor incluyen igualación, decodificación del canal... etc. También existen otra serie de funcionalidades que permiten la medida del BER, medidas en el dominio de la modulación y la visualización orientada a las

comunicaciones. Un sistema de comunicaciones puede ser simulado y analizado con degradaciones de canal para transmitir o recibir señales en vivo.

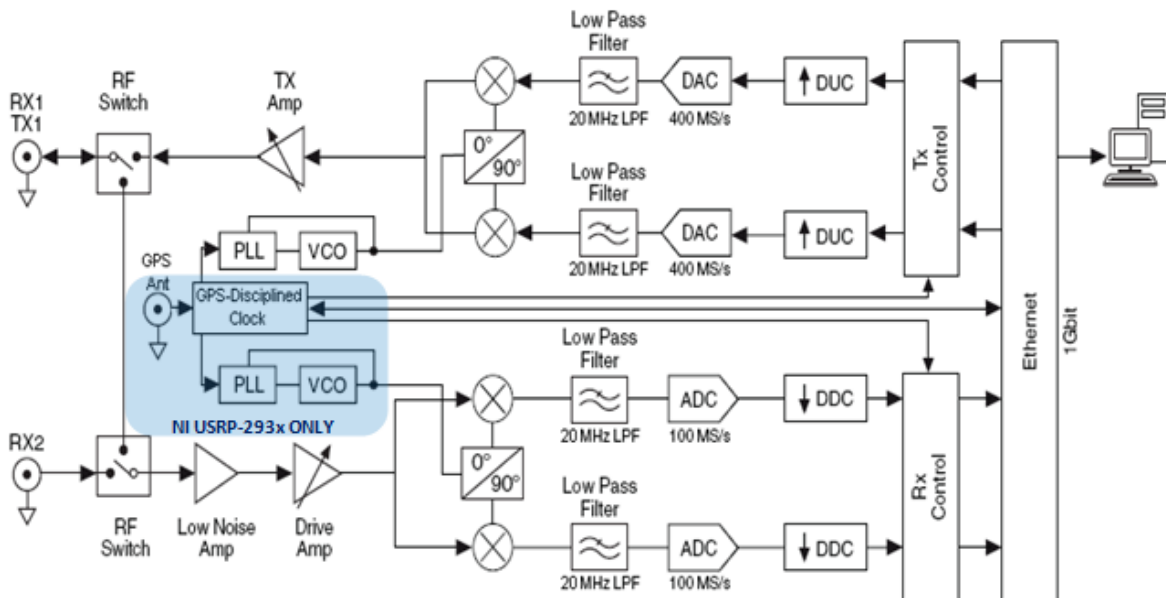
El LabVIEW MathScript RT Module permite utilizar la sintaxis de los archivos .m en un diagrama LabVIEW para combinar modelos de programación gráficos y de texto. Además, permite diseñar e implementar algoritmos de comunicaciones utilizando la sintaxis de los archivos .m.



**Figura 3.6: Ejemplo MathScript RT**

### 3.2.2 HARDWARE

Los USRP son transceptores radio de RF programables por software diseñados para la investigación de las comunicaciones inalámbricas, como se ha indicado anteriormente. Cuando el USRP se conecta a un ordenador funciona como un SDR con capacidades de procesamiento de señales digitales basados en host. Cada dispositivo USRP es capaz de proporcionar una transmisión independiente y un canal de recepción que puede trabajar en modo full-dúplex en algunas configuraciones hardware. En la Figura 3.7 se puede apreciar el diagrama de bloques del sistema USRP. [6]



**Figura 3.7: Diagrama de bloques de USRP**

### 3.2.3 TRANSMISOR USRP

Cuando el USRP actúa como transmisor recibe una forma de onda del ordenador con 16 bits de resolución muestreada de hasta 25Mmuestras por segundo. Esta señal se sube a radiofrecuencia (RF) antes de enviarla a un amplificador para luego ser transmitida por el aire.

|   |   |
|---|---|
| Rango de frecuencias  | 50 MHz a 2.2 GHz  |
| Paso de frecuencia  | <1 kHz  |
| Potencia máxima a la salida ( $P_{out}$ )   |   |
| <ul style="list-style-type: none"> <li>• 50 MHz a 1.2 GHz</li> <li>• 1.2 GHz a 2.2 GHz</li> </ul>             | <ul style="list-style-type: none"> <li>• 50 mW a 100 mW (17 dBm a 20 dBm)</li> <li>• 30 mW a 70 mW (15 dBm a 18 dBm)</li> </ul> |
| Rango de ganancia   | 0 dB a 31 dB  |
| Paso de ganancia  | 1 dB  |
| Precisión de frecuencia   | 2.5 ppm   |
| Ancho de banda máximo instantáneo   |   |
| <ul style="list-style-type: none"> <li>• 16-bit ancho de muestra</li> <li>• 8-bit ancho de muestra</li> </ul> | <ul style="list-style-type: none"> <li>• 25 MHz</li> <li>• 40 MHz</li> </ul>  |
| Velocidad máxima de muestreo I/Q  |   |
| <ul style="list-style-type: none"> <li>• 16-bit ancho de muestra</li> <li>• 8-bit ancho de muestra</li> </ul> | <ul style="list-style-type: none"> <li>• 25 MS/s</li> <li>• 50 MS/s</li> </ul>  |
| DAC   | 2 canales, 400 MS/s, 16 bit   |
| DAC SFDR  | 80 dB   |

**Tabla 3.1: Transmisor USRP**

### 3.2.4 RECEPTOR USRP

El USRP también es capaz de recibir señales y lo hace de la siguiente manera. La señal recibida se mezcla con la frecuencia portadora deseada con el objetivo de bajar en frecuencia la señal y convertirla en una señal banda base IQ compleja muestreada a 100 Mmuestras por segundo. Seguidamente, la señal digital se muestrea a una tasa especificada por el usuario y se envía al ordenador para su procesamiento.

Cuando el ADC de las muestras del NI USRP llega a la tasa máxima del digitalizador (100 Mmuestras/seg) se puede adquirir un ancho de banda de 20 MHz. El muestreo con una tasa tan alta con alta resolución (14 bits) produce una gran cantidad de datos. Sin embargo, para adquirir una señal con menor ancho de banda, es decir, una señal de banda estrecha, es suficiente muestrear a una velocidad doble que el ancho de banda de la señal (teorema de Nyquist para señales reales).

La operación de diezmado es el proceso de reducción de la tasa de muestreo en un tiempo discreto de la señal. En la práctica, esto significa la aplicación de un filtro anti-aliasing y perder algunas muestras. El NI USRP cuenta con un DDC (Digital Down-Converter) que le permite realizar el diezmado en hardware en lugar de hacerlo en el software, lo que suele resultar más lento.

Mediante el uso del DDC, la USRP es capaz de adquirir señales de banda ancha a una tasa de muestreo menor que la tasa máxima del digitalizador (100 MS/s), lo que reduce la memoria requerida para almacenar la forma de onda. [7]

|   |  |
|---|--|
| Rango de frecuencias  | 50 MHz a 2.2 GHz   |
| Paso de frecuencia  | <1 kHz   |
| Rango de ganancia   | 0 dB a 31.5 dB   |
| Paso de ganancia  | 0.5 dB   |
| Potencia máxima a la entrada ( $P_{in}$ )   | 0 dBm  |
| Figura de ruido   | 5 dB a 7 dB  |
| Precisión de frecuencia   | 2.5 ppm  |
| Ancho de banda máximo instantáneo   |  |
| <ul style="list-style-type: none"> <li>• 16-bit ancho de muestra</li> <li>• 8-bit ancho de muestra</li> </ul> | <ul style="list-style-type: none"> <li>• 20 MHz</li> <li>• 40 MHz</li> </ul>   |
| Velocidad máxima de muestreo I/Q  |  |
| <ul style="list-style-type: none"> <li>• 16-bit ancho de muestra</li> <li>• 8-bit ancho de muestra</li> </ul> | <ul style="list-style-type: none"> <li>• 25 MS/s</li> <li>• 50 MS/s</li> </ul> |
| ADC   | 2 canales, 100 MS/s, 14 bit  |
| ADC SFDR  | 88 dB  |

**Tabla 3.2: Receptor USRP**

### 3.3 LABVIEW

El LabVIEW es un lenguaje de programación gráfico llamado G, para crear código con el uso de diagramas de bloques, lo que le hace muy adecuado para el diseño y la implementación de algoritmos de comunicaciones.

El software LabVIEW utiliza el controlador NI USRP para especificar la configuración del hardware NI USRP explicado anteriormente y para enviar y recibir en banda base con el formato adecuado de datos I/Q. Además, con los módulos adicionales del kit NI USRP (MathScript RT Module y Modulation Toolkit) se proporcionan funciones específicas de las comunicaciones así como la opción de utilizar la sintaxis de los archivos .m. [7]

Este lenguaje de programación presenta diversas ventajas con respecto a otros:

- Se reduce el tiempo de desarrollo de las aplicaciones puesto que es muy fácil de aprender e intuitivo.
- Facilidad en la interfaz con instrumentos externos, especialmente los fabricados por National Instruments.
- Permite que los usuarios creen soluciones completas y complejas.
- Presenta librerías de automatización, procesamiento digital, para la adquisición de dato, análisis, etc.
- Posibilidad de implementar aplicaciones escritas en otros lenguajes.
- Se logra una mayor velocidad de ejecución debido al compilador gráfico del que está dotado el sistema.

#### 3.3.1 ENTORNO DE LABVIEW

LabVIEW es un programa que originalmente estaba orientado a aplicaciones de control de equipos electrónicos utilizados en el desarrollo de sistemas de instrumentación, lo que se conoce como instrumentación virtual. Los programas creados en LabVIEW se guardan en ficheros llamados VI (Virtual Instrument) ya que su apariencia y funcionamiento imitan a los de

un instrumento real. Todos los VI tienen dos partes, una interactiva por el usuario y otra de código fuente y aceptan parámetros de otros VIs.

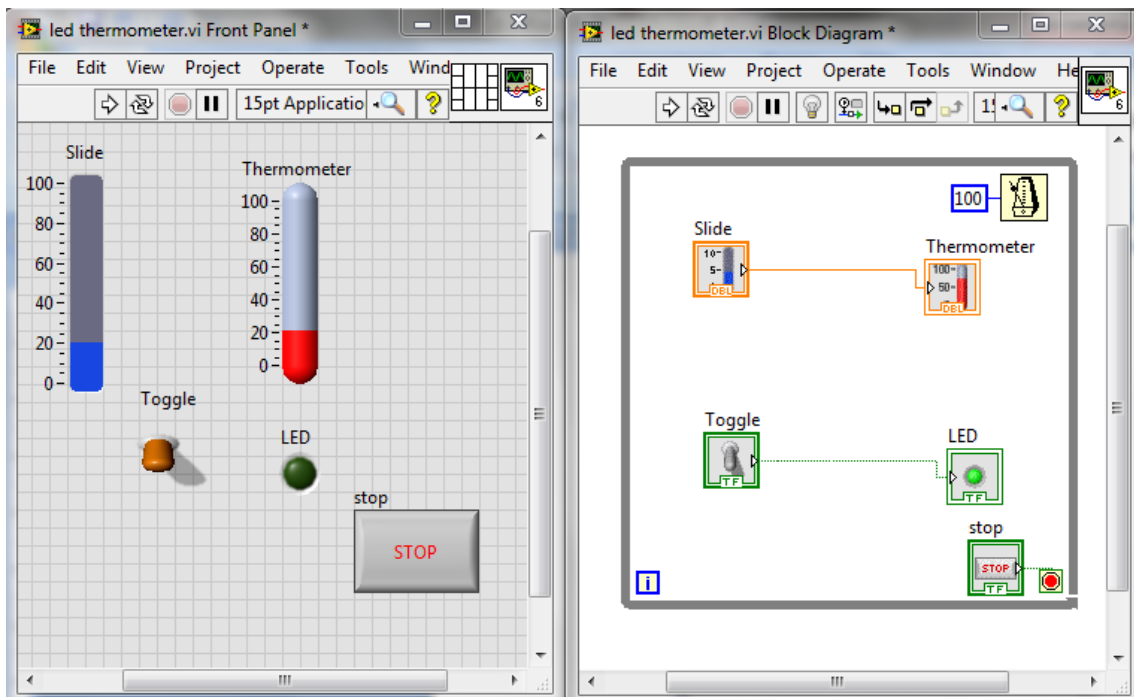
Los VIs contienen dos interfaces: Panel Frontal y Diagrama de Bloques. Éstas cuentan con paletas que contienen los objetos necesarios para implementar y desarrollar tareas.

Por un lado, el Panel Frontal es la interfaz gráfica con el usuario. En esta interfaz, el programa muestra las salidas que vienen determinadas por las entradas que introduce el usuario al ejecutar la aplicación.

El Panel Frontal está formado por pulsadores, gráficas, botones... etc. Cada uno puede estar definido o bien como un control o como un indicador. Los primeros son objetos que se utilizan para introducir datos al programa y el usuario es el responsable de su manipulación para darle los valores deseados. Son por tanto, las variables de entrada. Los indicadores por su parte, se utilizan para presentar los resultados, son variables de salida por lo que el usuario no puede modificarlos.

Por otro lado, el Diagrama de Bloques es donde se realiza la programación, es decir, contiene el código del programa para realizar el procesamiento de las entradas y salidas creadas en el Panel Frontal. En el Diagrama de Bloques se incluyen funciones y estructuras que, en el lenguaje G son nodos elementales. Son totalmente análogos a los operadores o librerías de funciones de los lenguajes convencionales. Las funciones y SubVIs realizan tareas concretas. Mientras que por otra parte, las estructuras y cables determinan el flujo de los datos en el programa.

El Panel Frontal y el Diagrama de Bloques se conectan a través de los terminales. Es decir, los controles e indicadores que se colocan en el Panel Frontal, en el Diagrama de Bloques se identifican mediante los terminales. [8]




**Figura 3.8: Panel Frontal y Diagrama de Bloques**


En LabVIEW, para crear y modificar tanto el Diagrama de Bloques como el Panel Frontal existen una serie de paletas que proporcionan las herramientas necesarias para realizar el programa.


La paleta de herramientas (Tools palette) se utiliza tanto en el Diagrama de Bloques como en el Panel Frontal y contiene los instrumentos necesarios realizar distintas acciones sobre los objetos. Las opciones que se incluyen son:


 *Automatic Tool Selection:* activado selecciona la herramienta automáticamente.


 *Operate Value:* cambia valores en los controles del Panel Frontal.


 *Position/ Size / Select:* selecciona, mueve y redimensiona objetos.


 *Edit Text:* crea y edita textos.

 *Connect Wire:* une elementos con cables en el Diagrama de Bloques.


 *Object Shortcut Menu:* despliega el menú asociado a un objeto.

 *Scroll Window:* mueve el contenido de las ventanas sin usar barras de desplazamiento.

 *Set/Clear Breakpoint:* fija puntos de interrupción en la ejecución del programa.

 *Probe Data:* crea puntos de prueba en los cables donde se visualiza el valor del dato que fluye por el cable en cada instante.

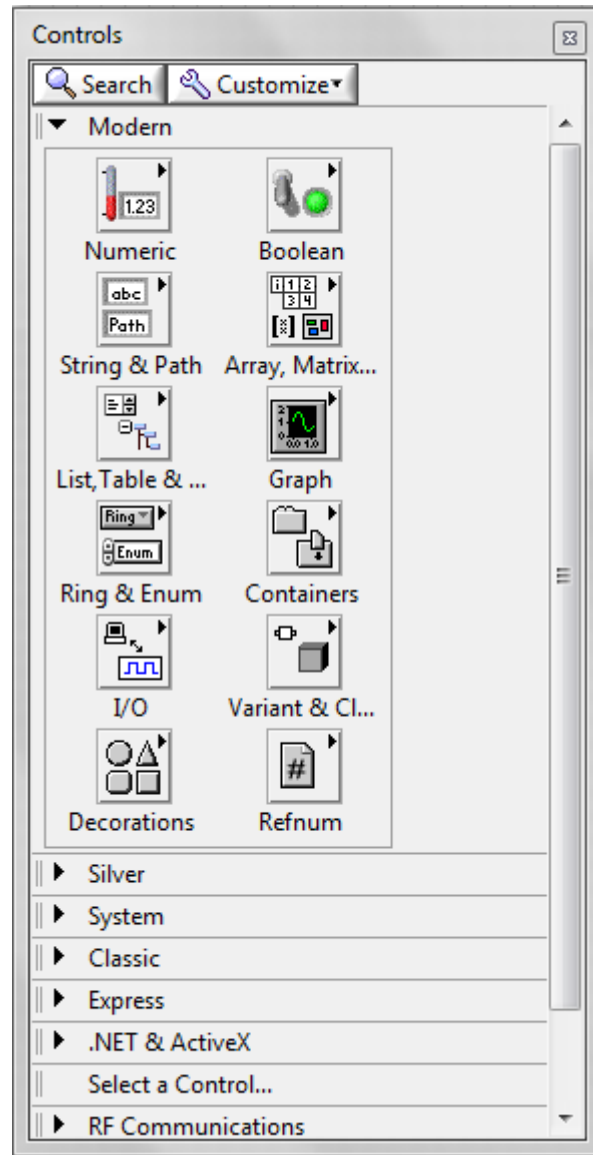
 *Get Color:* obtiene el valor del color de un elemento.

 *Set Color:* colorea un objeto o el fondo.

Por otra parte, la paleta de controles (Controls Palette) únicamente se usa en el Panel Frontal y contiene los objetos necesarios para crear una interface de entrada y salida de datos.

En la Figura 3.9 se muestra la paleta de controles donde cada submenú de la paleta contiene indicadores y controles característicos del tipo de objeto.

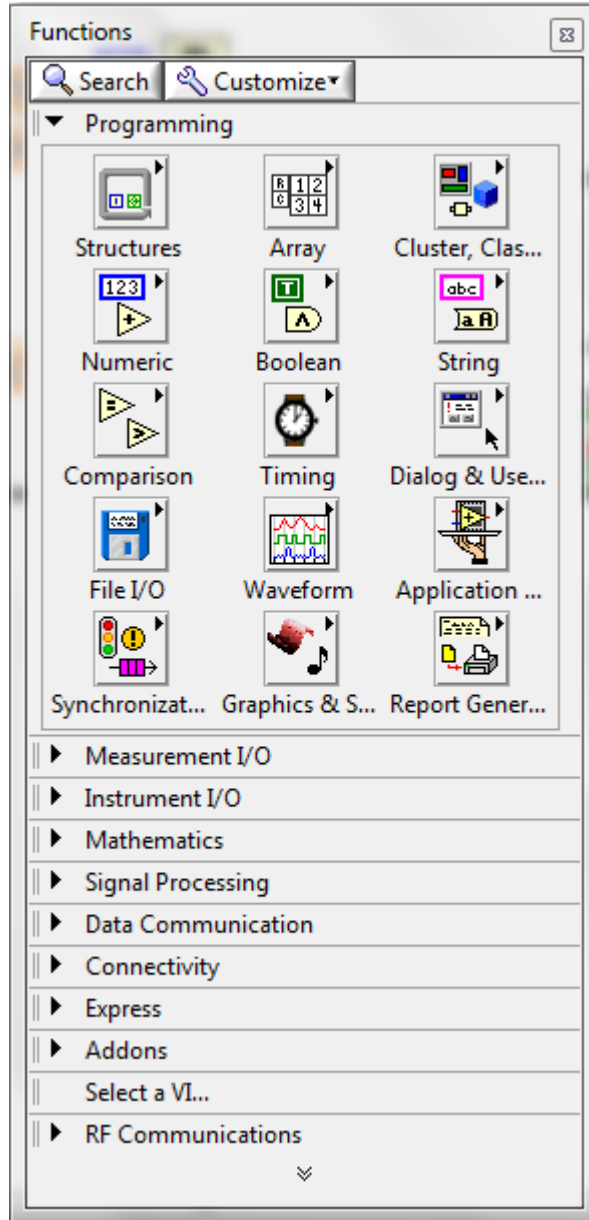
Además, es importante destacar que existen submenús que se corresponden a *toolkits* que únicamente aparecen una vez han sido instalados. Estos *toolkits* son herramientas adicionales de software que se han desarrollado con unos fines concretos y especializados que se suministran como productos por separado.



**Figura 3.9: Paleta de controles**



Por otro lado la paleta de funciones (Functions Palette) se utiliza en el diseño del Diagrama de Bloques, con ella se puede acceder a las distintas funciones, subVI y estructuras disponibles. Es importante añadir que los submenús que se corresponden con los *toolkits* no estarán presentes hasta que se hayan instalado. [9]

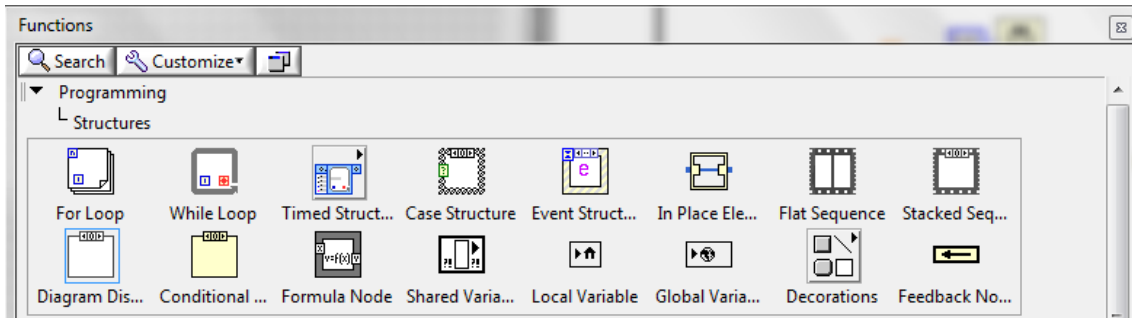


**Figura 3.10: Paleta de funciones**

Cada submenú de la paleta de funciones contiene distintas funcionalidades para las diversas tareas que se pueden realizar.

Uno de los submenús más importantes en este aspecto es el de estructuras que son las que controlan el flujo del programa mediante la ejecución de bucles, secuenciación de acciones, etc.

Cada estructura ejecuta su subdiagrama en función de las reglas que rigen su comportamiento particular. Un subdiagrama es un conjunto de nodos, cables y terminales que constituyen la estructura. Hay estructuras que únicamente tienen un subdiagrama como el *For Loop* y el *While Loop*. Mientras que hay otras que pueden tener múltiples subdiagramas superpuestos como en el caso del *Case Structure* y el *Sequence Structure*. Estas estructuras tienen un comportamiento similar al de otro lenguaje de programación con la salvedad de que en LabVIEW se utiliza un lenguaje de programación gráfico. [10]



**Figura 3.11: Estructuras**

### 3.3.2 PROGRAMACIÓN

En LabVIEW es recomendable empezar a programar a partir del Panel Frontal aunque realmente donde se realizará toda la escritura del programa será en el Diagrama de Bloques.

