



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

IMPLEMENTACIÓN DE TRANSMISOR SATELITAL
EN RADIO DEFINIDA POR SOFTWARE PARA
ALERTA DE EMERGENCIAS

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

Ingeniero en Telecomunicaciones

PRESENTAN:

Armando Camilo Flores Huelgas
Eloísa de Lourdes García García



DIRECTOR DE TESIS:
Dr. José María Matías Maruri

Ciudad Universitaria, CDMX, 2022



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*A mi mamá, por las enseñanzas y el cariño que me brindaste
A Mauricio por siempre creer en mí
Eloisa García*

*Dedico este trabajo a mi familia, a quienes he aprendido a valorar mucho más en este
tiempo de pandemia.
A mis profesores por su gran labor y por saberme transmitir sus conocimientos.
A mis amigos por ser la chispa de motivación todos los días
Camilo Flores*

Agradecimientos

Gracias al proyecto PAPIIT “Propuesta de sistema de alerta de emergencias global y escalable para México” con la clave IT102519 por el incentivo económico para el desarrollo de este trabajo.

Agradezco a mi tutor, mentor y gran amigo, José María, por haberme introducido a las telecomunicaciones y haberme acompañado en este viaje.

A las personas que más admiro y a quienes dedico todos mis logros, mis padres, que me han guiado hasta el punto en el que me encuentro.

Agradezco a mi papá Armando, porque nunca le faltó un consejo para darme, incluso cuando no tuve la humildad para pedir uno.

A mi mamá Patricia, por ser mi soporte emocional y la persona en quien más confío.

A quien me acompañó estudiando por las noches, mi hermana Aura, por ser la persona que me escucha cuando más lo necesito.

Y por último agradezco a mi compañera y amiga Eloísa, por atreverse a formar parte de este trabajo conmigo.

Camilo Flores

Gracias a mi mamá por alentarme a ser mejor, por brindarme las herramientas para hacer posible desarrollarme en mi vida académica y personal. Gracias a mi familia, por ser pilar de mi formación.

Agradezco a Mauricio por compartir conmigo esta etapa, siempre brindarme su apoyo para seguir adelante y motivar a fijarnos nuevas metas juntos.

Agradezco a Camilo por ser un gran amigo y compañero, por la confianza de aventurarnos a desarrollar este trabajo.

Agradezco especialmente al Dr. José María Matías por su confianza al integrarme a este proyecto, por los conocimientos compartidos y consejos brindados, por hacer posible la realización de este trabajo.

Agradezco al proyecto PAPIIT ”Propuesta de sistema de alerta de emergencias global y escalable para México” con clave IT102519 por las enseñanzas que me deja formar parte de este proyecto y el apoyo económico otorgado, así como a todas las personas que han colaborado en él.

Eloisa García

Declaración de autenticidad

La información presentada en este trabajo se obtuvo de diversas fuentes que se consideran fidedignas y se consignan puntualmente en las referencias. El uso dado a la información es de naturaleza estrictamente de investigación académica y de divulgación, sin fines de lucro o de otra índole. Se ha hecho también el mayor esfuerzo por acreditar debidamente datos, opiniones y contenidos presentados, por lo que cualquier error u omisión en ello, es del todo involuntario

ARMANDO CAMILO FLORES HUELGAS
ELOÍSA DE LOURDES GARCÍA GARCÍA
Ciudad Universitaria, CDMX, 2022

Índice general

Índice de figuras	ix
Acrónimos	1
Resumen	3
1. Introducción	5
1.1. Objetivos	6
1.2. Metodología	6
2. Estado del Arte	9
2.1. Sistemas de Alerta de Emergencia	9
2.1.1. Sistemas de Alerta de Emergencias en México	10
2.2. Descripción y uso de Protocolo CAP	11
2.2.1. Estructura del mensaje de alerta CAP	11
2.2.2. Ejemplo de alerta CAP	13
2.3. Los SDR y GNU Radio	14
3. Rehabilitación de la Estación Terrena de Seguimiento Satelital de la Facultad de Ingeniería.	17
3.1. Estado inicial	17
3.2. Acciones de Rehabilitación	19
3.2.1. Limpieza de los equipos	20
3.2.2. Diagrama de Conexiones	20
3.2.3. Configuración de Software	23
3.2.4. Configuración del TNC	24
3.2.5. Calibración mecánica de los rotores.	25
3.2.6. Comprobación de la antena UHF	25
3.3. Seguimiento y Recepción de Satélites LEO	26
4. Transmisor CAP Satelital	29
4.1. Requerimientos	29
4.1.1. Generador de mensajes CAP	29

ÍNDICE GENERAL

4.1.2. Hardware	31
4.1.3. Plataforma de desarrollo	33
4.1.4. Modulaciones	33
4.1.5. Desplazamiento Doppler	33
4.2. Desarrollo del transmisor SDR	35
4.2.1. Estructura del transmisor	35
4.2.1.1. Transmisor	37
4.2.1.2. Secciones de prueba y visualización	42
4.3. Efecto Doppler	45
4.3.1. Cálculo del efecto Doppler	46
4.3.2. Corrección de Frecuencia	47
4.3.2.1. Análisis del protocolo de comunicación entre Hamlib y Gpredict	48
4.3.2.2. Desarrollo de la interfaz entre Gpredict y el transmisor SDR	50
4.4. Unión de la interfaz con GNU Radio	53
4.4.1. Entrada de Comandos del USRP Sink	53
4.4.2. ZMQ	54
4.4.3. XML-RPC	57
5. Pruebas de funcionamiento y resultados obtenidos	61
5.1. Pruebas de los datos transmitidos	62
5.2. Verificación de la generación de modulaciones	65
5.3. Pruebas de transmisión en VHF y UHF	69
6. Conclusiones	77
Anexo	81
Anexo I. GNU Radio	81
Anexo II. Gpredict	84
Anexo III. Hamlib	92
Referencias	95

Índice de figuras

2.1. Decesos a causa de ciclones tropicales antes y después de la instalación del Sistema de Alerta temprana para Ciclones Tropicales (SIAT-CT) Fuente: CENAPRED [1]	10
3.1. Estación Terrena FI	18
3.2. Radio transceptor ICOM 910-H. Antes y después de la limpieza	20
3.3. Diagrama de conexiones de la Estación Terrena	21
3.4. TNC Kantronics Packet Communicator 9612+ y fuente de poder Astron RS-35A	21
3.5. Antenas tipo yagi VHF y UHF de la estación terrena	22
3.6. Interfaz principal del Orbitron	24
3.7. Ejemplo de archivo con datos TLE	24
3.8. Izquierda: Resultados medidos de VSWR a la antena de UHF. Derecha: valores de VSWR especificados por el fabricante. Fuente: hy-gain[2]	26
3.9. Imagen de la recepción de datos digitales de la ISS	28
4.1. Herramienta Generadora de mensaje CAP Fuente: cap.org [3]	30
4.2. Diagrama de estación terrena con USRP.	32
4.3. Diagrama de Bloques del Transmisor	36
4.4. Secciones del Diagrama de Bloques del Transmisor	37
4.5. Sección 1 del transmisor	38
4.6. Vector Source Properties	39
4.7. Propiedades de el bloque “File Source”	40
4.8. Sección 2 del transmisor	40
4.9. Sección 3 del transmisor	41
4.10. Sección 4 del transmisor	42
4.11. Interfaz de “Time Sink”	43
4.12. Interfaz de “Frequency Sink”	43
4.13. Interfaz de “Waterfall sink”	44
4.14. Sección 5 del transmisor	44
4.15. Representación gráfica del efecto Doppler	45
4.16. Ejemplo de la comunicación entre Hamlib y Gpredict	48
4.17. Conversación entre Hamlib y Gpredict	49

ÍNDICE DE FIGURAS

4.18. Ejecución del programa	52
4.19. Comunicación Gpredict y programa en Python	52
4.20. Conversación entre Gpredict y programa en Python	53
4.21. Bloque USRP sink	53
4.22. Comandos disponibles Fuente: GNU Radio[4]	54
4.23. Bloque ZMQ Pull	55
4.24. Comunicación entre Gpredict y GNU Radio por medio del programa en Python	56
4.25. Comunicación por XMLRPC entre Gpredict y GNU Radio	59
5.1. Diagrama del sistema de pruebas en simulación de GNURadio	61
5.2. Vector de bits definido para pruebas	62
5.3. Simulación de transmisión de vector de bits por modulación BPSK sin ruido en el canal	63
5.4. Simulación de transmisión de vector de bits por modulación BPSK con ruido en el canal	64
5.5. Simulación de transmisión de vector de bits por modulación BPSK con error en demodulación	64
5.6. Pruebas de Modulación BPSK	65
5.7. Pruebas de Modulación QPSK	66
5.8. Pruebas de Modulación DQPSK	67
5.9. Pruebas de Modulación 8PSK	68
5.10. Pruebas de Modulación 16QAM	69
5.11. Diagrama del sistema de pruebas para transmisión RF	69
5.12. Sistema de Captura en GNURadio [5]	70
5.13. Ruido en Receptor	71
5.14. GPredict y programa en Python transmitiendo la frecuencia al transmisor.	71
5.15. Desplazamiento de la señal por el efecto Doppler captada en el receptor	72
5.16. Posición inicial del satélite VHF	73
5.17. Corrimiento de la frecuencia en VHF	74
5.18. Gpredict: Posición inicial de satélite para prueba UHF	74
5.19. Desplazamiento de la frecuencia de la señal por efecto Doppler en UHF	75
5.20. Esquema tipo Waterfall del desplazamiento de frecuencia en la señal UHF	76
A1. Interfaz del Gpredict	85
A2. Datos de un satélite mostrados en GPredict	86
A3. Menú "Preferencias" del GPredict	87
A4. Agregar una estación terrena	88
A5. Configuración de radio	88
A6. Creación de un nuevo módulo	89
A7. Seguimiento del satélite	90
A8. Radio control	90

Acrónimos

- ACK: Acknowledge
- API: Application Programming Interface
- ATSC: Advanced Television Systems Committee
- CAP: Common Alerting Protocol
- CENAPRED: Centro Nacional de Prevención de Desastres
- DSP: digital signal processors
- EAS: Emergency Alert System
- FM: Frecuencia Modulada
- FPGA: field programmable gate arrays
- GPP: General purpose processors
- GPS: Global Positioning System
- GUI: Graphical User Interface
- ISIS: Innovative Solutions In Space
- LEO: Low Earth Orbit
- OOT: Out of Tree
- PMT: Polimorphic Types
- PSH: Push
- PSK: Phase Shift Keying
- PSoC: programmable system on chip
- ROE: Relación de Onda Estacionaria

ACRÓNIMOS

- RRC: Root Raised Cosine
- SASMEX: Sistema de Alerta Sísmica Mexicano
- SDR: Software Defined Radio
- SFN: Single Frequency Network
- SIAT-CT: Sistema de Alerta Temprana para Ciclones Tropicales
- SINAPROC: Sistema Nacional de Protección Civil
- SSN: Servicio Sismológico Nacional.
- TLE: Two Line Element
- TNC: Terminal Node Controller
- UHF: Ultra High Frequency
- USRP: Universal Software Radio Peripheral
- VHF: Very High Frequency
- VSWR: Voltage Standing Wave Ratio
- XML: Extensible Markup Language
- ZMQ: Zero Message Queue

Resumen

Los sistemas de alerta de emergencia tienen un papel fundamental en la prevención y manejo de daños en casos de eventos con posibilidad de tener efectos catastróficos, como sismos, huracanes o tsunamis. En México, los mayores daños se presentan en las localidades más marginadas [1], por lo que contar con un sistema de alerta de emergencias satelital de alta cobertura, donde no llegan los medios de comunicación tradicionales, supondría un gran beneficio para la sociedad.

En el presente trabajo se expondrá el desarrollo de un transmisor satelital de mensajes de alerta de emergencia. Para el desarrollo este transmisor se optó por usar radio definida por software, que es una tecnología que permite implementar en software componentes dedicados a realizar tareas de capa física.

El desarrollo del transmisor está pensado para formar parte de un sistema completo de transmisión y recepción de alertas de emergencia vía satelital, por lo cual se adaptó para uso con el equipo de la estación terrena de la Facultad de Ingeniería.

Para poder hacer uso de la estación terrena se hicieron trabajos de rehabilitación. Las actividades realizadas para la rehabilitación de la estación terrena comenzaron con el reconocimiento y evaluación de la misma, continuando con la calibración de equipos, y hasta la actualización o reemplazo de software. Para comprobar que la estación se encontraba en estado funcional, se realizaron ejercicios de seguimiento de satélites de órbita baja, en las bandas de UHF y VHF. Y con ello se logró recibir con éxito tanto señales digitales como analógicas.

A continuación se hizo un análisis de los requerimientos del transmisor satelital y se seleccionaron diversas herramientas tanto de software como de hardware.

Dentro del software se hizo uso de GNU Radio como plataforma de desarrollo, Gpredict como herramienta para el seguimiento de satélites, incluyendo el cálculo del posicionamiento de las antenas y la corrección del efecto Doppler. Por último, se desarrolló un software especializado en python para realizar la comunicación entre Gpredict y GNU Radio, lo que permitió la corrección automática de la frecuencia.

Como parte del sistema de alerta de emergencias se decidió usar el protocolo CAP, que es un estándar internacional para el intercambio de alertas de emergencia, por lo que se realizó una investigación de herramientas para la generación de mensajes en protocolo CAP, finalmente se hizo uso del generador provisto por CENAPRED, la autoridad responsable de la prevención de desastres en México.

RESUMEN

Además de lo anterior se desarrolló un transmisor SDR que recibe a la entrada los mensajes de protocolo CAP en formato xml, permite seleccionar el tipo de modulación digital entre BPSK, QPSK, DQPSK, 8PSK y 16QAM, así como seleccionar la frecuencia de transmisión, potencia y seleccionar el satélite al que se desea transmitir el mensaje, con ayuda de Gpredict.

Se realizaron pruebas para corroborar el buen funcionamiento del transmisor, y se verificó la correcta generación de las modulaciones, el ancho de banda de la señal transmitida y se observó el correcto funcionamiento del sistema implementado para la corrección del efecto Doppler.

Introducción

Las tareas de alertamiento ante desastres por fenómenos naturales y actividades humanas toman gran relevancia en el mundo debido al alto riesgo que estos representan. En México, durante los años 2000-2018 se perdieron 10,228 vidas debido a este tipo de desastres, los daños materiales en este periodo ascendieron a 2,357 millones de dólares en promedio por año. Una constante que resalta es que los mayores daños se presentan en las localidades más vulnerables y altamente marginadas del país [1].

Los Sistemas de Alertas de Emergencia tienen como objetivo principal proveer información oportuna para la protección de la población y la mitigación de daños. Accidentes con pirotecnia y explosiones por tomas clandestinas de hidrocarburos, son algunos de los eventos que recientemente han requerido del uso de un sistema de alerta en nuestro país [6].

Actualmente, México cuenta con siete sistemas de alertamiento temprano¹, entre ellos el Sistema de Alerta Temprana para ciclones, por cual el número de fallecimientos debido a Ciclones tropicales ha disminuido considerablemente, de acuerdo con datos de CENAPRED. [1]

En el presente trabajo se presentará la implementación de un transmisor parte de un Sistema de Alerta de Emergencia por comunicación satelital, en donde se hace uso de la tecnología de Software Defined Radio (SDR).

El desarrollo de este sistema de alerta de emergencia por comunicación satelital representaría un avance significativo en la infraestructura disponible en nuestro país, ya que permitiría tener alcance en las zonas con mayores afectaciones, mismas que suelen ser de difícil acceso y con alcance limitado a infraestructura de telecomunicaciones o energía eléctrica.

En el capítulo 1 se presentan los objetivos del presente trabajo, así también se describe la motivación y metodología que se han seguido en el proceso. En el Capítulo 2 se abordan los antecedentes y últimos avances en sistemas de alertas de emergencia vía satelital, el uso de protocolos estandarizados y se muestra un panorama general

¹Los sistemas de alertamiento temprano con los que cuenta México son para fenómenos sísmicos, volcánicos, ciclones tropicales, tsunamis, incendios forestales y meteorológicos

sobre el Radio Definida por Software y la plataforma de desarrollo GNU Radio.

En el capítulo 3 se explican las labores realizadas para el acondicionamiento del equipo disponible en la estación terrena de seguimiento satelital de la Facultad de Ingeniería; evaluación, rehabilitación, configuración y actualización, para su integración como parte del transmisor satelital.

En el capítulo 4 se muestra el proceso llevado a cabo para el desarrollo del transmisor satelital en Radio Definida por software, desde el planteamiento de requerimientos hasta la obtención de un sistema completo y funcional.

En el capítulo 5 se presentan los resultados obtenidos, mediante las pruebas realizadas con el transmisor satelital, resaltando las características deseadas contra las obtenidas.

En el capítulo 6 se resume el trabajo realizado, y se exponen las principales conclusiones del mismo. Finalmente, en los anexos se incluye información avanzada de configuración e instalación del software utilizado.

A continuación se expondrán los objetivos de la tesis y la metodología usada.

1.1. Objetivos

El objetivo principal de este trabajo es desarrollar un transmisor satelital mediante el uso de radio definida por software (SDR) para el envío de mensajes de alerta de emergencia.

A partir de este, surgen los siguientes objetivos secundarios:

- Realizar el diagnóstico y rehabilitación de la estación terrena de seguimiento satelital de la Facultad de Ingeniería.
- Realizar seguimiento de satélites de órbita baja.
- Establecer los requerimientos para un transmisor satelital para el envío de mensajes de alerta de emergencias usando el protocolo internacional CAP.
- Acoplar el transmisor satelital desarrollado al sistema de la estación terrena del grupo de investigación.

1.2. Metodología

Para llevar a cabo el desarrollo de este proyecto, se realizó una investigación sobre el estado del arte en materia de Sistemas de Alerta de Emergencia, del protocolo de alertamiento común (protocolo CAP) y del uso de la tecnología de radio definida por software en conjunto con la plataforma de desarrollo GNU Radio, comprendiendo así la factibilidad del desarrollo del transmisor.

Se llevó a cabo la rehabilitación de la estación terrena de comunicación con satélites de órbita baja de la Facultad de Ingeniería. Por lo que primeramente se identificaron las

condiciones iniciales de dicha estación, para después realizar las tareas de rehabilitación que incluyeron: limpieza de los equipos, generación de documentación, actualización y configuración de software, calibración de equipos y pruebas de recepción. Esto resultó necesario debido a que el transmisor en SDR es capaz de generar la señal a transmitir, pero requiere del sistema de antenas y del sistema de seguimiento para llevar a cabo la comunicación con satélites.

Para el desarrollo principal del transmisor satelital de protocolo CAP, se requirió en primer instancia el establecimiento de los requerimientos del mismo, tomando en cuenta los recursos disponibles y las necesidades a cumplir.

Una vez establecido lo anterior, se comenzó con el desarrollo del transmisor en la plataforma GNU Radio, especializada en radio definida por software. Para esto se analizó la programación por bloques, característica de GNU radio, y se evaluó si era necesario el desarrollo de bloques fuera de la biblioteca de GNU.

Por último se unieron los diversos componentes desarrollados para hacer un sistema de transmisión completo y se realizaron las pruebas necesarias para comprobar su correcto funcionamiento.

Estado del Arte

2.1. Sistemas de Alerta de Emergencia

Los Sistemas de Alerta de Emergencia (EAS) se han creado con el propósito de proteger a la población y mitigar los daños provocados por desastres naturales y actividades humanas, proveyendo a la población con información previa sobre posibles riesgos, permitiendo que se preparen para una respuesta efectiva.

Los sistemas de Alerta de Emergencia requieren de 4 componentes fundamentales:[7]

1. Conocimiento previo e identificación de riesgos, para conocer los riesgos asociados a los fenómenos y así llevar a cabo la elaboración de medidas de preparación y protección.
2. Sistemas de medición y monitoreo de los fenómenos, con el fin de realizar pronósticos o emitir avisos.
3. Planes de contingencia, para realizar acciones oportunas y ordenadas ante el impacto de algún fenómeno.
4. Difusión de alertas públicas, en donde se requieren medios de difusión y protocolos preestablecidos, que sean operados por las autoridades correspondientes.

Las telecomunicaciones tienen injerencia de manera directa en los sistemas de medición y monitoreo de los fenómenos naturales, así como en la difusión de las alertas. La radiodifusión fue por mucho tiempo el método más común para la transmisión de Alertas de Emergencias, hoy en día han surgido diferentes protocolos y métodos de transmisión que aprovechan las redes de transmisión de internet, tv y satelitales.

Los satélites dentro de los Sistemas de Alerta de Emergencias pueden tomar parte de la medición y monitoreo, por ejemplo, para la obtención de imágenes satelitales para los sistemas de prevención y erradicación de incendios forestales. [8]

Por otro lado, las comunicaciones satelitales también forman parte de los sistemas de difusión, tomando ventaja de su amplia cobertura, ya que las comunicaciones por

satélite son el medio más efectivo para lograr llegar a áreas fuera de la cobertura terrestre, así como llegar a pasajeros en trenes, aeronaves y buques, o como respaldo cuando los sistemas terrestres no sean fiables o sufran intermitencias.

En los sistemas de alertas de emergencias se requieren sistemas con redundancia para asegurar la transmisión de las alertas de emergencia, a pesar de cualquier condición que se presente.

2.1.1. Sistemas de Alerta de Emergencias en México

México cuenta con diversas Instituciones encargadas de la difusión de alertas de emergencia, como lo es el Servicio Sismológico Nacional SSN, el Centro Nacional de Prevención de Desastres CENAPRED y el Sistema Nacional de Protección Civil SINAPROC, estos organismos se encargan de crear mecanismos para la prevención de desastres, así como planes de respuesta.

Debido a la ubicación geográfica de nuestro país se tiene una alta susceptibilidad a desastres naturales por lo que México cuenta con 7 sistemas de alerta, para fenómenos sísmicos, volcánicos, ciclones tropicales, tsunamis, incendios forestales y meteorológicos en general.[7] Estos son de gran importancia y han demostrado su efectividad en la disminución de daños, como lo muestra la figura 2.1 en dónde se observa la disminución de decesos después de la instalación del Sistema de Alerta por Ciclones Tropicales en el año 2000.

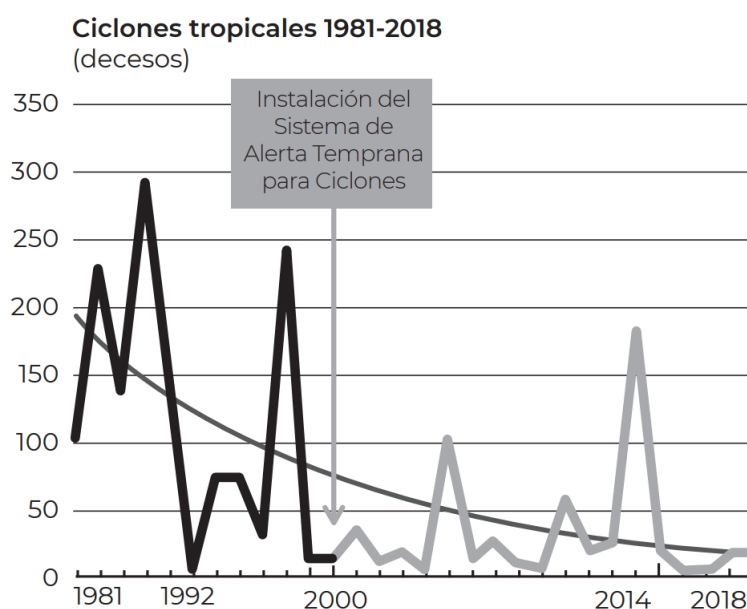


Figura 2.1: Decesos a causa de ciclones tropicales antes y después de la instalación del Sistema de Alerta temprana para Ciclones Tropicales (SIAT-CT)

Fuente: CENAPRED [1]

Una desventaja de estos sistemas es que basan su difusión en boletines y medios de comunicación públicos, como en noticieros de radio y televisión. El único con un sistema de alerta dedicado es el Sistema de Alerta Sísmica Mexicano SASMEX, sin embargo, su cobertura está limitada a ciertas ciudades y requiere de un receptor especial [9].

Debido a la orografía y situación socio-económica de distintas regiones de México, muchas poblaciones aún no cuentan con servicios de comunicación fiables, por lo que resulta muy difícil tener una cobertura amplia en la difusión de las alertas. Lo que ha contribuido a que las zonas más marginadas del país sean las más susceptibles ante los desastres, concentrando la mayor cantidad de daños materiales y humanos [1].

2.2. Descripción y uso de Protocolo CAP

El protocolo CAP (Common Alerting Protocol) es un estándar internacional para el intercambio de alertas de emergencia, permite que la tecnología moderna de información y comunicaciones sean aprovechadas para transmitir alertas de emergencia [10].

A pesar de que se pueden transmitir alertas de emergencia en medios tradicionales de comunicación como la radio y la televisión, se tiene un alcance limitado, por la cobertura de estos medios y por la sustitución por de estos medios por el internet.

En la actualidad se están creando soluciones para los usuarios en línea, por ejemplo Google ha permitido que las alertas de emergencia lleguen a usuarios que usen “Google tools”. Esto posibilita que en muchas herramientas online se distribuya la emergencia, por ejemplo en páginas web que contengan anuncios, se cambiarán por la alerta de emergencia si el usuario se encuentra en una zona cercana [11].

Así como estas muchas otras soluciones se están creando y todo esto es gracias al protocolo CAP. Ya que anteriormente a este estándar las alertas se transmitían por medio de texto plano sin ninguna estructura.

CAP provee un formato de mensaje digital para cualquier tipo de alerta. Además es de propiedad libre. Y no esta creado para alguna aplicación o método de comunicación específico. Y es compatible con diversas tecnologías preexistentes de alertas de emergencia [12].

Además ofrece otras capacidades como la posibilidad de usar información geográfica para dirigir la alerta por medio de coordenadas y representaciones geoespaciales, la posibilidad de mandar el mensaje con distintos lenguajes o para distinto tipo de audiencia, entre otras.

El principal objetivo de usar el protocolo CAP es tener un solo mensaje para una emergencia sin importar la plataforma en la que se vaya a reproducir la alerta, y de esta manera que se active cualquier tipo de sistema de alerta, ya sea público o privado.

2.2.1. Estructura del mensaje de alerta CAP

El mensaje de alerta CAP se guarda con un formato de archivo XML (Extensible Markup Language), lo que quiere decir que es un archivo de texto que consta de etique-

2. ESTADO DEL ARTE

tas y valores para dichas etiquetas. Las etiquetas también se conocen como segmentos puesto que una etiqueta puede contener a su vez otras etiquetas.

Las etiquetas básicas del mensaje CAP son <alert>, <info>, <resource>, <area>.

- El segmento <alert> provee información básica acerca del mensaje, su propósito, fuente y su estatus. La mayoría de los segmentos <alert> contendrán por lo menos un segmento <info>.
- La etiqueta <info> describe algún evento que se presenta en el futuro o que se este presentando en ese momento en términos de urgencia, severidad y certeza. También puede proveer instrucciones para una correcta respuesta conteniendo datos como la duración del fenómeno, información de contacto, links a otro tipo de información. Se pueden usar múltiples etiquetas <info> para describir diferentes parámetros o también para escribir la alerta en distintos idiomas. Un bloque <info> puede contener una o mas etiquetas de <area> y/o <resource>
- La etiqueta <resorce> es utilizada opcionalmente si es que en el bloque <info> se requiere algún recurso digital como una imagen o un audio.
- La etiqueta <area> describe el área geográfica donde aplica el mensaje de alerta. Aquí se pueden introducir tanto descripciones textuales, direcciones o códigos postales, así como con representaciones geoespaciales, con coordenadas de latitud, longitud y altitud.

Existen más etiquetas que se derivan de estas cuatro principales, es decir, subetiquetas que contienen más información sobre la alerta que se quiere transmitir. Por ejemplo hay tres etiquetas muy importantes que se encuentran dentro del bloque <info>, que son: “urgency”, “severity”, “certainty”.