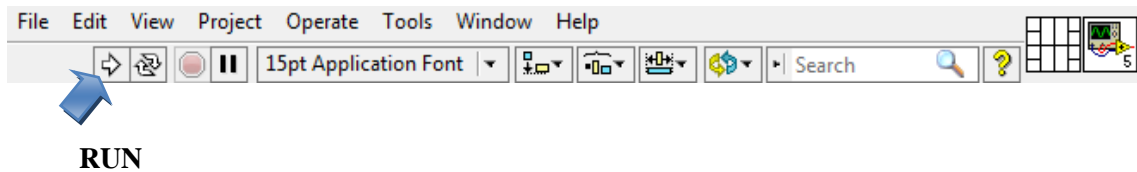
En primer lugar, se deben definir las entradas y salidas que tendrá el programa. Es decir, en la paleta de controles se deberán seleccionar las entradas que introducirá el usuario (controles) y las salidas que se presentarán en el programa (indicadores).

Cuando ya están definidas las entradas y salidas del programa se empieza a programar propiamente en el Diagrama de Bloques. Aquí es donde se introducen las funciones, estructuras, etc. que sean necesarias para el desarrollo del programa y que deberán estar correctamente cableadas a los terminales.

Además, para que resulte más fácil la programación existe la ventana *Context Help* donde se muestra la información básica de los objetos de LabVIEW cuando se sitúa el cursor encima de cada objeto. De esta forma, se muestran los iconos de las funciones, controles, subVIs e indicadores con los cables conectados a cada terminal. Cuando se mueve el cursor encima de la caja de opciones el *Context Help* muestra una descripción de estas opciones. [11]

### 3.3.3 EJECUCIÓN DE UN VI

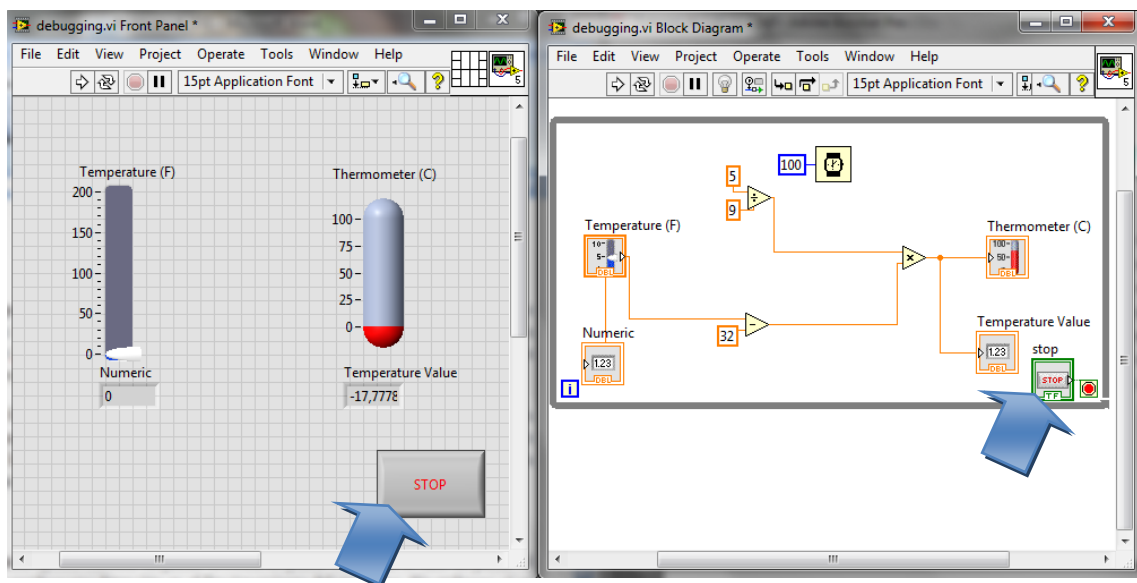
Una vez acabada la programación del VI se debe ejecutar para comprobar su correcto funcionamiento. Para esto, será necesario utilizar la ventana del Panel Frontal, dónde se pulsa el botón *Run*, que se encuentra en la barra de herramientas. En la Figura 3.12 se puede observar dónde se encuentra dicho botón para comenzar a ejecutar el programa.



**Figura 3.12: Ejecución de un VI- RUN**

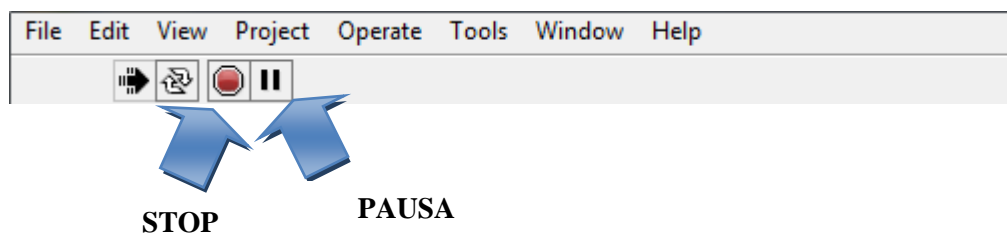
Una vez pulsado el botón *Run* el programa comenzará a ejecutarse, sin embargo, sólo se ejecutará una vez. En el caso de que se desee una ejecución continua se pulsa el botón situado a la derecha del de *Run* (*Continuous Run*). Si cuando se está produciendo la ejecución continua del programa se vuelve a pulsar dicho botón, se parará el VI.

Hay dos formas posibles de finalizar un programa. La primera de ellas consiste en utilizar un botón en el Panel Frontal del VI de tal forma que al pulsarlo se interrumpa el bucle de ejecución de la aplicación, es decir, un botón creado por el programador dentro de la estructura interna del bucle.



**Figura 3.13: Botón para la finalización de un VI**

Una segunda opción consiste en pulsar el botón de pausa o stop. En el caso de pulsar el botón de pausa se produce una detención en el funcionamiento del programa. Sin embargo, si se pulsa stop la ejecución del programa finaliza inmediatamente.

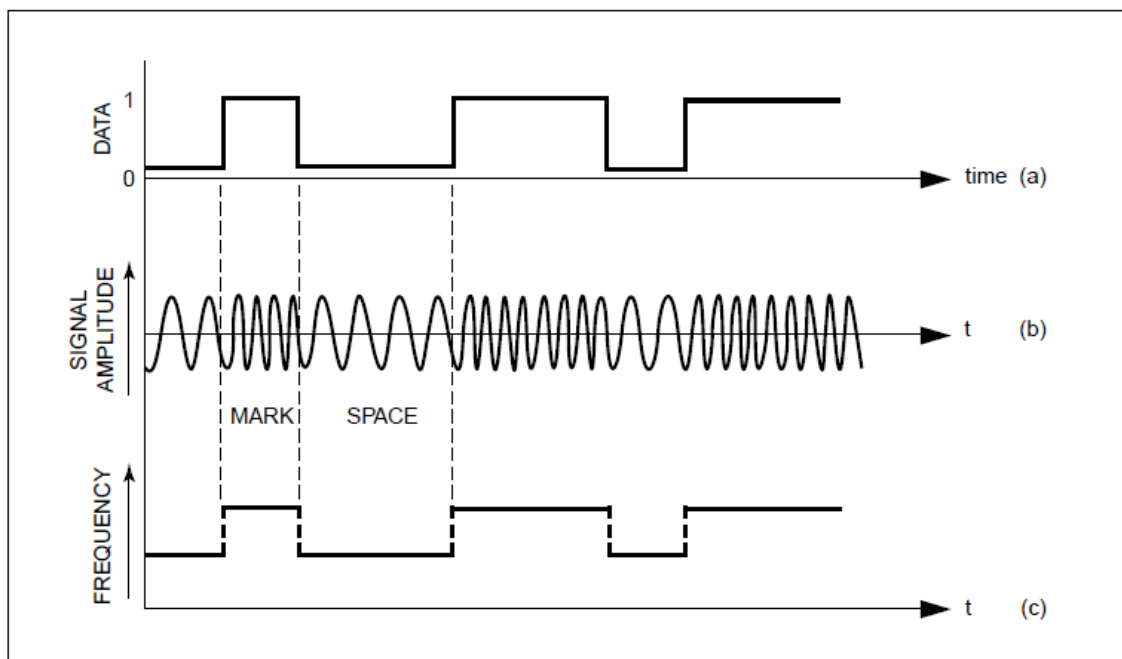


**Figura 3.14: Botón de pausa y stop**

### 3.4 MODULACIÓN FSK

La modulación FSK (Frequency Shift Keying) también conocida como FSK binaria es la forma más utilizada de modulación digital en el espectro radio de alta frecuencia. Se trata de una técnica de transmisión digital donde la señal moduladora es un tren de pulsos binarios que únicamente varía entre dos valores de tensión discretos. [14]

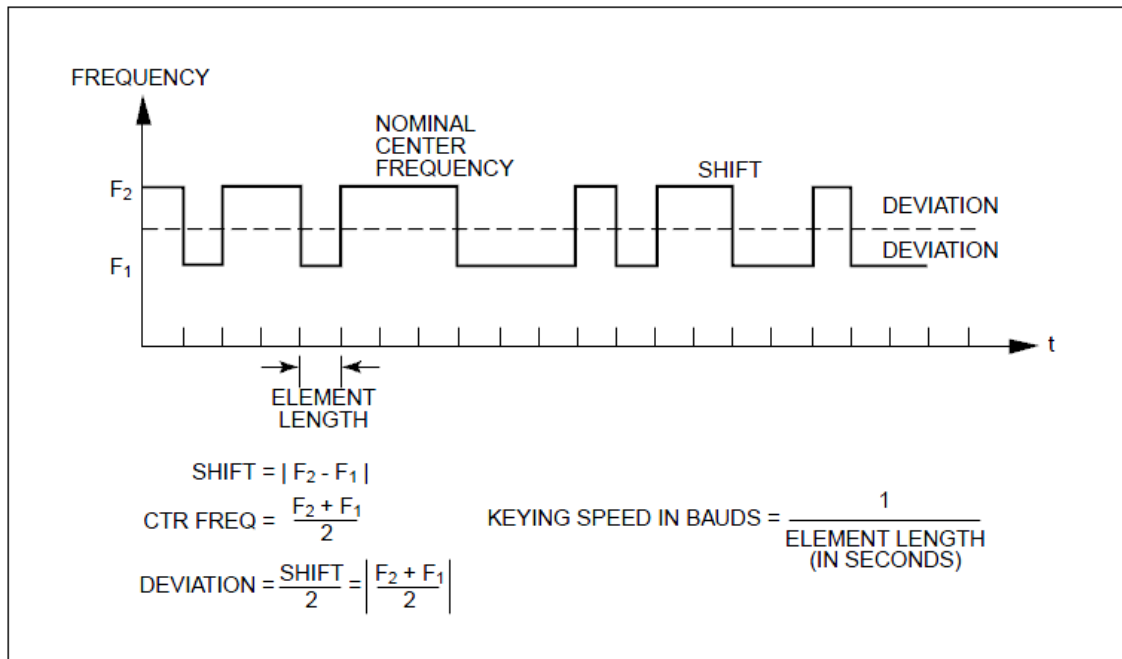
En este tipo de modulación, la señal moduladora hace variar la frecuencia de portadora, así la señal resultante codifica la información asociándola a distintos valores de frecuencia. Una frecuencia se designa como la frecuencia de “mark” y la otra como la de “space” y se corresponden a uno binario y cero, respectivamente. Por convenio, “mark” corresponde a las frecuencias de radio mayores. En la Figura 3.15 podemos observar la relación entre los datos y la señal modulada.



**Figura 3.15: Modulación FSK**

La modulación FSK tiene diversos parámetros para describir la señal. Entre ellos, se encuentra la longitud del elemento que es básicamente la duración mínima de una condición de “mark” o “space”. Los valores típicos de la longitud del elemento oscilan entre 5 y 22 milisegundos aunque se han utilizado longitudes menores a 1 microsegundo y mayores a 1 segundo. Otra forma de especificar la longitud del elemento es en términos de velocidad (en baudios), lo que se conoce como “keying speed” y se define como la inversa de la longitud del elemento.

Otro de los parámetros importantes en la modulación FSK es la desviación de frecuencia que es igual al valor absoluto de la diferencia entre la frecuencia central y las frecuencias de “mark” o “space”. Además es un valor constante del que depende el ancho de banda de la señal modulada. [13]



**Figura 3.16: Parámetros de la modulación FSK**

Esta modulación se fundamenta en la idea de asignar una frecuencia distinta a cada estado de la señal de datos. Para ello se distinguen dos tipos de modulación FSK: coherente y no coherente.

Por un lado, en la FSK coherente la fase de la señal se mantiene en el instante de asignar la frecuencia. Este método se denomina Modulación de Frecuencia Coherente y su obtención se realiza a partir de un único oscilador sobre el que se efectúa un cambio de frecuencias, con arreglo a los datos.

Por otro lado, en la FSK no coherente la fase de la señal no permanece en el momento de asignar la frecuencia. Es decir, hay un cambio de fase en la señal cuando se produce una conmutación de frecuencia, esto significa que cuando se produce un cambio de la frecuencia que corresponde a “space” a la frecuencia “mark”, la fase de la primera no permanece. De igual manera ocurre en el caso inverso. Este método es conocido como Modulación de Frecuencia No Coherente y se obtiene mediante dos osciladores independientes. Este tipo de modulación resulta más simple de generar por lo que se utiliza en la mayoría de las transmisiones FSK.

Para el caso de un tipo de modulación FSK binaria, la expresión general es:

$$V(t) = V_c \times \cos \left[ \left( w_c + vm(t) \times \frac{D_w}{2} \right) t \right] \quad (1)$$

Siendo:

- $V(t)$  : forma de onda FSK binaria.
- $V_c$  : amplitud de pico de la portadora no modulada.
- $w_c$  : frecuencia de la portadora en radianes.
- $vm(t)$  : señal moduladora digital binaria.
- $D_w$  : cambio de frecuencia de salida en radianes. [12]

Los métodos de demodulación de FSK se dividen en dos categorías principales: demoduladores detectores de FM y demoduladores de filtro-tipo FSK.

### 3.4.1 DEMODULADORES DETECTORES DE FM

Este método de demodulación consiste en tratar la señal FSK como una señal FM con modulación binaria. En la Figura 3.17 se muestra un diagrama de bloques para un tipo de detector FM que elimina la interferencia de fuera de banda y seguidamente está limitado para eliminar la interferencia AM. La señal limitada que es detectada en el discriminador de FM produce una salida positiva para una condición de “mark” y una salida negativa para una condición de “space”. Se utiliza un filtro paso bajo para eliminar las componentes de ruido de la señal que se encuentran a frecuencias superiores a la velocidad de transmisión. Finalmente, el circuito de decisión hace que los voltajes positivos sean unos binarios y los voltajes negativos sean ceros.

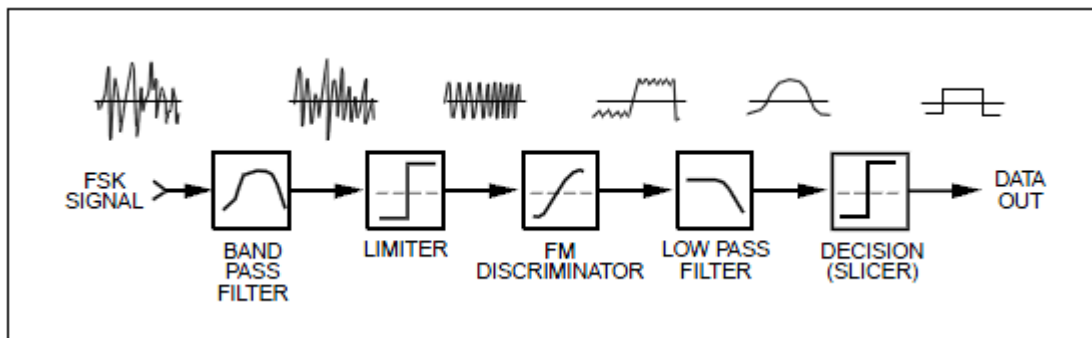


Figura 3.17: Demodulador detector de FM

Los detectores de FM no son óptimos en el sentido en que su rendimiento es más pobre que lo que predice como posible la teoría de detección de señal. Para comprobar que esto es así es preciso examinar el espectro de la señal FSK. En la Figura 3.18 se observa que prácticamente toda la energía de la señal de los tonos “mark” y “space” se encuentra en un ancho de banda igual al doble de la velocidad en baudios, centrada alrededor de las frecuencias de “mark” y “space” respectivamente.

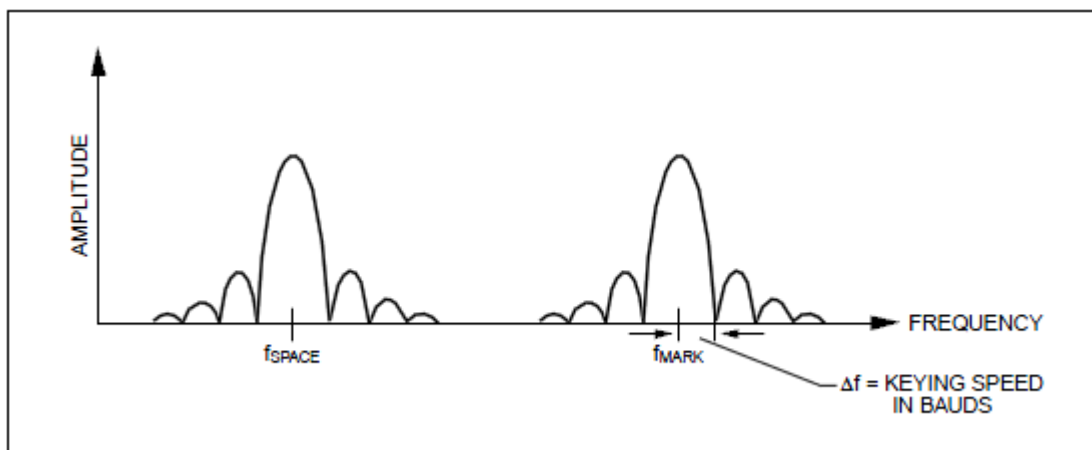
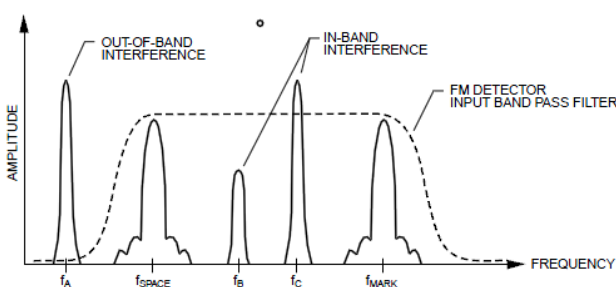


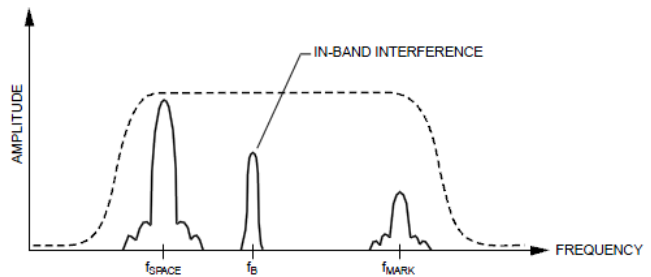
Figura 3.18: Espectro de la señal FSK

En la Figura 3.19 se muestra el espectro de la señal FSK con interferencias. La señal interferente  $f_A$  se rechaza al encontrarse fuera de la banda de paso del filtro de entrada del detector de FM. La señal interferente  $f_B$  es rechazada debido a la acción del limitador de FM. Sin embargo, la señal interferente  $f_C$  es más fuerte y se encuentra dentro de la banda de paso del filtro lo que impide la demodulación de la señal deseada.

En el caso de la Figura 3.20, a pesar de que la señal de “space” es más fuerte que la interferencia en banda, la amplitud de la señal “mark” es menor que la interferencia en banda debido a que ha sido reducida a causa del desvanecimiento por múltiples trayectos. Por consiguiente, en las transmisiones de la señal “mark” la demodulación se verá gravemente afectada.



**Figura 3.19: Interferencias en señal FSK**



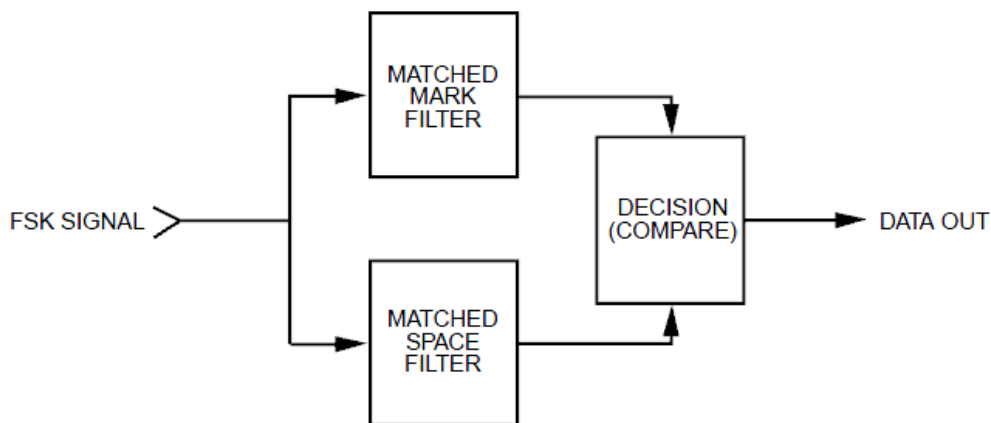
**Figura 3.20: Desvanecimiento selectivo**

Otro efecto a tener en cuenta que es de la interferencia de banda ancha de ruido blanco. Si la envolvente de ruido excede la señal FSK se evita la demodulación.

### 3.4.2 DEMODULADORES DE FILTRO-TIPO FSK

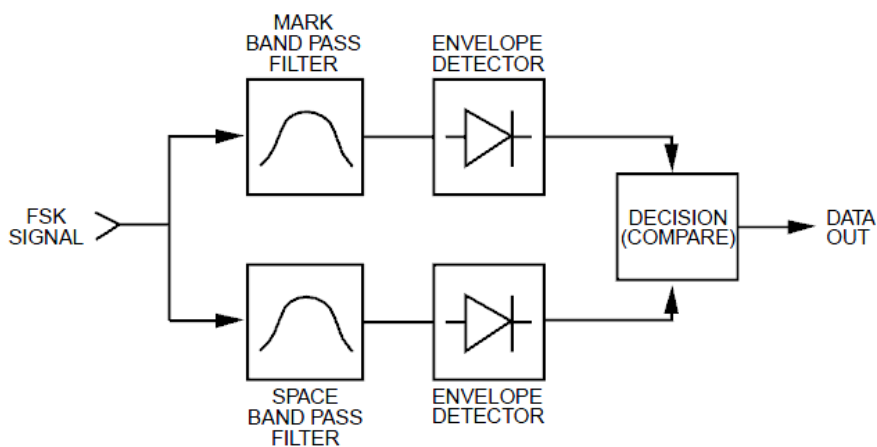
En este método de demodulación se pretende minimizar el error en el demodulador optimizando los parámetros de la señal FSK en la configuración del demodulador. Aunque es importante tener en cuenta que el diseño adecuado de un filtro no solo depende de los parámetros de la señal sino también de la naturaleza de la interferencia de la señal.