- Urgency: Hace referencia a qué tan pronta debe ser la toma de acción. Puede tomar valores de inmediatamente, esperado (en la siguiente hora), futuro, pasado (ya no se necesita acción) y desconocido.
- Severity: Se refiere a qué tan peligrosa es la amenaza que se está alertando. Toma valores de extrema, severa, moderada, menor y desconocida.
- Certainty: Indica con qué certeza ocurriría la amenaza que se alerta, y toma valores de: Observada (si está ocurriendo en ese momento), probable ($p > 50\%$), posible ($p < 50\%$), improbable ($p \approx 0$), y desconocido.

Para obtener información acerca de otras etiquetas o datos relacionados con el mensaje CAP se le invita al lector a revisar la documentación¹ del protocolo.

¹<http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2.html>

2.2.2. Ejemplo de alerta CAP

El siguiente código es un ejemplo de mensaje CAP que alerta de una inundación en Rwanda. Fue tomado del mismo generador antes mencionado.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
  <?xml-stylesheet type="text/xsl" href="https://cap-sources.s3.amazonaws.
    com/mx-cenapred-es/alert-style.xsl" ?>
3 <cap:alert xmlns:cap="urn:oasis:names:tc:emergency:cap:1.2" >
  <cap:identifier>urn:oid:2.49.0.1.484.1.2021.6.28.23.38.28</
    cap:identifier>
5 <cap:sender>meteo.rwanda@gmail.com</cap:sender>
  <cap:sent>2021-06-28T18:38:28-05:00</cap:sent>
7 <cap:status>Test</cap:status>
  <cap:msgType>Alert</cap:msgType>
9 <cap:scope>Public</cap:scope>
  <cap:info>
11 <cap:language>es</cap:language>
  <cap:category>Met</cap:category>
13 <cap:event>flood</cap:event>
  <cap:urgency>Expected</cap:urgency>
15 <cap:severity>Minor</cap:severity>
  <cap:certainty>Possible</cap:certainty>
17 <cap:expires>2021-06-30T18:38:28-05:00</cap:expires>
  <cap:senderName>Meteo Rwanda</cap:senderName>
19 <cap:headline>Flood warning for Musanze Northern Province , Rwanda</
    cap:headline>
  <cap:description>Heavy rains in Musanze Northern Province of Rwanda
    will lead to flooding , with likely extensive damage to houses and
    property in an area of 1,000 square kilometers. Rwanda Red Cross
    estimates that 1,000 households may be affected.</cap:description>
21 <cap:instruction>If you are in the warning area , avoid low-lying areas
    or move to higher ground immediately. Residents living along streams
    and creeks should take immediate precautions to protect life and
    property. Do not attempt to cross swiftly flowing waters or waters of
    unknown depth. Continue listening to local media as updates will be
    provided as conditions change. For emergency assistance , call 112.</
  
```

```
cap:instruction>
<cap:web>http://www.meteorwanda.gov.rw</cap:web>
23 <cap:contact>meteo.rwanda@gmail.com</cap:contact>
<cap:area>
25 <cap:areaDesc>Musanze Northern Province of Rwanda</cap:areaDesc>
<cap:polygon>-1.63,29.41 -1.63,29.74 -1.38,29.74 -1.38,29.41
-1.63,29.41</cap:polygon>
27 </cap:area>
</cap:info>
29 </cap:alert>
```

2.3. Los SDR y GNU Radio

El transmisor realizado para este proyecto de tesis con desarrolló en una tecnología conocida como radio definida por software o SDR por *Software Defined Radio*. Un SDR se define como un radio, cuyos componentes dedicados a realizar funciones de capa física, están implementados en software [13].

La ciencia y la tecnología avanzan rápidamente y sobre todo las tecnologías de la información y comunicaciones van desarrollandose de manera exponencial por lo que al tener un radio convencional, el cual tiene sus componentes en hardware, es difícil que se adapte a las nuevas tecnologías. En un radio convencional para poder modificar el protocolo, tipo de modulación o frecuencia de trabajo habría que cambiar componentes físicos, lo que no solo es complicado sino también costoso.

Un radio convencional tiene la limitante de que no puede trabajar con múltiples plataformas, es decir, si el radio transmisor es un cierto tipo, el radio receptor debe ser del mismo. Y no puede cambiar su modo de trabajo al momento. En ciertas aplicaciones, como por ejemplo en telefonía celular, esto es sumamente importante pues la estación base debe cambiar ciertos parámetros de la transmisión como la modulación, dependiendo de la localización del usuario [14].

Para cambiar el modo de operación en una radio convencional se necesitaría forzosamente de un cambio manual. En contraste un SDR puede cambiar entre múltiples protocolos de radio en tiempo real, simplemente haciendo la modificación correspondiente en el software de configuración.

Los SDRs realizan distintas tareas de procesamiento de señales en una computadora de uso general o con componentes electrónicos reconfigurables. Es decir, contienen diversas piezas de hardware, que junto con el software pueden ser programadas para realizar los trabajos de capa física de un radio convencional.

Entre los componentes principales de un SDR se encuentran la FPGAs *field programmable gate arrays*, los DSPs *digital signal processors*, los GPPs *general purpose processors* y los PSoCs *programmable system on chip*, entre otros elementos.

En el mercado existen diversas marcas y tipos de SDRs, la elección del modelo que se utilizó para el desarrollo del transmisor se expondrá en el capítulo 4.

Para el desarrollo de programas implementados en SDR existen varias plataformas. A continuación se hablará de GNU Radio, que es la plataforma que se utilizó para desarrollar el transmisor CAP satelital. La razón por la cual se optó por esta plataforma en particular se expondrá de igual manera en el capítulo 4.

GNU Radio es un software de código libre, es una plataforma de desarrollo que provee bloques de procesamiento de señales para implementar radios definidas por software [15]. También puede utilizarse sin hardware para tener un ambiente de simulación [16].

GNU Radio realiza el trabajo de procesamiento de señales, puede ser utilizado para recibir o transmitir flujos digitales y procesarlos para posteriormente ser transmitidos mediante hardware.

Dentro de su biblioteca contiene distintos bloques de procesamiento, como filtros, ecualizadores, demoduladores, codificadores. Estos bloques se conectan entre sí para formar el flujo por el cual los datos digitales van siendo procesados. Entre cada bloque se pueden usar distintos tipos de datos como bits, bytes, y números complejos [15].

Aparte de los bloques que incluye la biblioteca base de GNU Radio, se pueden agregar bloques creados por terceros, o se pueden desarrollar bloques propios. La mayoría de bloques están desarrollados en C++, mientras que la interfaz gráfica esta desarrollada con Python [15].

GNU Radio provee una interfaz gráfica donde se pueden manipular los bloques y conectarlos entre sí, con esto es suficiente para desarrollar diversas aplicaciones de procesamiento de señales.

El uso de SDR permite trabajar con gran flexibilidad en la elección de modulaciones, protocolos y frecuencias, por esta razón se decidió usar esta tecnología en el presente trabajo, haciendo que este desarrollo sea fácilmente adaptable en un futuro para nuevos requisitos.

Rehabilitación de la Estación Terrena de Seguimiento Satelital de la Facultad de Ingeniería.

En la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, en el edificio Q “Luis G. Valdés Vallejo”, perteneciente a la División de Ingeniería Eléctrica, se encuentra una Estación Terrena para satélites de órbita baja. Cuenta con equipo apto para trabajar en las bandas VHF y UHF y la recepción y transmisión analógica y digital.

Dicha estación se encontraba en desuso y sin un mantenimiento periódico por lo que se requirió llevar a cabo acciones de rehabilitación para ponerla de nuevo en funcionamiento, esto con el fin de adaptar toda la infraestructura para su uso con el equipo de Radio Definida por Software SDR (Software Defined Radio). El uso de éste sistema embebido supone mucha más flexibilidad para trabajar con distintos tipos de mensajes, modulaciones y protocolos.

3.1. Estado inicial

La estación terrena fue encontrada con el siguiente equipo, mostrado en la figura 3.1:

- Fuente de Poder Astron RS-35A.
- Terminal Node Controller Kantronics Packet Communicator TNC KPC 9612+.
- Controlador digital de posicionamiento EA4TX ARS-USB.
- Transceiver Icom IC-910H.

3. REHABILITACIÓN DE LA ESTACIÓN TERRENA DE SEGUIMIENTO SATELITAL DE LA FACULTAD DE INGENIERÍA.



Figura 3.1: Estación Terrena FI

- Controlador dual elevación-azimut G5500 Yaesu.
- Regulador de voltaje TDE MAX 1000.
- Antena VHF Hy-Gain 216 SAT.
- Antena UHF Hy-Gain UB-7030 SAT.
- Computador.
- Periféricos de PC.
- Cables de conexión.
- Adaptadores.
- Mobiliario.

En una primera instancia se hizo un reconocimiento general sobre el estado del equipo mostrado en la figura 3.1 para verificar su funcionamiento, por lo que se procedió

a hacer una prueba sin modificar el estado en el que se encontraba la estación, principalmente sin modificar ninguna conexión entre los equipos, ya que esto nos ayudaría a conocer la configuración que se tenía en ese momento.

Con la realización de la prueba se logró verificar el funcionamiento de todos los equipos mencionados anteriormente, excepto el TNC KPC debido a que requeriría una configuración más extensa por medio de software, que se abordará más adelante.

Se puso en funcionamiento el programa Orbitron, un software gratuito que permite la visualización y el seguimiento de satélites con ayuda de bases de datos públicas. Además es capaz, con los drivers adecuados, de controlar una amplia variedad de sistema de rotores y transceptores, lo que permite controlar los equipos y realizar el seguimiento automatizado de los satélites de órbita baja.

Con ello se logró poner en movimiento los rotores de las antenas y sincronizar la frecuencia en el transceptor Icom 910-H, teniendo así el control automático del desplazamiento de la frecuencia debido al efecto Doppler¹ y comprobando con esta prueba qué equipos se encontraban operando adecuadamente.

3.2. Acciones de Rehabilitación

Una vez realizada la prueba descrita anteriormente se establecieron las siguientes acciones a partir de lo observado:

1. Realizar una limpieza de los equipos
2. Establecer y documentar los diagramas de conexiones de los equipos de la estación terrena
3. Conocer el flujo de comunicación ente el software utilizado para la interconexión de los equipos.
4. Actualización de las bases de datos orbitales en el software Orbitron
5. Configurar el controlador de datos TNC Kantronics Packet Communicator
6. Calibrar mecánicamente los rotores.
7. Comprobar el correcto funcionamiento de las antenas
8. Pruebas de la estación terrena mediante el seguimiento y recepción de satélites LEO (analógica y digitalmente)

A continuación se explicarán más detalladamente las acciones realizadas en cada punto.

¹El efecto Doppler es el cambio aparente de la frecuencia debido a la velocidad relativa del satélite con respecto al observador.

3. REHABILITACIÓN DE LA ESTACIÓN TERRENA DE SEGUIMIENTO SATELITAL DE LA FACULTAD DE INGENIERÍA.

3.2.1. Limpieza de los equipos

Los equipos se encontraban en desuso y sin mantenimiento, por lo que era muy notable la cantidad de polvo que los cubría como se muestra en la figura 3.2, afortunadamente ningún equipo mostró un mal funcionamiento debido a ello.

La remoción de polvo de todos los equipos fue una actividad sencilla de realizar, sin embargo fue un paso importante para poder trabajar de manera cómoda con los equipos. Además, se llevó a cabo una selección y organización de los materiales que se encontraban en el espacio destinado para la estación terrena y así se despejó el lugar de trabajo.



Figura 3.2: Radio transceptor ICOM 910-H. Antes y después de la limpieza

3.2.2. Diagrama de Conexiones

La estación terrena se encontró prácticamente con todas las conexiones funcionales, por lo que con la elaboración del diagrama de la figura 3.3 se pretende registrar el arreglo encontrado para comprender el funcionamiento y generar una documentación para su uso en proyectos futuros.

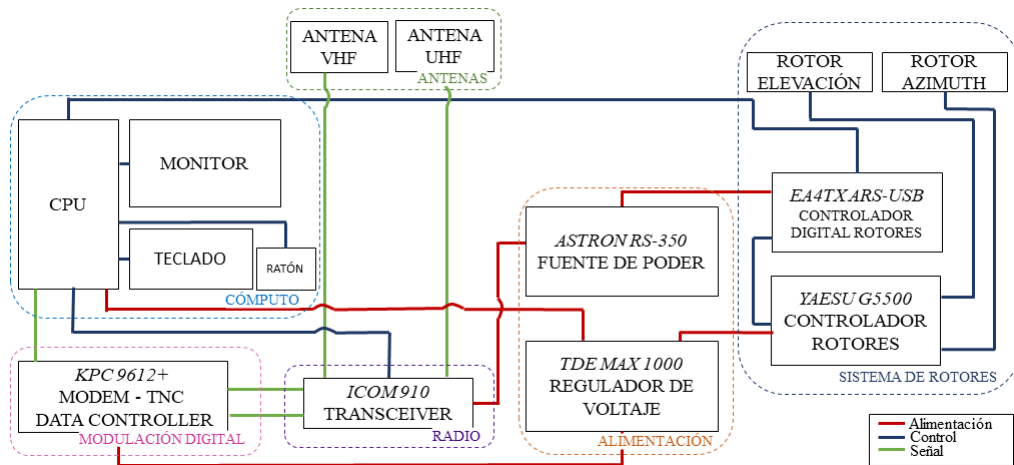


Figura 3.3: Diagrama de conexiones de la Estación Terrena

Para el mejor entendimiento del sistema, el diagrama se dividió en seis grupos, según su función; Computo, Modulación Digital, Radio, Antenas, Alimentación y Sistema de Rotores.

En el equipo de cómputo es dónde se encuentra instalado el software Orbitron, con el que se monitorea el paso de satélites de órbita baja, se realiza el control de rotores para el apuntamiento de las antenas y se sintoniza la frecuencia en el radio, para el seguimiento del paso de los satélites. De igual forma se realiza la conexión con el TNC Data Controller, mostrado en la figura 3.4, para llevar a cabo la modulación y/o demodulación digital.

El radio Icom IC-910H mostrado en la figura 3.2 es un equipo enfocado a radioaficionados por satélite. Cuenta con dos receptores compatibles con DSP de alta sensibilidad, y para la transmisión con 100 watts de potencia de salida en VHF y 75 watts en UHF.



Figura 3.4: TNC Kantronics Packet Communicator 9612+ y fuente de poder Astron RS-35A

3. REHABILITACIÓN DE LA ESTACIÓN TERRENA DE SEGUIMIENTO SATELITAL DE LA FACULTAD DE INGENIERÍA.

La estación cuenta con dos antenas instaladas en el techo de el edificio Q del Anexo de la Facultad de Ingeniería como se muestra en la figura 3.5, una antena para VHF y otra para UHF. Ambas son antenas tipo yagi de polarización circular. La antena para VHF un modelo "216 SAT" de marca Hy-Gain con 16 elementos y 10.7 dBd de ganancia. La antena de UHF es el modelo UB-7030SAT de la misma marca. Tiene 30 elementos y una ganancia de 14 dBd.



Figura 3.5: Antenas tipo yagi VHF y UHF de la estación terrena

El sistema de rotores se compone del sistema mecánico y del sistema de control. El sistema mecánico se compone de dos rotores que controlan la elevación y azimut de las antenas y se encuentran instalados en conjunto con las antenas. El sistema de control se compone del controlador digital EA4TX ARS-USB que recibe las instrucciones enviadas por el software de control, y por el controlador analógico YAESU G5500 que lleva directamente las directivas de movimiento a los rotores.

Por último, para la alimentación de los equipos se cuenta con el regulador TDE MAX 1000, para evitar que las variaciones de voltaje puedan dañar a los equipos, y con la Fuente de Poder Regulada Astron (figura 3.4) para cumplir con los requerimientos especiales del Radio ICOM y el Controlador Digital de Rotores.

3.2.3. Configuración de Software

Para el control y monitoreo de la estación se cuenta con Software especializado para estas tareas. El Software principal de la estación es Orbitron¹, el cual es un software gratuito disponible para Windows. Es un software de seguimiento de satélites en tiempo real que, con ayuda de los datos orbitales permite visualizar la posición del satélite y su huella.

Conociendo los datos de la posición de la estación terrena; el software proporciona los datos de elevación y azimut que la antena deberá tener para optimizar la recepción o transmisión y también hará el cálculo de la frecuencia Doppler, sin embargo, no proporciona la frecuencia central del satélite por lo que el usuario debe definir esta información al software de manera manual.

Para que se pueda enviar las directivas de posicionamiento de las antenas se utiliza la aplicación WispDDE de Windows; que se comunica con el software de predicción Orbitron utilizando el protocolo DDE por un lado y por el otro envía los datos al software de control de los rotores ARSVCOM a través del puerto serie. Se debe configurar el protocolo, el puerto y la tasa de transmisión de datos, por lo que antes de iniciar el seguimiento de cualquier satélite es importante inicializar con anterioridad estas configuraciones.

Los TLE (Two-Line Element)² son un formato de datos que contiene información sobre un objeto en órbita terrestre, esta información se utiliza para proyectar las futuras trayectorias orbitales. Los TLE son públicos, se actualizan con cierta frecuencia para aumentar la precisión de las predicciones y se pueden consultar en diversas páginas web.

En el panel de la derecha de la interfaz principal de Orbitron figura 3.6 se pueden observar una lista con nombres y botones tipo checkbox; esta lista contiene satélites que si los activamos se podrán mostrar en la interfaz gráfica sobre el mapa mundial.

Orbitron provee varios archivos de datos TLE en los que se incluyen todos los satélites que se pueden ver en el panel de la derecha. Para ver esas listas basta con seleccionar la opción del panel derecho "Load TLE" y se mostrarán varios archivos, cada uno contiene sus propios satélites con alguna temática en específico, como por ejemplo, satélites de radio amateur, o satélites de GPS.

En ocasiones es difícil encontrar los TLE de un satélite en específico en las listas predeterminadas de Orbitron, por lo que se recomienda agregar un archivo de datos donde se puedan agregar datos TLE de otras bases de datos³.

Para agregar un archivo propio basta con dirigirse a la carpeta TLE del Orbitron, que por lo general se encuentra en "C:/Program Files (x86)/Orbitron/Tle", y agregar otro archivo de texto. El formato debe coincidir con el mostrado en la figura 3.7 que consiste en el nombre del satélite y a continuación las dos líneas con los datos orbitales que conforman el TLE.

¹Más información sobre Orbitron y descarga en <http://www.stoff.pl/>

²Más información sobre el formato de datos orbitales TLE en <https://celestrak.com/>

³Ejemplo de repositorio de información y datos TLE de satélites <https://www.n2yo.com/>

3. REHABILITACIÓN DE LA ESTACIÓN TERRENA DE SEGUIMIENTO SATELITAL DE LA FACULTAD DE INGENIERÍA.

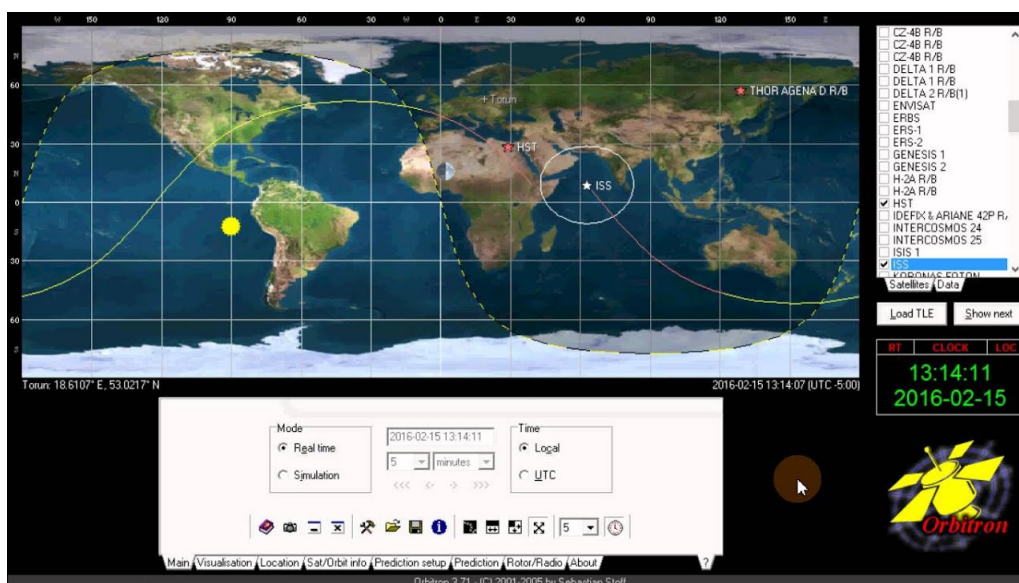


Figura 3.6: Interfaz principal del Orbitron

```

AO-7
1 07530U 74089B 05248.91610499 -.00000028 00000-0 10000-3 0 4935
2 07530 101.6179 293.4407 0012187 77.5622 282.6814 12.53570674409766
AO-10
1 14129U 83058B 05247.26265108 -.00000203 00000-0 10000-3 0 2127
2 14129 27.4417 14.8735 6030438 218.5542 77.4828 2.05866928139198
UO-11
1 14781U 84021B 05249.31115080 .00000497 00000-0 74164-4 0 8721
2 14781 98.2053 248.5365 0010327 98.3944 261.8445 14.79301502153790
    
```

Figura 3.7: Ejemplo de archivo con datos TLE

3.2.4. Configuración del TNC

El TNC (Terminal Node Controller) es un dispositivo que permite a los radios trabajar con modulaciones digitales, ya que puede tanto codificar como decodificar los mensajes que pasen por él. Este equipo se puede conectar y controlar desde una computadora.

El equipo TNC con el que se trabajo tenía como medio de comunicación un cable serie, por lo que hubo que encontrar un adaptador serial a USB y simplemente instalar el driver correspondiente para dicho adaptador.

Ya que este dispositivo no tiene ninguna clase de interfaz propia para su configuración se necesita de un programa para hacer esta tarea. Se decidió realizarlo con un emulador de terminal, en este caso se utilizó el hyperterminal. Para conectar el hyperterminal todo se puede dejar como la configuración por defecto, pero para configurar el puerto COM por donde se realiza la comunicación con el TNC hay que configurarlo

para que se comunique a una tasa de 9600 bits por segundo y con un tamaño de palabra de 8 bits.

Para la comunicación entre el TNC y el radio, el TNC tiene disponibles dos puertos. El puerto 1 es de tipo DSUB9, es considerado un puerto de baja de velocidad ya que solo admite una tasa de 1200 baudios. El puerto 2 es del tipo DSUB15 y acepta tasas de 4800, 9600, 19200 y 38400 baudios, [17] por lo que se considera un puerto de alta velocidad. Es importante verificar qué velocidades admite el radio con el que se trabaja para seleccionar el puerto adecuado.

Para empezar a recibir hay que entrar al modo de comandos ¹y cambiarle algunas configuraciones básicas, como el nombre de la estación, que en el caso de tener un permiso, éste sería el indicativo del radioaficionado.

3.2.5. Calibración mecánica de los rotores.

La calibración de los motores fue necesaria debido al desuso de las antenas, ya que el viento, la lluvia y el peso propio de las antenas hicieron que se desajustara el mecanismo. Es por esto que las dos antenas que estaban sujetas a los rotores no estaban del todo alineadas entre si, mientras una estaba elevada a un cierto ángulo la otra estaba ligeramente caída teniendo un ángulo de elevación distinto.

La calibración se realizó subiendo al poste de las antenas y ajustando sus posiciones manualmente. Se realizaron las pruebas de calibración para verificar que el ángulo que tienen las antenas en elevación y azimut fuera el mismo que se tiene detectado, tanto en el software como en el controlador de rotores. Estas pruebas fueron exitosas quedando las antenas ajustadas tanto en azimuth como en elevación.

3.2.6. Comprobación de la antena UHF

Como se mencionó con anterioridad, la estación cuenta con dos antenas para la recepción de bandas UHF y VHF. Para comprobar el funcionamiento de ambas antenas se realizaron pruebas de recepción en distintas frecuencias y teniendo como objetivo distintos transmisores. En estas pruebas se lograron obtener buenos resultados en la banda de VHF, logrando obtener nivel de recepción de satélites LEO y de una transmisión local de radiocomunicación.

En la recepción en UHF en primera instancia no se logró la recepción y debido al desuso y falta de mantenimiento de la estación no se tenía la certeza de que la antena de UHF funcionara correctamente, por lo que se decidió llevar a cabo una prueba de VSWR (Voltage Standing Wave Ratio). Esta prueba fue de utilidad para tener una medida de la cantidad de potencia que es radiada por la antena y la cantidad de potencia que es reflejada hacia el radio, proporcionando un indicador del desempeño de la antena.

¹Para mayor información de como configurar el TNC consultar el manual de usuario: <https://kantronics.com/wp-content/uploads/2018/10/KPC-9612Plus-Manual-RevG.pdf>.

3. REHABILITACIÓN DE LA ESTACIÓN TERRENA DE SEGUIMIENTO SATELITAL DE LA FACULTAD DE INGENIERÍA.

“El VSWR se define como el cociente del voltaje máximo de la onda estacionaria sobre el voltaje mínimo, también conocido como Relación de Onda Estacionaria ROE”[18] y se calcula de la siguiente forma.

$$VSWR = ROE = \frac{V_{max.}}{V_{min.}} = \frac{1 + |\rho_v|}{1 - |\rho_v|} \quad (3.1)$$

Donde:

ρ_v es el coeficiente de reflexión en la carga

$V_{max.}$ es el voltaje máximo de la onda estacionaria

$V_{min.}$ es el voltaje mínimo de la onda estacionaria

Para llevar a cabo esta prueba se hizo uso de un analizador de redes vectoriales. El analizador de redes vectoriales utiliza el concepto de medir las ondas transmitidas y reflejadas cuando una señal pasa a través de un dispositivo bajo prueba, mismo que nos sirvió para caracterizar el funcionamiento de las antenas en las banda de interés.

Para corroborar que el resultado encontrado fuera correcto, se sustrajo de las hojas de especificaciones del fabricante el correspondiente diagrama de VSWR para su comparación con los resultados obtenidos.

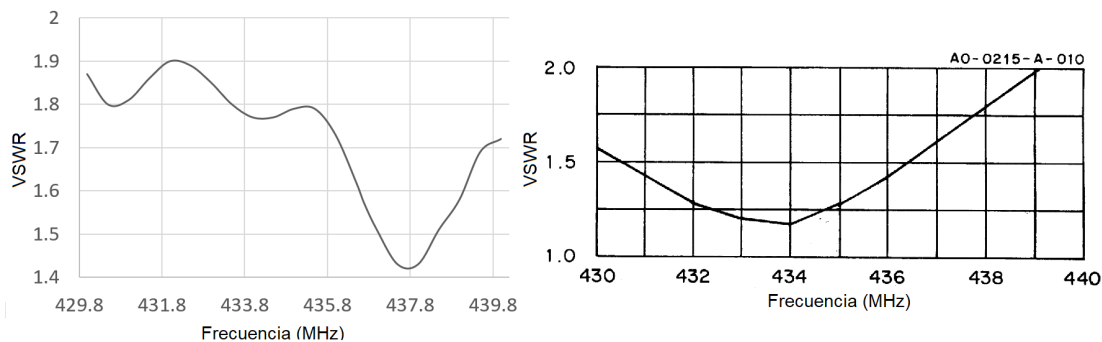


Figura 3.8: Izquierda: Resultados medidos de VSWR a la antena de UHF.

Derecha: valores de VSWR especificados por el fabricante. Fuente: hy-gain[2]

Con los resultados obtenidos podemos corroborar un buen funcionamiento de la antena debido a que en toda la banda analizada se tuvo un relación menor a 2, con lo que se podría proceder a realizar pruebas futuras para comenzar con la recepción en la banda UHF.

3.3. Seguimiento y Recepción de Satélites LEO

Una vez que ya se había realizado un mantenimiento general de la estación y corroborado el funcionamiento de los equipos, el siguiente paso fue hacer un seguimiento activo de satélites de orbita baja, tanto en la anda de VHF como en la de UHF.

En primer lugar se probó la recepción de VHF, por lo que se seleccionó uno de los repositorios incluidos en el Orbitron de satélites de radio amateur. Primero se confirmaba que éste transmitiera con modulaciones analógicas, lo que permitiría observar el buen funcionamiento de la antena de VHF y del radio Icom. Una vez que alguno de estos satélites comenzaba a tener huella sobre la posición de recepción se comenzaba con el seguimiento, ingresando la frecuencia especificada en la documentación disponible del satélite y la modulación (FM, N-FM, AM) así sintonizando el radio con los valores de frecuencia con la corrección Doppler.

La primera recepción exitosa que tuvimos fue en la banda VHF fue con la modulación FM, recibimos información de voz que logramos escuchar a través de radio ICOM. Debido a que el satélite era de uso amateur escuchamos datos como los indicativos de los radioaficionados y saludos entre ellos. Con esta prueba realizada fue hora de comenzar con la recepción de datos digitales en ambas bandas.

Para la prueba de la banda UHF se facilitaron datos sobre el satélite Aistechsat 2, un CUBESAT de dos unidades, transmitiendo su paquete de verificación (beacon) en 436.6 MHz. Se realizó un itinerario de paso del satélite por la posición deseada para poder realizar la mayor cantidad de intentos de recepción. La forma de cerciorarse de que se estaba recibiendo de manera correcta el beacon fue con el monitoreo de la potencia en el ICOM, se realizaron varios intentos de recepción sin éxito por lo que se decidió reducir el rango de pasos a aquellos dónde el satélite pasara casi por encima de la posición deseada para poder receptar con una mayor potencia el beacon, medida que ayudó a lograr en un par de ocasiones registrar el aumento de potencia en dicha frecuencia, asegurando la correcta recepción. Cabe señalar que es usual tener problemas de sensibilidad en este tipo de instalaciones en UHF. Para solucionar este inconveniente se suele usar un preamplificador de antena. El transceiver Icom, que se encuentra instalado, tiene la posibilidad de proveer alimentación para preamplificadores en la banda UHF.

La última prueba que se realizó fue la recepción de datos digitales, en este caso se seleccionó la Estación Espacial Internacional debido a la variedad de tipos de transmisiones y bandas en las que funciona, así como en la frecuencia de paso. Para llevar a cabo esta prueba se requirió configurar el TNC por medio del hyperterminal mostrado en la figura 3.9, así como indicar en el Icom que se estaría recibiendo un tipo de modulación Digital.

3. REHABILITACIÓN DE LA ESTACIÓN TERRENA DE SEGUIMIENTO SATELITAL DE LA FACULTAD DE INGENIERÍA.

```
cmd:PORT 2
PORT was 1
cmd:
KANTRONICS KPC9612PMX VERSION 9.1
(C) COPYRIGHT 2002-2005 BY KANTRONICS INC. ALL RIGHTS RESERVED.
DUPLICATION PROHIBITED WITHOUT PERMISSION OF KANTRONICS.
cmd:TRANS

port
PORT      2
cmd:port 1
PORT was 2
cmd:
KANTRONICS KPC9612PMX VERSION 9.1
(C) COPYRIGHT 2002-2005 BY KANTRONICS INC. ALL RIGHTS RESERVED.
DUPLICATION PROHIBITED WITHOUT PERMISSION OF KANTRONICS.
cmd:trans
cmd:hbaud
HBAUD    1200/9600      |
cmd:trans
>>/1 82A0AA646A9CE0 B08A6682A4AC60 A4A66092A6A6E0 82A0A4A682A860 AE92888A624063
      3E4152495353202D 20496E7465726E61 74696F6E616C2053 7061636520537461 746
96F6E
```

Figura 3.9: Imagen de la recepción de datos digitales de la ISS

De los datos recibidos la segunda línea de datos hexadecimales se convierte según el código ASCII obteniendo ” > *ARISS - International Space Station*” lo que confirma el origen de los datos obtenidos y con esto se asegura el buen funcionamiento de la demodulación Digital.

El seguimiento de satélites se planeó como una actividad que se realizaría de manera recurrente durante la elaboración de este proyecto, por lo que se realizó la configuración de una IP pública para poder acceder remotamente al equipo, sin embargo por la contingencia sanitaria de principios de 2020 no se tuvo acceso de nuevo a la estación terrena y estas actividades quedaron suspendidas.

Al final de la ejecución de las actividades antes descritas, se pudo concluir que todos los equipos de la estación terrena fueron probados y adaptados de manera satisfactoria al sistema, por lo que la estación terrena quedó operativa en su totalidad.

Transmisor CAP Satelital

En este capítulo se explica el procedimiento que se llevó a cabo para el desarrollo del transmisor CAP satelital. Se aborda desde el planteamiento de requerimientos para el transmisor, el generador de mensajes CAP, la creación del transmisor en el software GNU Radio y los métodos utilizados para llevar a cabo la corrección Doppler.

4.1. Requerimientos

Como se mencionó en el Capítulo 3, para tener un transmisor satelital se necesitan de varios componentes, que van desde distintos dispositivos de hardware como el transceiver, las antenas y hasta distintos tipos de software para poder interactuar con el hardware como el Orbitron o el Hyperterminal.

Para el diseño del transmisor satelital se deben seleccionar los componentes de acuerdo a las necesidades del proyecto, de igual forma se deben tener en cuenta otros aspectos como el equipo disponible; que los componentes seleccionados sean de ayuda para futuros proyectos, el costo que tendría comprar nuevo equipo, software o el costo de mantenimiento de estos mismos.

Tomando estos aspectos en cuenta se seleccionaron distintos componentes para el diseño del transmisor satelital, del cuál las partes mas importantes son el transmisor SDR y el control de la frecuencia Doppler. Para un mejor entendimiento la explicación se dividió en cinco categorías, donde se exponen los criterios de selección y la elección final.

4.1.1. Generador de mensajes CAP

Como se mencionó en el capítulo 2, el protocolo CAP establece la transmisión de la alerta mediante un archivo XML. Para generar el mensaje de alerta se investigaron diversas alternativas, desde instalar un software que genera alertas, hasta desarrollar software que completara los campos del mensaje en XML.

4. TRANSMISOR CAP SATELITAL

Debido a que se desea que el presente trabajo se centre en el desarrollo del transmisor SDR se buscó un generador CAP existente. Algo que es importante destacar es que las herramientas que existen para la creación de alertas requieren de un permiso, puesto que no es posible que cualquier persona publique una alerta. Es labor de la autoridad de atención a emergencias de cada país, en México el Centro Nacional de Prevención de Desastres CENAPRED. Por lo que este fue un impedimento para obtener algunas herramientas mas automatizadas.

No obstante, el propio protocolo CAP, con el fin de facilitar el uso del mismo, provee una herramienta en la nube para la creación de alertas. Esta herramienta permite que las autoridades de los países se inscriban y puedan generar mensajes con el protocolo. Además, cualquier persona puede acceder a la herramienta como invitado y ver ejemplos de alertas y experimentar con la herramienta modificando los campos del mensaje.

La herramienta se encuentra en la pagina web [https://cap.alert-hub.org/\(25/11/2021\)](https://cap.alert-hub.org/(25/11/2021)). Hay que seleccionar el país de origen e indicar que se quiere acceder como invitado y se nos mostrará algunos ejemplos de alerta, se selecciona alguno de ellos y se podrá ver una interfaz como la de la figura 4.1.

The image shows a web-based CAP message generator interface. On the left is a form with various fields for creating a message. On the right is a text area displaying the XML code generated from the form inputs.

Form Fields (Left Panel):

- Mostrar/Ocultar XML** | **Validación externa** | **Finalice esta sesión**
- identifier:** urn:oid:2.49.0.1.484.1.2021.6.29.0.31.41
- sender:** christian@usgs.gov | **sent:** 2021-06-28T19:31:41-05:00
- status:** Test | **msgType:** Alert | **scope:** Public
- language:** es | **category:** Met | **responseType:** [dropdown]
- event:** Message from USGS Earthquakes Program
- urgency:** Past | **severity:** Severe | **certainty:** Observed
- onset:** [input] | **expires:** 2021-06-30T19:31:41-05:00
- senderName:** National Earthquake Information Center (NEIC)
- Plantillas de texto de 'headline', 'description', 'instruction' IFRC messaging**
- headline:** Magnitude 7.8 Rat Islands, Aleutian Islands, Alaska
- description:** Felt (IV) on Shemya. A local tsunami generated with recorded wave heights (peak-to-trough) of 50 cm at Shemya and 12 cm at Adak.
- instruction:** [empty text area]
- web:** http://neic.usgs.gov/neis/eq_depot/2003/eq_031117/
- image uri:** http://neic.usgs.gov/neis/poster/2003/20031117.html
- image mimeType:** image/jpeg
- contact:** Roger Hansen, 1-907-474-5533
- areaDesc:** 95 km (60 miles) S of Little Sitkin Island, Alaska
105 km (65 miles) SW of Semisopchnoi Island,
- circle:** 51.13,-178.74 0 | **geocode:** [input]
- format:** lat,long<space>radius | **format:** 'type' = 'value'
- polygon:** [input] | **format:** SW SE NE NW SW (lat,lon points)

XML Code (Right Panel):

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/xsl" href="https://cap-sources.s3.amazonaws.com/mx-cenapred-es/alert-style.xsl" ?>
<cap:alert xmlns:cap="urn:oasis:names:tc:emergency:cap:1.2" >
  <cap:identifier>urn:oid:2.49.0.1.484.1.2021.6.29.0.31.41</cap:identifier>
  <cap:sender>christian@usgs.gov</cap:sender>
  <cap:sent>2021-06-28T19:31:41-05:00</cap:sent>
  <cap:status>Test</cap:status>
  <cap:msgType>Alert</cap:msgType>
  <cap:scope>Public</cap:scope>
  <cap:info>
    <cap:language>es</cap:language>
    <cap:category>Met</cap:category>
    <cap:event>Message from USGS Earthquakes Program</cap:event>
    <cap:urgency>Past</cap:urgency>
    <cap:severity>Severe</cap:severity>
    <cap:certainty>Observed</cap:certainty>
    <cap:expires>2021-06-30T19:31:41-05:00</cap:expires>
    <cap:senderName>National Earthquake Information Center (NEIC)
  </cap:senderName>
  <cap:headline>Magnitude 7.8 Rat Islands, Aleutian Islands,
  Alaska</cap:headline>
  <cap:description>Felt (IV) on Shemya. A local tsunami generated with
  recorded wave heights (peak-to-trough) of 50 cm at Shemya and 12 cm at Adak.
  </cap:description>
  <cap:web>http://neic.usgs.gov/neis/eq_depot/2003/eq_031117/</cap:web>
  <cap:contact>Roger Hansen, 1-907-474-5533</cap:contact>
  <cap:resource>
    <cap:resourceDesc>Image file</cap:resourceDesc>
    <cap:mimeType>image/jpeg</cap:mimeType>
    <cap:uri>http://neic.usgs.gov/neis/poster/2003/20031117.html</cap:uri>
  </cap:resource>
  <cap:area>
    <cap:areaDesc>95 km (60 miles) S of Little Sitkin Island, Alaska
    105 km (65 miles) SW of Semisopchnoi Island, Alaska
    2220 km (1380 miles) WSW of Anchorage, Alaska
    3060 km (1900 miles) W of JUNEAU, Alaska</cap:areaDesc>
    <cap:circle>51.13,-178.74 0</cap:circle>
  </cap:area>
  </cap:info>
</cap:alert>
```

Figura 4.1: Herramienta Generadora de mensaje CAP

Fuente: cap.org [3]

El panel izquierdo de la herramienta, contiene varios cuadros de texto donde se

pueden modificar los campos del mensaje de alerta CAP y los cambios se observan del lado derecho donde se observa como quedaría el archivo XML final.

Esta herramienta satisfizo la necesidad de tener un generador de mensaje CAP, puesto que este proyecto se enfoca en el transmisor de dicho mensaje. Y con este generador se pueden obtener mensajes CAP para usar y probar el transmisor. En un futuro el transmisor se podría acoplar a un sistema automatizado de alertas de emergencia que tenga permiso de publicar alertas.

4.1.2. Hardware

El equipo central de la estación terrena es el transceiver cuya función la cumplía el Radio Icom 910H. Este equipo, como se mencionó anteriormente, está especializado para radioaficionados por satélite, por lo que está limitado a ciertas funcionalidades, en cuanto a los tipos de modulación o las tasas de transmisión. Con respecto a los rangos de frecuencia, opera en las bandas de radioaficionado de VHF y UHF.

Con el objetivo de tener un transceiver más flexible, y que, en el futuro se pueda adaptar a nuevas configuraciones, se decidió diseñar el transmisor en una radio definida por software, ya que con esta tecnología se amplía el rango de posibilidades tanto para la recepción como para la transmisión, puesto que los componentes del transceiver que solían permanecer en hardware, como mezcladores, filtros, amplificadores, moduladores/demoduladores y detectores, ahora serán implementados mediante software en el sistema embebido.

Además, el uso de SDR permite que, en un futuro, se pueda modificar la estación para ser adaptada a proyectos con distintos requerimientos de ancho de banda, modulación, o tipo de mensaje.

Cabe mencionar que la sustitución del transceiver por el SDR, como se muestra en la figura 4.2, es la única sustitución de hardware que se realizó para este proyecto, ya que todos los demás componentes de la estación terrena, que se mencionaron en el capítulo 3, se aprovecharon y adaptaron para el diseño del transmisor.

4. TRANSMISOR CAP SATELITAL

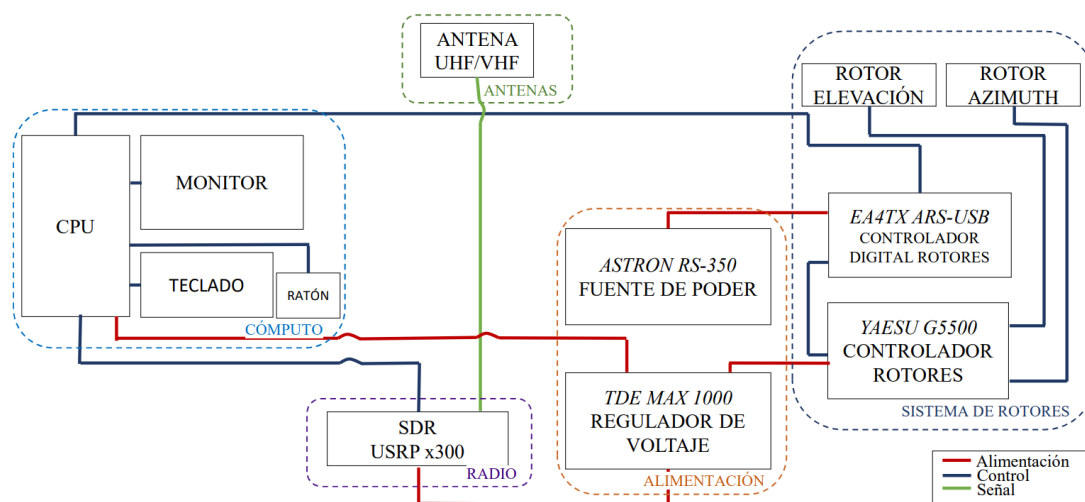


Figura 4.2: Diagrama de estación terrena con USRP.

En cuanto a la selección del modelo específico de SDR, se tienen disponibles en el grupo de investigación los SDR's de la marca Ettus Research, modelos USRP X300 y X310. Por lo que se decidió trabajar con esos modelos, además de que ya se tenía experiencia con dichos dispositivos.

Los USRP son plataformas de SDR de alto rendimiento, escalables, para el desarrollo de sistemas de comunicación inalámbricas. En la tarjeta madre del USRP se encuentra el corazón del radio SDR, que es una FPGA (field-programmable gate array). El USRP cuenta con múltiples interfaces de alta velocidad como PCIe, y puertos duales de 1/10 GigE. Además la tarjeta madre tiene ranuras disponibles para dos tarjetas hijas. Las tarjetas hijas se adquieren por separado y se eligen de acuerdo a la frecuencia con la que se requiera trabajar. Se encuentran disponibles en el mercado tarjetas que trabajan desde corriente directa hasta 6 GHz. En el grupo de investigación se cuenta con la tarjeta hija modelo WBX 120 que trabaja desde los 50 MHz hasta los 2.2 GHz y tiene un ancho de banda de 120 MHz.

Además, los drivers y bibliotecas que Ettus provee, hacen que el equipo sea compatible con distintas plataformas y lenguajes de software como: GNU Radio, NI LabVIEW, MathWorks Simulink, REDHAWK, OpenBTS GSM, Amarisoft LTE, Software Radio Systems LTE, C++, Python, y otras plataformas desarrolladas por terceros. Se hablará precisamente de la plataforma de desarrollo en el siguiente apartado.

Como se mencionó, la sustitución del transceiver Icom-910 por el SDR USRP hace que la estación terrena sea mucho más flexible, sin embargo, surge el requerimiento de un amplificador de potencia a la salida del SDR, debido a que este dispositivo solo puede transmitir hasta 20 [dBm] de potencia, por lo que, en un futuro, se requerirá adquirir un amplificador para poder cumplir correctamente con la funcionalidad de transmisor.

4.1.3. Plataforma de desarrollo

Existen diferentes plataformas para programar los SDR. Para algunas de estas se debe adquirir una licencia con costo para poder utilizarlas. Por ejemplo Simulink, que es parte de Matlab, o LabView que es un software desarrollado por National Instruments. Ambas son plataformas que sirven para desarrollar programas en SDR, pero ambas conllevan un costo anual.

Por otro lado, GNU Radio es una plataforma que permite el desarrollo de programas en SDR y es un software libre, por lo que se optó por utilizar esta plataforma libre de licenciamiento.

La principal diferencias de GNU Radio con respecto a otras plataformas, es que está desarrollada para sistemas operativos basados en Linux. Por consiguiente, se requirió elegir una distribución de Linux para el desarrollo del transmisor satelital. Se seleccionó Ubuntu ya que es una distribución de uso popular en la comunidad de usuarios de Linux, por lo que existe mucha información disponible en cuanto a soporte. Además, se tenía experiencia trabajando con dicho sistema operativo. En cuanto a la versión de Ubuntu se eligió la 18.04 ya que en el momento de la selección era la versión estable más actual.

Cabe mencionar que existe una versión de GNU Radio compatible con el sistema operativo Windows. Sin embargo, GNU Radio es originalmente desarrollado sobre Linux, por lo que la versión de Windows es limitada en cuestiones de compatibilidad. Además, la mayoría de desarrollos de GNU Radio se hacen en Linux, ya que ambos son de software libre. Debido a esto, la información disponible en documentación, comunidades y foros es más amplia.

4.1.4. Modulaciones

La fuente de datos del transmisor satelital es el mensaje de alerta de emergencia del protocolo CAP que, como se mencionó en el Capítulo 2, es un archivo XML, por lo que se decidió que se debían utilizar modulaciones digitales. En las comunicaciones satelitales en bandas de radiofrecuencia de VHF y UHF se usa una gran variedad de modulaciones, como BPSK, QPSK, GMSK o GFSK, entre otros. Se comenzó utilizando la modulación BPSK, pero aprovechando la posibilidad de implementar varias modulaciones debido al uso de SDR, se seleccionaron también QPSK y 16QAM.

4.1.5. Desplazamiento Doppler

Como se mencionó en el Capítulo 3, otro aspecto importante de una estación satelital, especialmente para satélites de órbita baja, es el desplazamiento de frecuencia que existe debido al efecto Doppler. Este desplazamiento se debe corregir para la transmisión y/o recepción en la estación calculando cuánto se desplaza la frecuencia en distintos instantes, tomando en cuenta la posición y velocidad del satélite respecto a la estación. Este cálculo es un tanto complicado de realizar debido a que la frecuencia Doppler varía constantemente, y por tanto, la corrección hay que realizarla en tiempo

real. En la sección 4.3 de este capítulo se muestra cómo se realiza el cálculo teórico del desplazamiento.

Existe una gran variedad de software especializado para seguimiento satelital, también hay software libre. La mayoría de estas herramientas realizan el cálculo del desplazamiento Doppler para el satélite seleccionado, por lo que se usará uno de ellos, en vez de desarrollar un programa dedicado para realizar estos cálculos.

En el capítulo 3 se mencionó que el software que se venía utilizando en la estación terrena es el Orbitron, sin embargo Orbitron es un software que funciona exclusivamente en Windows. Por este motivo se optó por utilizar un software libre que funcionara en Linux.

Es importante recordar que el Orbitron no solo calculaba el desplazamiento de la frecuencia sino también la posición de las antenas así como el control de los rotores y de la frecuencia del transceiver, posibilitando la automatización del seguimiento. Por lo que la sustitución de este software debía de ser con un programa con funcionalidades similares. Para esto se evaluaron distintas opciones.

Al igual que con las plataformas de desarrollo existe software de licenciamiento no gratuito, como son por ejemplo quantumGND, desarrollado por Kratos/RT Logic, este software es capaz no solo de hacer el seguimiento del desplazamiento de frecuencia y el movimiento de rotores automatizado, sino que también puede comunicarse directamente con el satélite. Sin embargo este software no es de uso gratuito y además está optimizado para su uso junto con hardware propio de Kratos.

Otra opción es ISIS GSKit, desarrollado por la empresa ISIS (Innovative Solutions In Space), al igual que el software anterior puede hacer la comunicación directa con el satélite y además el seguimiento junto con la automatización de los rotores, pero este como su nombre lo indica es un kit, por lo que de no comprar el hardware de la misma empresa no se provee del software, lo que hace que esta opción quede descartada, puesto que ya se cuenta con equipo.

Se exploraron otras opciones, sin embargo, se seleccionó el software llamado Gpredict. Este programa permite seleccionar satélites de interés del usuario, para hacer seguimiento de su posición. Los datos orbitales necesarios para calcular el posicionamiento de los satélites son tomados de bases de datos públicas al igual que lo hacía el Orbitron. También realiza el cálculo del desplazamiento de la frecuencia debido al efecto Doppler y el cálculo de los ángulos de posicionamiento de las antenas para una correcta recepción y/o transmisión, y junto con Hamlib, que son bibliotecas que actúan como drivers, se pueden configurar gran variedad de radios y rotores para la automatización del seguimiento.

Para conocer acerca de la instalación y configuración de Gpredict se puede consultar el Anexo I, y para la instalación y configuración de Hamlib se puede consultar el Anexo II de esta tesis. El proceso que se llevó a cabo para poder enlazar Gpredict con el transmisor desarrollado en SDR para corregir el desplazamiento de frecuencia, se expone en la sección 4.4.

4.2. Desarrollo del transmisor SDR

En esta sección se mostrará el programa transmisor SDR desarrollado en el software GNU Radio. La programación en GNU Radio se lleva a cabo por bloques o módulos, esto significa que cada uno de los recuadros mostrados representa un bloque que lleva a cabo una función específica dentro del programa, en este caso puede ser una acción o procesamiento al flujo de datos, dichos bloques se interconectan entre si para formar el programa final.

La plataforma GNU Radio cuenta con un amplio catálogo de bloques predeterminados, que abarcan diversas funciones en distintas categorías como “Channel Models”, “Equalizers”, “Filters”, “Modulators”, “Networking Tools”, entre otras. Además de estos predefinidos, existen distintas opciones para crear bloques personalizados; los bloques OOT (Out of Tree), se pueden desarrollar en distintos lenguajes de programación y después incorporarse a las bibliotecas de GNU Radio, otra opción es el bloque de “Python Code” que permite adjuntar código, en lenguaje de programación python, al bloque para su ejecución dentro del flujo del programa.

4.2.1. Estructura del transmisor

La figura 4.3 muestra un diagrama de bloques en una vista general del transmisor desarrollado en GNU Radio. En este diagrama se observan algunos bloques en color gris opaco, este color denota cuando un bloque se encuentra deshabilitado. La habilitación y deshabilitación de bloques permite, durante el desarrollo, llevar a cabo pruebas o alternar funcionalidades.

4. TRANSMISOR CAP SATELITAL

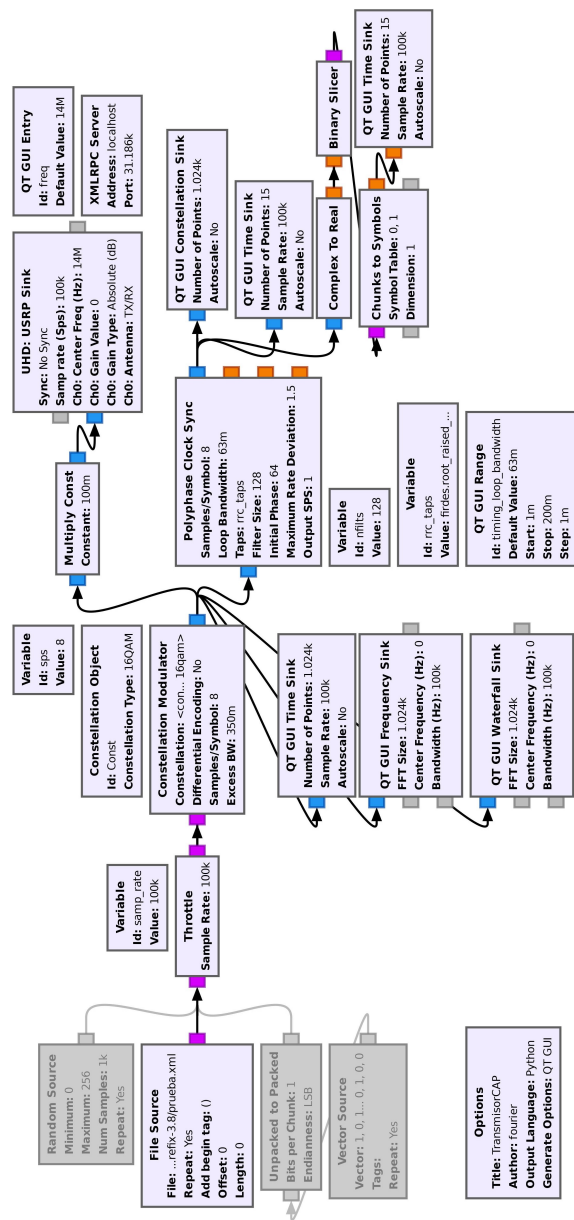


Figura 4.3: Diagrama de Bloques del Transmisor

Para facilitar el entendimiento, en la sección 4.2.1.1 se explicará la funcionalidad de cada bloque que conforma el diagrama.

La figura 4.4 se muestra el diagrama de bloques del transmisor por secciones según su función.

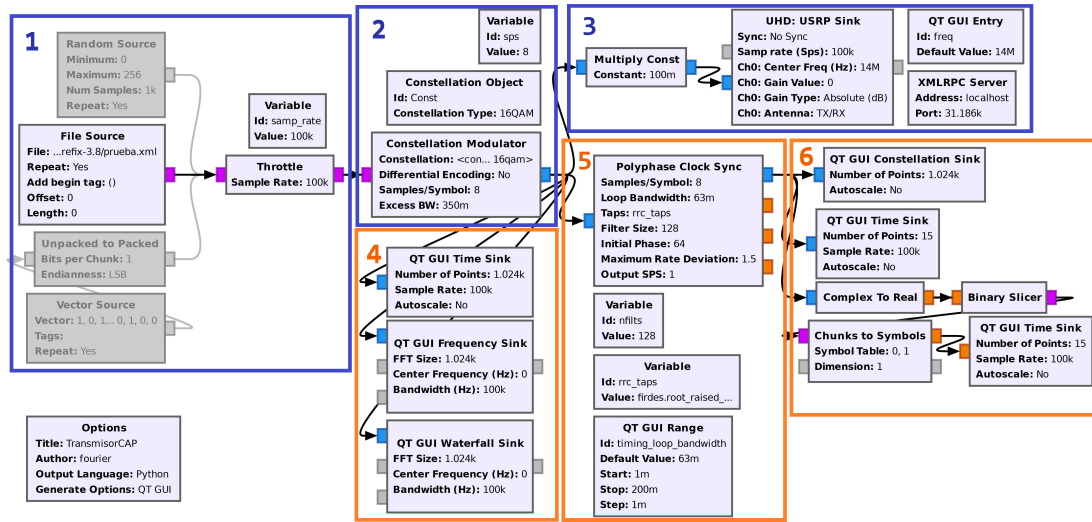


Figura 4.4: Secciones del Diagrama de Bloques del Transmisor

El diagrama de bloques se conforma por dos partes principales; los bloques dentro de los recuadros azules cumplen las funciones del transmisor y en las secciones en naranja se encuentran los bloques utilizados para hacer pruebas de funcionamiento, y visualización de señales en etapas intermedias y finales del transmisor.

4.2.1.1. Transmisor

Los bloques del transmisor, resaltados en azul en la figura 4.4, se dividen a su vez en tres secciones:

Sección 1

En la sección 1, figura 4.5 se muestran los bloques que representan las fuentes de datos o información que se envía a través del transmisor. La fuente de datos definitiva son los datos del archivo XML del protocolo CAP.