En la Figura 3.21 se muestra un diagrama de bloques de un demodulador para FSK coherente. En el bloque de decisión se compara la salida de los filtros y si la salida del filtro de “mark” es mayor que el del filtro de “space” se toma la decisión de que se transmitió una señal de “mark”. En el caso de que la salida del filtro “space” sea mayor se toma la decisión de que se transmitió “space”. Un demodulador de filtro adaptado es óptimo debido a que sus filtros están adaptados a la señal transmitida de modo que su respuesta a la señal deseada se maximiza con respecto a su respuesta de ruido.



**Figura 3.21: Demodulador de filtro adaptado FSK**

Debido a que FSK no coherente es mucho más común que la FSK coherente, es necesario tener un tipo de demodulador que no dependa de la información de la fase. Una demodulación óptima de FSK no coherente se puede lograr por detección de envolvente de las salidas de los filtros de la señal en un demodulador. La Figura 3.22 es un ejemplo de este tipo de demoduladores.



**Figura 3.22: Demodulador de FSK no coherente**

Las salidas de los filtros de “mark” y “space” se detectan y luego se comparan mediante el circuito de decisión para comprobar cuál es de mayor magnitud. Con la forma del filtro adecuada, el rendimiento del demodulador se acerca al óptimo teórico para FSK no coherente. Para el caso de un entorno con interferencia de ruido blanco el filtro adecuado es aquel que tiene la misma forma espectral de la señal transmitida. Sin embargo, en la mayoría de los entornos no se puede asumir que la interferencia es de ruido blanco.

En el caso de FSK no coherente con interferencias de ruido no blanco, el problema del diseño es más complejo. Existen dos tipos de interferencias que son la interferencia de canal adyacente y la interferencia CW. Esta clase de interferencias son impredecibles y para

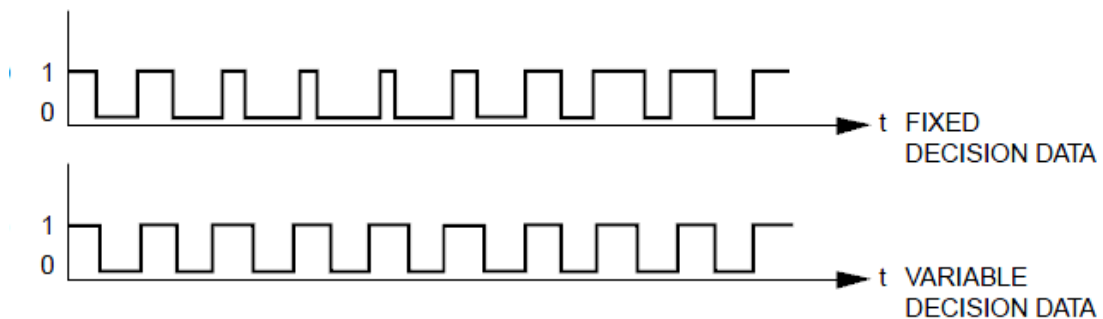


minimizar su efecto se utiliza un filtro paso banda con faldas de atenuación relativamente abruptas y sin lóbulos laterales. Es deseable que el filtro funcione bien además con el ruido blanco.

Para cada forma de filtro existe un ancho de banda óptimo. En general, si el ancho de banda del filtro es demasiado ancho se incluirá exceso de energía de ruido. En el caso de que el ancho de banda del filtro sea demasiado estrecho, los elementos de la señal consecutivos interferirán entre sí, produciéndose de esta manera interferencia entre símbolos (ISI).

El módulo de decisión de la Figura 3.17 puede ser realizado por distintas configuraciones de circuito. El más simple consiste en un comparador de tensión que decide que la señal de “mark” se recibió si la salida del detector de “mark” es mayor que el “ruido” de salida del detector de “space”. Este es el circuito de decisión óptimo teóricamente cuando la amplitud de las señales transmitidas son iguales y sin desvanecimiento selectivo. Si la señal está sujeta a desvanecimiento selectivo (estadísticas de amplitud de Rayleigh), el rendimiento del demodulador se ve gravemente degradado.

Otro tipo de configuración es la de suma, el circuito de decisión compara la suma diferencial con una referencia de tensión, que determina la decisión de umbral. Cuando la suma es mayor a dicho umbral se toma una decisión “mark” y cuando la suma es menor que el umbral se toma una decisión “space”. Cuando la señal de tiene amplitudes iguales de “mark” y “space” el umbral óptimo es una referencia de voltaje cero. Sin embargo, durante las condiciones de desvanecimiento selectivo es necesario variar dinámicamente el umbral para alcanzar así la demodulación óptima [13]. En la Figura 3.23 se aprecia el efecto de la señal con umbrales fijos y variables.



**Figura 3.23: Efecto de la señal con umbrales fijos y variables**

### 3.5 TRANSCEIVER ML5805

Los dispositivos utilizados para la transmisión y la recepción serán transceptores. De esta forma se podrá utilizar el mismo dispositivo para la emisión (desde el satélite en modo transmisión) como para la recepción en la estación terrestre (en modo recepción), lo cual facilitará el enlace.

El transceptor utilizado será el modelo ML5805 de la marca RFMD ya que es el dispositivo que se encuentra en el mercado que más se ajusta a las necesidades del presente proyecto. Este tipo de transceptores consumen muy poca potencia, requisito necesario para ir a bordo del satélite dadas las elevadas pérdidas del enlace y resisten temperaturas suficientemente extremas para el entorno de trabajo. Son capaces de trabajar a diferentes tasas de velocidad y soportan DSSS (Direct-Sequence Spread Spectrum), una técnica de modulación mediante la cual se puede reducir aún más la tasa de transmisión consiguiendo incrementos de ganancia sin las cuales no se cumpliría el balance de potencias exigido.

El transceptor está integrado en un solo chip y funciona con un tipo de modulación FSK. Está pensado para una variedad de aplicaciones (teléfonos inalámbricos digitales, transmisión inalámbrica de video y audio y enlaces de datos inalámbricos) que operan desde 5.725 GHz hasta 5.850 GHz en la banda ISM sin licencia. Tiene una sensibilidad a la entrada de -96 dBm (a una tasa de transmisión de 2.048 Mbps) y un BER (Bit Error Rate) del 0.1%.

Este dispositivo tiene un mezclador de conversión ascendente, un buffer/predriver y un amplificador de potencia para producir una salida típica de 21 dBm. Incluye un sintetizador integrado que se utiliza tanto en los modos de transmisión como de recepción. La regulación de la fuente de alimentación está incluida en el modelo ML5805, proporcionando aislamiento al circuito y ofreciendo un rendimiento constante para tensiones entre 2.8 V y 3.6 V.

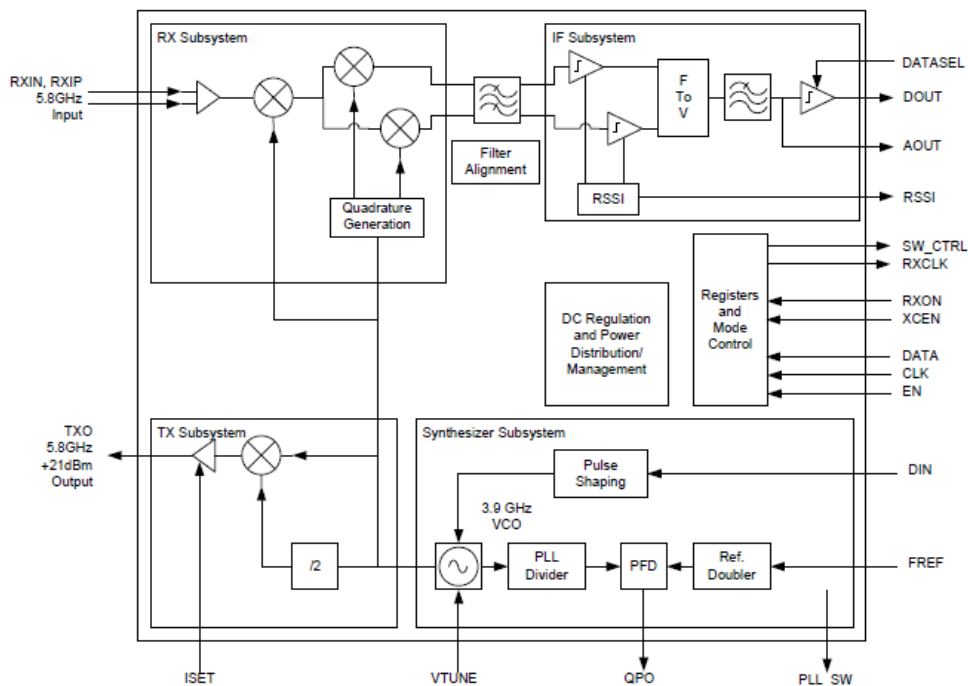


Figura 3.1: Diagrama de bloques funcional

El transceptor se puede configurar para que actúe como transmisor o como receptor. Para llevar a cabo este trabajo se necesita que el ML5805 se configure para transmitir ya que la recepción la llevará a cabo el USRP.

### 3.5.1 MODO TRANSMISIÓN DEL ML5805

El transmisor ML5805 consiste en un mezclador de conversión ascendente seguido de un amplificador de ganancia programable, para permitir la calibración de fábrica de la potencia de salida, y un amplificador de potencia (PA).

En el modo de transmisión, el bucle PLL está cerrado para eliminar la deriva de frecuencia. Un modulador de FSK de bucle cerrado modula tanto el VCO y el fraccional-N PLL. El VCO es modulado directamente mediante una modulación FSK con filtrado de datos de transmisión. El PLL es impulsado por un modulador sigma-delta, el cual asegura que el PLL sigue la frecuencia media de la VCO modulada.

La salida del modulador es convertida a alta frecuencia mediante un mezclador y amplificada con un PA (Power Amplifier) para proporcionar una potencia de salida de 21 dBm. [15]

Las características técnicas del ML5805 cuando actúa como transmisor se detallan en la siguiente tabla:

| TRANSMISOR                                     |                |           |        |          |                        |
|--|----------------|-----------|--------|----------|------------------------|
| Parámetro                                      | Especificación |           |        | Unidad   | Condición              |
|  | Mínimo         | Típico    | Máximo |          |                        |
| Rango $f_{\text{entrada}}$                     | 5.725          |           | 5.850  | GHz      |                        |
| $P_{\text{salida}}$                            |                | 21        |        | dBm      | Adaptado a 50 $\Omega$ |
| Desviación                                     |                | $\pm 400$ |        | kHz      | 1.1520 Mbps            |
|  |                | $\pm 512$ |        | kHz      | 1.5360 Mbps            |
|  |                | $\pm 596$ |        | kHz      | 1.7554 Mbps            |
|  |                | $\pm 680$ |        | kHz      | 2.048 Mbps             |
| Impedancia <sub>salida</sub>                   |                | TBD       |        | $\Omega$ | En el pin TXO          |
| BW filtro transmisión/Tasa de ratio de símbolo |                | 0.5       |        |          | 1.1520 Mbps            |
|  |                | 0.9       |        |          | 1.5360 Mbps            |
|  |                | 0.8       |        |          | 1.7554 Mbps            |
|  |                | 0.7       |        |          | 2.048 Mbps             |

Tabla 3.1: Características del ML5805 como transmisor

### 3.5.2 STARTER KIT ML5805

El Starter Kit es la placa de evaluación para realizar las pruebas necesarias con el transceptor ML5805. Los Micro Linear Starter Kits permiten testear y verificar el dispositivo fácil y rápidamente. De esta manera es posible testear tanto el enlace de radiocomunicación entre los transceptores idénticos (uno en transmisión y otro en recepción) como para testear el

enlace entre el transceptor (en transmisión) y el USRP (en recepción) que es el que atañe al presente proyecto.

El SK-03 (Starter Kit) utiliza un estándar que se ejecuta en un ordenador que tenga Windows como sistema operativo. Así, los clientes son capaces de modificar los parámetros del enlace y el programa de los dispositivos de manera interactiva.

Los Starter Kit contienen: dos placas de banda base, dos placas de radio, dos antenas, dos fuentes de alimentación, ejemplos de código y documentación completa (guía de usuario, esquemático, lista de materiales, teoría de la operación y código de ejemplo).

El Starter Kit realiza la comunicación con el ordenador mediante un USB. Contiene un módulo de diseño modular radio y un microcontrolador PIC de Microchip de bajo coste así como un PC GUI.

Ofrece una velocidad de datos de 2.048 Mbps, por lo que, como se deduce de la Tabla 3.1, la desviación en frecuencia será de  $\pm 680$  kHz y el ancho de banda del filtro de transmisión/tasa de ratio de símbolo (es decir el producto BT) será de 0.7. La potencia de salida oscila entre 16 dBm y 22 dBm y la sensibilidad es de -95 dBm para un BER (Bit Error Rate) del 0.1%.

Otra de las prestaciones del SK03 es su potente software que es capaz de realizar la evaluación de cada canal, la evaluación de la Densidad Espectral de Potencia (incluyendo un visualizador del analizador de espectro) y ofrece también un protocolo inalámbrico simple que permite el testeo del error de paquete (PER). [4]

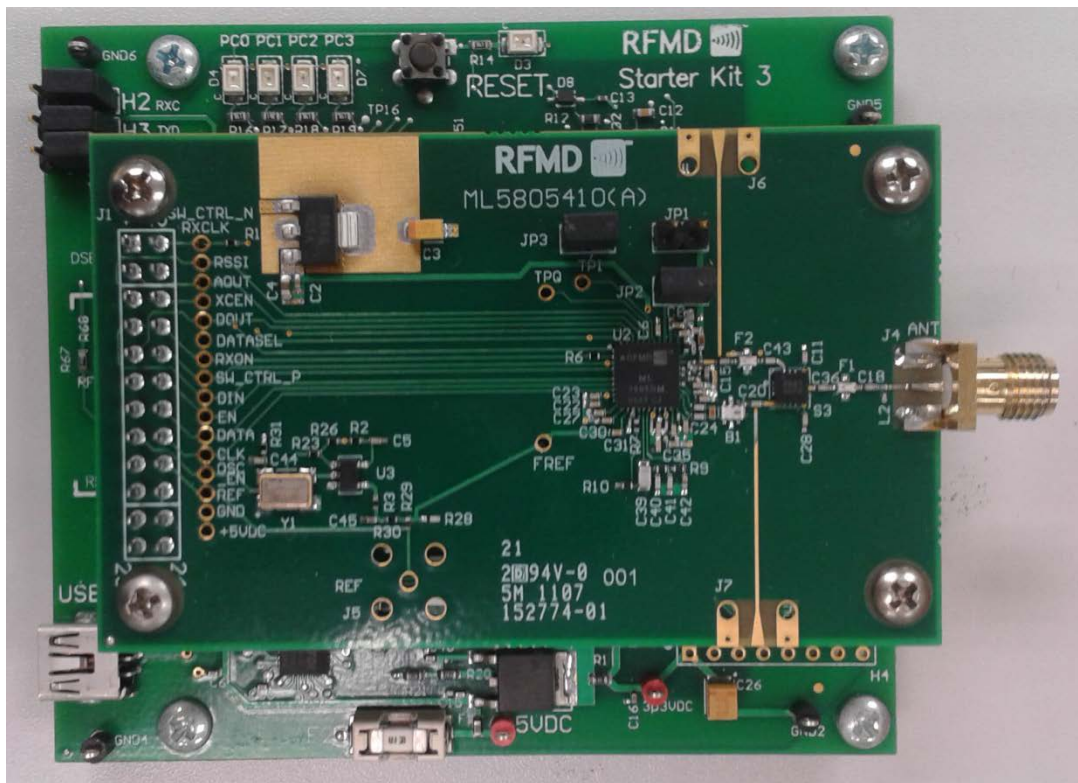


Figura 3.2: Starter Kit 03

## 4 DESARROLLO

En este capítulo se explicarán en primer lugar las funciones básicas para realizar las tareas de control del USRP así como la demodulación FSK. A continuación, se pasará a describir las distintas versiones que se llevaron a cabo para conseguir finalmente un programa para realizar la recepción de los datos enviados a través del transceptor ML5805.

### 4.1 CONTROL CON EL USRP

El LabVIEW es la herramienta necesaria para poder interactuar con el hardware NI-URSP para poder realizar el desarrollo y la exploración de distintos algoritmos de comunicación que procesan las señales recibidas y sintetizan las señales para la transmisión.

El controlador software del NI-URSP ofrece diversas funciones para la configuración de hardware y software mediante herramientas para realizar la apertura y cierre de las sesiones así como ejecutar operaciones de lectura y escritura.

Para realizar la recepción de una señal se necesitan unos módulos básicos para interactuar con el USRP. En la Figura 4.1 se muestra cómo abrir una sesión de recepción, configurar la señal, iniciar la adquisición, recibir los datos, analizar y representar los datos, detener la adquisición y cerrar la sesión de recepción.

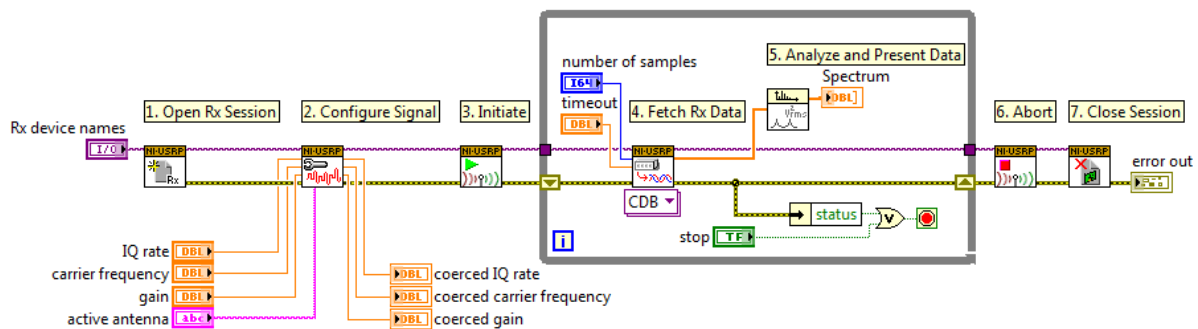


Figura 4.1: Diagrama de recepción con el USRP

En la figura anterior se puede observar que los módulos uno, dos, tres, cuatro, seis y siete son los correspondientes al control con el USRP mientras que el cinco analiza y representa los datos obtenidos en la recepción. A continuación, se explican cuáles son las características básicas de cada una de estas funciones necesarias para realizar la interacción con el USRP.



*niUSRP Open Rx Session*: abre una sesión de recepción para el dispositivo que se especifica en el parámetro de entrada “device name” y devuelve a la salida “sesión handle out” que se utiliza para identificar esta sesión del instrumento en los posteriores NI-USRP VIs.



*niUSRP Configure Signal*: configura la propiedades de la señal recibida y devuelve el valor actual del dispositivo. Los parámetros que se configuran a la entrada son:

- *IQ rate*: especifica la velocidad de muestreo de la banda base I/Q de datos de recepción en muestras por segundo. Es el resultado de multiplicar la tasa de símbolo (symbol rate) por las muestras/símbolo.
- *Carrier frequency*: frecuencia de portadora.
- *Gain*: ganancia agregada en dB aplicada a la señal de RF.
- *Active antenna*: especifica el puerto de la antena que se utilizará para este canal.



*niUSRP Initiate*: inicia la adquisición de la sesión de recepción.



CDB Cluster ▾

*niUSRP Fetch Rx Data (poly)*: obtiene los datos de la lista de canales especificada. En este módulo se tiene como entrada el parámetro “number of samples” donde se indica el número de muestras que se obtienen desde la adquisición del canal.



*niUSRP Abort*: detiene la adquisición de datos una vez ha sido iniciada ésta previamente.



*niUSRP Close Session*: cierra el identificador de sesión del dispositivo.

Las funciones descritas anteriormente son las que se utilizan para controlar el USRP utilizando el LabVIEW para así poder recibir y recomponer la señal transmitida. [6]

## 4.2 DEMODULACIÓN FSK

En este proyecto se trata de demodular una señal FSK (Frequency Shift Keying). Para ello, existen unas herramientas en el Modulation Toolkit que serán de utilidad para realizar este trabajo.

En primer lugar, será necesario generar los parámetros de la FSK para lo que tenemos el siguiente bloque:



*MT Generate FSK System Parameters*: este VI calcula los parámetros que se usan para realizar bien la modulación o la demodulación FSK. Se deben introducir los siguientes parámetros de entrada:

- *Samples Per Symbol*: especifica un número par y positivo de muestras dedicadas para cada símbolo. Multiplica este valor por el “Symbol Rate” para determinar el “IQ Rate”.
- *M-FSK*: especifica el número M-aria, es decir, el número de desviaciones de frecuencia distintas para utilizar como símbolos. En este caso, el valor es 2.
- *Expected FSK deviation*: especifica la desviación máxima de frecuencia FSK. A frecuencias banda base, las desviaciones para símbolos individuales se espacian uniformemente en el intervalo  $[-f_d, f_d]$ , donde  $f_d$  representa la desviación en frecuencia. En este caso, el valor es 680 kHz.
- *Symbol Phase Continuity*: especifica si las transiciones de fase entre símbolos son continuas. En este caso, existe continuidad de fase entre símbolos.

A la salida de esta función se obtiene un clúster de los parámetros FSK donde se incluyen las muestras por símbolo (“Samples per Symbol”), el mapa de símbolos (“Symbol Map”) y la continuidad de fase de símbolo (“Symbol Phase Continuity”).

A continuación será necesario calcular los coeficientes del filtro para realizar la demodulación FSK, para lo cual se utiliza la siguiente función de LabVIEW:



*MT Generate Filter Coefficients*: calcula los coeficientes del filtro de conformación de pulsos (“pulse-shaping filter”) y los coeficientes del filtro adaptado (“matched filter”) aplicados a la demodulación. Los coeficientes de filtro calculados se utilizan durante la modulación para reducir el ancho de banda de la señal transmitida y durante la demodulación para reducir la interferencia entre símbolos. El filtro de conformación de impulsos se puede utilizar tanto en la transmisión como para la demodulación de FSK, mientras que el filtro adaptado sólo se utiliza para la demodulación.

Los parámetros que se introducen a la entrada de dicho módulo son:

- *Modulation type*: especifica el tipo de modulación o demodulación para la cual se generan los coeficientes del filtro. En este caso, el valor de esta entrada es FSK.
- *Pulse shaping filter*: especifica el tipo de filtro a generar. Se puede elegir entre: ninguno, “raised cosine”, “root raised cosine” y “gaussian”. Para el caso que nos ocupa el tipo de filtro es “raised cosine” dado que es el que utiliza el transmisor.
- *Pulse shaping samples per symbol*: especifica el número de muestras deseadas por símbolo para el filtro de conformación de pulsos. En este caso, será el mismo valor que el de “samples per symbol”, es decir, el hallado en la salida de la función *MT Generate FSK System Parameters*.
- *Matched samples per symbol*: especifica el número de muestras deseadas por símbolo para el filtro adaptado de demodulación. En este caso, será el mismo valor que el de la entrada *pulse shaping samples per symbol*.
- *Filter parameter*: especifica el producto BT (BW filtro transmisión/Tasa de ratio de símbolo). En este caso, el valor es 0.7, es decir, el utilizado en el transmisor.
- *Filter length (symbols)*: especifica la longitud deseada del filtro de conformación de impulsos, en símbolos. La longitud del filtro adaptado puede variar según el tipo de filtro y el tipo de modulación. Para este caso, se utilizan cuatro símbolos de longitud de filtro.

Los parámetros que da este módulo a la salida son:

- *Pulse shaping filter coefficients*: devuelve un array ordenado de coeficientes del filtro correspondiente a la respuesta del filtro deseada para el filtro de conformación de impulsos utilizado en la modulación. El número de coeficientes depende de las características del filtro de conformación de impulsos especificadas en el parámetro de filtro de conformación de impulsos.
- *Matched filter coefficients*: devuelve un array ordenado de coeficientes del filtro correspondiente a la respuesta del filtro deseada para el filtro adaptado utilizado en la demodulación.



Por otra parte, antes de realizar la demodulación FSK propiamente dicha, es interesante realizar un remuestreo de los datos para volver a muestrear a la tasa deseada. Para ello se utiliza el módulo que sigue:



**MT Resample (Complex Cluster):** especifica una forma de onda y el nuevo muestreo a valores complejos. Se vuelve a muestrear la señal ya que el demodulador requiere de un número entero de muestras por símbolo. Para volver a muestrear una forma de onda es necesario especificar la frecuencia de muestreo deseada que difiere de la frecuencia de muestreo actual de la forma de onda compleja de entrada. La frecuencia de muestreo deseada es la “IQ Rate” (Symbol Rate x Samples/symbol).

Como parámetro de entrada principal de este bloque tenemos la forma de onda compleja que especifica los datos de forma de onda compleja en banda base. Esta entrada es un cluster de tres componentes:

- $t_0$ : especifica el “trigger” (inicio) del tiempo del array Y.
- $dt$ : especifica el intervalo de tiempo entre puntos de datos en el array Y.
- Y: especifica un array con los valores complejos del dominio del tiempo de los datos. Las partes real e imaginaria de este array complejo de datos corresponde a sus componentes en fase (I) y los datos en cuadratura de fase (Q), respectivamente.

En último lugar, se conecta la salida de este módulo al demodulador FSK para realizar la recuperación de los datos enviados mediante el transmisor.



**MT Demodulate FSK:** demodula la forma de onda compleja banda base de la señal FSK modulada y devuelve la forma de onda demodulada, el flujo de bits de información demodulada y los resultados de la medición obtenidos durante la demodulación. Este VI trata de eliminar el desplazamiento de portadora y de fase mediante el bloqueo de la señal portadora.

Las entradas que se introducen en este bloque son las siguientes:

- *Input complex waveform*: especifica los datos de la forma de onda banda base compleja modulada en FSK.
- *FSK system parameters*: especifica los valores de los parámetros que definen el sistema FSK. Esta entrada debe estar unida con la salida del bloque *MT Generate System Parameters*.
- *Matched filter coefficients*: especifica un array ordenado que contiene los coeficientes del filtro adaptado deseado. Esta entrada debe ser conectada con la salida *matched filter coefficients* de la función *MT Generate Filter Coefficients*.
- *Synchronization parameters*: especifica los valores de los parámetros que describen la secuencia de sincronización y el rango de bits sobre el que se realiza la búsqueda de la secuencia.
- *Pulse shaping filter coefficients*: especifica un array ordenado que contiene los coeficientes del filtro de conformación de pulsos deseado. Se debe conectar a la salida *pulse shaping filter coefficients* de la función *MT Generate Filter Coefficients*.
- *Reset*: especifica si el VI demodula utilizando las colas de la iteración anterior. En este caso, como se requiere de una adquisición continua, este valor debe establecerse a FALSE. [6]



### 4.3 DESARROLLO DEL DEMODULADOR EN LABVIEW

En este apartado se explicarán los distintos programas, las diferentes estrategias de programación y los diversos problemas que han surgido a lo largo de la construcción del demodulador FSK.

Cabe destacar que para realizar un análisis exhaustivo de la correcta funcionalidad del demodulador, fue necesario realizar un montaje en el que se utilizó un transmisor para enviar un patrón binario. De esta manera, se podía comprobar si el flujo de datos de salida del demodulador coincidía con el patrón enviado en el transmisor. Es decir, se realizó un sistema de comunicación donde había un transmisor que enviaba una señal al receptor USRP que se encargaba de realizar la correcta demodulación de los datos enviados. Al comparar los datos enviados con los recibidos se comprobaba si existía fallo y a continuación se indagaba sobre en qué parte del programa se producía error.

#### 4.3.1 PRIMERA VERSIÓN

Para realizar esta primera versión se parte de una realizada previamente por un técnico de National Instruments, en la que proporciona un programa para realizar una demodulación en FSK. A continuación, se pasará a explicar en qué consistía esta primera versión del demodulador y cuáles fueron los obstáculos que se presentaron y las soluciones llevadas a cabo para solventarlos.

En esta primera versión se utilizaban dos bucles. El primero de ellos para poder modificar los datos de entrada aunque más adelante se comprobó que no era necesario. El segundo de ellos para realizar la adquisición de las muestras por bloques como se explicará más adelante. En la Figura 4.2 se puede observar cómo quedó el diagrama de bloques.

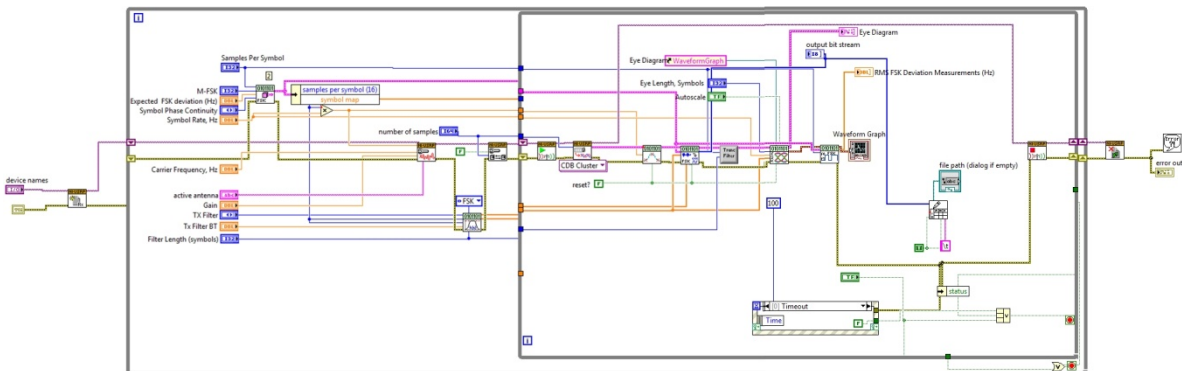


Figura 4.2: Diagrama de bloques de la primera versión

En la Figura 4.3 se pueden apreciar las funciones que se incluyen en la primera parte del programa, en el primer bucle. El módulo *niUSRP Open Rx Session.vi* debe encontrarse fuera del bucle ya que sólo se abrirá una sesión de recepción cada vez que se ejecute del programa.

La sesión de recepción se cableará a la entrada del módulo *niUSRP Configure Signal.vi* donde se introducirán los parámetros de entrada necesarios que ya se explicaron en el apartado anterior.

Después de este módulo, la sesión se cablea a la entrada el módulo *niUSRP Configure Number of Samples.vi* donde se especifica si el funcionamiento del dispositivo es finito o continuo y el número de muestras a adquirir. Se tiene una entrada booleana que si se pone a TRUE el dispositivo adquiere un número finito de muestras mientras que si se indica FALSE el dispositivo está adquiriendo las muestras de manera finita. Otra de las entrada es “number of samples” donde se especifica el número de muestras a adquirir por el dispositivo. Evidentemente, si la entrada booleana se pone a FALSE, el valor de “number of samples” será ignorado.

Por otra parte, el cálculo de los parámetros de la FSK y el cálculo de los coeficientes de los filtros sólo será necesario realizarlo al iniciar la sesión de recepción. Es por esto, que en esta parte se han de incluir las funciones *MT Generate FSK System Parameters.vi* y *MT Generate Filter Coefficients.vi* con sus correspondientes entradas que serán introducidas por el cliente en el Panel Frontal del LabVIEW.

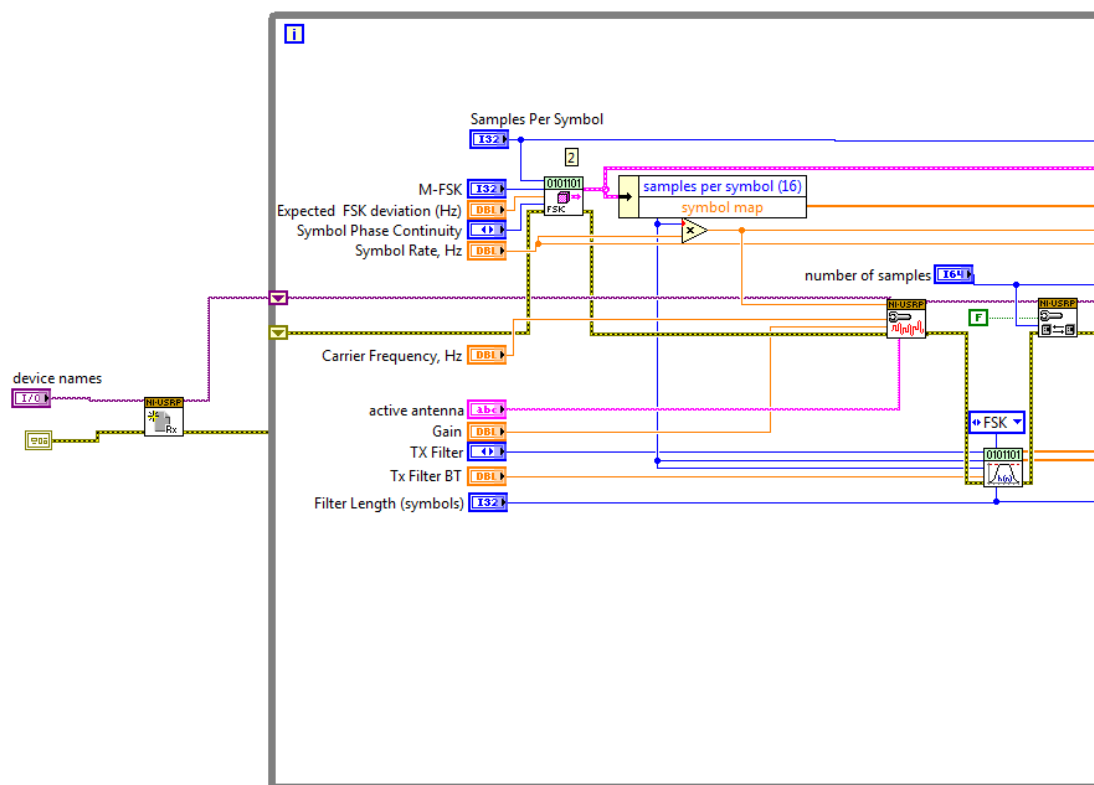


Figura 4.3: Diagrama de bloques de la primera versión (I)

Las funciones de la segunda parte, es decir, las incluidas en el bucle interno se pueden apreciar en la Figura 4.4. En esta sección del diagrama de bloques se observa que lo primero que se realiza al ejecutar este bucle es la función *niUSRP Initiate.vi* para poder iniciar la adquisición de la sesión de recepción.

A continuación mediante el módulo *niUSRP Fetch Rx Data (poly).vi* se obtienen los datos de la adquisición. Como anteriormente se indicó, a esta función hay que introducirle el número de muestras que se obtienen en la adquisición del canal. De esta manera, la adquisición no se realiza de manera continua si no que cada “number of samples” indicado en la entrada de este módulo se realizará una nueva adquisición.

Una vez obtenidos los datos complejos se introduce la señal en la función *MT Resample (Complex Cluster).vi* para hacer un remuestreo de la señal a la tasa deseada (Symbol Rate x Samples per symbol). Una vez hecho esto, la señal se introduce en *MT Demodulate FSK.vi* para realizar la demodulación en FSK de la señal. A la salida se obtiene el flujo de datos de información demodulada además de la forma de onda demodulada.

Después de la demodulación se realiza un filtrado de la señal para luego introducirla en la función *Plot FSK Eye Diagram.vi* de donde se obtiene a la salida una gráfica que representa el diagrama de ojos de la señal y a continuación se tiene el módulo *FSK Deviation Measurement.vi* de donde se obtiene a la salida una gráfica con la forma de onda demodulada y el valor RMS de la desviación en frecuencia de la señal FSK. Estas funciones no son estrictamente necesarias si no que se han incluido para poder detectar con mayor claridad cuáles podrían ser los fallos iniciales del demodulador.

Para poder comprobar que la salida de los datos demodulados es la correcta se efectúa una escritura en fichero de la salida de bits del demodulador FSK. Para realizarlo se utiliza la función *Write to Spreadsheet File.vi* donde le indicamos el nombre del fichero y se escribe de tal forma que todos los datos irán a continuación unos de otros.

Para terminar este bucle interno se utiliza el módulo *niUSRP Abort.vi* para detener la adquisición de datos. Sin embargo esto va unido a un registro de desplazamiento, de tal forma que hasta que no se pare este bucle interno el programa estará adquiriendo y deteniendo la adquisición.

Se demostró que esto genera un fallo muy evidente, ya que al iniciar y detener la adquisición de datos del USRP continuamente se pierden bits porque sólo recogemos un número determinado de muestras con cada pasada del bucle.

Se intentó poner a FALSE el parámetro de entrada booleano de la función *niUSRP Configure Number of Samples.vi*, sin embargo, como en el módulo *niUSRP Fetch Rx Data (poly).vi* también se pide el número de muestras, la adquisición se seguía realizando a bloques y no de manera continua.

También se comprobó en esta versión que es necesario que el parámetro de entrada booleano de la función *MT Demodulate FSK.vi* debe estar a FALSE dado que si se pone a TRUE el demodulador trunca las colas de la señal y no las guarda para la siguiente iteración (el siguiente bloque) por lo que se produce la pérdida de bits.

Siguiendo con la línea del código, lo último que se debe realizar es cerrar la sesión mediante la función *niUSRP Close Session.vi* que como es evidente se realizará una única vez a lo largo de la ejecución del programa, razón por la cual se encuentra fuera de los dos bucles.

Debido a estos problemas se decidió realizar una nueva versión de demodulador donde se iniciara una única vez la adquisición de los datos para así poder realizar de manera continua la recepción sin pérdida de bits.

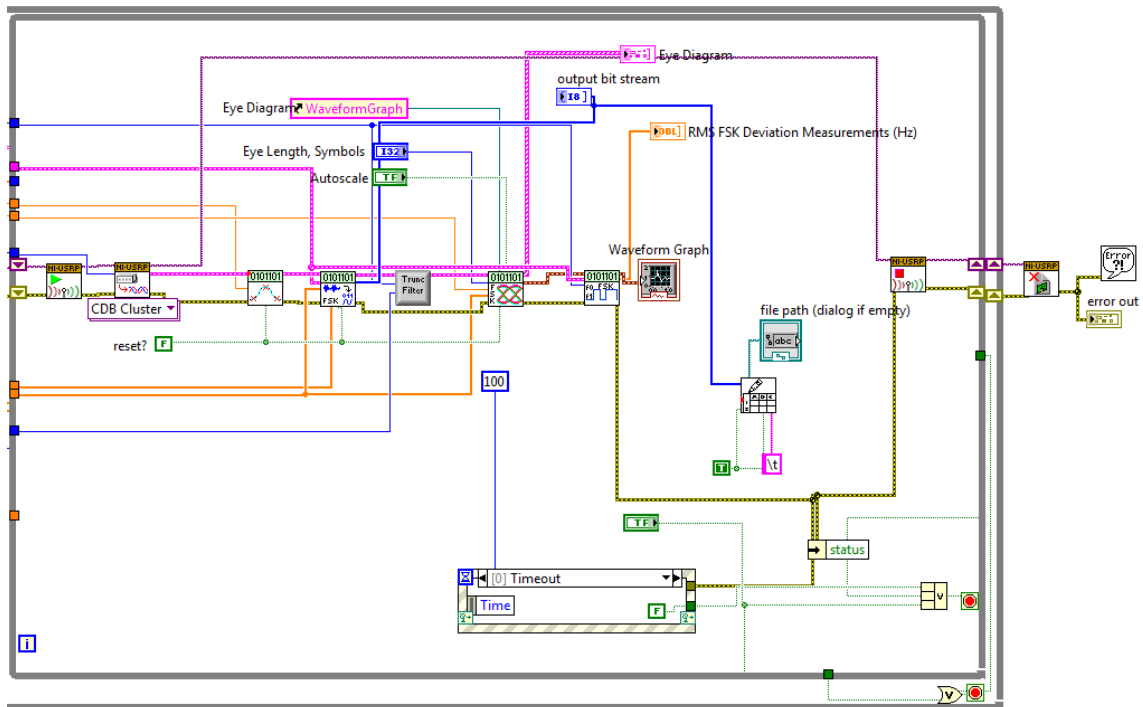


Figura 4.4: Diagrama de bloques de la primera versión (II)

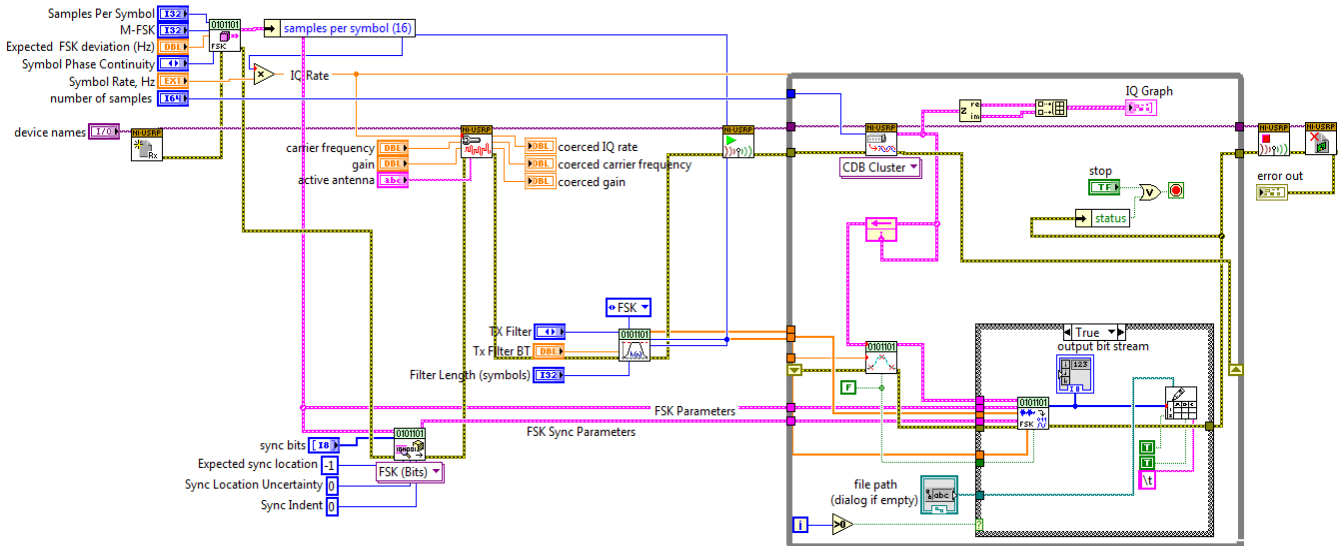
### 4.3.2 SEGUNDA VERSIÓN

En primera instancia se realizó otra versión para tratar de recuperar los bits perdidos debido a que la adquisición se realizaba con cada pasada del bucle, sin embargo, después de varias pruebas se acabaron solucionando otros problemas que fueron surgiendo.

Por un lado, la adquisición de datos que se pretendía realizar tenía unos valores que se pretendían mantener a lo largo de la ejecución del programa por lo que ya no era necesario el primer bucle inicial que se realizó en la primera versión. De esta manera, los valores de la frecuencia de portadora, la desviación en frecuencia, la tasa de símbolo, las muestras por símbolo...entre otros, no variarán a lo largo de la ejecución del programa si no que se podrían cambiar cada vez que se volviera a ejecutar el programa pero no mientras se produjera la adquisición y demodulación de los datos.

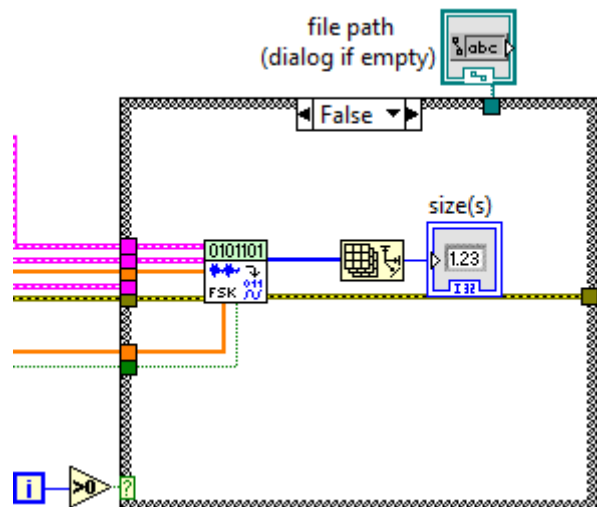
Se comprobó tras varias ejecuciones que la velocidad de muestreo ("IQ rate") que se introduce en el módulo *niUSRP Configure Signal.vi* no siempre es posible que la admita el USRP. Es decir, el USRP tiene una serie de valores de velocidades de muestreo que son las que puede soportar. Por lo tanto, a la hora de realizar las pruebas del BER que se detallarán más adelante en la memoria, habrá que escoger un valor adecuado de "IQ Rate" para que coincida con el que es capaz de soportar el dispositivo.