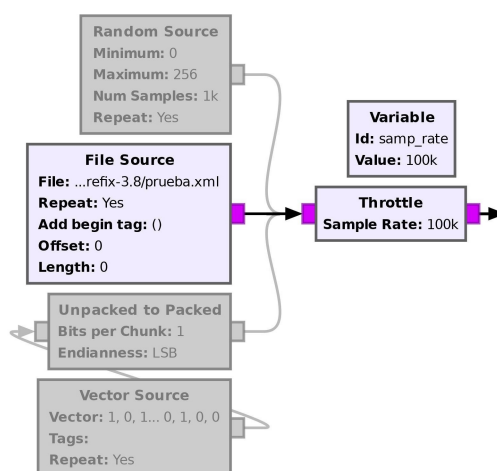


Figura 4.5: Sección 1 del transmisor

En esta sección hay bloques deshabilitados debido a que no se puede tener más de una fuente de datos al mismo tiempo. En este caso los bloques sombreados son formas de generar señales para realizar pruebas de funcionamiento al transmisor.

Los bloques “Vector Source” y “Chunks to Symbols” en conjunto permiten definir un patrón de datos arbitrario, es decir, el usuario puede definir en el bloque Vector Source un patrón conocido; por ejemplo un “10101010” o una serie de únicamente “unos” o únicamente “ceros”. Esto con la finalidad de poder usar secuencias de datos conocidas para probar secciones del transmisor donde se modularán y transmitirán los datos. Probando con distintos patrones se observó y verificó la formación de la constelación de la modulación.

En la figura 4.6 se observan los datos de configuración que requiere este bloque. En el campo “Vector” se pueden introducir valores separados por comas y este será el patrón arbitrario utilizado. Se resalta de las propiedades el campo denominado “Output Type” configurado con valor de byte, lo que quiere decir que cada valor que está introducido entre las comas equivale a un byte (8 bits), por lo que bien se podría seleccionar un valor entre 0 y 255. Por lo tanto, la salida de este bloque son bytes, pero debido a que la intención es que el patrón sea de bits, se requiere del siguiente bloque denominado “Chunks to Symbols”.

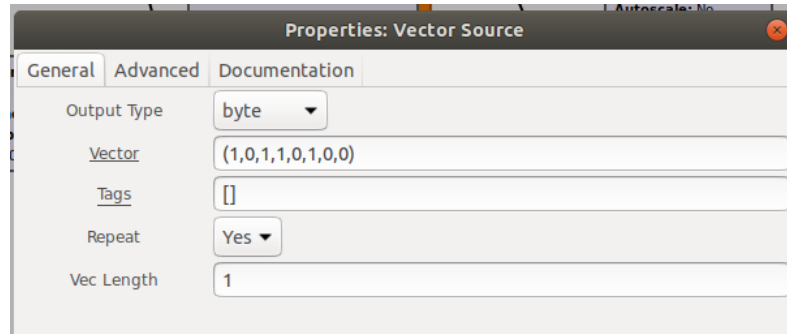


Figura 4.6: Vector Source Properties

El bloque “Chunks to Symbols” toma como entrada los bytes provenientes del bloque “Vector Source” y toma el bit menos significativo de cada byte y lo guarda hasta obtener ocho bits y así volver a formar un byte a la salida. En las propiedades de este bloque hay que especificar qué tipo de dato queremos a la salida, pero en este caso es conveniente el byte ya que, hay que recordar que, estos bloques son para hacer pruebas con datos controlados. En el transmisor final estos bytes son caracteres del archivo XML del mensaje CAP.

El parámetro “Repeat” de la figura 4.6 se activa para que cada que se termine de enviar el patrón, se vuelva a repetir la misma secuencia.

El bloque “Random Source”, como su nombre indica, permite hacer pruebas con datos aleatorios y así verificar que se estén formando todos los símbolos esperados dentro de la constelación.

El bloque “File Source” es la fuente principal de datos del transmisor ya que permite seleccionar el archivo que se requiere transmitir, en este caso el XML proveniente del protocolo CAP. La figura 4.7 muestra las propiedades de este bloque. En el campo “File” se selecciona la ubicación del archivo a enviar, otro parámetro importante es el “Output Type” en donde se debe seleccionar el formato byte, acorde con el formato del archivo que se requiere enviar. En esta configuración se puede seleccionar si enviar el archivo una sola ocasión, o repetirse en el campo “Repeat”.

4. TRANSMISOR CAP SATELITAL

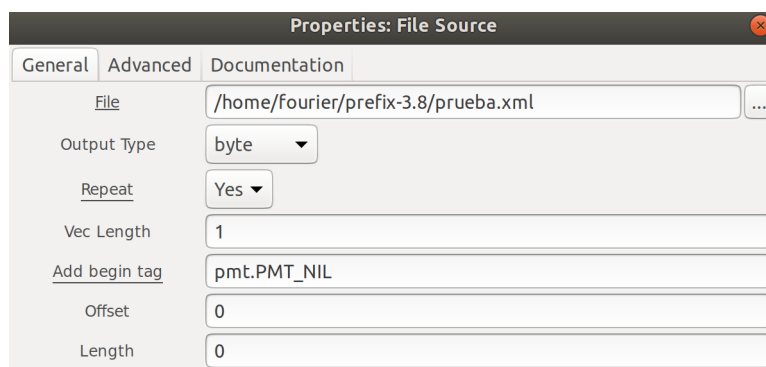


Figura 4.7: Propiedades de el bloque “File Source”

El bloque “Throttle” sirve para limitar la tasa con la que fluyen los datos, se debe poner siempre que no haya otro bloque que limite dicha tasa. Por lo general, un bloque que tenga la tasa limitada tendrá que ver con hardware, como un micrófono o una bocina. En este caso las fuentes son de software por lo que de no poner este bloque, surgirían distintos problemas, ya que, de no poder controlar la tasa de datos tampoco se podría limitar el ancho de banda de la señal a transmitir. Otro problema relacionado es que el CPU trabajaría a su máxima capacidad generando intermitencias en la ejecución del programa.

Sección 2

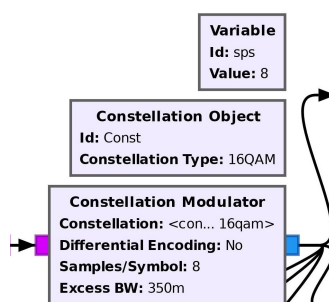


Figura 4.8: Sección 2 del transmisor

La segunda sección, figura 4.8 se compone principalmente por el bloque modulador “Constellation Modulator” que lleva a cabo la modulación digital. Este bloque recibe como entrada, datos en formato byte sin signo y entrega a la salida una señal compleja en banda base a la que es aplicada un filtro de raíz de coseno alzado (RRC root raised cosine). El filtro RRC se agrega con el propósito de disminuir el ancho de banda, en conjunto con un segundo filtro RRC que se debe agregar en la recepción, permite

minimizar la interferencia entre símbolos (ISI). La implementación de este filtro en el transmisor implica que se utilice un filtro igual en la recepción.

Dentro de este bloque también se requiere definir el número de muestras por símbolo que se desean tomar a la entrada, por lo que se ha utilizado un bloque de tipo variable para definir esta cantidad y poderlo manipular de acuerdo a nuestros requerimientos.

El bloque “Constellation Object” sirve para definir el tipo de modulación digital que llevará a cabo el bloque “Constellation Modulator”. Se puede realizar definiendo manualmente el tipo de constelación que se quiere generar o eligiendo entre los tipos de modulaciones ya definidas dentro del bloque: BPSK, QPSK, 8PSK o 16QAM.

Sección 3

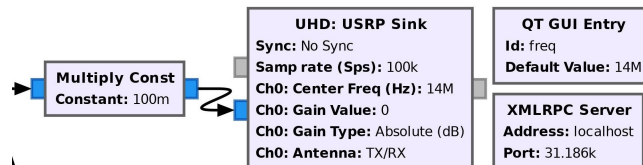


Figura 4.9: Sección 3 del transmisor

La tercera sección definida, figura 4.9 contiene los bloques utilizados para la transmisión de la señal en la banda pasante. Se compone de un bloque multiplicador por una constante que funciona como regulador de potencia, ya que al ser unitaria la salida en el “Constellation Modulator” podemos definir en este bloque la magnitud con la que deseamos emitir la señal.

Para realizar la transmisión de la señal se requiere de un bloque que lleve a cabo la interacción con el hardware del SDR, en el caso de los SDR USRP el fabricante proporciona la biblioteca de drivers y controladores con los bloques requeridos para llevar a cabo ésta conexión, mas adelante se profundizará en el funcionamiento de estos bloques.

Para la transmisión de la señal al USRP se utiliza el bloque “UHD:USRP Sink” que recibe la señal en la banda base y la sube a la banda pasante en la frecuencia que sea definida en el bloque y finalmente lleva a cabo la transmisión de la señal a través del equipo USRP.

Como se ha explicado anteriormente, el transmisor para satélites de órbita baja requiere variar la frecuencia, debido al desplazamiento Doppler. Para realizar el cálculo de la frecuencia se usará el software de seguimiento satelital Gpredict, es decir, la frecuencia es definida externamente en el Gpredict, y el transmisor SDR debe tomarla por lo que se utiliza un bloque “QT GUI ENTRY” para definir la variable que guardara la frecuencia y un servidor de red “XMLRPC”.

El servidor de red funciona como receptor de la nueva frecuencia que se requiere establecer y mediante el bloque “ENTRY” se lee y guarda este valor en una variable

local para que pueda ser leída dentro del bloque “UHD: USRP Sink” y modificar la frecuencia de transmisión, para esto se requiere configurar adecuadamente el Gpredict para que la frecuencia pueda ser enviada correctamente. En secciones subsecuentes se explicará a profundidad el funcionamiento de este mecanismo para el cual se ha requerido de la conexión por medio de un programa en Python.

4.2.1.2. Secciones de prueba y visualización

Los bloques pertenecientes a esta sección se han utilizado durante el desarrollo del transmisor para realizar pruebas de funcionamiento y troubleshooting. Dichos bloques permiten visualizar el flujo de datos y transformación de la señal en distintas etapas del transmisor.

Sección 4

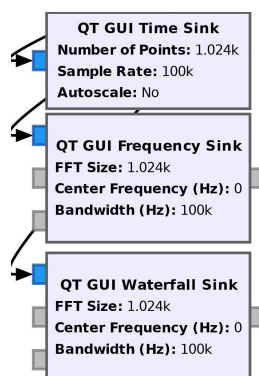


Figura 4.10: Sección 4 del transmisor

Los tres bloques mostrados en la figura 4.10 se utilizan para visualizar la señal a la salida del modulador y son del tipo “GUI”. Todos los bloques que llevan en su nombre “GUI” hacen referencia a “Graphical User Interface” lo que significa que mostrarán algo con lo que el usuario podrá interactuar en tiempo de ejecución. En este caso se seleccionaron 3 bloques:

- “Time Sink”: Muestra la señal en el tiempo, se puede ver la forma de onda. Figura 4.12

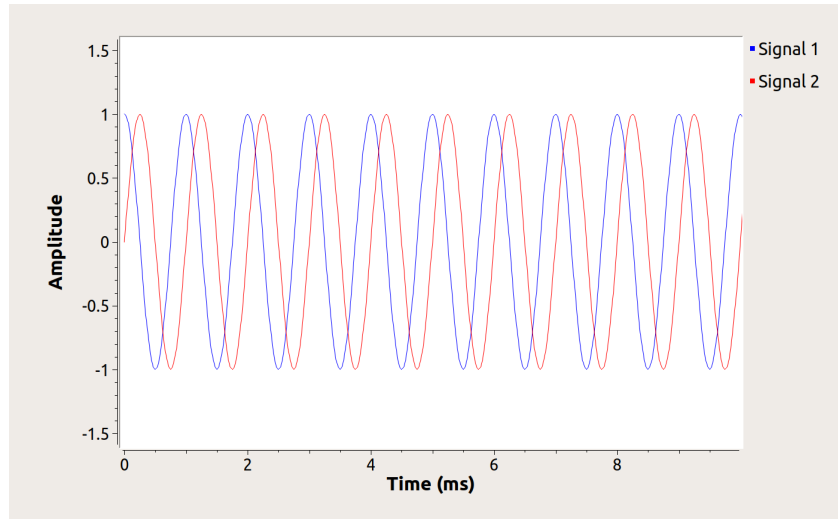


Figura 4.11: Interfaz de "Time Sink"

- "Frequency Sink": Se visualiza el espectro de frecuencia de la señal en el tiempo, a cada instante se va actualizando. Figura 4.12

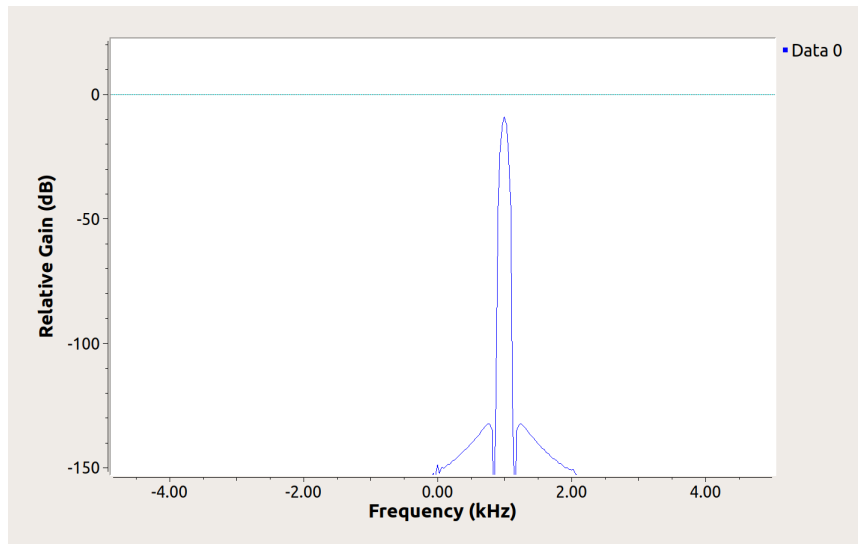


Figura 4.12: Interfaz de "Frequency Sink"

- "Waterfall sink": Este bloque también sirve para visualizar el espectro de la señal, pero aquí la frecuencia se muestra con una escala de colores y va quedando un registro de los datos pasados. Este modo de visualización servirá mas adelante para poder apreciar el cambio de frecuencia debido al efecto Doppler. Figura ??

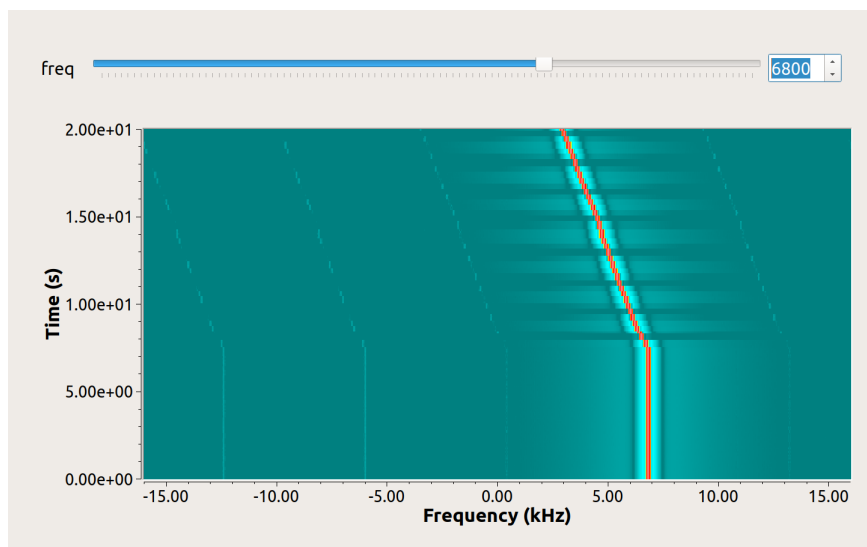


Figura 4.13: Interfaz de “Waterfall sink”

Sección 5

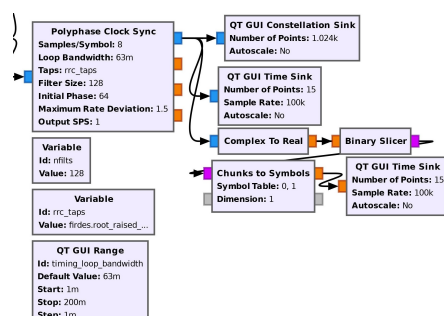


Figura 4.14: Sección 5 del transmisor

La sección 5, mostrada en la figura 4.14, no es parte del transmisor, sino que funge como un receptor de pruebas. Si bien esta sección del programa no pretende ser un receptor funcional, ni hacer la demodulación completa de la señal, se decidió incluirla para poder corroborar el buen funcionamiento del transmisor.

El elemento principal de esta sección es el bloque Polyphase Clock Sync, que es el bloque que inicia el proceso de filtrado, sincronización y demodulación de la señal para así verificar que se este enviando de la forma correcta.

Este bloque sincroniza en tiempo la señal de entrada, es decir, sincroniza los relojes entre la transmisión y la recepción. El bloque “QT GUI RANGE” permite modificar el valor del “loop_bandwidth” de manera visual y durante la ejecución del programa. Esta

es un variable que permite ajustar el factor de amortiguación que ayuda determinar y sincronizar la frecuencia y la fase de la señal, por lo que si notamos un desajuste en la recepción, este es un parámetro que ayuda a corregirlo.

Dentro del bloque se incluyen varias etapas de filtros, determinadas por medio de la variable “rrc_taps” y “ntaps”, que en conjunto complementan el efecto del filtro de raíz de coseno alzado RRC empleado en la transmisión, cumpliendo el propósito de tener una mínima interferencia entre símbolos, finalmente permitiendo recuperar la constelación enviada en la transmisión.

Al igual que en la sección 4, aquí se han incluido bloques para tener una referencia del funcionamiento del programa, así se puede comparar los datos que se transmiten con los que se logran recibir. Con estos bloques podemos observar gráficamente la constelación que se ha formado, la señal en el tiempo y por último los datos recuperados en el tiempo.

4.3. Efecto Doppler

El efecto Doppler se presenta cuando se trata de transmitir a un cuerpo en movimiento o recibir desde un cuerpo en movimiento, es decir, cuando hay un movimiento relativo entre el transmisor y el receptor. Cuando se transmite un mensaje, este tarda en llegar al destino un cierto tiempo, debido a que debe recorrer una cierta distancia. Cuando el objeto está en movimiento, ésta distancia cambia con respecto al tiempo. Esto provoca que, para un observador, la frecuencia del mensaje cambie. El ejemplo mas común de este efecto es cuando se está en una calle y un coche pasa muy rápido sonando la bocina, al principio se escucha agudo el sonido, pero conforme cambia de posición y se aleja de uno, pareciera una nota grave, sin embargo para el conductor del auto el sonido siempre fue el mismo.

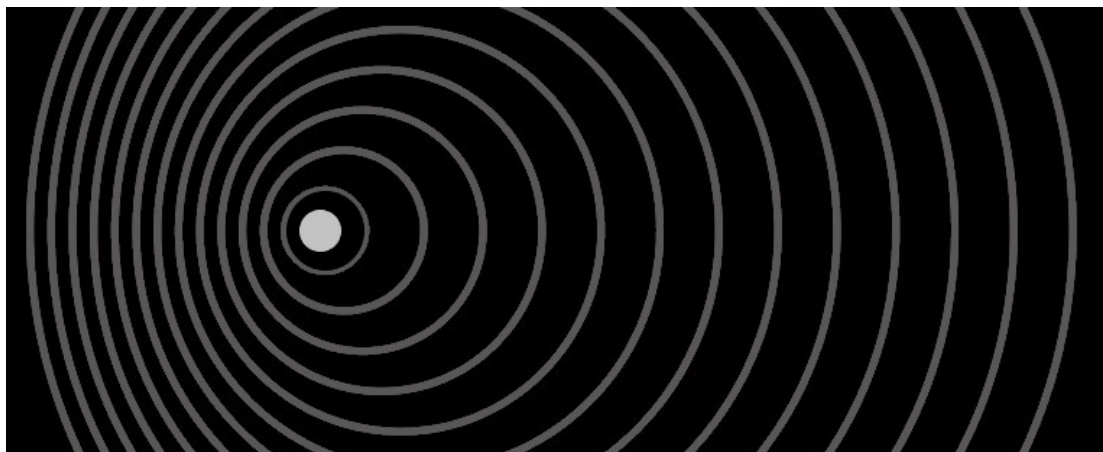


Figura 4.15: Representación gráfica del efecto Doppler

La figura 4.15 simula que el transmisor del mensaje, podría ser el coche del ejemplo, está en movimiento hacia la izquierda, y las ondas representan dicho mensaje viajando en el espacio. Se pueden observar que las ondas de la izquierda están más juntas entre sí, ya que al mismo tiempo que viajan hacia la izquierda el coche también lo está haciendo, y esto se traduce en un incremento de frecuencia, en cambio las ondas que quedan a la derecha o atrás del carro están más separadas entre sí, lo que se traduce en una frecuencia más baja o grave.

Este fenómeno también afecta a la transmisión satelital sobre todo en órbitas bajas o elípticas puesto que el satélite, con respecto a la estación terrena, se está moviendo a una cierta velocidad.

Para el cálculo del desplazamiento de frecuencia se tiene la siguiente expresión matemática [19].

$$\Delta f = \frac{v_t f_t}{c} \quad (4.1)$$

Donde Δf es la variación de la frecuencia con respecto a la frecuencia a la que se transmite o recibe el mensaje, f_t es la frecuencia del mensaje, c es la velocidad de la luz y v_t es la velocidad relativa del satélite con respecto a la estación terrena.

Es importante mencionar que la velocidad v_t no es la misma velocidad con la que se mueve el satélite, es mas bien la velocidad relativa del satélite con respecto a la estación terrena, en la dirección de la línea de vista entre ambos. Es decir, la velocidad del satélite multiplicada por el coseno del ángulo que hay entre el vector de velocidad del satélite y la dirección de propagación de la onda.

Como el satélite no va en línea recta, puesto que va en una órbita, el ángulo que se forma con el vector velocidad y el trayecto de propagación cambian constantemente, la velocidad relativa entre el satélite y la estación terrena también cambia constantemente, es por esto que es necesario el uso del software Gpredict para que calcule y monitoree a cada momento el desplazamiento Doppler, y se lo indique a la estación terrena.

4.3.1. Cálculo del efecto Doppler

Para ejemplificar y corroborar los cálculos del desplazamiento de la frecuencia debido al efecto Doppler, se realizarán los cálculos estimados para este efecto, haciendo uso de la expresión 4.1, en las bandas de frecuencia VHF y UHF.

Se considerarán los casos en los que el desplazamiento de la frecuencia es máximo y como ejemplo se toma la velocidad de la Estación Espacial Internacional, y se compara los valores obtenidos con los que calcula el software. De hecho muchos nanosatélites se lanzan desde la ISS, por lo que suelen tener órbitas y velocidades muy parecidas a la estación.

Para ello primero se obtiene la velocidad relativa entre el satélite y el punto de observación, que es la estación terrena. Se considera que el ángulo de elevación mínimo es 0° , en dónde se presentaría la máxima variación de la frecuencia por el efecto Doppler.

En realidad la velocidad máxima relativa será ligeramente menor a la velocidad del satélite, pero, como aproximación, se tomará la velocidad del satélite.

$$v_{rel} = 7.65 \cdot 10^3 \left[\frac{m}{s} \right] \cos(0) \quad (4.2)$$

$$v_{rel} = 7.65 \cdot 10^3 \left[\frac{m}{s} \right] \quad (4.3)$$

Una vez se conoce la velocidad relativa, se puede obtener la frecuencia Doppler con la ecuación 4.1. En cuanto la frecuencia de la señal emitida por el satélite, se utiliza para este ejemplo frecuencias de interés en las banda VHF y UHF.

$$\Delta f_{VHF} = \frac{7.65 \cdot 10^3 \left[\frac{m}{s} \right] \cdot 145.89 [MHz]}{3 \cdot 10^8 \left[\frac{m}{s} \right]} \quad (4.4)$$

$$\Delta f_{VHF} = 3.72 [kHz] \quad (4.5)$$

$$\Delta f_{UHF} = \frac{7.65 \cdot 10^3 \left[\frac{m}{s} \right] \cdot 437.02 [MHz]}{3 \cdot 10^8 \left[\frac{m}{s} \right]} \quad (4.6)$$

$$\Delta f_{UHF} = 11.14 kHz \quad (4.7)$$

Con los resultados, se verifica que los valores obtenidos están dentro del rango de los valores calculados por el software, ya que, aunque existe una variación, se asemejan.

Los valores obtenidos del orden de 3 kHz para VHF y 11 kHz para UHF demuestran que el desplazamiento Doppler en los satélites de órbita baja es muy importante ya que son valores del mismo orden del ancho de banda de las señales usadas en estas bandas de frecuencia (10 kHz).

Hay que considerar que este ejemplo es una simplificación del modelo completo que se utiliza para calcular el efecto Doppler, ya que para obtener un resultado más certero se requiere tomar en cuenta el desplazamiento del satélite en tres dimensiones, para así poder calcular correctamente el ángulo que existe entre el desplazamiento del satélite y la dirección de propagación de la señal.

4.3.2. Corrección de Frecuencia

Como ya se ha mencionado, en el diseño del transmisor satelital se tiene la necesidad de corregir en tiempo real el desplazamiento de la frecuencia debido al efecto Doppler. El software de seguimiento que se seleccionó, Gpredict, realiza la tarea de calcular dicho desplazamiento tomando en cuenta el satélite que se quiera seguir y la localización de la estación terrena. En algunos casos Gpredict puede funcionar como un controlador directo para distintos transceivers, con la ayuda del driver correspondiente puede controlar la frecuencia y el movimiento de los rotores.

4. TRANSMISOR CAP SATELITAL

En este proyecto, Gpredict proporcionará el cálculo de la magnitud del desplazamiento Doppler, pero se requiere hacerle llegar dicha información al tranceiver, que en este caso es el transmisor satelital implementado en SDR.

Existe una herramienta muy utilizada junto con el Gpredict llamada Hamlib. Hamlib es una biblioteca que provee una API para controlar varios modelos de tranceiver y rotores, hace la labor de un controlador o driver para comunicar a la computadora con los dispositivos de hardware. Esta herramienta, aunque no fue directamente utilizada, fue muy útil para comprender como es que Gpredict envía los cálculos de la frecuencia Doppler a otros equipos. Para este punto ya se tenía configurado el Gpredict junto con Hamlib, esa información esta documentada en el Anexo III de esta tesis.

4.3.2.1. Análisis del protocolo de comunicación entre Hamlib y Gpredict

Para probar la comunicación entre Gpredict y Hamlib se tiene que poner a funcionar Gpredict siguiendo algún satélite para que calcule el desplazamiento de frecuencia y poner a escuchar al Hamlib dicho desplazamiento. Para esto hay que introducir por terminal el siguiente código para que Hamlib espere una conexión.

```
1 rigctld -m 1
```

En el menú de “Radio Control” de Gpredict hay que hacer clic en “Track” y “Engage” y se observa que el botón “engage” se queda activado, esto quiere decir que la conexión se realizó correctamente.

Para observar el protocolo de comunicación entre Gpredict y Hamlib se hizo uso del software Wireshark. Wireshark es un analizador de paquetes de software libre. Es muy utilizado para hacer troubleshooting en el ámbito de redes. En este caso, permite ver el contenido de los mensajes enviados por los dos programas, Gpredict y Hamlib, para transmitirse el desplazamiento de frecuencia.

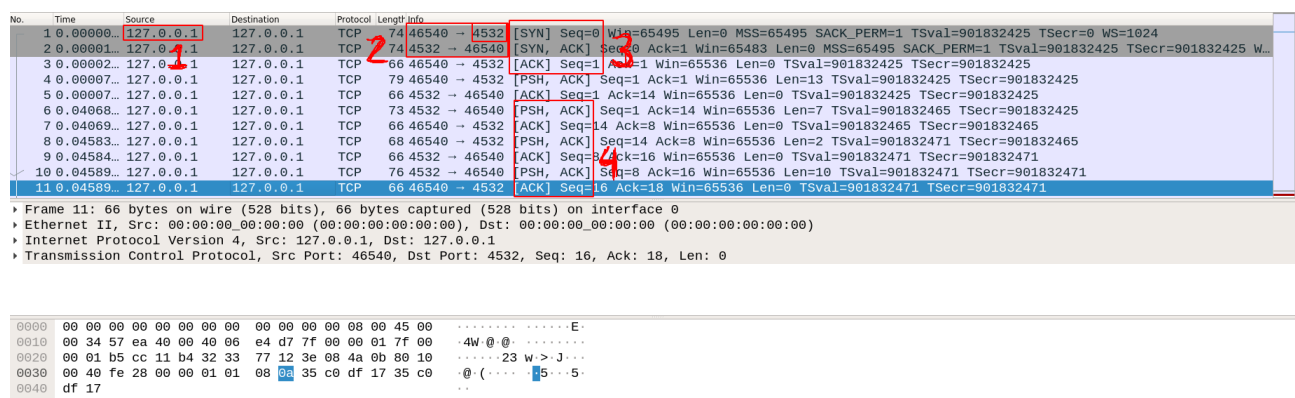


Figura 4.16: Ejemplo de la comunicación entre Hamlib y Gpredict

00000000	46 20 20 31 34 35 37 39 37 32 30 30 0a	F 14579 7200.
00000000	52 50 52 54 20 30 0a	RPRT 0.
0000000D	66 0a	f.
00000007	31 34 35 37 39 37 32 30 30 0a	14579720 0.
0000000F	66 0a	f.
00000011	31 34 35 37 39 37 32 30 30 0a	14579720 0.
00000011	66 0a	f.
0000001B	31 34 35 37 39 37 32 30 30 0a	14579720 0.
00000013	66 0a	f.
00000025	31 34 35 37 39 37 32 30 30 0a	14579720 0.
00000015	46 20 20 31 34 35 37 39 37 32 30 32 0a	F 14579 7202.
0000002F	52 50 52 54 20 30 0a	RPRT 0.
00000022	66 0a	f.
00000036	31 34 35 37 39 37 32 30 30 0a	14579720 0.
00000024	66 0a	f.
00000040	31 34 35 37 39 37 32 30 32 0a	14579720 2.
00000026	66 0a	f.
0000004A	31 34 35 37 39 37 32 30 32 0a	14579720 2.
00000028	66 0a	f.
00000054	31 34 35 37 39 37 32 30 32 0a	14579720 2.
0000002A	46 20 20 31 34 35 37 39 37 32 30 33 0a	F 14579 7203.
0000005E	52 50 52 54 20 30 0a	RPRT 0.
00000037	66 0a	f.
00000065	31 34 35 37 39 37 32 30 32 0a	14579720 2.
00000039	66 0a	f.
0000006F	31 34 35 37 39 37 32 30 33 0a	14579720 3.
0000003B	66 0a	f.
00000079	31 34 35 37 39 37 32 30 33 0a	14579720 3.
0000003D	66 0a	f.
00000093	31 34 35 37 39 37 32 30 33 0a	14579720 3.
0000003F	46 20 20 31 34 35 37 39 37 32 30 34 0a	F 14579 7204.
0000008D	52 50 52 54 20 30 0a	RPRT 0.
0000004C	66 0a	f.
00000094	31 34 35 37 39 37 32 30 33 0a	14579720 3.
0000004E	66 0a	f.
0000009E	31 34 35 37 39 37 32 30 34 0a	14579720 4.
00000050	66 0a	f.
000000A3	31 34 35 37 39 37 32 30 34 0a	14579720 4.
00000052	66 0a	f.
000000B2	31 34 35 37 39 37 32 30 34 0a	14579720 4.
00000054	46 20 20 31 34 35 37 39 37 32 30 35 0a	F 14579 7205.
000000BC	52 50 52 54 20 30 0a	RPRT 0.
00000061	66 0a	f.
000000C3	31 34 35 37 39 37 32 30 34 0a	14579720 4.
00000063	66 0a	f.
000000CD	31 34 35 37 39 37 32 30 35 0a	14579720 5.
00000065	66 0a	f.
000000D7	31 34 35 37 39 37 32 30 35 0a	14579720 5.
00000067	66 0a	f.
000000E1	31 34 35 37 39 37 32 30 35 0a	14579720 5.
00000069	46 20 20 31 34 35 37 39 37 32 30 36 0a	F 14579 7206.
000000EB	52 50 52 54 20 30 0a	RPRT 0.
00000076	66 0a	f.
000000F2	31 34 35 37 39 37 32 30 35 0a	14579720 5.
00000078	66 0a	f.
000000FC	31 34 35 37 39 37 32 30 36 0a	14579720 6.
0000007A	66 0a	f.
00000106	31 34 35 37 39 37 32 30 36 0a	14579720 6.
0000007C	71 0a	q.
00000110	52 50 52 54 20 30 0a	RPRT 0.

Figura 4.17: Conversación entre Hamlib y Gpredict

En la figura 4.16 se puede observar un ejemplo de comunicación entre Gpredict y Hamlib donde se pueden destacar los siguientes aspectos importantes (la explicación que sigue se refiere a los números en rojo mostrados en la figura 4.16).

1. La dirección IP a la que se está accediendo, en este caso por las condiciones del experimento esta es el local host (127.0.0.1), pero existe la posibilidad de controlar la estación terrena de manera remota, poniendo otra dirección IP.
2. Se observa que la conversación se inicia por un puerto cualquiera hacia el puerto ya conocido característico de Gpredict 4532
3. Se inicia la conversación TCP.
4. Se comparte la información acerca del desplazamiento y luego se manda una confirmación de recibido.

En la figura 4.17 se hace seguimiento de la información como tal para comprender su comunicación. Se puede ver al Hamlib en azul y al Gpredict en rojo. Gpredict

manda un mensaje que actualiza la frecuencia cuando empieza por una "F" mayúscula y a continuación la frecuencia actual. El radio responde a esto con un mensaje RPRT X, donde, en este caso, X es cero indicando que se recibió correctamente, en el resto del tiempo Gpredict solicita al radio que le dé su frecuencia actual con un mensaje "f" minúscula y el radio responde con su frecuencia actual.¹

Hamlib fue de mucha ayuda para comprender el protocolo de comunicación que habría entre Gpredict y el radio, y provee las herramientas para conectar dicho software al equipo de una estación terrena. Ahora la tarea es llevar esa información al transmisor satelital en SDR. Por lo que se necesita un intermediario de comunicación parecido al Hamlib pero que lleve la información a GNU Radio. Ese intermediario es un programa que se desarrolló en Python, a continuación se expondrá.

4.3.2.2. Desarrollo de la interfaz entre Gpredict y el transmisor SDR

Se tenía la necesidad de recuperar la frecuencia que proporciona el Gpredict en el puerto 4532, y para atenderla se exploraron múltiples opciones y se hicieron varias pruebas para encontrar soluciones, por ejemplo una opción era crear un bloque de GNU, o instalar bloques ya creados por terceras personas, sin embargo se tuvieron muchos problemas de compatibilidad puesto que las herramientas para unificar dichos bloques con el transmisor satelital, tenían diferentes versiones de GNU y Linux, y esto dificultaba mucho el correcto funcionamiento.

Por esta razón se optó por desarrollar un programa en Python ya que al ser externo a GNU no dependería de la versión de GNU o de la compatibilidad con los USRP pues sería un proceso aparte, por lo que el programa serviría para cualquier futuro cambio en el desarrollo del transmisor.

Se encontró el módulo gr-gpredict desarrollado por terceros² para GNU, que llevaba a cabo una tarea similar a la que se necesitaba. Con el estudio de este módulo, junto con lo aprendido de la comunicación entre Hamlib y Gpredict, se pudo desarrollar un software propio en Python para comunicar el Gpredict con el SDR. Este es el programa desarrollado:

```
1 import socket , time
3 HOST = '127.0.0.1' # (localhost)
  PORT = 4532      # non-privileged ports > 1023
5
7 bind_to = (HOST, PORT)
```

¹Para más información acerca de este protocolo se recomienda visitar la página del manual de comandos <http://manpages.ubuntu.com/manpages/xenial/en/man8/rotctld.8.html>.

²<https://github.com/wnagele/gr-gpredict-doppler>

```

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
9 server.bind(bind_to)
server.listen(0)
11
time.sleep(0.5)
13
while True:
15     print ("Waiting for connection on: %s:%d" % bind_to)
    sock, addr = server.accept()
17     print ("Connected from: %s:%d" % (addr[0], addr[1]))

    cur_freq = 0
    while True:
21         data = sock.recv(1024)
            if not data:
23                 break

                if data.startswith('F'):
                    freq = int(data[1:].strip())
27                     if cur_freq != freq:
                        print ("New frequency: %d" % freq)
29                         cur_freq = freq
                            sock.sendall("RPRT 0\n")
31                     elif data.startswith('f'):
                        sock.sendall("f: %d\n" % cur_freq)
33
sock.close()
35     print ("Disconnected from: %s:%d" % (addr[0], addr[1]))

```

Este programa se ejecuta con Python (puede ser desde la terminal de Linux), y de manera similar al Hamlib espera una conexión, una vez que la encuentra empieza a conversar con Gpredict. En la figura 4.18 se puede observar con bastante claridad lo que esta haciendo el programa, recibe las actualizaciones de la nueva frecuencia y la va imprimiendo. Por lo tanto ya se tiene recuperada la información de Gpredict.

La figura 4.19 muestra la comunicación entre el Gpredict y el programa, donde se observan los cambios de frecuencia que está enviando el Gpredict.

4. TRANSMISOR CAP SATELITAL

```

~/Documents/GNURAd
File Edit View Search Terminal Help
~/Documents/GNURAd$ python doppl.py
Waiting for connection on: 127.0.0.1:4532
Connected from: 127.0.0.1:40434
New frequency: 145801303
New frequency: 145801304
New frequency: 145801306
New frequency: 145801308
New frequency: 145801309
New frequency: 145801312
New frequency: 145801313
New frequency: 145801317
New frequency: 145801318
New frequency: 145801320
Disconnected from: 127.0.0.1:40434
Waiting for connection on: 127.0.0.1:4532

```

Figura 4.18: Ejecución del programa

La figura 4.20 muestra los paquetes de red de la conversación entre Gpredict y el programa de python, conservando el mismo proceso de comunicación que se mostró en la figura 4.16 con Hamlib.

La información contenida en los paquetes enviados entre Gpredict y el programa se muestra en la figura 4.20.

```

... 56.11... 127.0.0.1 127.0.0.1 TCP 74 46570 → 4532 [SYN] Seq=0 Win=65495 Len=
... 56.11... 127.0.0.1 127.0.0.1 TCP 74 4532 → 46570 [SYN, ACK] Seq=0 Ack=1 Win=
... 56.11... 127.0.0.1 127.0.0.1 TCP 66 46570 → 4532 [ACK] Seq=1 Ack=1 Win=6553
... 56.11... 127.0.0.1 127.0.0.1 TCP 79 46570 → 4532 [PSH, ACK] Seq=1 Ack=1 Win=
... 56.11... 127.0.0.1 127.0.0.1 TCP 66 4532 → 46570 [ACK] Seq=1 Ack=14 Win=645
... 56.13... 127.0.0.1 127.0.0.1 TCP 73 4532 → 46570 [PSH, ACK] Seq=1 Ack=14 Wi
... 56.13... 127.0.0.1 127.0.0.1 TCP 66 46570 → 4532 [ACK] Seq=14 Ack=8 Win=655
.....0..... = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
0000  00 00 00 00 00 00 00 00 00 00 00 08 00 45 00  .....E.
0010  00 41 db 66 40 00 40 06 61 4e 7f 00 00 01 7f 00  .A.f@.@.aN.....
0020  00 01 b5 ea 11 b4 2d 37 f0 9e bd ce 6e 80 80 18  .....-7.....n...
0030  00 40 fe 35 00 00 01 01 08 0a 35 d1 66 d8 35 d1  .@.5.....5.f.5.
0040  66 cd 46 20 20 31 34 35 37 39 38 35 35 30 0a    f.F 145 798550.

```

Figura 4.19: Comunicación Gpredict y programa en Python

```

00000000 46 20 20 31 34 35 37 39 38 35 35 30 0a F 14579 8550.
00000000 52 50 52 54 20 30 0a RPRT 0.
0000000D 66 0a f.
00000007 60 3a 20 31 34 35 37 39 38 35 35 30 0a f: 14579 8550.
0000000F 46 20 20 31 34 35 37 39 38 35 35 30 0a F 14579 8550.
00000014 52 50 52 54 20 30 0a RPRT 0.
0000001C 66 0a f.
0000001B 66 3a 20 31 34 35 37 39 38 35 35 30 0a f: 14579 8550.
0000001E 46 20 20 31 34 35 37 39 38 35 35 31 0a F 14579 8551.
00000028 52 50 52 54 20 30 0a RPRT 0.
0000002B 66 0a f.
0000002F 60 3a 20 31 34 35 37 39 38 35 35 31 0a f: 14579 8551.
0000002D 46 20 20 31 34 35 37 39 38 35 35 33 0a F 14579 8553.
0000003C 52 50 52 54 20 30 0a RPRT 0.
0000003A 66 0a f.
00000043 66 3a 20 31 34 35 37 39 38 35 35 33 0a f: 14579 8553.
0000003C 46 20 20 31 34 35 37 39 38 35 35 34 0a F 14579 8554.
00000050 52 50 52 54 20 30 0a RPRT 0.
00000049 66 0a f.
00000057 60 3a 20 31 34 35 37 39 38 35 35 34 0a f: 14579 8554.
0000004B 46 20 20 31 34 35 37 39 38 35 35 36 0a F 14579 8556.
00000064 52 50 52 54 20 30 0a RPRT 0.
00000058 66 0a f.
0000006B 66 3a 20 31 34 35 37 39 38 35 35 36 0a f: 14579 8556.
0000005A 46 20 20 31 34 35 37 39 38 35 35 38 0a F 14579 8558.
00000078 52 50 52 54 20 30 0a RPRT 0.
00000067 66 0a f.
0000007F 60 3a 20 31 34 35 37 39 38 35 35 38 0a f: 14579 8558.
00000069 46 20 20 31 34 35 37 39 38 35 35 39 0a F 14579 8559.
0000008C 52 50 52 54 20 30 0a RPRT 0.

```

Figura 4.20: Conversación entre Gpredict y programa en Python

Con esto queda resuelto el tema de obtener la información del desplazamiento de frecuencia, ahora se tiene que trasladar esta información a GNU Radio donde se encuentra el transmisor satelital.

4.4. Unión de la interfaz con GNU Radio

Ahora que ya se puede recuperar el cálculo del desplazamiento de la frecuencia debido al efecto Doppler, la siguiente tarea consiste en indicarle al bloque encargado de transmitir al aire, del transmisor en GNU, que cambie la frecuencia conforme le indique el programa de Python. Para dar solución a este problema se exploraron y consideraron diversas opciones, a continuación se exponen las más viables.

4.4.1. Entrada de Comandos del USRP Sink

Dentro del transmisor satelital, el bloque encargado de transmitir al aire el mensaje es el “USRP sink” como se muestra en la figura 4.21. Dicho bloque tiene dos entradas, una de ellas llamada “in”, es por donde entra la señal a transmitir, la otra entrada llamada “Command”, sirve para introducir comandos y que estos realicen ciertas acciones en el USRP.

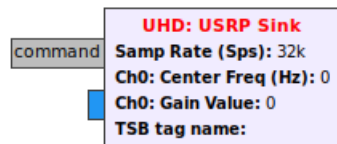


Figura 4.21: Bloque USRP sink

4. TRANSMISOR CAP SATELITAL

Los comandos deben de estar en un formato específico llamado “PMT” que significa “polimorphic types”, estos tipos de datos se utilizan para llevar información entre bloques. Esta información puede ser de distintos tipos de datos, desde cadenas hasta diccionarios. Para el caso específico del bloque “USRP sink” se tiene que pasar los comandos en un sistema de etiquetas y valor.

GNU Radio provee un manual de la API dónde están documentados los comandos, con sus respectivas etiquetas, que se pueden utilizar.

Command name	Value Type	Description
chan	int	Specifies a channel. If this is not given, either all channels are chosen, or channel 0, depending on the action. A value of -1 forces 'all channels', where possible.
gain	double	Sets the Tx or Rx gain (in dB). Defaults to all channels.
freq	double	Sets the Tx or Rx frequency. Defaults to all channels. If specified without <code>lo_offset</code> , it will set the LO offset to zero.
lo_offset	double	Sets an LO offset. Defaults to all channels. Note this does not affect the effective center frequency.
tune	tune_request	Like <code>freq</code> , but sets a full tune request (i.e. center frequency and DSP offset). Defaults to all channels.
lo_freq	double	For fully manual tuning: Set the LO frequency (RF frequency). Conflicts with <code>freq</code> , <code>lo_offset</code> , and <code>tune</code> .
dsp_freq	double	For fully manual tuning: Set the DSP frequency (CORDIC frequency). Conflicts with <code>freq</code> , <code>lo_offset</code> , and <code>tune</code> .
direction	string	Used for timed transceiver tuning to ensure tuning order is maintained. Values other than 'TX' or 'RX' will be ignored.
rate	double	See <code>usrp_block::set_samp_rate()</code> . Always affects all channels.
bandwidth	double	See <code>usrp_block::set_bandwidth()</code> . Defaults to all channels.
time	timestamp	Sets a command time. See <code>usrp_block::set_command_time()</code> . A value of <code>PMT_NIL</code> will clear the command time.
mboard	int	Specify mboard index, where applicable.
antenna	string	See <code>usrp_block::set_antenna()</code> . Defaults to all channels.

Figura 4.22: Comandos disponibles

Fuente: GNU Radio[4]

En la figura figura 4.22 se muestran los comandos disponibles para el bloque USRP sink. En este caso el comando de interés es el que sirve para cambiar la frecuencia.

Tanto C++ como Python tienen su biblioteca para trabajar con los PMT. A continuación se muestra un ejemplo en C++ de como se formatearía el mensaje para cambiar la frecuencia del USRP.

```
1 pmt::pmt_t command = pmt::cons(  
  pmt::mp("freq"),  
3 pmt::mp(1.1e9) )
```

Como se mencionó los comandos van en forma de etiqueta y valor, por lo que un PMT lleva pares de datos conformados por etiqueta y valor. Esto da la ventaja de introducir varios comandos en una sola instrucción.

Esta solución se descarto debido a que resultaba complejo tener dos sistemas (el de creación de comandos y el de introducción de comandos al bloque), siendo ineficiente a comparación de la solución que se selecciono finalmente. Esta solución podría servir en caso de que se requieran cambiar mayor número de parámetros en la transmisión.

4.4.2. ZMQ

ZMQ es una biblioteca de comunicaciones, orientada a mensajes. Funciona como agente intermedio entre el emisor y receptor del mensaje. Se puede ver como un tipo

de socket, que tiene sus propios tipos de mensajes implementados, estos son:

1. Pair
2. Request-Reply
3. Publish-Subscribe
4. Pipeline (push-pull)

Cada tipo de mensaje tiene sus características de cómo se transmite y recibe el mensaje, el que se decidió a utilizar en esta opción es el push-pull, donde el mensaje push es el emisor, y el pull el receptor. Cabe mencionar que esta opción encaja perfecto con lo que se desea, ya que GNU Radio ya tiene implementados bloques con este protocolo. Por lo que para recibirlo en el GNU solo hay que configurar el bloque que se muestra en la figura 4.23.

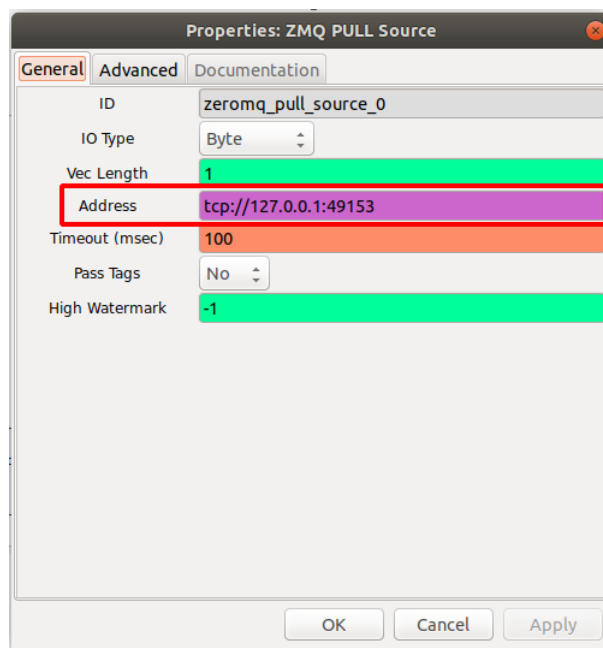


Figura 4.23: Bloque ZMQ Pull

Lo más importante al configurar en el bloque PULL es la dirección, que en éste caso seguimos utilizando el localhost y el puerto teniendo en cuenta que debe estar libre. Para la transmisión de la frecuencia se tiene que hacer un socket del tipo ZMQ en el programa en Python.

```
1 import zmq # hay que importar la biblioteca para poder usar estas
    funciones
```

4. TRANSMISOR CAP SATELITAL

Time	Source	Destination	Protocol	Length	Info
1 0.0000	127.0.0.1	127.0.0.1	TCP	74	33578 → 4532 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2357798002 TSecr=0 WS=1
2 0.0000	127.0.0.1	127.0.0.1	TCP	74	4532 → 33578 [SYN, ACK] Seq=9 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=2357797954
3 0.0000	127.0.0.1	127.0.0.1	TCP	66	33578 → 4532 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2357798002 TSecr=2357797954
4 0.0002	127.0.0.1	127.0.0.1	TCP	79	33578 → 4532 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=13 TSval=2357798002 TSecr=2357797954
5 0.0002	127.0.0.1	127.0.0.1	TCP	66	4532 → 33578 [ACK] Seq=1 Ack=14 Win=64512 Len=0 TSval=2357798002 TSecr=2357798002
6 0.0003	127.0.0.1	127.0.0.1	TCP	77	49153 → 36222 [PSH, ACK] Seq=1 Ack=1 Win=64 Len=11 TSval=2357798002 TSecr=2357756141
7 0.0003	127.0.0.1	127.0.0.1	TCP	66	36222 → 49153 [ACK] Seq=1 Ack=12 Win=64 Len=0 TSval=2357798002 TSecr=2357798002
8 0.0003	127.0.0.1	127.0.0.1	TCP	73	4532 → 33578 [PSH, ACK] Seq=1 Ack=14 Win=64512 Len=7 TSval=2357798002 TSecr=2357798002
9 0.0004	127.0.0.1	127.0.0.1	TCP	66	33578 → 4532 [ACK] Seq=14 Ack=8 Win=65536 Len=0 TSval=2357798002 TSecr=2357798002
10 0.0057	127.0.0.1	127.0.0.1	TCP	68	33578 → 4532 [PSH, ACK] Seq=14 Ack=8 Win=65536 Len=2 TSval=2357798008 TSecr=2357798002
11 0.0057	127.0.0.1	127.0.0.1	TCP	66	4532 → 33578 [ACK] Seq=8 Ack=16 Win=64512 Len=0 TSval=2357798008 TSecr=2357798008
12 0.0058	127.0.0.1	127.0.0.1	TCP	79	4532 → 33578 [PSH, ACK] Seq=8 Ack=16 Win=64512 Len=13 TSval=2357798008 TSecr=2357798008
13 0.0058	127.0.0.1	127.0.0.1	TCP	66	33578 → 4532 [ACK] Seq=16 Ack=21 Win=65536 Len=0 TSval=2357798008 TSecr=2357798008
14 0.0058	127.0.0.1	127.0.0.1	TCP	79	33578 → 4532 [PSH, ACK] Seq=16 Ack=21 Win=65536 Len=13 TSval=2357798008 TSecr=2357798008
15 0.0058	127.0.0.1	127.0.0.1	TCP	66	4532 → 33578 [ACK] Seq=21 Ack=29 Win=64512 Len=0 TSval=2357798008 TSecr=2357798008
16 0.0059	127.0.0.1	127.0.0.1	TCP	73	4532 → 33578 [PSH, ACK] Seq=21 Ack=29 Win=64512 Len=7 TSval=2357798008 TSecr=2357798008
17 0.0059	127.0.0.1	127.0.0.1	TCP	66	33578 → 4532 [ACK] Seq=29 Ack=28 Win=65536 Len=0 TSval=2357798008 TSecr=2357798008
18 0.0130	127.0.0.1	127.0.0.1	TCP	68	33578 → 4532 [PSH, ACK] Seq=29 Ack=28 Win=65536 Len=2 TSval=2357798015 TSecr=2357798008
19 0.0130	127.0.0.1	127.0.0.1	TCP	66	4532 → 33578 [ACK] Seq=28 Ack=31 Win=64512 Len=0 TSval=2357798015 TSecr=2357798015
20 0.0130	127.0.0.1	127.0.0.1	TCP	79	4532 → 33578 [PSH, ACK] Seq=28 Ack=31 Win=64512 Len=13 TSval=2357798015 TSecr=2357798015
21 0.0130	127.0.0.1	127.0.0.1	TCP	66	33578 → 4532 [ACK] Seq=31 Ack=41 Win=65536 Len=0 TSval=2357798015 TSecr=2357798015
22 1.0010	127.0.0.1	127.0.0.1	TCP	79	33578 → 4532 [PSH, ACK] Seq=31 Ack=41 Win=65536 Len=13 TSval=2357798015 TSecr=2357798015
23 1.0010	127.0.0.1	127.0.0.1	TCP	66	4532 → 33578 [ACK] Seq=41 Ack=44 Win=64512 Len=0 TSval=2357798015 TSecr=2357798015
24 1.0011	127.0.0.1	127.0.0.1	TCP	77	49153 → 36222 [PSH, ACK] Seq=12 Ack=1 Win=64 Len=11 TSval=2357798003 TSecr=2357798002
25 1.0011	127.0.0.1	127.0.0.1	TCP	66	36222 → 49153 [ACK] Seq=1 Ack=23 Win=64 Len=0 TSval=2357798003 TSecr=2357798003
26 1.0012	127.0.0.1	127.0.0.1	TCP	73	4532 → 33578 [PSH, ACK] Seq=41 Ack=44 Win=64512 Len=7 TSval=2357798003 TSecr=2357798003
27 1.0014	127.0.0.1	127.0.0.1	TCP	66	33578 → 4532 [ACK] Seq=44 Ack=48 Win=65536 Len=0 TSval=2357798003 TSecr=2357798003
28 1.0069	127.0.0.1	127.0.0.1	TCP	68	33578 → 4532 [PSH, ACK] Seq=44 Ack=48 Win=65536 Len=2 TSval=2357798009 TSecr=2357798003
29 1.0069	127.0.0.1	127.0.0.1	TCP	79	4532 → 33578 [PSH, ACK] Seq=48 Ack=46 Win=64512 Len=13 TSval=2357798009 TSecr=2357798009
30 1.0069	127.0.0.1	127.0.0.1	TCP	66	33578 → 4532 [ACK] Seq=46 Ack=61 Win=65536 Len=0 TSval=2357798009 TSecr=2357798009
31 2.0019	127.0.0.1	127.0.0.1	TCP	79	33578 → 4532 [PSH, ACK] Seq=46 Ack=61 Win=65536 Len=13 TSval=2357800004 TSecr=2357798009
32 2.0019	127.0.0.1	127.0.0.1	TCP	73	4532 → 33578 [PSH, ACK] Seq=61 Ack=59 Win=64512 Len=7 TSval=2357800004 TSecr=2357800004
33 2.0019	127.0.0.1	127.0.0.1	TCP	66	33578 → 4532 [ACK] Seq=59 Ack=68 Win=65536 Len=0 TSval=2357800004 TSecr=2357800004
34 2.0072	127.0.0.1	127.0.0.1	TCP	68	33578 → 4532 [PSH, ACK] Seq=59 Ack=68 Win=65536 Len=2 TSval=2357800009 TSecr=2357800004
35 2.0072	127.0.0.1	127.0.0.1	TCP	79	4532 → 33578 [PSH, ACK] Seq=68 Ack=61 Win=64512 Len=13 TSval=2357800009 TSecr=2357800009
36 2.0072	127.0.0.1	127.0.0.1	TCP	66	33578 → 4532 [ACK] Seq=61 Ack=81 Win=65536 Len=0 TSval=2357800009 TSecr=2357800009
37 3.0027	127.0.0.1	127.0.0.1	TCP	79	33578 → 4532 [PSH, ACK] Seq=61 Ack=81 Win=65536 Len=13 TSval=2357800105 TSecr=2357800009
38 3.0029	127.0.0.1	127.0.0.1	TCP	77	49153 → 36222 [PSH, ACK] Seq=23 Ack=1 Win=64 Len=11 TSval=2357801005 TSecr=2357798003
39 3.0029	127.0.0.1	127.0.0.1	TCP	66	36222 → 49153 [ACK] Seq=1 Ack=34 Win=64 Len=0 TSval=2357801005 TSecr=2357801005
40 3.0029	127.0.0.1	127.0.0.1	TCP	73	4532 → 33578 [PSH, ACK] Seq=81 Ack=74 Win=64512 Len=7 TSval=2357801005 TSecr=2357801005
41 3.0029	127.0.0.1	127.0.0.1	TCP	66	33578 → 4532 [ACK] Seq=74 Ack=88 Win=65536 Len=0 TSval=2357801005 TSecr=2357801005
42 3.0086	127.0.0.1	127.0.0.1	TCP	68	33578 → 4532 [PSH, ACK] Seq=74 Ack=88 Win=65536 Len=2 TSval=2357801011 TSecr=2357801005
43 3.0087	127.0.0.1	127.0.0.1	TCP	79	4532 → 33578 [PSH, ACK] Seq=88 Ack=76 Win=64512 Len=13 TSval=2357801011 TSecr=2357801011
44 3.0087	127.0.0.1	127.0.0.1	TCP	66	33578 → 4532 [ACK] Seq=76 Ack=101 Win=65536 Len=0 TSval=2357801011 TSecr=2357801011
45 4.0039	127.0.0.1	127.0.0.1	TCP	79	33578 → 4532 [PSH, ACK] Seq=76 Ack=101 Win=65536 Len=13 TSval=2357802006 TSecr=2357801011
46 4.0041	127.0.0.1	127.0.0.1	TCP	77	49153 → 36222 [PSH, ACK] Seq=34 Ack=1 Win=64 Len=11 TSval=2357802006 TSecr=2357801005
47 4.0041	127.0.0.1	127.0.0.1	TCP	66	36222 → 49153 [ACK] Seq=1 Ack=45 Win=64 Len=0 TSval=2357802006 TSecr=2357802006
48 4.0042	127.0.0.1	127.0.0.1	TCP	73	4532 → 33578 [PSH, ACK] Seq=101 Ack=89 Win=64512 Len=7 TSval=2357802006 TSecr=2357802006

Figura 4.24: Comunicación entre Gpredict y GNU Radio por medio del programa en Python

```

3 context = zmq.Context()
  zmq_socket = context.socket(zmq.PUSH)
5 zmq_socket.bind("tcp://127.0.0.1:49153")

```

En el código anterior se está declarando el socket del tipo ZMQ y se conecta a la dirección que ya se había configurado. Y dentro del programa ya que se tiene la frecuencia actualizada se manda al bloque en GNU con la siguiente instrucción.

```

1 zmq_socket.send_json(freq)

```

Se realizó una prueba de comunicación ya de todo el sistema, es decir, Gpredict mandando la información y recibíendola en GNU Radio. Y se observa la comunicación en la figura 4.24. Se puede ver en morado la conversación entre Gpredict y el programa y cuando hay un cambio de frecuencia, es cuando se ven mensajes en rojo.