Para evitar que se perdieran bits se comprueba que la adquisición de los datos se debe realizar fuera del bucle, así el módulo *niUSRP Inicie.vi* se queda fuera para evitar la pérdida de bits. Sin embargo se comprobó que se seguía produciendo el problema aunque no a tan gran escala. La solución fue utilizar un “Feedback Node” para que se guardaran los datos de una iteración del bucle a la siguiente para que así fuera capaz de utilizar las colas de los bits de la iteración anterior.



**Figura 4.5: Diagrama de bloques de la segunda versión**

Tras diversas pruebas se comprobó que en el primer bloque de bits, es decir, en la primera iteración del bucle, se perdían bits debido a que no existía realimentación anterior y no tiene información de las colas. Por lo tanto, se opta por una solución en la que escriba en el fichero a partir del segundo bloque utilizando una estructura “case”. En la Figura 4.6 se aprecia cómo en el caso de que la condición “ $i > 0$ ” sea FALSE, es decir, en el caso de que se encuentre en la primera iteración del bucle while únicamente saca por pantalla el tamaño del flujo de bits de información demodulada.



**Figura 4.6: Estructura Case (FALSE)**

Para el caso de que la condición sea verdadera, es decir, a partir del segundo bloque de datos se produce la estructura en fichero mediante el mismo procedimiento que se utilizó en la versión anterior, es decir, mediante la función *Write to Spreadsheet File.vi*.

De esta manera, se evita finalmente el fallo entre bloques, sin embargo, al no tener un sincronismo el programa empieza a adquirir datos directamente cuando se ejecuta el programa. Es decir, si se envía por ejemplo la trama 00000001 al ejecutar el programa no empezará recogiendo esta trama si no que dependiendo del instante en el que empezamos a adquirir los datos podría empezar con la trama 001 por ejemplo.

Así, se utiliza la función *MT Generate Synchronization Parameters.vi* para construir un clúster de los parámetros de sincronización de la FSK que seguidamente serán enviados al demodulador. Esta función tiene como entradas:

- *FSK system parameters*: especifica los valores de los parámetros que definen el sistema FSK. Debe estar conectado al clúster de salida de la función *MT Generate FSK System Parameters.vi*.
- *Sync bits*: especifica un “array” de bits de sincronización que se utiliza para generar el clúster de los parámetros de sincronización FSK.
- *Expected sync location*: especifica la ubicación prevista de los parámetros de sincronización. En este caso el valor es -1 para que se busque en toda la forma de onda los parámetros de sincronización.
- *Sync location uncertainty*: especifica el número de símbolos antes o después de la posición de sincronización esperada donde puede estar situado el primer símbolo de la secuencia de sincronización. Este parámetro se omite si la entrada *expected sync location* se establece a -1.
- *Sync indent*: especifica la distancia que la secuencia de sincronización dista del bloque de información. La distancia es el número de símbolos demodulados anteriores a la secuencia de sincronización. Es decir, un valor de 10 indica que el flujo de bits de salida se compone de 10 símbolos de datos, seguido por la secuencia de sincronización, seguido de los símbolos de datos restantes. El valor por defecto es 0, que es el que se ha establecido en el programa.

Como se puede observar en la Figura 4.5 se realiza una gráfica de la salida del módulo *niUSRP Fetch Rx Data (poly).vi* que es el que obtiene a la salida un clúster con los datos de la adquisición. El objetivo de esta gráfica fue comprobar si se producía un cambio en ella cuando se cambiaba la trama a enviar, sin embargo, se comprobó que esto no ocurría. La razón es que el USRP no puede realizar internamente todas las funciones de manera inmediata así que la solución por la que se optó fue dividir el programa en dos partes. Una de ellas realizará la función de adquirir las muestras y la otra se encargará de la demodulación propiamente dicha de la señal. De esta forma se pretendía averiguar cuál era el bloque que producía ese retardo a la hora de realizar la recepción de los bits. En el siguiente apartado se explica la estrategia de programación consumer/producer por la que se optó.

En otra de las pruebas que se realizaron se comprobó que al disminuir la velocidad de símbolo (“Symbol Rate”) sí se observaba un cambio en la gráfica, es decir, la recepción se realizaba de manera prácticamente instantánea. Por lo tanto, otra de las opciones que se barajaron fue reducir la velocidad de símbolo en el transmisor dado que el USRP no es capaz de realizar una recepción inmediata con una tasa muy elevada.

Otro de los problemas que surgieron fue que el módulo *niUSRP Fetch Rx Data (poly).vi* se desbordaba a causa de que se llenaba uno de los buffers de recepción antes de que los datos pudieran ser devueltos. Como consecuencia de todos estos problemas, se decidió dividir el programa en dos partes como se ha indicado anteriormente.

### **4.3.3 TERCERA VERSIÓN**

La idea principal para realizar esta versión fue separar el programa en dos partes, una para recibir los datos y otra para realizar todo el proceso de remuestreo de la señal y la demodulación.

Esta estrategia de programación está basada en el patrón de diseño “Producer/Consumer” que a su vez está basado en el patrón Maestro/Esclavo. En programación, este problema también es conocido como el problema del buffer limitado.

Esta nueva forma de programación está orientada a separar en múltiples bucles procesos que comparten información con el objetivo de realizar cada iteración de cada proceso a velocidades distintas. Se distinguen dos casos diferenciados: el productor de los datos y el consumidor de los mismos, que han sido previamente producidos.

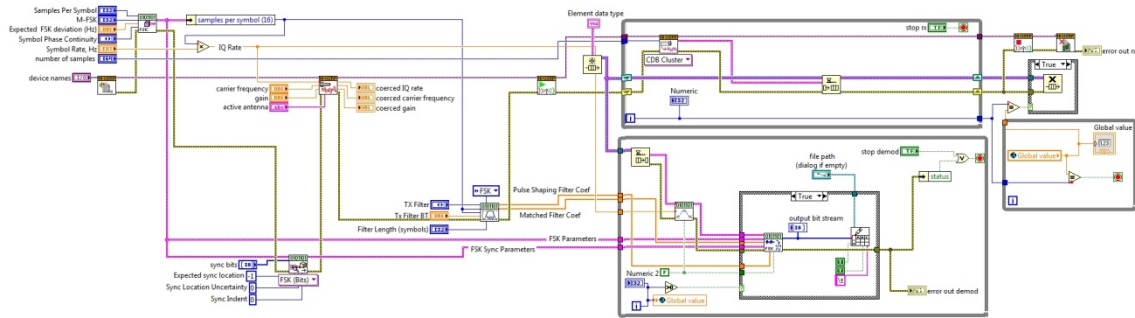
El trabajo del productor consiste en generar los datos, colocarlos en el buffer y realizar de nuevo el proceso. Al mismo tiempo, el consumidor se encarga de sacar los datos de la cola (buffer) donde los ha introducido el productor y de consumir los datos. Es decir, el productor y el consumidor comparten las colas de datos, por tanto, éstas comunican datos entre los bucles del “Producer/Consumer”. Estas colas tienen la ventaja de poder almacenar de manera temporal los datos entre los bucles del productor y consumidor.

En el caso que se plantea en este proyecto se quiere utilizar este patrón de diseño para realizar una comunicación digital. Este tipo de aplicación requiere dos procesos simultáneos a distintas velocidades. El primer proceso (productor) está continuamente recibiendo los datos del sistema de comunicación. El segundo proceso (consumidor) recoge los datos recibidos por el productor y los analiza. Los bucles paralelos del productor y consumidor realizan la recepción y el análisis de los datos de la red y la comunicación por medio de las colas entre los dos permite el almacenamiento temporal de los datos recibidos. Con el almacenamiento temporal, los paquetes pueden ser recibidos y enviados para analizarlos con mayor rapidez.

Como se ha explicado anteriormente se utiliza esta filosofía para llevar a cabo esta versión. Así, a modo de productor se tiene un primer bucle donde se reciben los datos. Por otro lado, como consumidor se tiene un segundo bucle donde se procesan estos datos que son recogidos de la cola, es decir, realiza las tareas de volver a muestrear la señal y demodularla en FSK.

En la Figura 4.7 se tiene una visión global del programa que se irá desglosando para explicar cada fragmento del mismo.





**Figura 4.7: Diagrama de bloques de la tercera versión**

Se puede observar que la primera parte del programa, es decir, la anterior a los bucles es exactamente igual a la de la segunda versión (Figura 4.5). Se tienen fuera el bucle los módulos necesarios para realizar el control con el USRP. De tal forma que en primer lugar está la función *niUSRP Open Rx Session.vi* para abrir la sesión de recepción del USRP. A continuación se tiene el módulo *niUSRP Configure Signal.vi* donde se introducirán los parámetros tales como la velocidad de muestreo (“IQ rate”), la frecuencia de portadora, la ganancia y la antena del USRP que se utiliza para realizar la recepción.

Finalmente, se tiene el módulo *niUSRP Initiate.vi* que se encarga de iniciar la sesión de adquisición y que como se ha explicado anteriormente es necesario que únicamente se realice una vez para evitar la pérdida de bits.

También en la primera parte correspondiente a las funciones anteriores a los bucles se encuentran los módulos que calculan los parámetros necesarios para ser luego utilizados en la demodulación en FSK. Se tiene así los módulos *MT Generate FSK System Parameters.vi* y *MT Generate Filter Coefficients.vi* cuyas funciones son calcular los parámetros de la FSK y los coeficientes de los filtros, respectivamente.

Por último se encuentra la función *MT Generate Synchronization Parameters.vi* que será la encargada de construir un clúster con los parámetros de sincronización de la FSK que serán enviados al demodulador.

La segunda parte del programa es donde se lleva a cabo la filosofía del “Producer/Consumer” explicada anteriormente, donde se crean dos bucles, uno para que haga las funciones de productor y otro que haga las de consumidor.

Para llevar a cabo este modelo de programación se parte de un proyecto realizado en Labview cuyo diagrama de bloques se puede observar en la Figura 4.8.



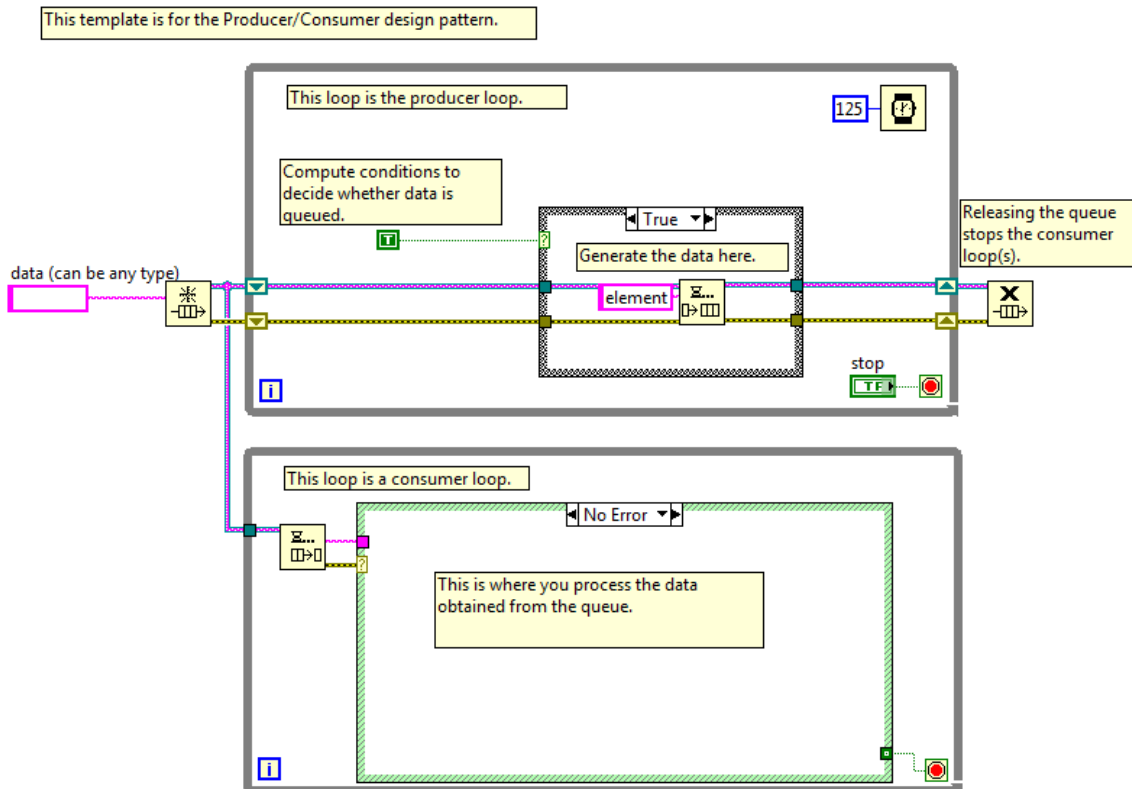


Figura 4.8: Producer/Consumer Design Pattern (Data)

Si siguiendo este modelo para realizar un patrón de diseño “Producer/Consumer” se obtiene la segunda parte del programa que se puede observar en la Figura 4.9.

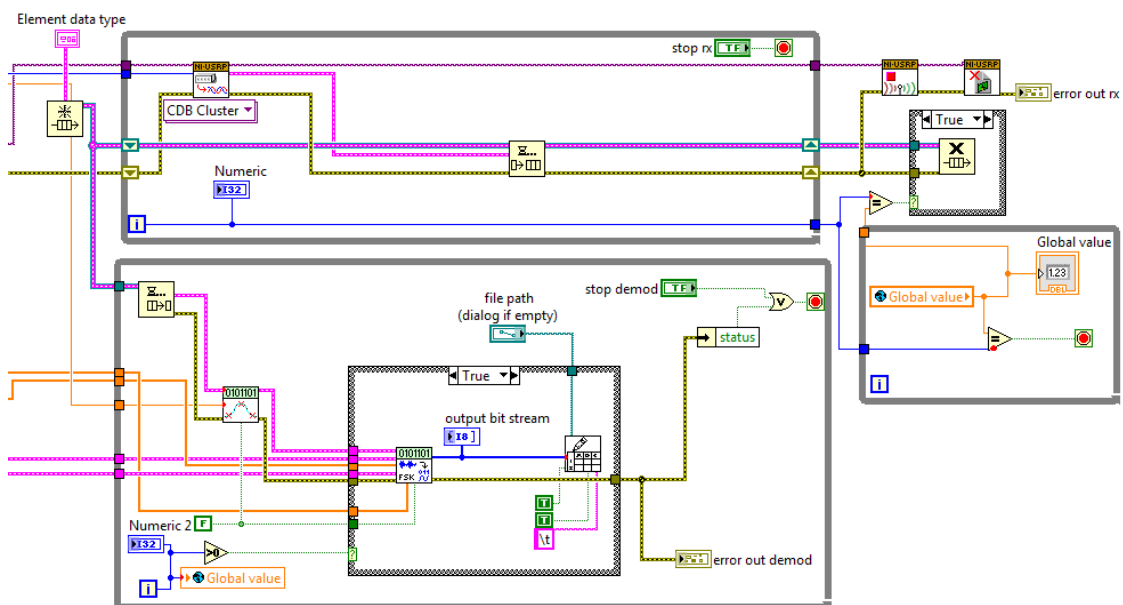


Figura 4.9: Modelo Producer/Consumer

Una cola mantiene el orden de los elementos de datos (FIFO=First in/First out). Las colas son útiles en estas situaciones de productor/consumidor donde una porción de código es la creación de datos para ser utilizada en otra parte. La ventaja principal de usar una cola es que el productor y la tasa de consumo no tienen que ser las mismas. Si el consumo es más lento que la producción, el código del productor se verá obligado a esperar hasta que el consumidor termine de poner en cola los elementos.

Para empezar, es necesario colocar fuera de los bucles la función *Obtain Queue Function.vi* cuyo objetivo será devolver la referencia de una cola. . De esta forma, el consumidor (segundo bucle) sabe dónde debe empezar a analizar los datos. Se le deberá introducir como entrada el tipo de datos que contiene la cola, se puede colocar cualquier tipo de datos. En este caso, el tipo de datos será el de la salida de la función *niUSRP Fetch Rx Data (poly).vi*, es decir, un clúster con los datos de la adquisición.

En el primer bucle, es decir, en el productor, será donde se realice la adquisición de los datos. Seguidamente estos datos se cablean a la entrada del módulo *Enqueue Element Function.vi* para ser añadidos a la parte posterior de la cola. A esta función también será necesario añadirle como entrada la referencia de la cola, que vendrá dada por la salida de la función *Obtain Queue Function.vi*.

Una vez se haya realizado una iteración en el primer bucle, se pasará al siguiente, es decir, el productor. La primera función que realizará este bucle será *Dequeue Element Function.vi* que será la encargada de obtener los datos de la cola para que posteriormente se realice el remuestreo y la demodulación de la señal. Como se puede observar en la Figura 4.9 el proceso de remuestreo, la demodulación y la escritura en fichero es el mismo que el utilizado en la versión anterior, Figura 4.5.

Para que todos los datos que se han adquirido sean procesados, es necesario que el bucle del consumidor realice las mismas iteraciones que el del productor. Por ello, creamos una variable global en el segundo bucle que indique las iteraciones realizadas en esta parte, de tal forma que cuando las iteraciones realizadas productor sean las mismas que las del consumidor se realice la función *Release Queue Function.vi* que sirve para liberar la referencia de la cola.

Es interesante explicar que se ha creado el programa de forma que cuando se pare la adquisición (el bucle del productor) se parará también el control con el USRP, ya que luego no es necesario a la hora de realizar el post procesamiento de los datos adquiridos.

Sin embargo, el problema de este programa viene de que el productor funciona a una tasa mucho mayor que el consumidor. Dicho de otro modo, el bucle encargado de la adquisición de datos se realiza a una velocidad mucho mayor que el otro bucle posterior encargado de post procesamiento de la señal. Esto es debido a que es más laborioso realizar el remuestreo, la demodulación FSK y la escritura en fichero que la simple adquisición de los datos realizada por el USRP.

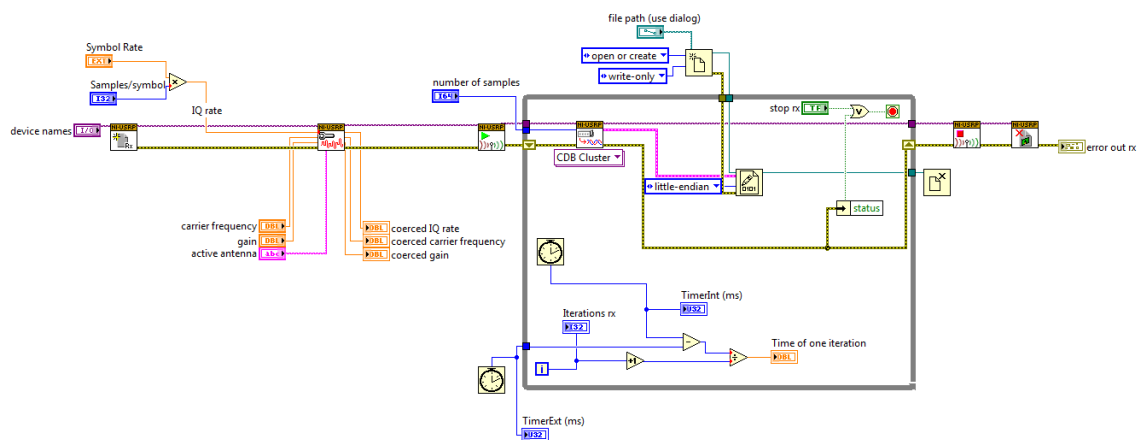
Debido a esto, se desborda uno de los buffers de recepción antes de que los datos puedan ser procesados. Como consecuencia de este problema se decide pasar a otra solución para realizar la recepción de los datos que no está basada en el modelo “Producer/Consumer”.

#### 4.3.4 ÚLTIMA VERSIÓN

Esta última versión se aleja del modelo “Producer/Consumer” aunque está pensada para realizar el proceso de adquisición y de análisis de los datos por separado. Los problemas de llenado del buffer y de las distintas tasas del productor y consumidor se solucionaron realizando un almacenamiento en ficheros.

Otra de las modificaciones importantes que se realizó fue dividir el programa en dos con el único objetivo de la comodidad a la hora de desarrollar este proyecto.

Por un lado, se realizó un programa para realizar la adquisición de la señal, cuyo diagrama de bloques se puede observar en la Figura 4.10.



**Figura 4.10: Diagrama de bloques de la última versión (Adquisición)**

Como se puede observar, la estructura inicial anterior al bucle consiste en el mismo esquema visto en versiones anteriores. Inicialmente se realiza la apertura de la sesión de recepción del USRP mediante la función *niUSRP Open Rx Session.vi*, se introducen los parámetros de la señal en el módulo *niUSRP Configure Signal.vi* y se inicia la adquisición de la sesión mediante *niUSRP Iniciate.vi*.

La diferencia radica en la forma de procesar los datos, en los casos anteriores, se realizaba todo en un mismo programa, la adquisición y el procesamiento. Sin embargo, ahora los datos son almacenados en un fichero para luego ser extraídos y procesados por un segundo programa que se explicará más adelante en la memoria.

Una vez en el interior del bucle se realiza la función *niUSRP Fetch Rx Data (poly).vi* tal y como se ha realizado en versiones anteriores para obtener los datos de la lista de canales que se ha especificado. A continuación los datos que han sido adquiridos en este módulo deben ser almacenados en un fichero. Para ello se utiliza la función *Write to Binary File.vi* que se encarga de escribir los datos en un fichero previamente creado (fuera del bucle) mediante la función *Open/Create/Replace File.vi*. Para terminar este proceso de escritura en fichero, es necesario utilizar la función *Close File.vi* para cerrar el fichero una vez haya terminado la adquisición, es decir, una vez fuera del bucle.

En última instancia se coloca el módulo *niUSRP Abort.vi* para detener la adquisición de los datos seguido de la función *niUSRP Close Session.vi* para cerrar la sesión de recepción del USRP.

Por otra parte, en la Figura 4.11 se puede observar que se realizó un examen de tiempos. El objetivo consistía en comprobar cuál era el número de muestras y las muestras por símbolo óptimas para que la velocidad de procesamiento coincidiera con la velocidad de muestreo. Así, en el siguiente apartado de los resultados se explicará para cada fuente de señal, cuales presentaron los mejores resultados. Es importante recalcar que este estudio es orientativo únicamente ya que el temporizador que se encuentra fuera del bucle no es exacto y puede variar en función del día, la hora,... etc que se realizó el estudio.