Estos mensajes se mandan con una bandera PSH que es el tipo de mensaje push del protocolo ZMQ, dicho mensaje llega al bloque en GNU y automáticamente le responde de recibido con un ACK.

Una vez que la información se encuentra en el bloque de GNU Radio esta se convierte en un flujo de datos a la salida del bloque “ZMQ Pull Source”. Estos datos se tienen que enviar al bloque USRP sink, por lo que se necesita implementar un canal de comunicación entre estos dos bloques. La solución que se expondrá a continuación obtiene la información de la frecuencia sin requerir canal extra de comunicación. Por este motivo se descartó ZMQ.

4.4.3. XML-RPC

El bloque “XMLRPC Server” se incluye dentro del flowgraph del transmisor con el objetivo de establecer una comunicación continua con el software GPredict. Esto permite obtener la información sobre la corrección de la frecuencia Doppler y guardar ésta dentro de una variable que pueda ser leída por el bloque “UHD: USRP Sink” para a su vez modificar la frecuencia de transmisión en tiempo real.

Este bloque fue el establecido para actuar como servidor debido a que con este protocolo de comunicación se permite especificar el valor de parámetros para ser enviados y leídos por una variable local dentro de GNU Radio. Esta solución consta de 3 partes primordiales para su funcionamiento:

1. El envío de la frecuencia desde el módulo de Radio del GPredict hasta el programa Python descrito en la sección 4.3.2.2
2. El reenvío de la frecuencia desde el programa Python hacia el servidor colocado dentro del programa transmisor SDR en GNU Radio
3. La recepción y lectura de una variable dentro del programa transmisor de GNU Radio

Para llevar a cabo la función descrita en el punto 2 se requirió agregar en el programa de python los comandos necesarios para el reenvío en este protocolo, obteniendo así el siguiente código:

```

1 import socket , time
  import xmlrpclib
3 HOST = '127.0.0.1'
  PORT = 4532
5 bind_to = (HOST, PORT)
  server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7 server.bind(bind_to)
  server.listen(0)
9 s=xmlrpclib.Server('http://localhost:31186')
  time.sleep(0.5) #Parametro de ajuste
11 while True:
```

```
13     print ("Waiting for connection on: %s:%d" % bind_to)
14     sock, addr = server.accept()
15     print ("Connected from: %s:%d" %(addr[0], addr[1]))
16     cur_freq = 0
17     while True:
18         data = sock.recv(1024)
19         if not data:
20             break
21         if data.startswith('F'):
22             freq = int(data[1:].strip())
23             if cur_freq != freq:
24                 print ("New frequency: %d" % freq)
25                 cur_freq = freq
26                 s.set_freq(cur_freq) #envio variable
27                 sock.sendall("RPRT 0\n")
28             elif data.startswith('f'):
29                 sock.sendall("f: %d\n" % cur_freq)
30 sock.close()
31 print ("Disconnected from: %s:%d" %(addr[0], addr[1]))
```

Se observa en la línea 9 la declaración del protocolo, dirección y puerto por el que se establecerá la comunicación con el servidor XMLRPC de GNU Radio. A partir de la línea 20 comienza el proceso de actualización de la frecuencia y la declaración de la variable a enviar en la línea 25, en donde vemos que se utiliza *set_freq* lo que indica que se establecerá el nuevo valor de la variable *freq* dentro de GNU Radio.

Para cumplir con el punto 3 antes descrito, dentro del flowgraph del transmisor de GNU Radio se agrega el bloque “XMLRPC Server” que solo recibe como parámetros de configuración la dirección y puerto de comunicación, en este caso *localhost* y *31186* respectivamente.

Mediante el bloque “QT GUI Entry” se declara la variable *freq* y el valor inicial de la frecuencia. Como se había mencionado anteriormente, la nomenclatura “GUI” indica que durante la ejecución se desplegará una interfaz gráfica, en este caso permitirá observar en tiempo real el valor que va tomando la variable *freq*.

Con este flujo establecido se realizaron las pruebas de comunicación, corroborando la correcta recepción del nuevo valor de la frecuencia de transmisión conforme al movimiento del valor de la frecuencia Doppler. En la figura 4.25 se observa en la parte inferior derecha, en la interfaz de GPredict, la frecuencia de Downlink y uplink relacionada al satélite seleccionado, en la parte superior derecha se observa la frecuencia

que se está enviando al transmisor CAP en GNU Radio para la corrección del efecto Doppler, en el recuadro inferior izquierdo se encuentra la señal en el tiempo y el valor que la variable freq que cambia para realizar la corrección y en el recuadro superior izquierdo se el mapa dónde identificamos diversos satélites en orbita y su huella sobre la tierra.

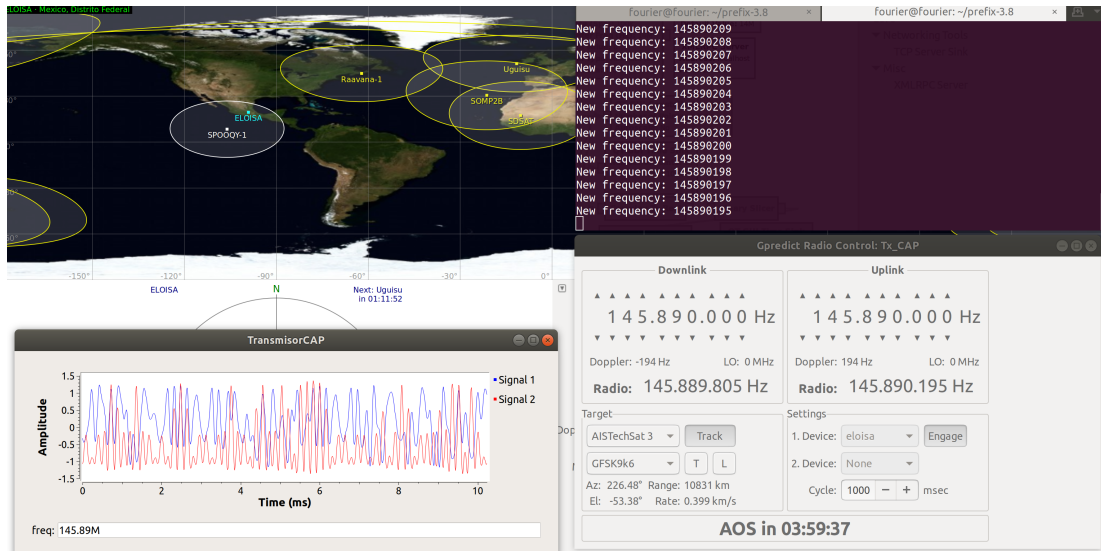


Figura 4.25: Comunicación por XMLRPC entre Gpredict y GNU Radio

Pruebas de funcionamiento y resultados obtenidos

En este capítulo se describirán las pruebas que se realizaron con el objetivo de verificar el correcto funcionamiento del transmisor CAP satelital, empezando desde etapas tempranas del desarrollo, como fue la comprobación de la entrada de datos del archivo CAP, pasando por las diferentes modulaciones y terminando con la prueba del sistema completo. Logrando visualizar las características de la señal transmitida y corroborar el corrimiento de frecuencia debido al efecto Doppler. El sistema de pruebas utilizado se muestra en la figura 5.1.

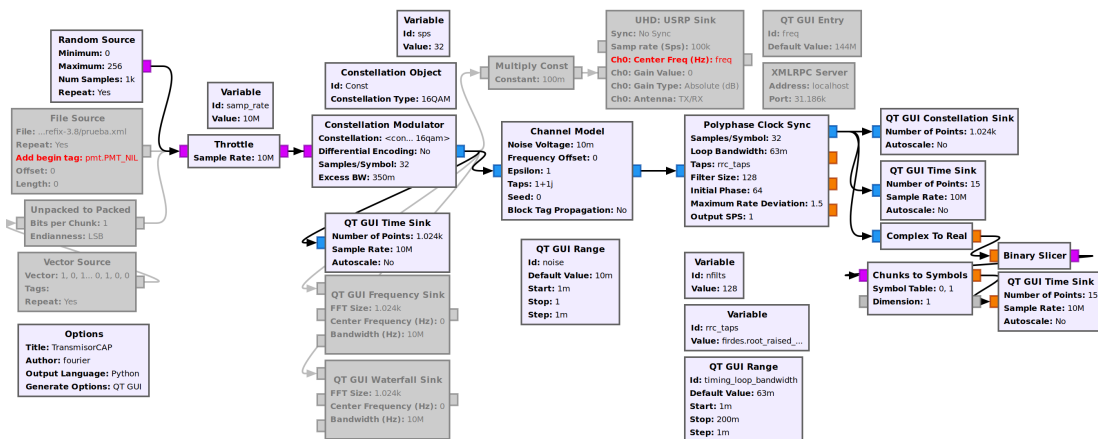


Figura 5.1: Diagrama del sistema de pruebas en simulación de GNURadio

Para corroborar el funcionamiento de los distintos tipos de modulación que se implementaron en el transmisor, se hizo uso de los bloques de prueba descritos en la sección 4.2.1.2, y adicionalmente se agregó un simulador de canal. Dichos bloques permiten observar la señal generada en distintos dominios: la señal compleja en el tiempo a la

salida del transmisor, la constelación formada en la modulación y la demodulación a nivel de bit.

5.1. Pruebas de los datos transmitidos

Con el objetivo de verificar que los datos con que se alimenta el transmisor son los correctos, se hizo uso de los módulos “Vector Source” y “Unpacked to Packed” para definir una serie de bits que se puedan identificar fácilmente en la demodulación. En la figura 5.2 se aprecia la configuración en el bloque “Vector Source” y la secuencia definida.

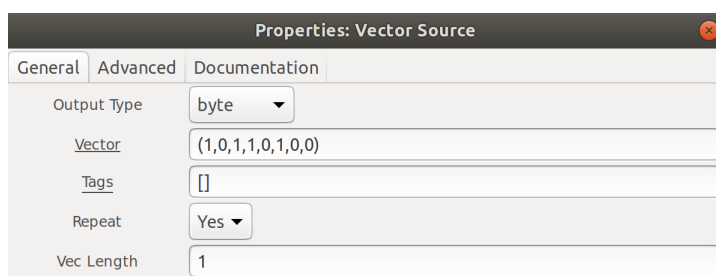


Figura 5.2: Vector de bits definido para pruebas

En esta primera prueba se usó la modulación BPSK por su robustez, y no se ha añadido ruido al canal. En las pantallas de visualización de las señales de la figura 5.3 se observa la señal recibida en el tiempo después de pasar por el canal, en banda base 5.3 (a), la constelación de la señal recibida 5.3 (b), los símbolos de la señal demodulada 5.3 (c) y por último los bits ya demodulados en el tiempo 5.3 (d). Resaltado en un cuadro rojo se puede observar el patrón que se ha enviado, y que es efectivamente el mismo patrón que se introdujo en el modulo “Vector Source”. Por lo que se comprueba que se están introduciendo de manera correcta los datos.

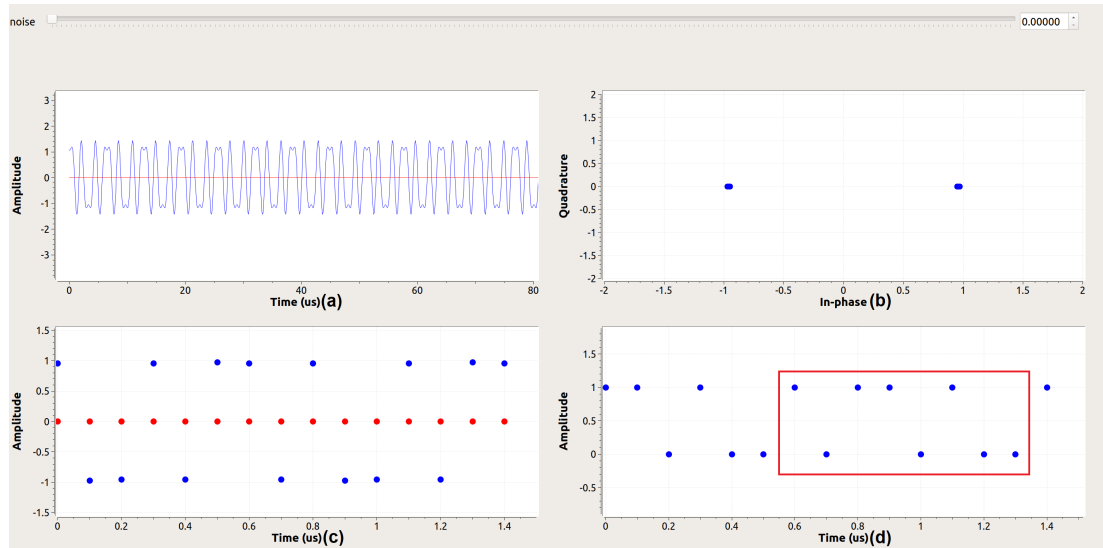


Figura 5.3: Simulación de transmisión de vector de bits por modulación BPSK sin ruido en el canal

En una segunda prueba, con el fin de probar el modulo de canal, que puede servir más adelante, se añadió el modulo con cierto ruido. En la primera prueba el ruido fue de 1 [V]. En la figura 5.4 de la simulación se aprecian los efectos del ruido en la señal en el tiempo y en la representación de la constelación se ven los símbolos mucho más dispersos, sin embargo aún bien definidos, por lo que en la demodulación se puede apreciar que se sigue recibiendo correctamente el vector.

5. PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS OBTENIDOS

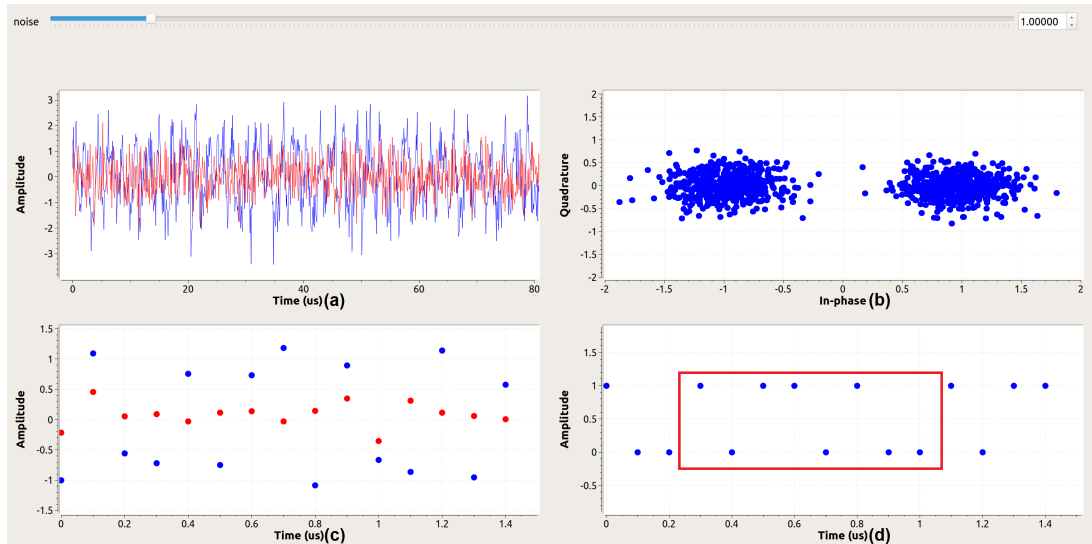


Figura 5.4: Simulación de transmisión de vector de bits por modulación BPSK con ruido en el canal

En la siguiente prueba, el ruido se aumentó a 2.5 [V]. En los resultados mostrados en la figura 5.5 se observan más severamente los efectos del ruido en la señal, en este caso ya se encuentran algunos errores en la demodulación.

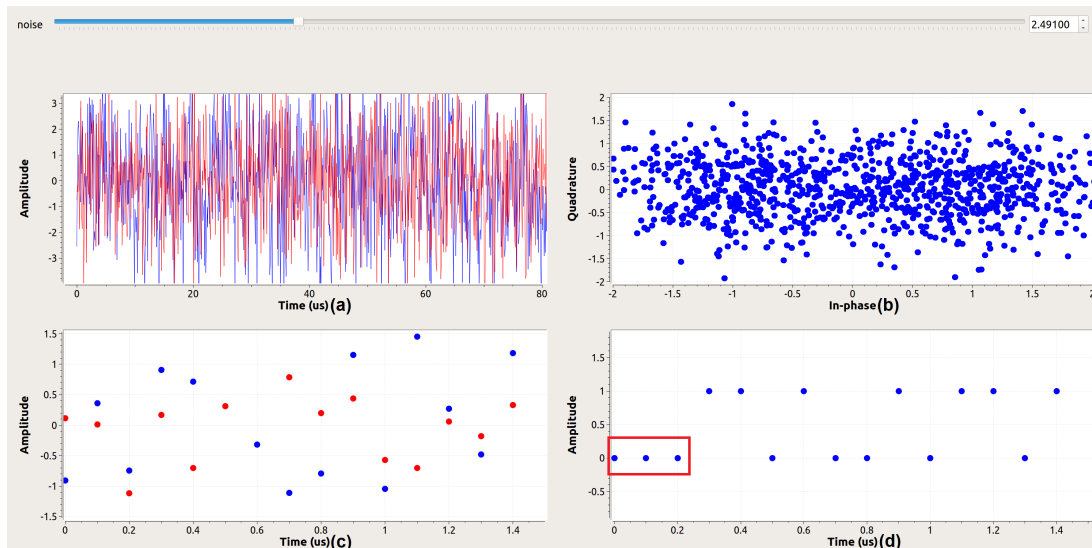


Figura 5.5: Simulación de transmisión de vector de bits por modulación BPSK con error en demodulación

5.2. Verificación de la generación de modulaciones

Las siguientes pruebas se hicieron con el propósito de verificar el correcto funcionamiento de los distintos tipos de modulación implementados, mediante el uso de bloques de prueba, se observó la señal modulada en distintas etapas. A continuación se explicarán y mostrarán las pruebas realizadas con cada modulación diferente:

BPSK

La figura 5.6 muestra los resultados de la modulación BPSK, en el apartado (a) se puede observar la señal en el tiempo a la salida del modulador, en esta modulación la fase de la señal portadora cambia entre dos estados, 0 y π , estos dos estados de fase se ven traducidos en dos puntos sobre la representación en cuadratura y fase que se puede observar en el apartado (b). En el apartado (c) observamos los símbolos de la señal modulada, en donde notamos como solo la parte real de la señal es la que contiene información, lo esperado en una modulación BPSK. Por ultimo, en el apartado (d), se presentan los bits, resultado de la decodificación de los símbolos del apartado (c).

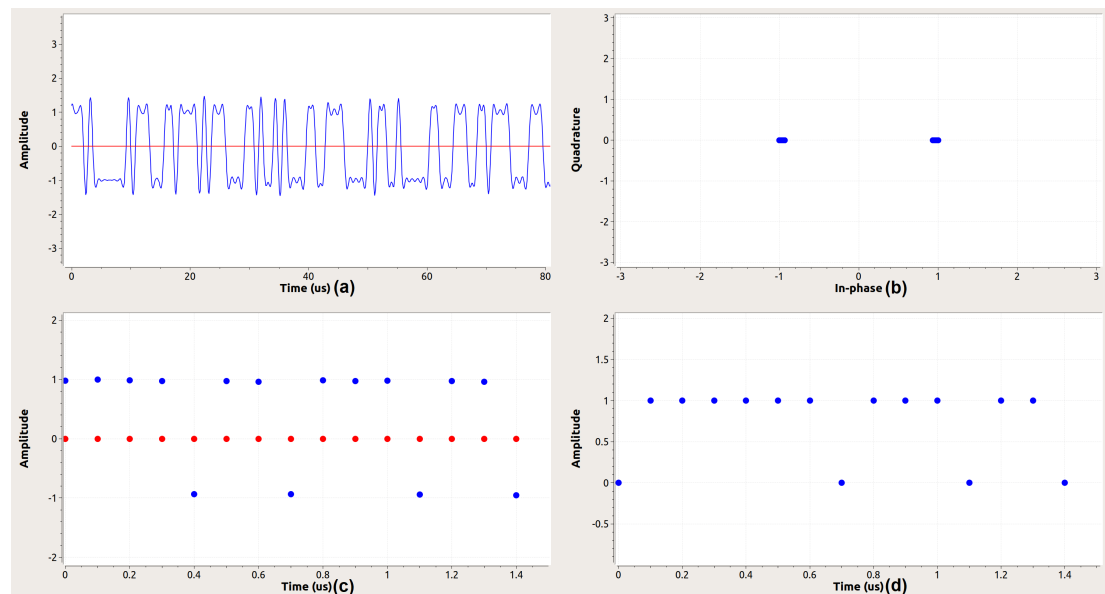


Figura 5.6: Pruebas de Modulación BPSK

QPSK

En la figura 5.7 se encuentran los resultados de las pruebas con la modulación QPSK. En el apartado (a), se tiene la señal real representada en color azul, y la señal

5. PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS OBTENIDOS

imaginaria representada en color rojo. Cada señal tiene dos cambios de estado de manera independiente, para tener un total de cuatro estados, es por esto que la modulación QPSK se puede ver como dos modulaciones BPSK en cuadratura. En el apartado (b) se presenta la constelación de la señal, donde se encuentran los cuatro símbolos en el plano de fase y cuadratura. La representación de los símbolos de la señal en el tiempo se encuentra en el apartado (c), observando cambios tanto en la parte real como imaginaria de la señal. Finalmente los bits decodificados en el apartado (d).

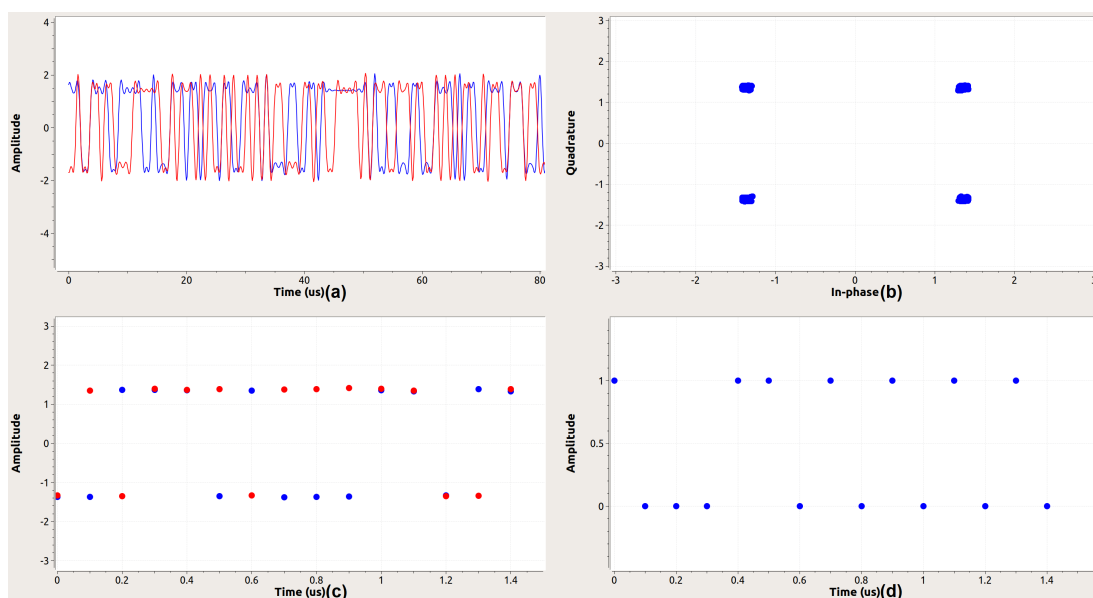


Figura 5.7: Pruebas de Modulación QPSK

DQPSK

El resultado de la modulación DQPSK se muestra en la figura 5.8, en los distintos apartados podemos notar una similitud a las características antes descritas para la modulación QPSK, sin embargo, DQPSK utiliza la diferencia de fase entre símbolos adyacentes para evitar problemas asociados a la falta de sincronización de fase entre transmisor y receptor, por lo que es útil en receptores no coherentes.

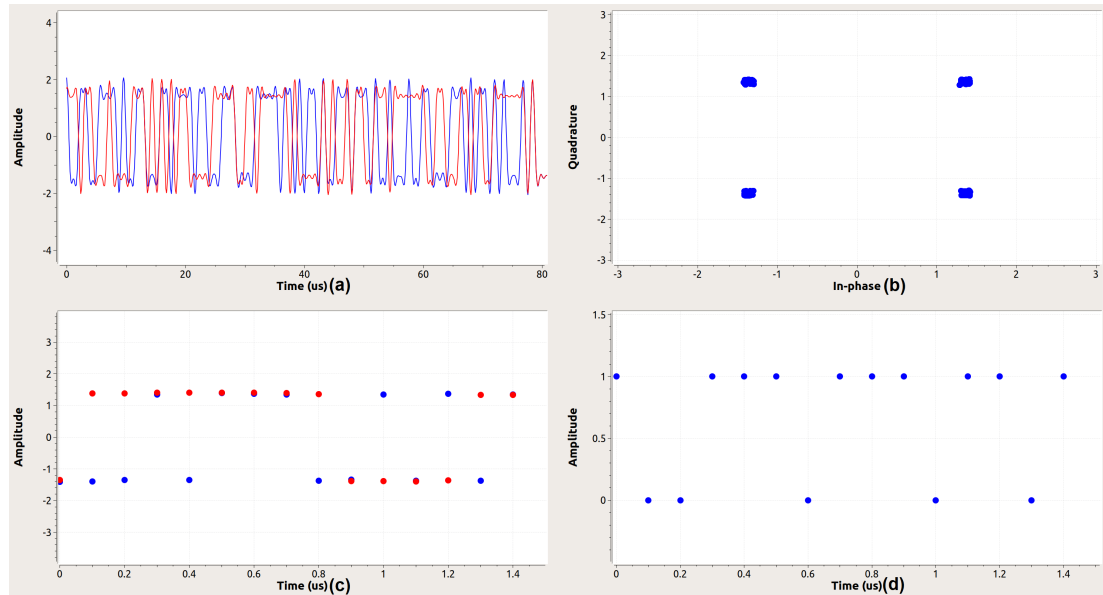


Figura 5.8: Pruebas de Modulación DQPSK

8PSK

En la figura 5.9 se encuentran los resultados de las pruebas con la modulación 8PSK. En el apartado (a) se observa la señal modulada en tiempo, se visualiza tanto la señal real como la imaginaria y se nota que va cambiando fase ambas señales. En el apartado (b) se presentan los 8 símbolos en el plano, se puede notar como cada símbolo se separa por un ángulo de 45 grados. En el apartado (c) se representa los símbolos de la señal modulada en el tiempo y en el apartado (d) los bits decodificados.

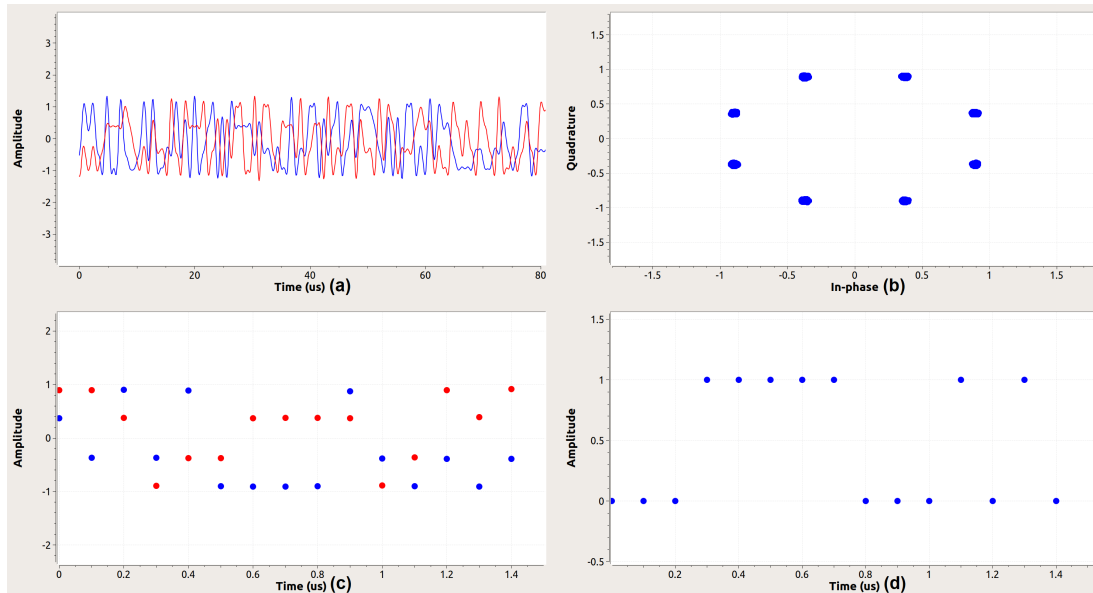


Figura 5.9: Pruebas de Modulación 8PSK

16QAM

Los resultados de la modulación 16QAM se muestran en la figura 5.10. Esta es una modulación en amplitud por cuadratura, que consiste en modular por desplazamiento de amplitud dos señales desfasadas entre sí 90° . En el apartado (a) observamos la señal en el tiempo, en donde tanto la señal real e imaginaria son utilizadas para la representación de los símbolos, mientras que en el apartado (b) se muestra la constelación con los 16 símbolos, en donde notamos que cada símbolo no tan solo se diferencia por el cambio de fase, como en las modulaciones anteriores, si no también por un cambio en la amplitud. Los símbolos de la señal se observan en la sección (c) y a nivel de bit en la sección (d).

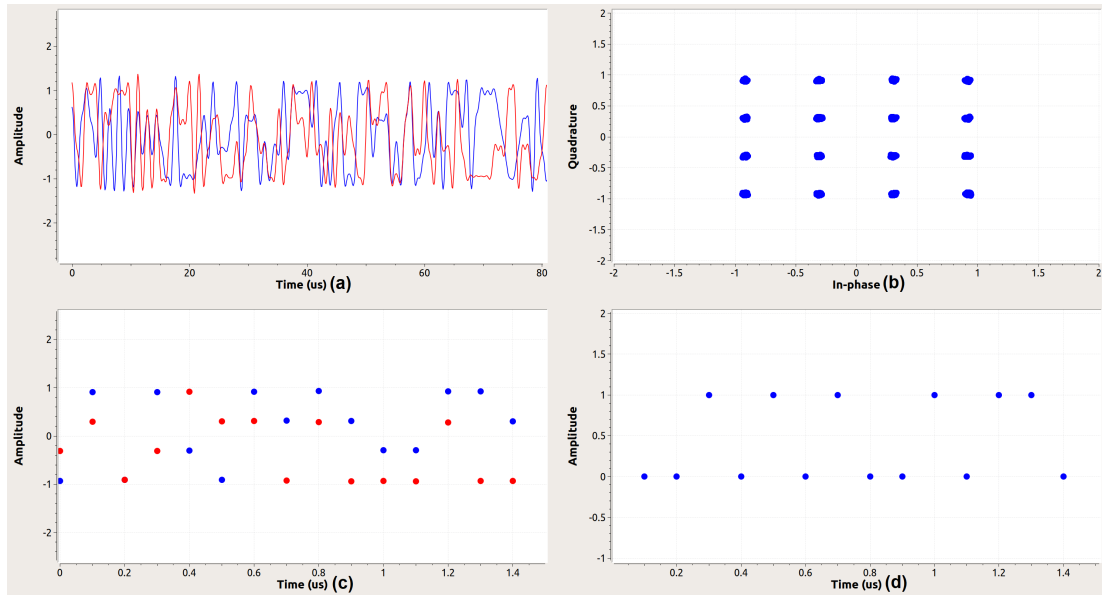


Figura 5.10: Pruebas de Modulación 16QAM

5.3. Pruebas de transmisión en VHF y UHF

Para llevar a cabo estas pruebas se montó el escenario de pruebas que se muestra en la figura 5.11, que consta esencialmente de dos secciones.

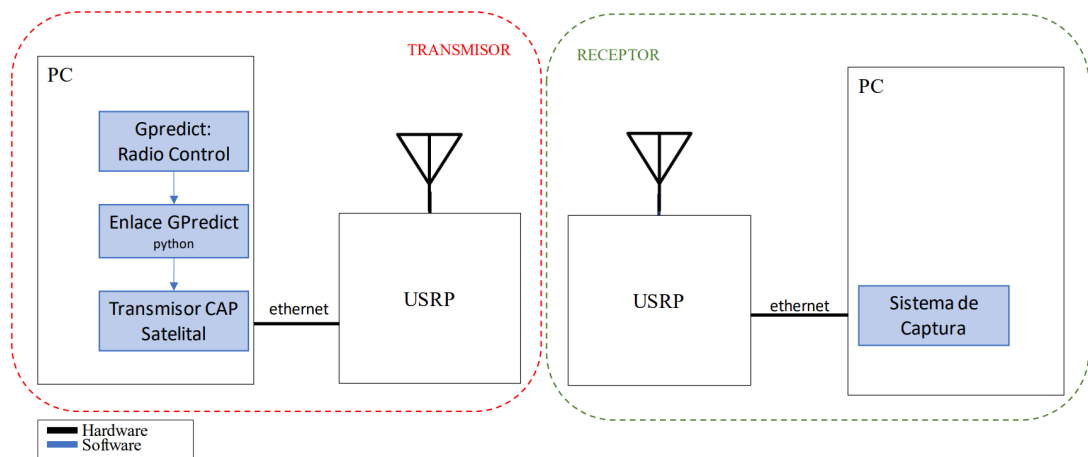


Figura 5.11: Diagrama del sistema de pruebas para transmisión RF

El transmisor está compuesto por la PC; dónde se ejecuta el Transmisor CAP Satelital de GNU Radio, el software de seguimiento satelital Gpredict y el programa

5. PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS OBTENIDOS

de conexión entre ambos, en Python. La PC se conecta con el SDR USRP X300, que cuenta con antena vertical omnidireccional tribanda de 144 MHz, 400 MHz y 1200 MHz *Comet Tri-Band HT SMA-703*.

La sección que representa el receptor está compuesta de igual manera por una PC conectado a un equipo USRP X300 con una antena *Comet Tri-Band HT SMA-703* como la anteriormente descrita. En la PC se ejecutó el programa de Sistema de Captura, desarrollado en el proyecto “Metodología para la evaluación en campo de redes SFN con ATSC en la Ciudad de México” del equipo de investigación [5].

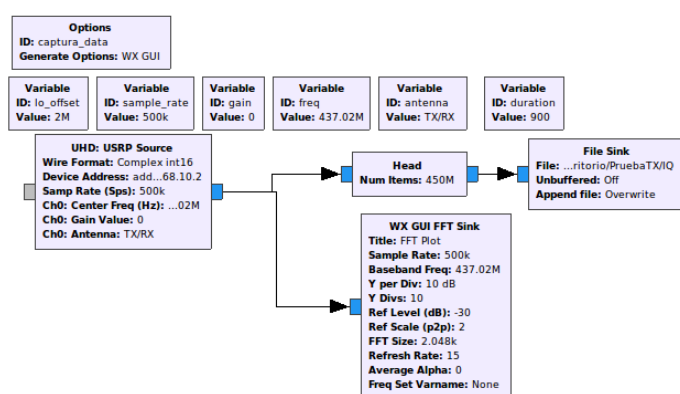


Figura 5.12: Sistema de Captura en GNURadio [5]

Para llevar a cabo las pruebas se selecciona en el programa de seguimiento satelital, GPredict, el satélite de interés, en este caso, se seleccionó un satélite próximo a tener huella sobre la posición deseada. Después en la PC del receptor ejecuta el programa de captura. En ese momento se puede apreciar en el programa receptor el nivel de ruido a través del analizador de frecuencia como se observa en la figura 5.13

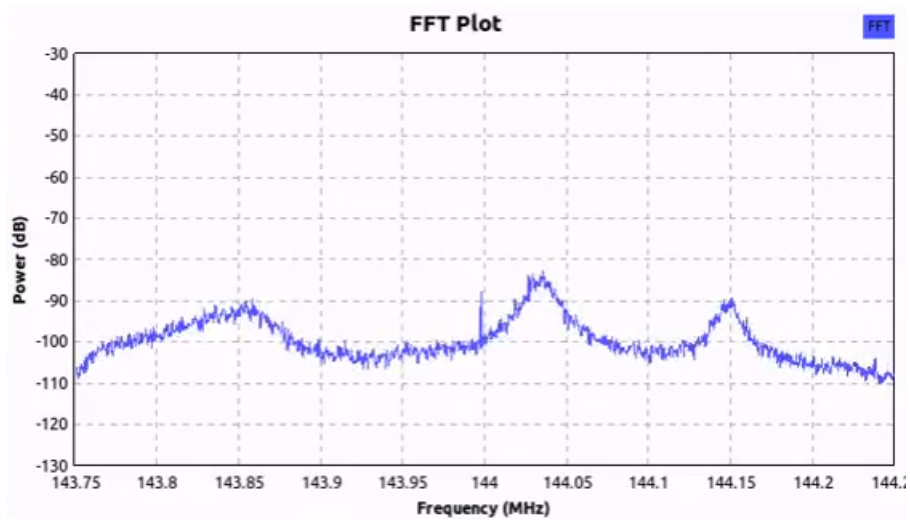


Figura 5.13: Ruido en Receptor

Una vez que el satélite comienza a tener huella sobre la posición cómo se muestra en la figura 5.14, se comienza la ejecución del transmisor CAP Satelital, el programa de conexión en Python y en la configuración del Gpredict se ejecuta en el módulo de radio control y se comienza la transmisión hacia el satélite al hacer clic en “engage” como se muestra en dicha figura, así como los valores de frecuencia que se están modificando.

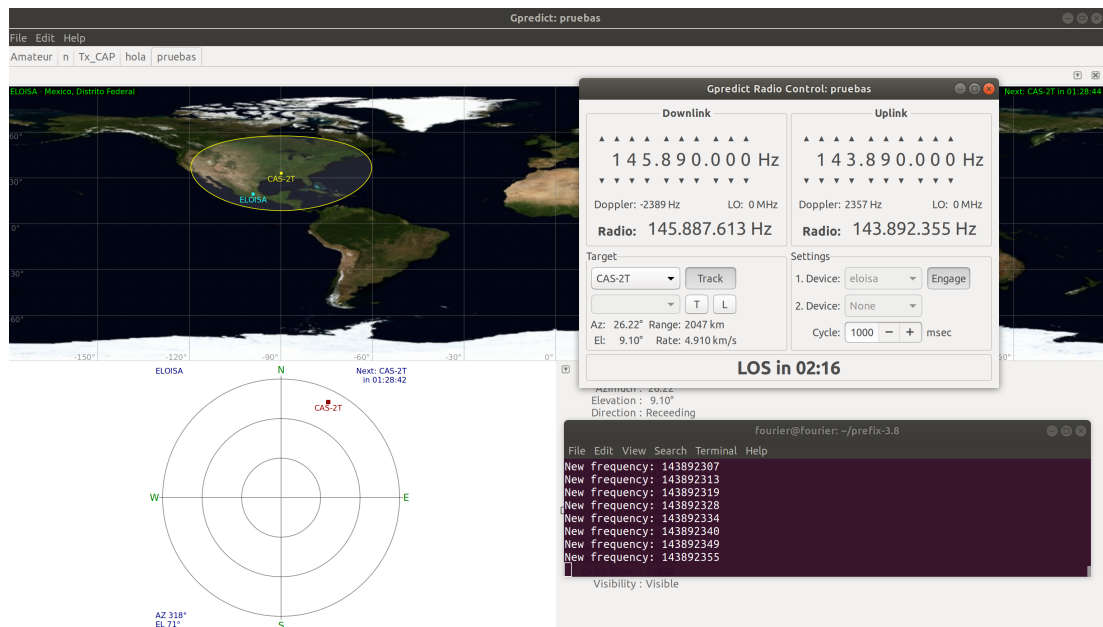
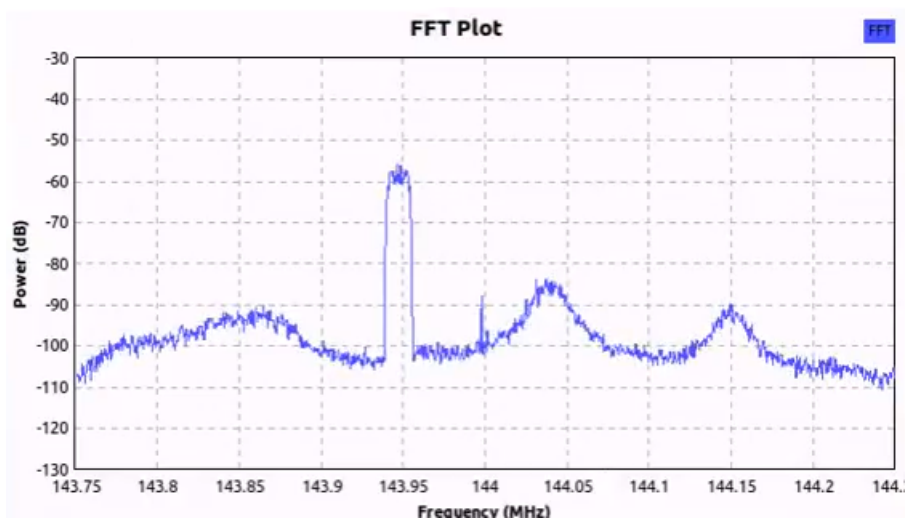


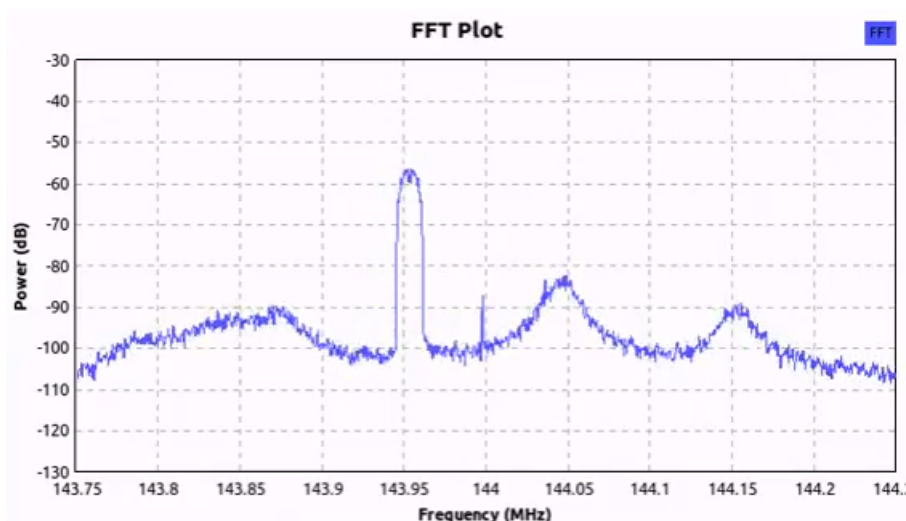
Figura 5.14: GPredict y programa en Python transmitiendo la frecuencia al transmisor.

5. PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS OBTENIDOS

La figura 5.15 muestra el espectro de frecuencia de la señal recibida en el receptor, donde se puede observar el corrimiento de frecuencia. En la primera imagen (a) se observa que la transmisión comienza con una frecuencia menor a la central y va aumentando gradualmente conforme el satélite se acerca a la posición central, hasta llegar al punto más cercano de la frecuencia central (b) y continua aumentando gradualmente hasta que se sale de la huella del satélite.



(a)



(b)

Figura 5.15: Desplazamiento de la señal por el efecto Doppler captada en el receptor

La segunda prueba se realizó en la banda de VHF con un satélite cuya posición

5.3 Pruebas de transmisión en VHF y UHF

inicial se observa en el mapa mostrado en la parte superior izquierda de figura 5.16 y se desplazará hacia el sur. En dicha figura también se muestran los datos del satélite seleccionado y el módulo de radio del Gpredict.

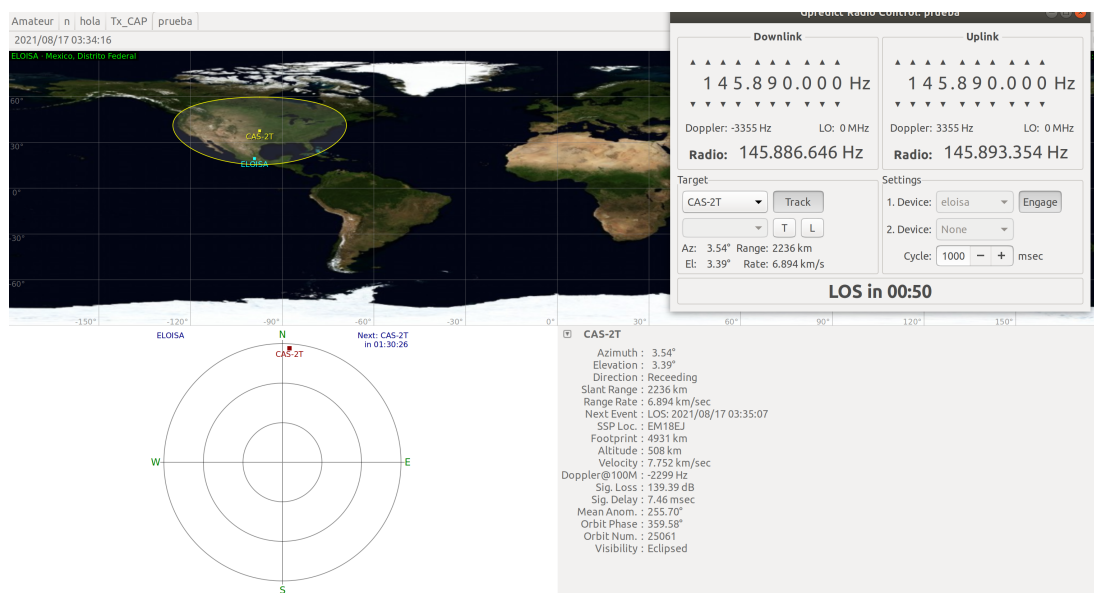


Figura 5.16: Posición inicial del satélite VHF

Esta vez se configuró el espectro de manera que quede registro de la medida más fuerte que se ha tenido. De esta forma quedará marcada la señal después de que se desplace por el efecto Doppler. En la figura 5.17 se puede observar el espectro de la señal después de que el satélite ya ha pasado por arriba de la estación terrena y como se puede observar la traza verde nos indica que la señal en un principio estuvo recorrida un poco a la izquierda, por lo que esto nos indica la correcta corrección del corrimiento de la frecuencia. Teniendo en la traza azul, la medición de la señal en el último momento de recepción de la señal.

5. PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS OBTENIDOS

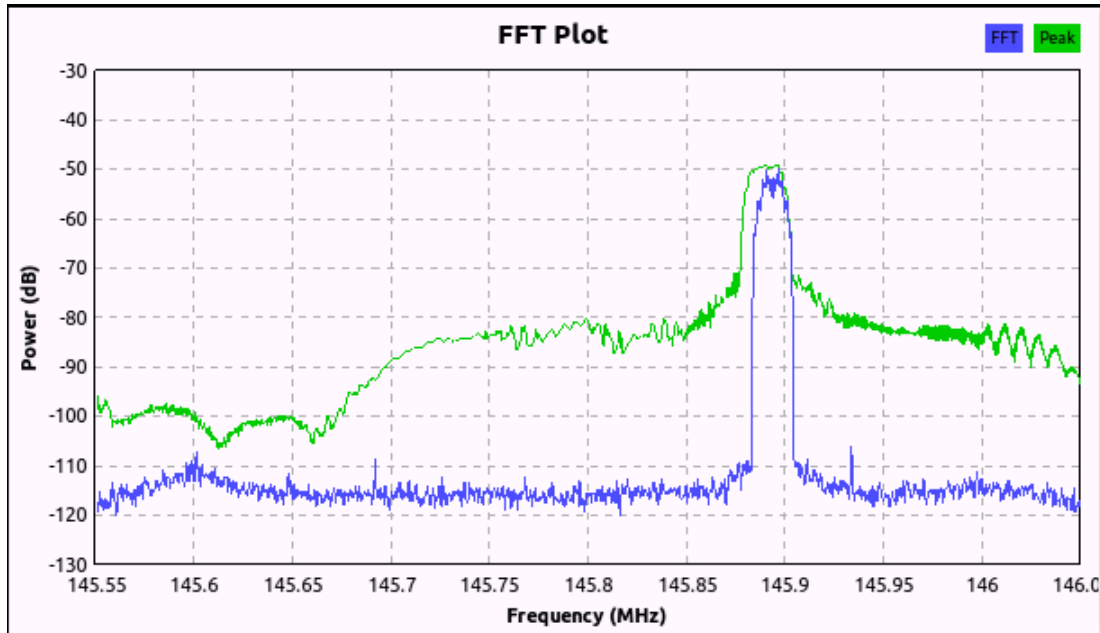


Figura 5.17: Corrimiento de la frecuencia en VHF

Para comprobar la capacidad del transmisor en distintas frecuencias, se realizó la prueba de transmisión en la banda UHF con la configuración mostrada en la figura 5.18. De igual forma, se seleccionó un satélite con frecuencia de “Uplink” en UHF próximo a pasar, tan pronto comienza a tener huella sobre la posición deseada se comienza con la transmisión.

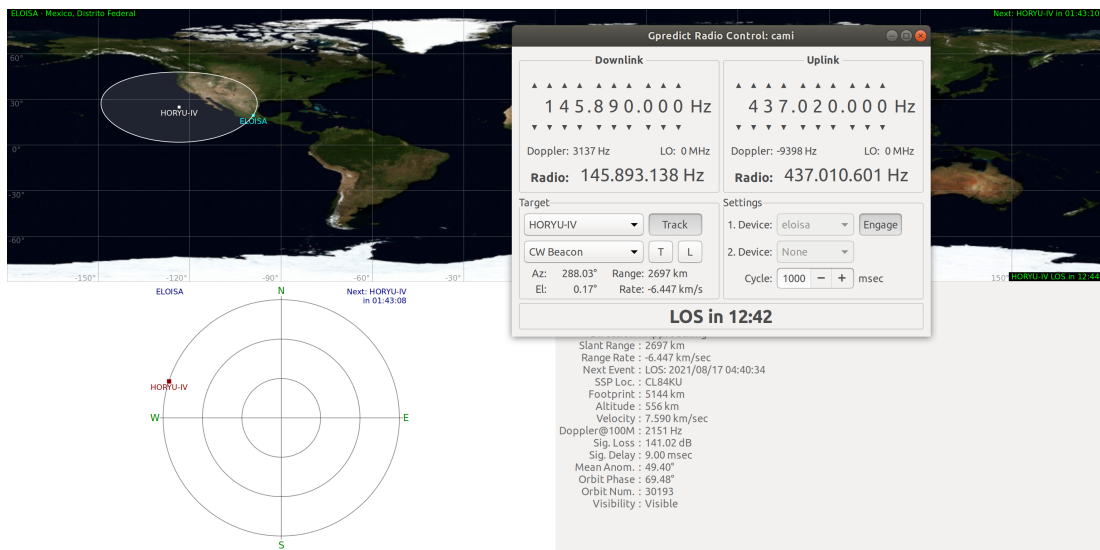


Figura 5.18: Gpredict: Posición inicial de satélite para prueba UHF

En el receptor, se comenzó a registrar la señal, en dónde podemos destacar el ancho de banda aproximado de 10 kHz. Conforme el satélite avanzaba sobre la posición de la estación terrena la frecuencia se modificaba correspondientemente con el efecto Doppler, quedando como posición última de la señal el espectro en color azul mostrado en la figura 5.19, mientras que en verde se muestra el espectro histórico de la señal observando los cambios en frecuencia en el tiempo.