Además hay que tener en cuenta que, como se explicó anteriormente, la velocidad de muestreo que se introduce en el módulo *niUSRP Configure Signal.vi* no siempre es capaz de soportarla el USRP, ya que tiene unas velocidades propias establecidas.

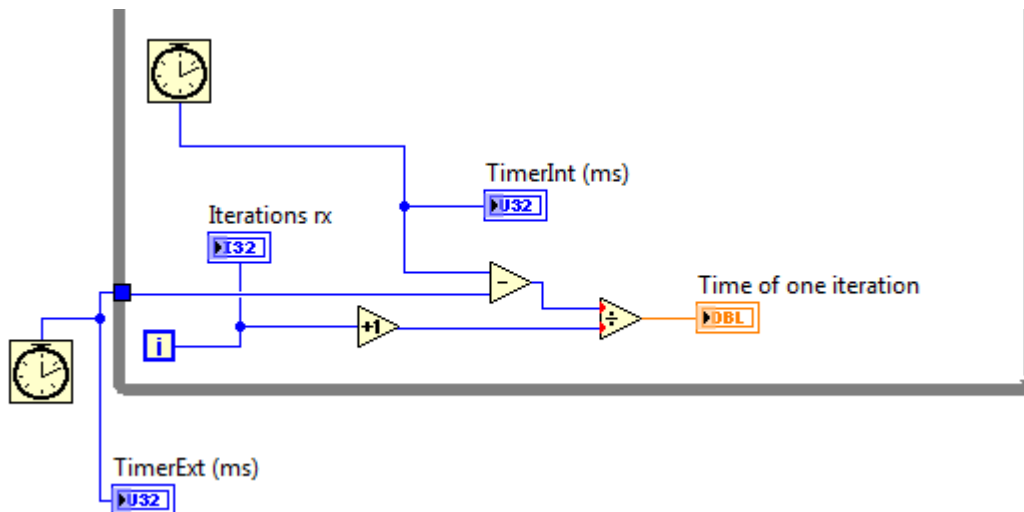


Figura 4.11: Examen de tiempos

Por otro lado, como se ha explicado anteriormente se realizó otro programa para realizar el procesamiento y el análisis de la señal recibida. En la Figura 4.12 se puede observar el diagrama de bloques del mismo.

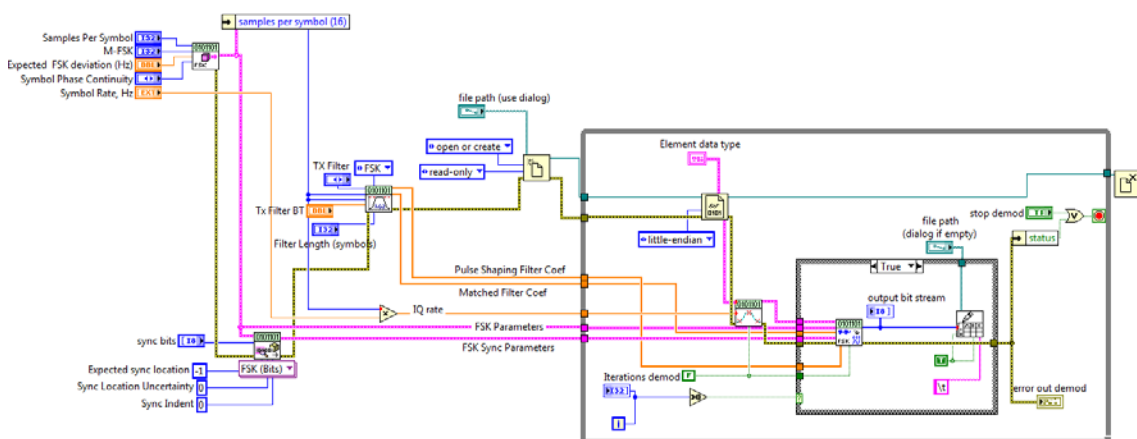


Figura 4.12: Diagrama de bloques de la última versión (Demodulación)

Como se puede observar, los módulos iniciales son los mismos que se utilizaban en versiones anteriores, se tiene el módulo para crear los parámetros de la FSK (*MT Generate FSK System Parameters.vi*), el encargado de generar el sincronismo (*MT Generate Synchronization Parameters.vi*) y el que calcula los coeficientes de los filtros (*MT Generate Filter Coefficients.vi*).

Dado que los datos que se desean procesar se encuentran en el fichero que se ha creado en el programa de adquisición, se necesita de la función *Read from Binary File.vi* para poder extraer los datos. Al igual que en el programa de adquisición se necesita de la función *Open/Create/Replace File.vi* para indicar el fichero del cual se han a leer los datos.

En lo referente al resto del programa, la estructura es la misma ya que se hace un remuestreo y luego se realiza la demodulación de los datos y se guarda en fichero los datos demodulados a partir del segundo bloque de datos, ya que, como se indicó anteriormente el primero no posee información de las colas.

Sin embargo, al realizar las distintas pruebas se comprobó que a pesar de los cambios realizados, el buffer se seguía llenando dado que los procesos realizados en este programa eran demasiado lentos. El fallo venía de uno de los buffers internos del módulo *MT Demodulate FSK.vi* así que se indagó en el programa del demodulador. De esta manera se comprobó que el bloque que estaba dando fallo era uno necesario para realizar unas salidas que en este programa no se utilizan. Por tanto, la solución consistió en crear un nuevo programa de demodulación que no obtuviera estas salidas innecesarias para este proyecto.

De esta manera se solucionó el fallo obtenido además de que se consiguió aumentar la velocidad del procesamiento y con ello se conseguían realizar más iteraciones en un tiempo mucho menor que lo obtenido en anteriores versiones.

## 5 RESULTADOS

En este apartado se explicarán las distintas pruebas realizadas para obtener las curvas de BER para las distintas fuentes de señal. Se utilizaron tres fuentes distintas para generar la señal transmitida: el Starter-Kit 03, un híbrido entre el SK-03 y el HiDAC y por último, el HiDAC propiamente.

### 5.1 STARTER-KIT 03

En primer lugar se realizaron las pruebas del BER utilizando como transmisor el SK-03, que como se explicó anteriormente es la placa de evaluación para realizar las pruebas necesarias con el transceptor ML5805.

Previamente se detallaron las características de dicho transmisor cuya velocidad de datos es de 2.048 Mbps, cuya desviación en frecuencia es de  $\pm 680$  kHz y cuyo producto BT es de 0.7. Sin embargo, se realizó una prueba para comprobar cuáles eran las frecuencias con las que recibiríamos la señal proveniente el SK-03.

Para ello, se realizó el siguiente montaje:

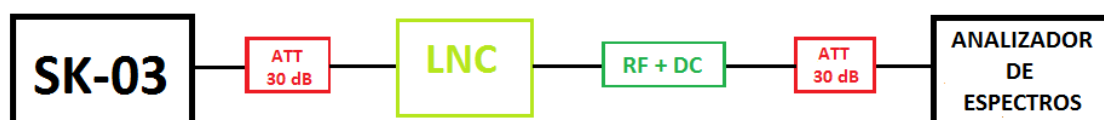


Figura 5.1: Montaje del SK-03 en el Analizador

Como se observa en la Figura 5.1 se tiene como transmisor el SK-03, cuya frecuencia de emisión es 5.8 GHz, seguido de una atenuación de 30 dB para evitar introducir más potencia al LNC de la que es capaz de soportar. Este último se trata de un convertidor de bajo ruido de la marca KUHNE que se utiliza para bajar la señal a frecuencia intermedia (FI) en un rango que oscila entre 400 MHz y 700 MHz. A continuación, se tiene el componente BIAS TEE ZFBT-4R2G-FT cuya función principal es alimentar al LNC, por lo que se le aplica una tensión de 14 voltios. Para seguir, se tiene otro atenuador de 30 dB y finalmente se encuentra el analizador de espectros.

Los datos obtenidos en el analizador fueron:

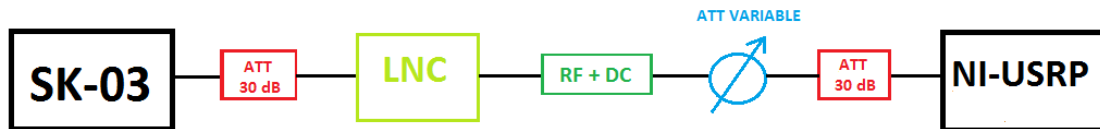
|            | Frecuencia (MHz) | Potencia (dBm) |
|------------|------------------|----------------|
| En el cero | 599.34           | -14            |
| En el uno  | 600.7            | -14            |

Tabla 5.1: Análisis de frecuencias del SK-03

Como se aprecia en la Tabla 5.1 se obtuvo una potencia a la entrada del analizador de unos -14 dBm. También se puede deducir de estos resultados que la frecuencia central es 600.02 MHz y que la desviación en frecuencia es, como bien se facilita en las especificaciones del Starter Kit-03, de  $\pm 680$  kHz.

Es importante añadir que se realizó otra prueba cuyo montaje consistía únicamente en el SK-03 seguido de un latiguillo y una atenuación de 30 dB y el analizador de espectros. De esta prueba se dedujo que la potencia a la salida del SK-03 era de 16 dBm aproximadamente, lo que concuerda con las especificaciones del transmisor. Sin embargo, esta prueba sólo fue necesaria para comparar los distintos resultados de las diferentes fuentes de señal, dado que lo que interesa en este proyecto para realizar los cálculos del BER es la potencia a la salida del montaje, es decir, a la entrada del USRP (receptor).

A continuación se detalla el montaje que se efectuó para realizar los cálculos del BER:



**Figura 5.2: Montaje del SK-03 para cálculo del BER**

En este escenario se realizan las pruebas para obtener el Bit Error Rate (BER) en función de la potencia de salida, es decir, la de entrada al USRP. Como se puede observar en la Figura 5.2, al montaje anterior se añade un atenuador variable, cuyo máximo son 66 dB, para ir variando la potencia de la señal de salida.

El objetivo de este experimento es comprobar cuál sería la sensibilidad máxima soportable por el USRP en este escenario. Inicialmente, la potencia de salida como se ha indicado anteriormente (sin poner ninguna atenuación en el atenuador variable) es de -14 dBm. A lo largo del experimento se irá aumentando la atenuación hasta comprobar cuál es la máxima sensibilidad que el USRP es capaz de tolerar teniendo como transmisor el SK-03.

Para demodular la señal se utilizó la última versión del programa de LabVIEW explicado en el apartado anterior. Los datos de entrada que se introdujeron tanto en programa de adquisición como en el de demodulación fueron los que se detallan a continuación:

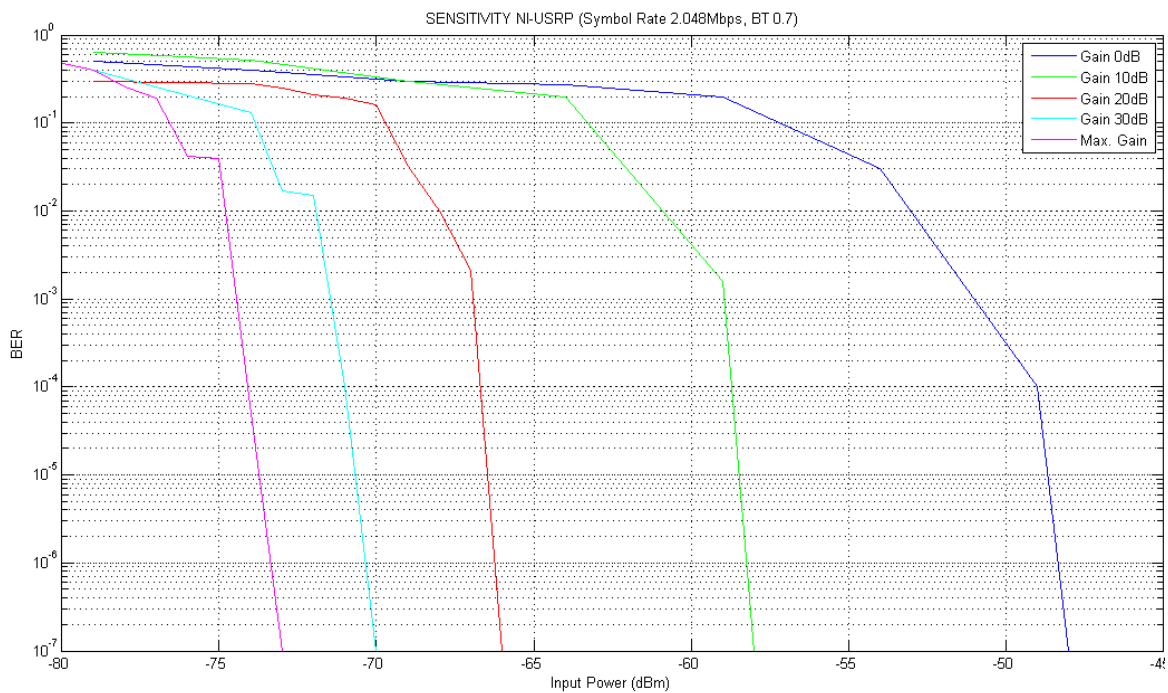
| STARTER KIT - 03       |               |
|------------------------|---------------|
| Symbol Rate            | 2.048Mbps     |
| Carrier Frequency      | 600.02MHz     |
| Number of samples      | 1000          |
| Samples/symbol         | 10            |
| TX filter              | Raised Cosine |
| Tx filter BT           | 0.7           |
| Expected FSK deviation | 680kHz        |
| M-FSK                  | 2             |

**Tabla 5.2: Parámetros de entrada del SK-03**

Los parámetros “Symbol Rate”, “TX filter”, “Tx filter BT”, “Expected FSK deviation” y “M-FSK” coinciden efectivamente con los incluidos en las especificaciones del Starter Kit – 03 que se detallaron en otro apartado previo en esta memoria. La “carrier frequency”, es decir, la frecuencia de portadora es la correspondiente a la frecuencia central calculada después de hallar mediante el analizador de espectros la frecuencia correspondiente al cero y al uno.

El valor de “Number of samples”, es decir, el número de muestras que se obtienen en la adquisición del canal se convino 1000 únicamente por adquirir una cantidad de muestras razonable. En el parámetro “Samples/symbol” se decidió poner un valor de 10 para que así el valor de la velocidad de muestreo fuera 20.48 Mbps (IQ rate = Symbol Rate x Samples/Symbol). El valor de la velocidad de símbolo no se podía modificar dado que era el indicado en las especificaciones del SK-03, sin embargo el número de muestras por símbolo sí que se podía variar con la intención de que la velocidad de muestreo introducida fuera aceptable por el USRP. Como se ha explicado anteriormente, el USRP tiene una serie de valores establecidos de velocidades de muestreo que son los que es capaz de soportar. Se realizaron varias pruebas hasta que se comprobó que el valor de “IQ rate” 20.48 Mbps, es decir en nuestro caso lo correspondiente a 10 “samples /symbol”, era un valor aceptable por el receptor USRP.

En las pruebas realizadas para calcular el BER se iba modificando el valor de la atenuación del circuito mediante el atenuador variable para las distintas ganancias internas del USRP que se introducían como parámetros en el programa de adquisición. Así se realizaron pruebas para una ganancia interna del USRP de 10 dB, 20 dB, 30 dB y 38 dB, que era la máxima ganancia que podía introducir el receptor. Así, se obtuvo la gráfica siguiente:



**Figura 5.3: Gráfica de la sensibilidad del NI-USRP con SK-03**

Para realizar estos cálculos de BER se envió un patrón binario en el SK-03 que el programa de LabVIEW demoduló. A continuación se comparó el patrón binario original con el fichero obtenido tras la demodulación de la señal para poder obtener el Bit Error Rate. Estos cálculos se llevaron a cabo mediante Matlab dado que era la herramienta que más se ajustaba a las necesidades.

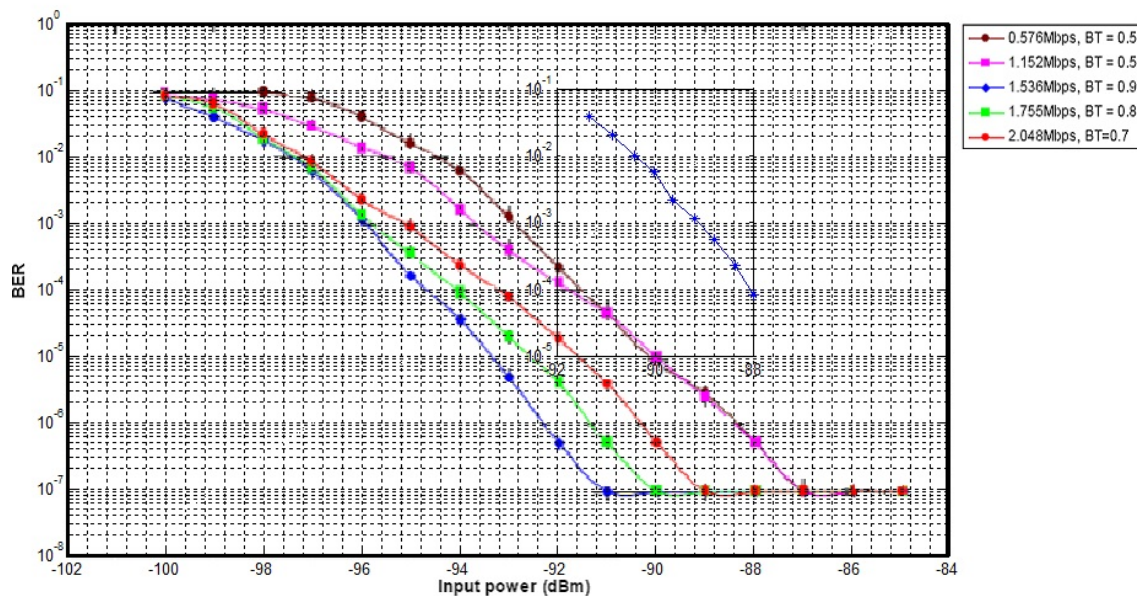
En la Figura 5.3 se tiene el Bit Error Rate (BER) en función de la potencia de entrada, es decir a la entrada del USRP. Como se puede observar, para la ganancia máxima implementada



en el USRP (38 dB) se obtiene la máxima sensibilidad admisible por el receptor. En este proyecto interesa cuál será la potencia para un BER del 0.1%, en este caso, se tiene una sensibilidad de -74.5 dBm. Es decir, en el caso de un escenario cuyo transmisor es el SK-03 y cuyo receptor es el USRP se obtiene una sensibilidad máxima de -74.5 dBm para un BER del 0.1%.

Según las especificaciones del fabricante en el caso de un escenario donde el SK-03 actúe como transmisor y receptor, la sensibilidad máxima para un BER del 0.1% debería ser -95 dBm en el caso de transmitir a una tasa de 2.048 Mbps.

Sin embargo, estos resultados fueron comprobados en el laboratorio con el SK-03 que se utiliza en este proyecto y se advirtió que la sensibilidad real que se obtenía en ese caso era de aproximadamente -89 dBm (como se observa en la gráfica solapada).



**Figura 5.4: Análisis sensibilidad SK-03 TX y RX**

Por lo tanto, comparando los datos obtenidos podemos decir que el utilizar el USRP como receptor empeora las condiciones de sensibilidad con respecto al SK-03. Es decir, en un escenario en el que el transmisor es el SK-03 y el receptor es el USRP se obtiene una sensibilidad máxima de -74.5 dBm. Mientras que si se utiliza el SK-03 como transmisor y receptor se logra una sensibilidad máxima de -89 dBm.

En consecuencia, el utilizar el USRP como receptor empeora la sensibilidad en un valor correspondiente a 14.5 dB.

Resta ahora por comprobar si en otros escenarios con otros transmisores se producen mejoras en la sensibilidad utilizando como receptor el USRP.

## 5.2 HÍBRIDO

Una vez realizadas las pruebas del BER con el SK-03 se pasó a utilizar otra fuente de señal. Este transmisor era un híbrido, donde la banda base era la del HiDAC y la placa de transmisión era la del Starter Kit-03.

Para hallar las frecuencias y las potencias de salida se realizaron dos montajes en el analizador de espectros. Por una parte se realizó un escenario que se puede observar en la Figura 5.5 y cuyos resultados se encuentran en la Tabla 5.3; donde la frecuencia de emisión es de 5.8 GHz.

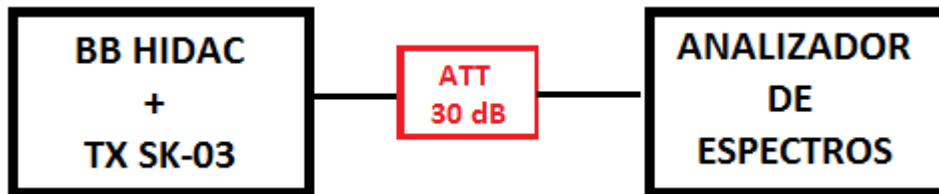


Figura 5.5: Montaje del Híbrido en el Analizador sin conversión

|            | Frecuencia (MHz) | Potencia (dBm) | Desviación en frecuencia |
|------------|------------------|----------------|--------------------------|
| En el cero | 5799.315         | -13.5          | ±680 kHz                 |
| En el uno  | 5800.675         | -13.5          |                          |

Tabla 5.3: Análisis de frecuencias del Híbrido sin conversión

De este montaje se calculó que la potencia de entrada, es decir, la ofrecida por el híbrido, era de 14 dBm. Por tanto, se aproxima a la potencia de entrada proporcionada por el SK-03 que era de 16 dBm.

Por otro lado, también se realizó un análisis de las frecuencias y las potencias para un escenario en el que se realizara la conversión a FI para una frecuencia de emisión de 5.8 GHz:

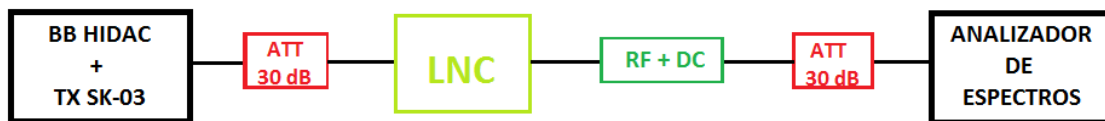


Figura 5.6: Montaje del Híbrido en el Analizador con conversión

|            | Frecuencia (MHz) | Potencia (dBm) | Desviación en frecuencia |
|------------|------------------|----------------|--------------------------|
| En el cero | 599.335          | -20.3          | ±680 kHz                 |
| En el uno  | 600.695          | -20.3          |                          |

Tabla 5.4: Análisis de frecuencias del Híbrido con conversión

Como se puede observar en la Figura 5.6, el montaje es el mismo que el realizado con el SK-03. Además observando la Tabla 5.4 se deduce que la frecuencia central es también 600.02 MHz como en el caso anterior.

La diferencia fundamental entre este transmisor y el analizado en el apartado anterior radica en la velocidad de símbolo. El SK-03 trabajaba a una tasa de 2.048 Mbps mientras que el híbrido al tener como banda base el HiDAC sólo alcanza los 400 kbps.

El escenario en el que se planteó la medida del BER para el híbrido fue el descrito a continuación:



**Figura 5.7: Montaje del Híbrido para cálculo del BER**

Para este montaje se plantea realizar el mismo procedimiento que en el caso del SK-03. Es decir, se tiene como objetivo calcular el Bit Error Rate (BER) en función de la potencia de salida, por tanto, a la entrada del USRP. El objetivo también será cuál es la sensibilidad máxima soportada por el USRP, pero en este caso teniendo como transmisor el híbrido. La potencia de salida es de -20.3 dBm (sin añadir atenuación en el atenuador variable) como se puede observar en la Tabla 5.4.

Por otro lado, los parámetros que se deben introducir en los programas de adquisición y demodulación son los mismos indicados en la Tabla 5.2, excepto en los indicados a continuación:

| <b>HÍBRIDO (BB HIDAC + TX SK-03)</b> |         |
|--------------------------------------|---------|
| <b>Symbol Rate</b>                   | 400kbps |
| <b>Number of samples</b>             | 400     |
| <b>Samples/symbol</b>                | 10      |

**Tabla 5.5: Parámetros de entrada del Híbrido**

La razón de que la tasa de símbolo disminuya es consecuencia de que la banda base es ahora el HiDAC y la máxima tasa a la que es capaz de transmitir es esta.

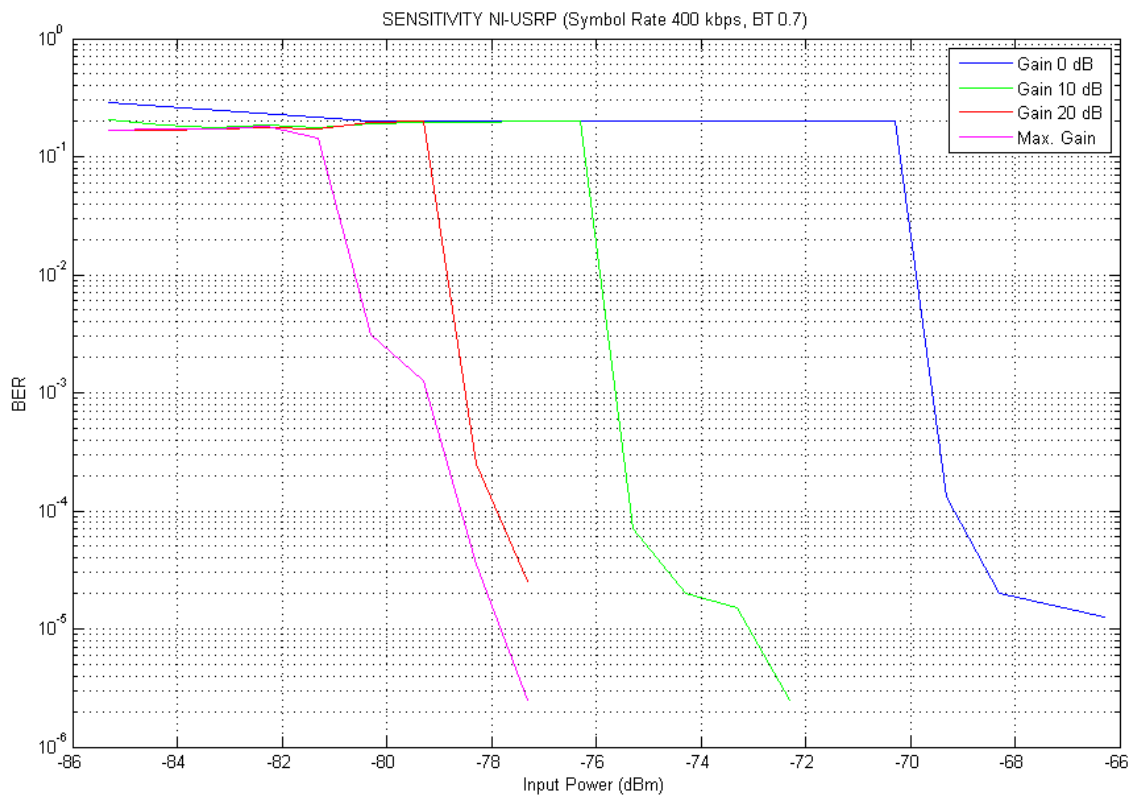
En lo que se refiere al valor de las muestras por símbolo ocurre lo mismo que en el caso en el que el transmisor era el SK-03. Dado que el USRP sólo admite unos valores de velocidades de muestreo, es necesario que el producto de las muestras por símbolo y la tasa de símbolo sea un valor válido para el USRP. Es por esto que se fija entonces que la velocidad de muestreo (“IQ rate”) sea de 4 Mbps, por lo tanto, 10 muestras por símbolo ya que el valor de la velocidad de símbolo viene determinado en la configuración del HiDAC.

Por otra parte, el valor del número de muestras deviene del examen de tiempos que explicó anteriormente y que puede observarse en la Figura 3.13. Se realizó este añadido en el programa de adquisición para comprobar que la velocidad de procesamiento coincidía con la velocidad de muestreo.

Tras realizar varias pruebas con distinto número de muestras se comprobó que para un valor de 400 muestras la velocidad de procesamiento calculada mediante el diagrama de la Figura 3.13 era el que más se aproximaba a la velocidad de muestreo.

Para calcular la velocidad de procesamiento se utilizó el indicador “time of one iteration” que calcula el tiempo que tarda en realizarse cada iteración y que, para el caso de 400 muestras tiende a 101 us. Para calcular la velocidad de procesamiento de cada muestra por tanto,  $1/((101 \times 10^{-6})/400) = 3.96$  Mbps. Si comparamos este resultado con el de la velocidad de muestreo, es decir, la velocidad de símbolo multiplicada por las muestras por símbolo, se obtienen 4Mbps. Por consiguiente, se aproximan lo suficiente como para obtener un buen resultado con los parámetros establecidos.

Para finalizar, se obtiene la siguiente gráfica dónde se indica el Bit Error Rate (BER) en función de la potencia de salida, es decir, la de entrada del USRP.



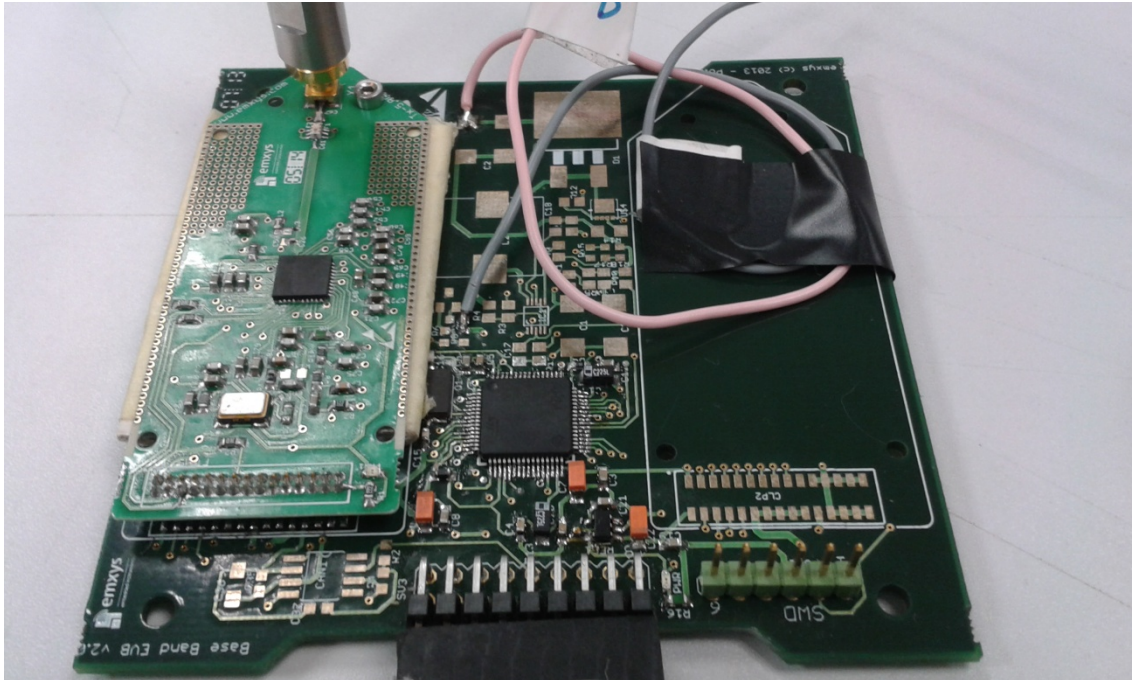
**Figura 5.8: Gráfica de la sensibilidad del NI-USRP con el Híbrido**

Para hallar estos resultados se realizó el mismo procedimiento que el caso en el que se tenía como transmisor el SK-03.

En la Figura 5.8 se aprecia que la máxima sensibilidad obtenida por el USRP, proporciona la curva de la ganancia máxima implementada en el USRP (38 dB). Se observa que para un BER del 0.1% se obtiene una sensibilidad máxima en el USRP de -79.3 dBm. Es decir, se produce una mejora con respecto al SK-03 que proporcionaba una sensibilidad máxima de -74.5 dBm. Esto, se explica porque la velocidad de símbolo del híbrido (400 kbps) es muy inferior a la ofrecida por el SK-03 (2.048 Mbps) lo cual influye en la sensibilidad.

### 5.3 HiDAC

Por último, se realizaron las mismas pruebas para otro transmisor. En este caso, se trata del HiDAC, tanto banda base como transmisor. Realmente, el HiDAC, creado por la empresa Emxys, debería tener las mismas prestaciones que el Starter Kit ya que tiene el mismo transceptor, el ML5805. Sin embargo, el HiDAC tiene más funcionalidades en cuanto al envío de patrones binarios más variados.



**Figura 5.9: HiDAC**

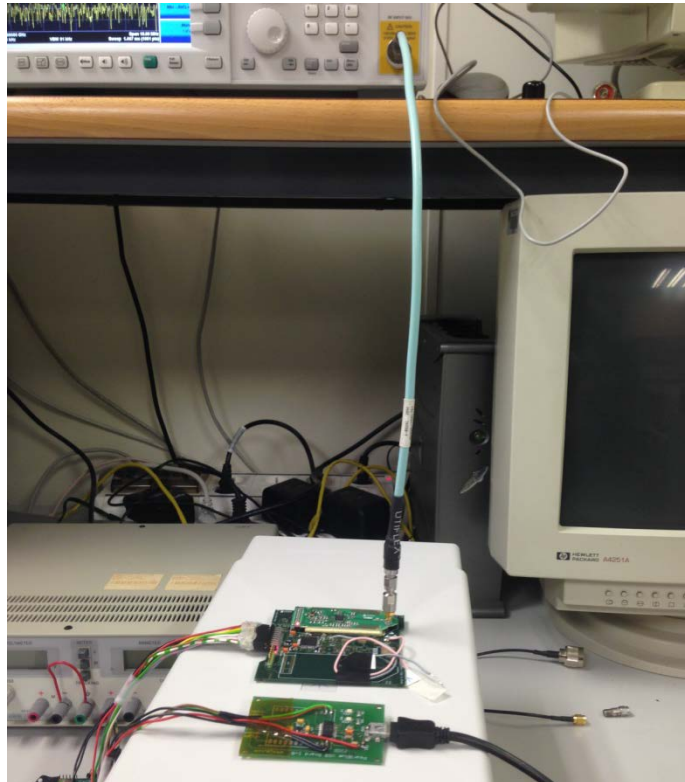
Como se puede observar en la Figura 5.9, el transceptor está incluido en la placa que sobresale a la izquierda del dibujo que se corresponde con la transmisión. La placa verde oscura que se encuentra debajo es la banda base. Ambas placas fueron desarrolladas por la empresa Emxys. La velocidad máxima de trabajo es de 400 kbps.

Al contrario de lo que ocurría en el Starter Kit, en el caso del HiDAC se puede programar la velocidad a la que se desea que trabaje. Es decir, se pueden modificar esos 400 kbps y cambiando algunos valores de la programación se puede disminuir hasta llegar incluso a 25 kbps.

Como en los dos casos previos, se realizó en un principio una serie de experimentos utilizando el analizador de espectros para comprobar cuáles eran las potencias y las frecuencias que era capaz de dar el montaje utilizado.

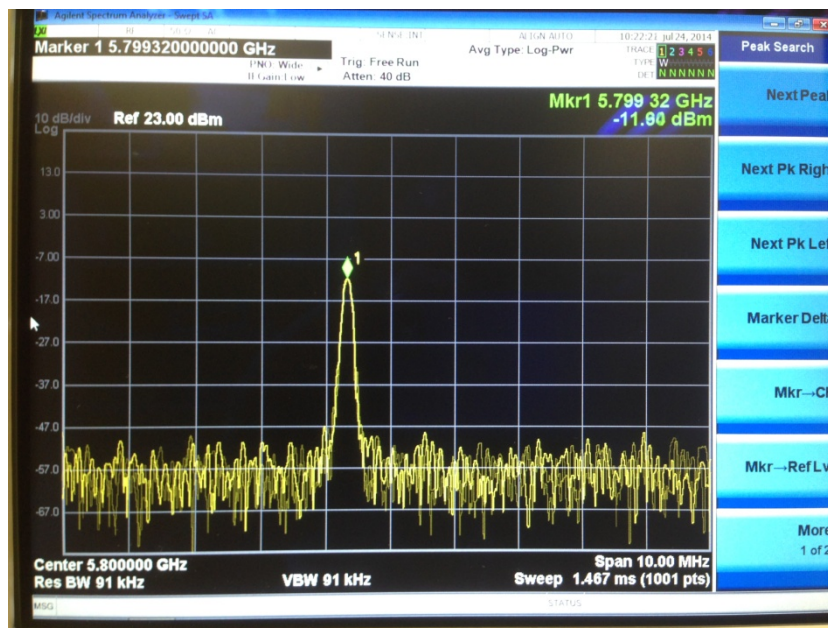


En primer lugar se realizó una prueba en el analizador para comprobar cuál era la potencia de entrada en el montaje con el USRP, es decir, la potencia de salida del HiDAC.

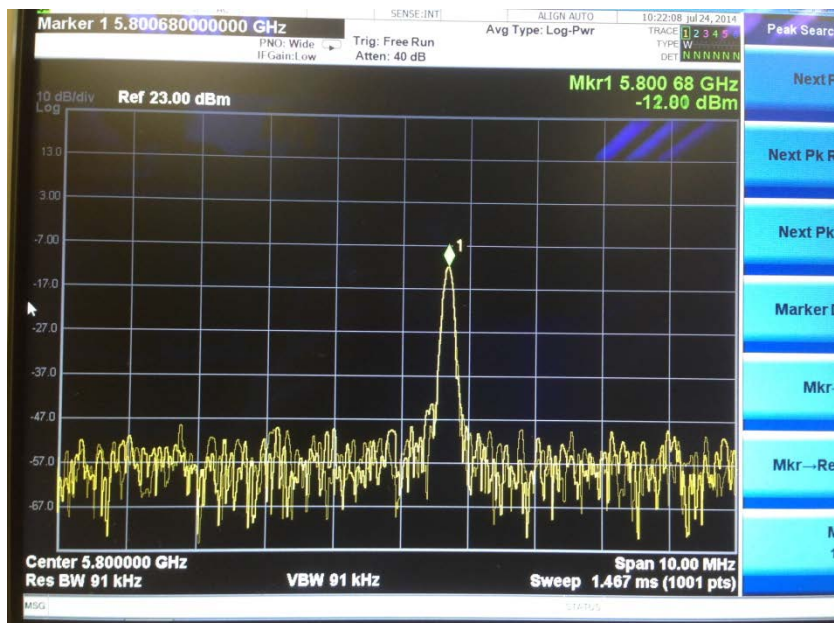


**Figura 5.10: Montaje del HiDAC en el Analizador sin conversión**

Como se puede observar en la Figura 5.10 en este montaje se tiene directamente el HiDAC conectado al analizador mediante un latiguillo. El objetivo de este experimento era calcular por un lado la frecuencia central así como la potencia que es capaz de suministrar el HiDAC.



**Figura 5.11: Frecuencia del cero con el HiDAC (sin conversión)**

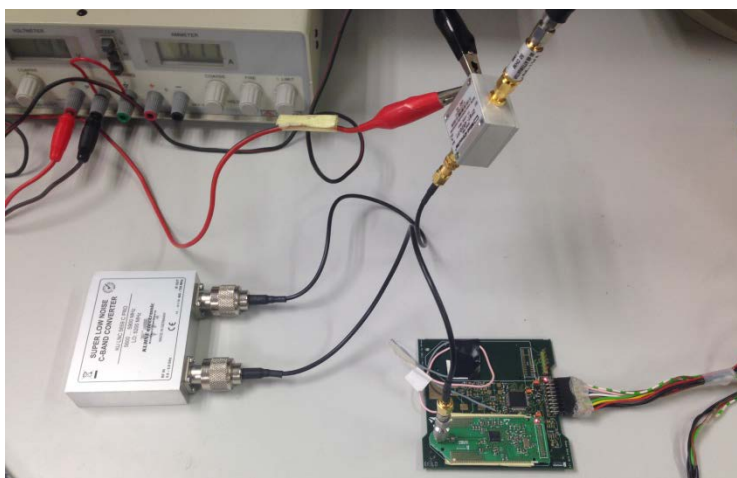


**Figura 5.12: Frecuencia del uno con el HiDAC (sin conversión)**

En la Figura 5.11 se aprecia que la frecuencia del cero es 5799.32 MHz y la potencia son aproximadamente -12 dBm. Para el caso de la Figura 4.12, la frecuencia del uno es 5800.68 MHz y la potencia también es de unos -12 dBm. Por tanto, la frecuencia central de esta señal se encuentra, como era de esperar, en 5.8 GHz y la desviación en frecuencia correspondiente es de  $\pm 680$  kHz.

Se deduce por tanto, que la potencia de salida del HiDAC es mucho menor que la conseguida mediante el híbrido o el Starter Kit, los cuales tenían una potencia de salida de aproximadamente unos 15 dBm en media. Por consiguiente, el HiDAC tiene una potencia de unos 20 dB menor que los otros dos transmisores, lo cual resultará desfavorable dado que la señal enviada será más débil.

Ahora bien, se realizó otro montaje para hallar las frecuencias y las potencias en el analizador de espectros. Este escenario es el que se utilizará más adelante para realizar las pruebas del BER, es decir, realizando la conversión a frecuencia intermedia mediante el LNC.



**Figura 5.13: Montaje del HiDAC en el analizador con conversión**

Del montaje de la Figura 5.13 se obtienen los siguientes espectros en el analizador:

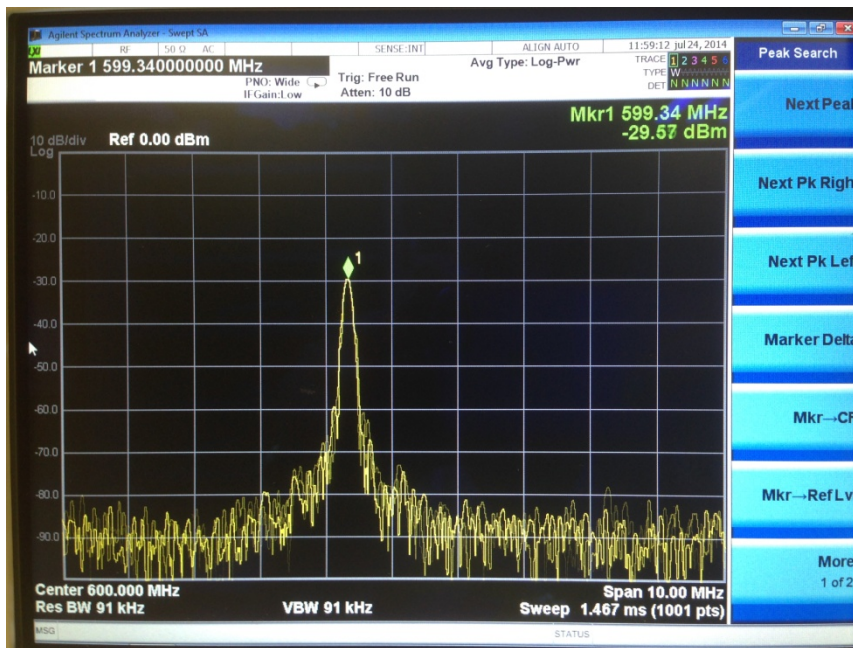


Figura 5.14: Frecuencia del cero con el HiDAC (con conversión)

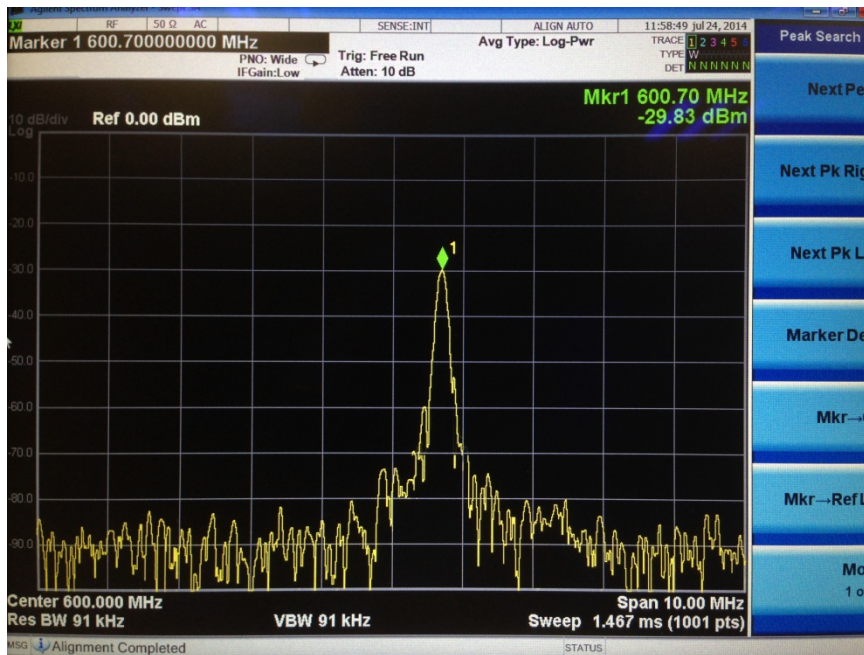


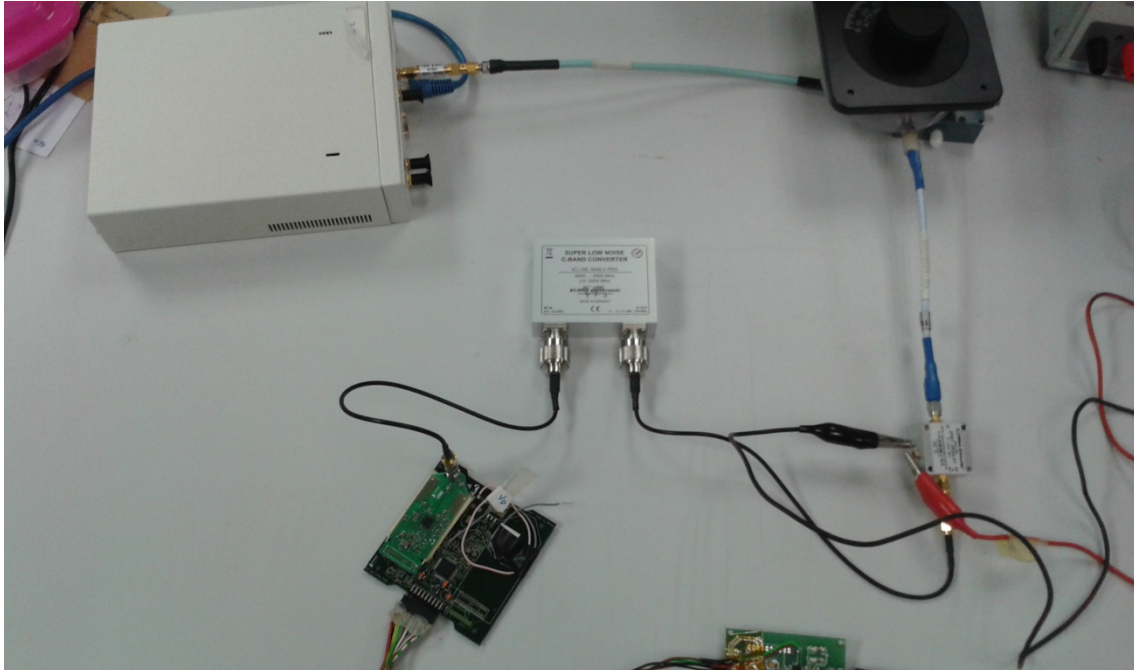
Figura 5.15: Frecuencia del uno con el HiDAC (con conversión)

De la Figura 5.14 se advierte que la frecuencia correspondiente al cero es 599.34 MHz, con una potencia de aproximadamente -30 dBm. En el caso de la Figura 5.15 se distingue que la frecuencia correspondiente al uno es 600.7 MHz cuya potencia se encuentra cerca de los -30 dBm. Con estos datos se calcula la frecuencia central que se encuentra en 600.02 MHz y la desviación en frecuencia que es de 1.36 MHz o lo que es lo mismo  $\pm 680$  kHz. Por consiguiente,

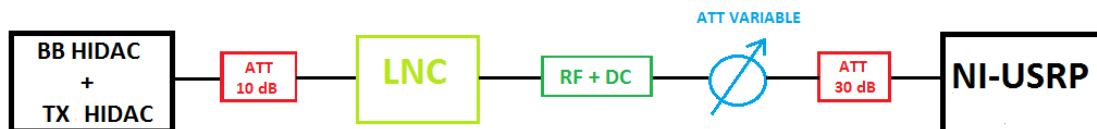


los resultados obtenidos de frecuencia de portadora y desviación en frecuencia han sido los esperados.

Por otro lado, se desarrolló otro montaje para calcular el Bit Error Rate (BER) como se realizó previamente en las dos anteriores fuentes de señal.



**Figura 5.17: Montaje del HiDAC para cálculo del BER**



**Figura 5.16: Esquema del montaje del HiDAC para cálculo del BER**

En las figuras anteriores se describe el escenario utilizado para realizar las pruebas del Bit Error Rate (BER) en función de la potencia de salida, es decir, la potencia de entrada al USRP. Como en los casos anteriores se utiliza el LNC para bajar a frecuencia intermedia y un atenuador variable para modificar la potencia de la señal de salida en las distintas adquisiciones de la señal.

Sin embargo, en la Figura 5.17 se advierte una diferencia con respecto a los montajes anteriores cuyo transmisor era el SK-03 o el híbrido. Esta distinción consiste en que en los dos montajes anteriores la atenuación que se situaba después del transmisor era de 30 dB, por el contrario en este caso la atenuación es de 10 dB. Esto es debido a que como se ha comentado anteriormente, la potencia de salida del HiDAC es 20 dB menor que la de los otros dos transmisores. Por tanto, por este problema de potencia, se ha tenido que aumentar el montaje en 20 dB.

Una vez realizado el montaje se pretende comprobar empíricamente cuál es la sensibilidad máxima que soporta el USRP cuando se tiene el HiDAC como transmisor. La potencia inicial de la que se parte son los -30 dBm que se obtienen a la entrada del USRP cuando no se suministra ninguna atenuación en el atenuador variable. Como en los dos casos anteriores, la atenuación se irá aumentando hasta verificar cuál es la máxima sensibilidad que admite el USRP cuando se utiliza como transmisor el HiDAC.

Los parámetros que se deben introducir para demodular la señal en los programas de adquisición y demodulación están indicados en la Tabla 5.2, sin embargo, existen algunas excepciones que se recogen en la siguiente tabla.

| HiDAC             |         |
|-------------------|---------|
| Symbol Rate       | 400kbps |
| Number of samples | 200     |
| Samples/symbol    | 10      |

Tabla 5.6: Parámetros de entrada del HiDAC

La tasa de símbolo es 400kbps en el caso del HiDAC debido a que no es capaz de transmitir a mayor velocidad como se ha explicado anteriormente. En el caso del número de muestras por símbolo ocurre lo mismo que en los casos previos, es necesario para que la velocidad de muestreo sea un valor que el USRP admita. En lo que se refiere al valor del número de muestras viene como en el caso del híbrido del examen de tiempos, ya que era el valor óptimo para que la velocidad de procesamiento coincidiera con la velocidad de muestreo.

Para concluir, se realizó la siguiente gráfica donde se representa el BER en función de la potencia de entrada al USRP.

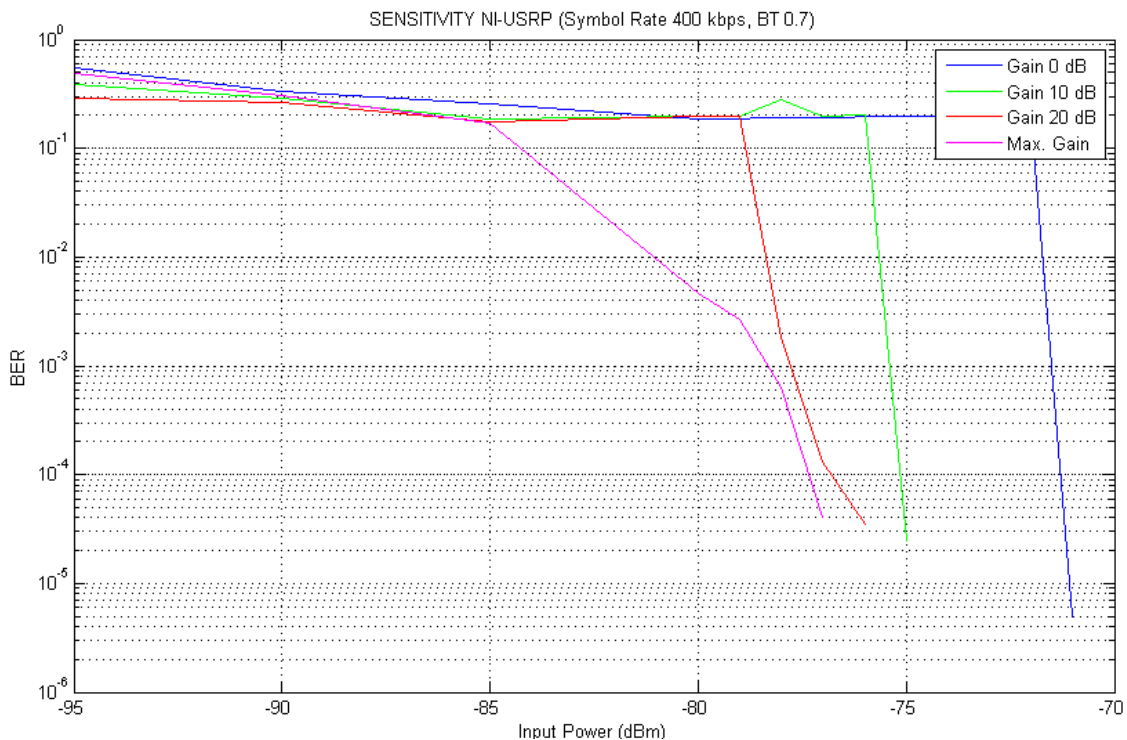


Figura 5.18: Gráfica de la sensibilidad del NI-USRP con el HiDAC

De la Figura 5.18 se deduce que la máxima sensibilidad del USRP se obtiene cuando se aumenta al máximo la ganancia interna el USRP, como era de esperar.

Por otra parte, se tiene que para un BER del 0.1% la sensibilidad máxima de USRP es -78.5 dBm. Es decir, se produce un empeoramiento con respecto al híbrido que proporciona una sensibilidad máxima de -79.3 dBm. Sin embargo mejora con respecto al SK-03 que presenta una sensibilidad de -74 dBm, aunque hay que añadir que las velocidades de símbolo son distintas en este caso, ya que para el SK-03 la velocidad es 2.048 Mbps mientras que en el caso del HiDAC son 400 kbps.

## 6 CONCLUSIONES

En la presente memoria se ha realizado una breve explicación del origen de los pequeños satélites con objetivos educativos para seguidamente centrarse en el análisis y la caracterización del Politech.1.

Se introdujo el satélite Politech.1, donde se explicó la plataforma NaoSAT así como las distintas cargas útiles que iban a bordo del satélite, incluyendo la antena en banda C. También se explicó la estructura para realizar la recepción en la estación terrestre, ya que este proyecto versa sobre una solución para llevar a cabo esta recogida de los datos.

A continuación se analizaron y estudiaron las distintas prestaciones que ofrece la utilización de un receptor SDR (Software Defined Radio), concretamente el NI-USRP fabricado por la empresa National Instruments. De esta manera se comprobaron las ventajas que ofrecía este transceptor, entre las que se encontraban su precio asequible así como su facilidad para utilizarlo en diversas aplicaciones de RF.

Sin embargo, este hardware precisa del software LabVIEW para desarrollar algoritmos de comunicación, motivo por el cual fue necesario estudiar los distintos métodos de programación así como el entorno de desarrollo del software.

Para llevar a cabo el diseño del demodulador FSK programado en LabVIEW fue necesario crear un sistema de comunicaciones con el transceptor ML5805, así que fue preciso realizar una explicación de las funcionalidades del transceptor así como de su placa de evaluación, el Starter Kit – 03.

En esta memoria se explicaron las distintas versiones a las que se llegaron, cuáles fueron sus problemas así como sus soluciones para llegar finalmente a una versión óptima del demodulador FSK programado en LabVIEW.

Finalmente, se llevaron a cabo tres escenarios distintos para realizar distintos sistemas de comunicación donde se estudió la tasa de error de bit, las distintas potencias y frecuencias recibidas. Por otra parte, también se analizó la sensibilidad máxima del USRP en función de estos distintos escenarios con sus diversos dispositivos de transmisión de la señal.

Se concluyó que para el escenario donde el transmisor era el Starter Kit–03 la sensibilidad máxima del USRP obtenida era de -74.5 dBm para un BER del 0.1% siendo la tasa de transmisión 2.048 Mbps.

Para el caso del escenario donde el transmisor era un híbrido (donde la banda base era la del HiDAC y la placa de transmisión era la del Starter Kit-03) la sensibilidad máxima del USRP obtenida era de -79.3 dBm para un BER del 0.1 % siendo la tasa de transmisión 400 kbps. En este caso, la sensibilidad mejora respecto al SK-03, la razón es que aunque la potencia de los transmisores sea la misma, unos 15 dBm, la tasa de transmisión no lo es. En el caso del híbrido, a una tasa de 400 kbps se realiza la demodulación de manera más fiable por lo que se produce menor error que en el caso de la tasa de transmisión del SK-03 que es mayor al doble de ésta, 2.048 Mbps.

Por último, para el caso del escenario donde el transmisor era el HiDAC se obtuvo que la sensibilidad máxima del USRP era de -78.5 dBm para el BER del 0.1 % siendo la tasa de

transmisión 400 kbps. En este caso, la sensibilidad empeora respecto al híbrido, esto es debido a que a pesar de que la tasa de transmisión es la misma, la potencia que el HiDAC proporciona es peor que la del híbrido, -12 dBm en comparación con los 15 dBm del híbrido. Debido a ello, la potencia de salida del HiDAC es muy débil, contiene mucho ruido que es a continuación amplificado por el LNC que hay en el montaje. Por consiguiente, en el caso del HiDAC se estaría amplificando mucho más ruido que en el caso del híbrido.

Es así como quedan analizados, explicados y estudiados los tres escenarios que se han llevado a cabo en la memoria.

En definitiva, se han cumplido todos los objetivos del presente proyecto. Se han estudiado las prestaciones del NI-USRP 2920 así como los métodos de programación y el entorno del software LabVIEW. También se ha diseñado el demodulador FSK para lo cual, se han solventado las muchas dificultades que han surgido debido al escaso material explicativo sobre el hardware de National Instruments.

Además, se han creado los diferentes escenarios con sus distintos dispositivos de transmisión de señal así como se ha estudiado y analizado la tasa de error de bit, las potencias y frecuencias recibidas y la sensibilidad máxima del USRP. Es decir, en este proyecto se ha realizado tanto trabajo de laboratorio como de simulación.

Se ha comprobado mediante este proyecto lo poco apropiado que resulta el entorno del NI-USRP para aplicaciones operativas. Por lo tanto, la solución de realizar la recepción de las imágenes del satélite mediante este dispositivo SDR (Software Defined Radio) no es factible.

Uno de los mayores inconvenientes de realizar una recepción con el USRP es que dada la baja potencia que proporciona actualmente el HiDAC, no es viable debido a la sensibilidad del USRP. Sin embargo, queda pendiente como línea futura de investigación el aumentar la potencia al HiDAC.

El paso que sigue a este proyecto es realizar las pruebas necesarias utilizando como receptor el tranceptor ML5805 para demostrar si esta solución de recogida de datos ofrece una mejor recepción que la obtenida mediante el USRP.

También sería interesante para un futuro realizar una comparación entre la recepción mediante el tranceptor ML5805 y la recepción mediante el USRP para así realizar un análisis de la utilidad que puede proporcionar este hardware en el proyecto del Politech.1.

Por último, recalcar también que se pretende que el satélite Politech.1 sea el primero de muchos satélites creados por la asociación de la Universidad Politécnica de Valencia y la empresa EMXYS, para así conseguir la experiencia necesaria en el ámbito espacial.

# ÍNDICE DE FIGURAS

|   |    |
|---|----|
| Figura 3.1: Plataforma NAOSat .....                                       | 13 |
| Figura 3.2: Esquema de diseño en el satélite.....                         | 15 |
| Figura 3.3: Esquema de diseño de la estación terrestre.....               | 16 |
| Figura 3.4: NI USRP .....   | 17 |
| Figura 3.5: Conexión NI-USRP con PC.....                                  | 18 |
| Figura 3.6: Ejemplo MathScript RT Module .....                            | 19 |
| Figura 3.7: Diagrama de bloques de USRP.....                              | 19 |
| Figura 3.8: Panel Frontal y Diagrama de Bloques .....                     | 22 |
| Figura 3.9: Paleta de controles .....                                     | 24 |
| Figura 3.10: Paleta de funciones .....                                    | 25 |
| Figura 2.11: Estructuras .....  | 26 |
| Figura 3.14: Botón de pausa y stop .....                                  | 27 |
| Figura 3.12: Ejecución de un VI- RUN.....                                 | 27 |
| Figura 3.13: Botón para la finalización de un VI.....                     | 27 |
| Figura 3.15: Modulación FSK.....  | 28 |
| Figura 3.16: Parámetros de la modulación FSK.....                         | 29 |
| Figura 3.17: Demodulador detector de FM.....                              | 30 |
| Figura 3.18: Espectro de la señal FSK .....                               | 30 |
| Figura 3.19: Interferencias en señal FSK .....                            | 31 |
| Figura 3.20: Desvanecimiento selectivo .....                              | 31 |
| Figura 3.21: Demodulador de filtro adaptado FSK.....                      | 32 |
| Figura 3.22: Demodulador de FSK no coherente.....                         | 32 |
| Figura 3.23: Efecto de la señal con umbrales fijos y variables .....      | 33 |
| Figura 3.1: Diagrama de bloques funcional .....                           | 34 |
| Figura 3.2: Starter Kit 03.....   | 36 |
| Figura 4.1: Diagrama de recepción con el USRP.....                        | 37 |
| Figura 4.2: Diagrama de bloques de la primera versión.....                | 41 |
| Figura 4.3: Diagrama de bloques de la primera versión (I).....            | 42 |
| Figura 4.4: Diagrama de bloques de la primera versión (II) .....          | 44 |
| Figura 4.5: Diagrama de bloques de la segunda versión.....                | 45 |
| Figura 4.6: Estructura Case (FALSE) .....                                 | 45 |
| Figura 4.7: Diagrama de bloques de la tercera versión .....               | 48 |
| Figura 4.8: Producer/Consumer Design Pattern (Data).....                  | 49 |
| Figura 4.9: Modelo Producer/Consumer .....                                | 49 |
| Figura 4.10: Diagrama de bloques de la última versión (Adquisición).....  | 51 |
| Figura 4.11: Examen de tiempos.....                                       | 52 |
| Figura 4.12: Diagrama de bloques de la última versión (Demodulación)..... | 52 |
| Figura 5.1: Montaje del SK-03 en el Analizador .....                      | 54 |
| Figura 5.2: Montaje del SK-03 para cálculo del BER.....                   | 55 |
| Figura 5.3: Gráfica de la sensibilidad del NI-USRP con SK-03 .....        | 56 |
| Figura 5.4: Análisis sensibilidad SK-03 TX y RX.....                      | 57 |
| Figura 5.5: Montaje del Híbrido en el Analizador sin conversión .....     | 58 |
| Figura 5.6: Montaje del Híbrido en el Analizador con conversión .....     | 58 |
| Figura 5.7: Montaje del Híbrido para cálculo del BER.....                 | 59 |

|   |    |
|---|----|
| Figura 5.8: Gráfica de la sensibilidad del NI-USRP con el Híbrido ..... | 60 |
| Figura 5.9: HiDAC .....   | 61 |
| Figura 5.10: Montaje del HiDAC en el Analizador sin conversión .....    | 62 |
| Figura 5.11: Frecuencia del cero con el HiDAC (sin conversión) .....    | 62 |
| Figura 5.12: Frecuencia del uno con el HiDAC (sin conversión) .....     | 63 |
| Figura 5.13: Montaje del HiDAC en el analizador con conversión .....    | 63 |
| Figura 5.14: Frecuencia del cero con el HiDAC (con conversión) .....    | 64 |
| Figura 5.15: Frecuencia del uno con el HiDAC (con conversión).....      | 64 |
| Figura 5.17: Esquema del montaje del HiDAC para cálculo del BER.....    | 65 |
| Figura 5.16: Montaje del HiDAC para cálculo del BER.....                | 65 |
| Figura 5.18: Gráfica de la sensibilidad del NI-USRP con el HiDAC .....  | 66 |



# ÍNDICE DE TABLAS

|   |    |
|---|----|
| Tabla 3.1: Transmisor USRP .....                                    | 20 |
| Tabla 3.2: Receptor USRP .....                                      | 21 |
| Tabla 3.1: Características del ML5805 como transmisor.....          | 35 |
| Tabla 5.1: Análisis de frecuencias del SK-03 .....                  | 54 |
| Tabla 5.2: Parámetros de entrada del SK-03 .....                    | 55 |
| Tabla 5.3: Análisis de frecuencias del Híbrido sin conversión ..... | 58 |
| Tabla 5.4: Análisis de frecuencias del Híbrido con conversión ..... | 58 |
| Tabla 5.5: Parámetros de entrada del Híbrido.....                   | 59 |
| Tabla 5.6: Parámetros de entrada del HiDAC.....                     | 66 |

# **BIBLIOGRAFÍA**

- [1] José Manuel Huidobro, “Historia de los satélites de comunicaciones”, COIT. Available: [http://www.coit.es/foro/pub/ficheros/historia\\_de\\_los\\_satelites\\_de\\_comunicaciones.\\_bit\\_134.\\_5c6c417a.pdf](http://www.coit.es/foro/pub/ficheros/historia_de_los_satelites_de_comunicaciones._bit_134._5c6c417a.pdf)
- [2] Emxys, “NaoSAT”, Available: <http://www.naosat.com/>
- [3] M. Bayarri, “Diseño del subsistema de comunicaciones del satélite Politech.1”, Universidad Politécnica de Valencia, 2011.
- [4] M. Ferrando, “Análisis y caracterización del sistema transceptor del satélite Politech.1”, Universidad Politécnica de Valencia.
- [5] “CubeSat Proposal – Politech.1”, Universidad Politécnica de Valencia.
- [6] National Instruments. Available: [www.ni.com](http://www.ni.com)
- [7] Robert W. Heath Jr., “Digital Wireless Communication. Physical Layer Exploration Lab using the NI USRP”, National Technology & Science Press.
- [8] Hector Cifuentes, “Tutorial LabVIEW”. Universidad Distrital “Francisco Jose de Caldas”, 2001. Available: <http://perso.wanadoo.es/jovilve/tutoriales/016tutorlabview.pdf>
- [9] G. A. Holguín, S. M. Pérez, A. A. Orozco, “Curso básico LabVIEW 6i”. Universidad Tecnológica de Pereira. Available: <http://www.ie.itcr.ac.cr/einteriano/control/Labview/ParaAprender/Curso%20LabVIEW6i.pdf>
- [10] H. P. Halvorsen, “Introduction to LabVIEW”. Telemark University College. Available: <http://home.hit.no/~hansha/documents/labview/training/Introduction%20to%20LabVIEW/Introduction%20to%20LabVIEW.pdf>
- [11] José Rafael Lajara, José Pelegrí. “LabVIEW: Entorno gráfico de programación”. National Instruments.
- [12] H. R. Rueda. “Modulación Digital FSK”, 2011.
- [13] Bob Watson. “FSK: Signals and Demodulation”. The Communications Edge.
- [14] D. Adionel. “Digital Transmission”. Signals and Communication Technology.
- [15] “RFMD ML5805 Data sheet”. Available: <http://www.rfmd.com/CS/Documents/ML5805DS.pdf>