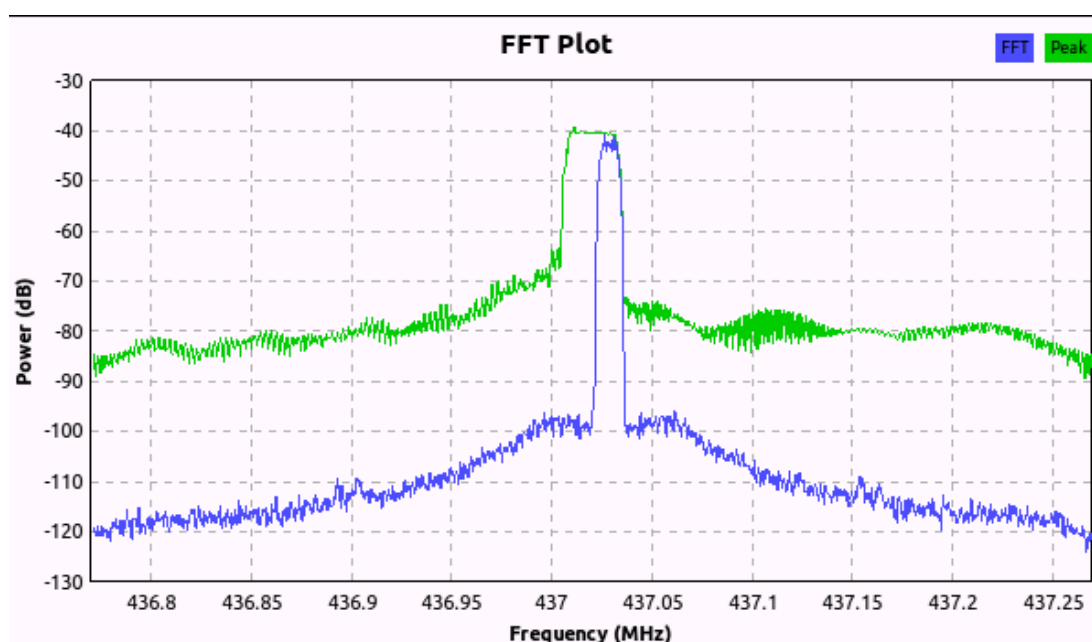


Figura 5.19: Desplazamiento de la frecuencia de la señal por efecto Doppler en UHF

Se incluyó el registro de la señal en el diagrama tipo waterfall. Este tipo de representación de la señal ayuda a visualizar sobre todo los cambios en frecuencia a lo largo del tiempo. cada línea horizontal representa el espectro de la señal en un instante, representando la intensidad o amplitud con una escala de colores que se indica a la derecha de la figura, el eje horizontal representa la frecuencia y el eje vertical el tiempo. Las medidas nuevas van saliendo de la parte de abajo y conforme aparecen nuevas muestras todas las anteriores se van recorriendo hacia arriba. En la figura 5.20, se puede observar el diagrama tipo waterfall donde se registra de manera más clara el desplazamiento de la frecuencia en el tiempo que ha durado la estación dentro de la huella del satélite. Si este desplazamiento no hubiera ocurrido se observaría en el diagrama una línea completamente vertical, sin embargo observamos como la frecuencia de transmisión fue aumentando gradualmente conforme el tiempo y la posición del satélite cambiaba.

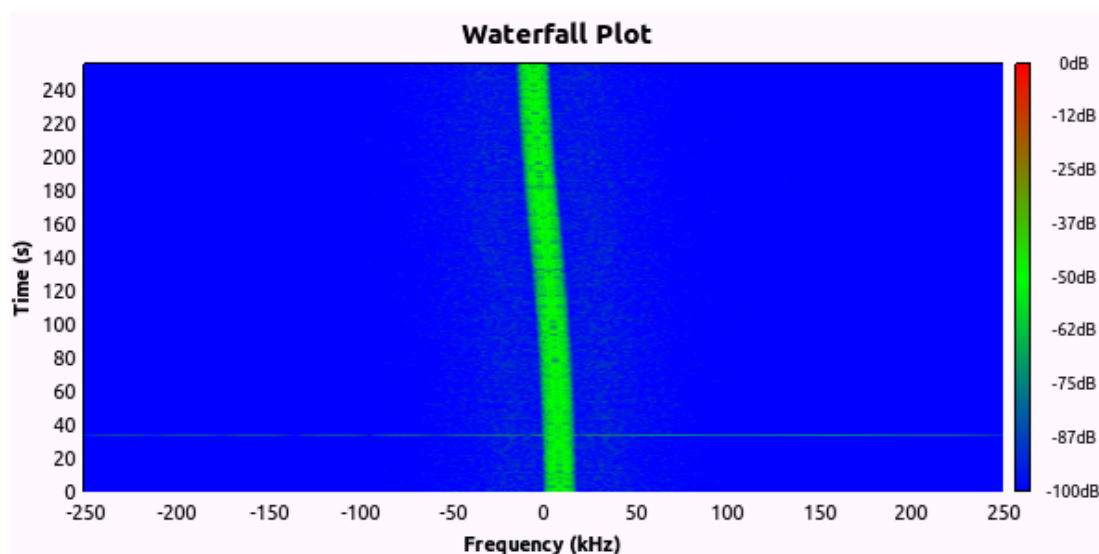


Figura 5.20: Esquema tipo Waterfall del desplazamiento de frecuencia en la señal UHF

Se observa que en este caso la variación por el efecto Doppler es más notoria que en la prueba de VHF, ya que el aumento de la frecuencia de transmisión tiene este efecto, como se ha explicado en la sección 4.3. En este caso el desplazamiento es aproximadamente de 18 kHz. Este valor obtenido en las pruebas de UHF es correspondiente en magnitud con el calculado teóricamente en el apartado 4.3.1.

Comparando en las dos bandas de frecuencias, como era de esperarse, el desplazamiento de la frecuencia debido al efecto Doppler fue mayor en la banda UHF, dado que es de mayor frecuencia, y por lo tanto es más notorio y se alcanza a apreciar mejor en el esquema de waterfall.

En las pruebas finales se puso en marcha tanto el seguimiento automático de los satélites, como la conexión con GNU Radio y el transmisor satelital, pudiendo ver la corrección automática del corrimiento de frecuencia. En todas las pruebas se logró mostrar que se llegó al funcionamiento esperado.

Conclusiones

El objetivo principal de esta tesis se cumplió al lograr desarrollar un transmisor satelital mediante el uso de radio definida por software (SDR) para el envío de mensajes de alerta de emergencia.

En las primeras etapas del desarrollo se llevó a cabo la rehabilitación de la estación terrena del Departamento de Telecomunicaciones de la Facultad de Ingeniería, con el propósito de integrar el transmisor CAP satelital a esta infraestructura. Para lo cual se realizó un diagnóstico del estado inicial de los equipos que componen la estación terrena, primeramente se realizaron labores de limpieza y organización de los equipos.

Posteriormente se estableció y documentó el diagrama de conexiones de los equipos para así contar con una fuente de consulta en futuros proyectos. Con esto se comprendió el flujo de comunicación entre equipos y software, lo que permitió resolver distintos fallos en el mismo, así como familiarizarnos con los diferentes componentes de una estación de transmisión.

Se configuró el TNC, dispositivo que posibilita a la estación terrena para recibir y transmitir mensajes digitales. Para ello se requirió utilizar un emulador de terminal para realizar la comunicación entre el equipo de cómputo y el dispositivo.

Se calibraron de manera manual las antenas y los controladores de los rotores que posibilitan el seguimiento de satélites y se realizaron pruebas confirmando el buen funcionamiento de las antenas.

Finalmente, con estas acciones se consiguió tener una estación terrena funcional, por lo que se comenzaron las pruebas de recepción con satélites de órbita baja en las bandas de VHF y UHF, logrando tener en completo funcionamiento la estación terrena.

Con la experiencia de haber trabajado con la estación terrena se definieron los requerimientos del transmisor CAP satelital. Se decidió utilizar radio definida por software en conjunto con la plataforma de desarrollo GNU Radio, para tomar provecho de su flexibilidad en el desarrollo del transmisor.

El uso de la plataforma de desarrollo GNU Radio de software libre nos benefició porque es una herramienta de uso gratuito y con amplia variedad de funcionalidades para el desarrollo de radio definida por software y además compatible con los equipos con los que contaba el grupo de trabajo, lo que facilitó la implementación del transmisor.

6. CONCLUSIONES

Sin embargo, en ocasiones la documentación de software libre puede parecer deficiente. En comparación con los equipos tradicionales de transmisión con los que contaba la estación, la implementación del transmisor con SDR nos dio muchas más opciones y funcionalidades, lo que se reflejó en la facilidad de implementación de los módulos de lectura de datos, modulación y filtros. Así como nos permitió realizar la conexión con otras herramientas y realizar pruebas de manera simulada, adicionando bloques de prueba.

Se eligió Gpredict como el software de seguimiento de satélites artificiales, mismo que realiza el cálculo de la posición que deben tener las antenas y el desplazamiento de frecuencia necesario para corregir el efecto Doppler. En comparación con otras plataformas de seguimiento de satélites, Gpredict es un software libre, disponible en Linux, que además incluye un controlador, llamado Hamlib, que es compatible con una gran variedad de equipos de radio. Este controlador fue una base importante para poder desarrollar la conexión entre Gpredict y el transmisor en SDR.

Para enlazar Gpredict con el transmisor desarrollado en SDR, se evaluaron distintas soluciones, finalmente se desarrolló una interfaz de comunicación mediante un programa en Python que envía los datos calculados por Gpredict hacia la plataforma de desarrollo de SDR, GNU Radio. Esto no fue una tarea sencilla, sin embargo con esta conexión se logró tener un seguimiento automático de los satélites, en dónde además Gpredict también realizará el control de los rotores.

La decisión de desarrollar el transmisor en SDR permitió usar la infraestructura ya existente de la estación terrena y los equipos USRP pertenecientes al grupo de investigación. Sin embargo, se requiere adquirir un amplificador de potencia para colocarse a la salida de los equipos SDR puesto que los radios no tienen la potencia suficiente para transmitir a un satélite de órbita baja.

En el desarrollo del transmisor CAP satelital se logró integrar la fuente de datos que nos permitiera enviar el mensaje de emergencia CAP, incluir distintas modulaciones digitales adecuadas para la transmisión satelital, así como obtener la corrección de frecuencia necesaria para contrarrestar el efecto Doppler y dentro del transmisor indicar la modificación de la frecuencia en tiempo real.

Para llevar a cabo las pruebas de funcionamiento del transmisor fue necesario desarrollar una sección del programa que realiza parte de las tareas de un receptor. Al realizar las pruebas se comprobó que los datos se estaban transmitiendo de manera correcta, ya que al visualizar la señal demodulada se corroboró que correspondía con los datos de entrada a nivel de bit. También se verificó que la modulación de la señal se realizó correctamente al visualizar la constelación correspondiente a cada tipo de modulación en los bloques de prueba. Finalmente, constatamos la corrección del desplazamiento Doppler, al observar que la frecuencia central de transmisión se desplazaba correspondientemente con el paso del satélite y con la frecuencia calculada por el Gpredict.

Si bien el alcance de este proyecto no fue específicamente mostrar los efectos de diferentes modulaciones, nos sirvió para reforzar conceptos como el manejo del nivel de ruido, el ancho de banda y los errores en la transmisión, pero sobre todo para probar

el funcionamiento del transmisor y su flexibilidad.

Como trabajo futuro, complementario a este proyecto, se podrá adicionar esta sección con el decodificador del mensaje para poder extraer el archivo XML y así obtener el receptor de protocolo CAP, además de la adquisición del amplificador de potencia para tener la estación completamente funcional.

En la elaboración de este trabajo se hizo notable que el uso de Radio Definido por software otorga beneficios sobresalientes en comparación del uso de sistemas de radio convencionales, como la reducción del tiempo de desarrollo. Sin embargo, algunas desventajas que notamos al realizar este proyecto es que los equipos SDR utilizados son costosos en comparación con equipos convencionales y requieren de fuentes externas para amplificación de potencia.

En los siguientes anexos se documentan los pasos concisos para la instalación de diferentes componentes de software que fueron necesarios para la realización del proyecto. Con el fin de que en futuros trabajos se pueda acceder rápidamente a esta información.

Anexo I. GNU Radio

GNU Radio es una herramienta de desarrollo de software de código abierto y gratuito que proporciona bloques de procesamiento de señales para implementar radios de software. Se usa ampliamente en entornos de aficionados, académicos y comerciales para respaldar tanto la investigación de comunicaciones inalámbricas como los sistemas de radio del mundo real.

Instalación

Al ser un software libre GNU Radio se encuentra disponible de manera gratuita en la plataforma Github¹, este software solamente es el compatible con los sistemas operativos basados en linux.

Se requiere instalar con antelación algunas bibliotecas base, mismas que se podrán instalar con ejecutar el siguiente comando directo en la terminal del sistema operativo Linux

```
sudo apt-get -y install git swig cmake doxygen build-essential libboost-  
all-dev libtool libusb-1.0-0 libusb-1.0-0-dev libudev-dev libncurses5-  
dev libfftw3-bin libfftw3-dev libfftw3-doc libcppunit-1.14-0  
libcppunit-dev libcppunit-doc ncurses-bin cpufrequtils python-numpy  
python-numpy-doc python-numpy-dbg python-scipy python-docutils qt4-bin  
-dbg qt4-default qt4-doc libqt4-dev libqt4-dev-bin python-qt4 python-  
qt4-dbg python-qt4-dev python-qt4-doc python-qt4-doc libqwt6abi1
```

¹<https://github.com/csete/Gpredict/releases/25/11/2021>

```
libfftw3-bin libfftw3-dev libfftw3-doc ncurses-bin libncurses5
libncurses5-dev libncurses5-dbg libfontconfig1-dev libxrender-dev
libpulse-dev swig g++ automake autoconf libtool python-dev libfftw3-
dev libcppunit-dev libboost-all-dev libusb-dev libusb-1.0-0-dev fort77
libstdl1.2-dev python-wxgtk3.0 git libqt4-dev python-numpy ccache
python-opengl libgsl-dev python-cheetah python-mako python-lxml
doxygen qt4-default qt4-dev-tools libusb-1.0-0-dev libqwtplot3d-qt5-
dev pyqt4-dev-tools python-qt5-qt4 cmake git wget libxi-dev gtk2-
engines-pixbuf r-base-dev python-tk liborc-0.4-0 liborc-0.4-dev
libasound2-dev python-gtk2 libzmq3-dev libzmq5 python-requests python-
sphinx libcomedi-dev python-zmq libqwt-dev libqwt6abi1 python-six
libgps-dev libgps23 gpsd gpsd-clients python-gps python-setuptools
```

En el presente trabajo se usó el método de instalación PyBOMBS. PyBOMBS es una biblioteca útil para la instalación de GNU Radio, cuando se requiere usar módulos externos (módulos Out of Tree) como UHD para la conexión con los equipos de hardware como se utilizó en este desarrollo. La instalación se realiza en un directorio de usuario específico en lugar de en los archivos del sistema.

Por medio de los siguientes comandos, al ser ejecutados desde la terminal, se realizará la instalación.

```
1 sudo apt-get install python3-pip
  sudo pip3 install pybombs
3 pybombs auto-config
  pybombs recipes add-defaults
5 pybombs prefix init ~/prefix-3.8 -R gnuradio-default
```

Para comenzar con GNU radio es necesario abrir una terminal en la carpeta designada para la instalación, en este caso “prefix-3.8” y desde la terminal del SO ejecutar los siguientes comandos, que generaran un ambiente para la correcta operación de GNU Radio.

```
1 source ~/prefix-3.8/setup_env.sh
  gnuradio-companion
```

El método presentado en este apartado ha sido el que mejor funcionamiento presentó entre la variedad de formas de instalación que ofrece la comunidad de GNU Radio, sin embargo, se invita al lector a elegir cualquier otro método de instalación que sea pertinente de acuerdo a sus necesidades. Se puede encontrar más información al respecto directamente en el repositorio de GNU Radio¹ y de empresas que utilizan esta tecnología y ofrecen guías de instalación².

¹<https://github.com/gnuradio/pybombs>

²[https://kb.ettus.com/Building_and_installing_the_UHF_SRP_Open-Source_Toolchain_\(UHD_and_GNU_Radio\)_on_Linux](https://kb.ettus.com/Building_and_installing_the_UHF_SRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux)

Anexo II. Gpredict

El Gpredict es un software libre capaz de calcular las trayectorias de los satélites, y la posición que deben tener las antenas para poder hacer el seguimiento, así como el cálculo del cambio de la frecuencia debido al efecto Doppler.

Instalación

Para la instalación del Gpredict se puede descargar el código fuente desde el repositorio en Github¹ y descomprimir el paquete con:

```
tar -xvfz Gpredict-x.y.z.tar.bz2
```

Se cambia al directorio “Gpredict-x.y.z” y se ejecutan los siguientes comandos

```
1 ./configure
make
3 sudo make install
```

Este proceso puede llegar a tardar un poco, después de eso ya se tiene instalado el programa. Ahora hay que configurarlo.

Configuración

Para obtener información más detallada se puede recurrir al manual de usuario.² Muchas de las siguientes configuraciones fueron recuperadas de ahí.

Una vez completada la instalación se puede ejecutar el programa simplemente buscando el Gpredict en le menú y se mostrará una interfaz como la mostrada en la figura A1.

¹<https://github.com/csete/Gpredict/releases>

²<https://github.com/csete/Gpredict/raw/master/doc/um/Gpredict-user-manual.odt>

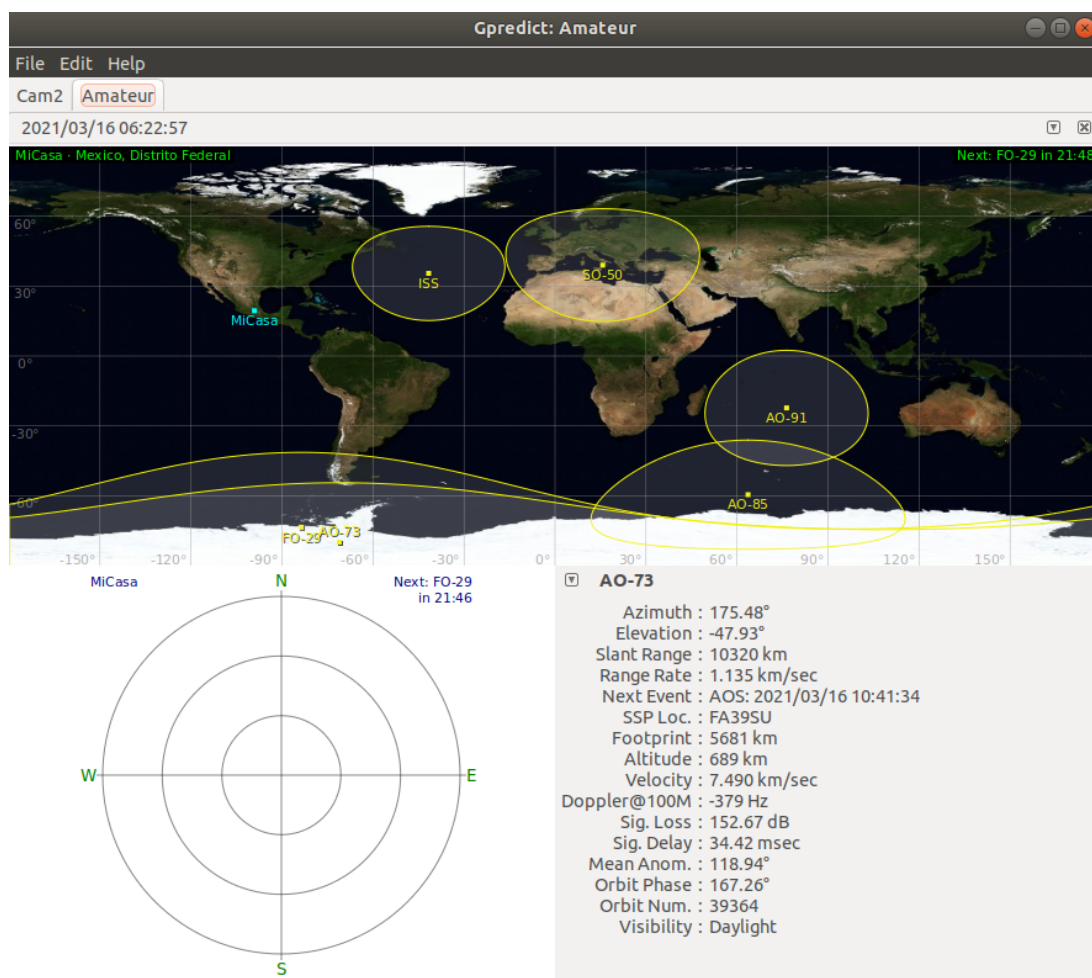


Figura A1: Interfaz del Gpredict

En la interfaz se puede ver un mapa con los distintos satélites que estén configurados para poder seguir y también dentro del mapa se muestran las estaciones terrenas configuradas. En el caso de la figura A1 se está viendo el módulo que viene ya configurado por defecto cuando se descarga y abre por primera vez Gpredict.

En el extremo inferior derecho viene información específica de un satélite en particular y se puede observar que ésta cambia conforme pasa el tiempo y el satélite se mueve de posición. Si se requiere ver la información de otro satélite se puede hacer clic en el nombre del satélite sobre el mapa.

También se puede hacer clic en la flecha que viene al lado del nombre del satélite como lo indica la figura A2 y ahí seleccionar el satélite.

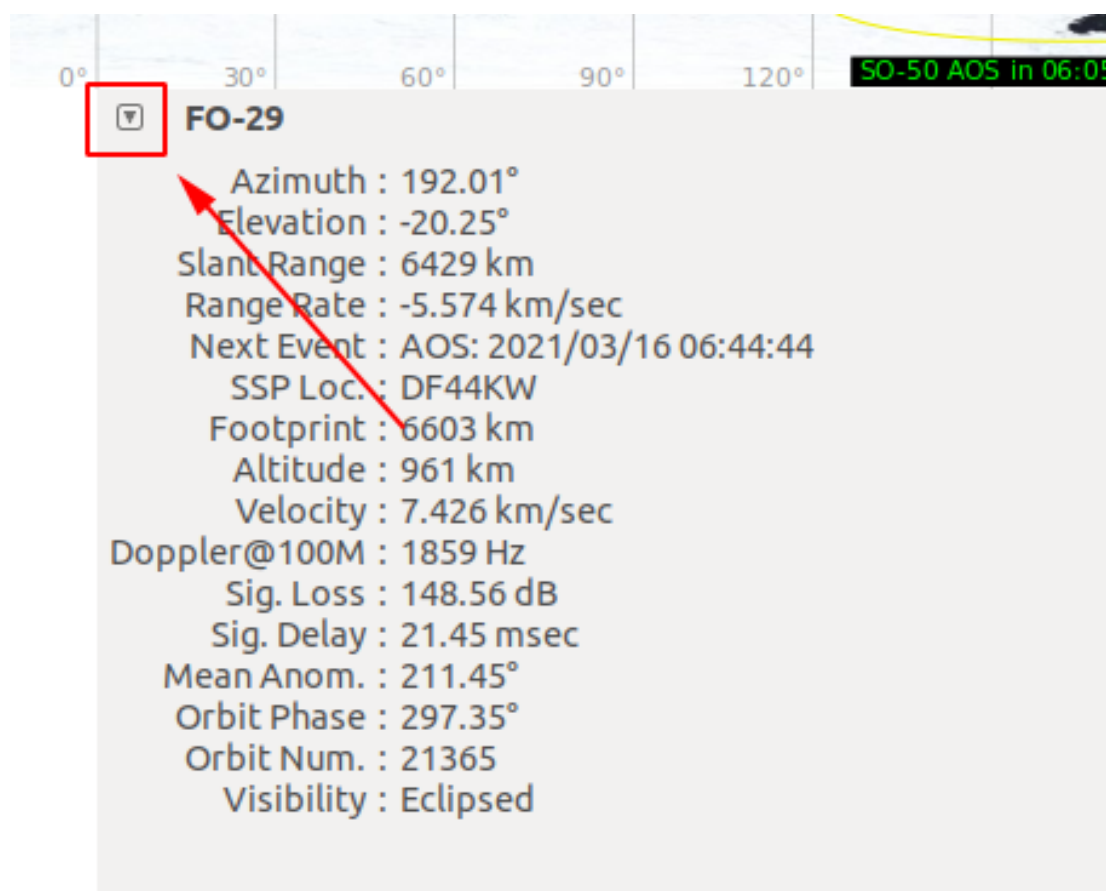


Figura A2: Datos de un satélite mostrados en GPredict

Se desplegará un menú en el que podemos seleccionar un satélite o, del satélite ya seleccionado calcular los momentos en que el satélite esté en línea de vista desde la estación terrena.

Es importante tener los datos TLE (two-line element set) actualizados para que el cálculo de las órbitas sea correcto, para esto Gpredict proporciona una función para actualizar dichos datos de internet automáticamente, sin la necesidad de buscar en otras fuentes externas. Solo se hace clic en “Edit / Update TLE data from network” y ya están actualizados los datos.

Después hay que configurar la estación terrena, para esto se va al menú “Edit / Preferences” y se mostrará una ventana como la mostrada en la figura A3.

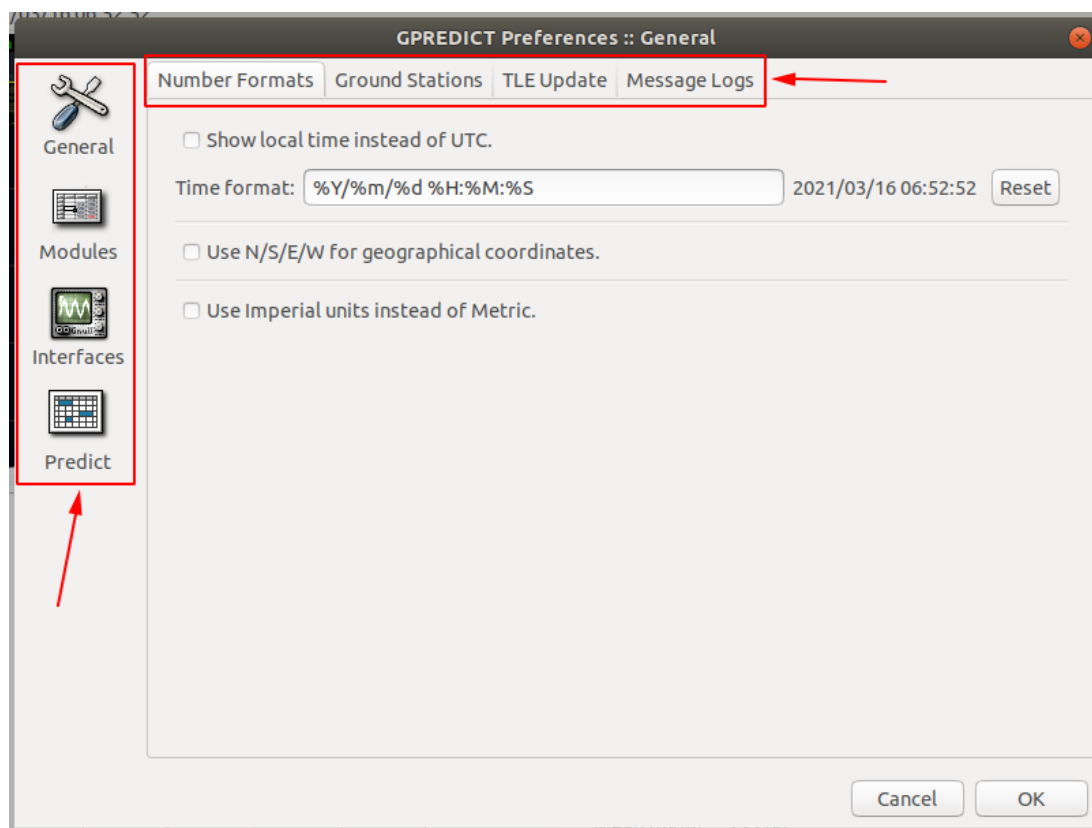


Figura A3: Menú "Preferencias" del GPredict

De lado izquierdo de la figura A3 se pueden ver varios menús que serán de ayuda más adelante, cada menú tiene diferentes pestañas que se muestran en la parte de arriba.

Para la estación terrena se selecciona del menú "General" la pestaña "Ground Station", se muestra una lista de las estaciones terrenas configuradas, si es la primera vez que se entra solo se debe tener una estación llamada sample, que viene en la configuración por defecto. Se hace clic en el botón "Add new" y aparece una ventana con las opciones para agregar la estación terrena, como la mostrada en la figura A4.

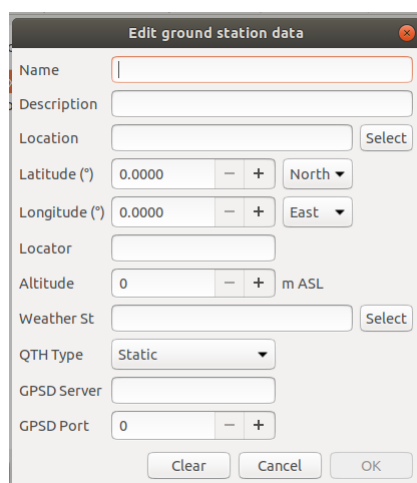


Figura A4: Agregar una estación terrena

- Aquí lo más importante son registrar las coordenadas de donde se tiene la estación terrena y algún nombre que la distinga por si se tienen registradas varias.

Después, en la misma ventana de preferencias en el menú “interfaces” se hace clic en “Add new” y aquí se llenan los datos del radio, aquí lo importante es la dirección a la que se enviara la información (host, port). Por defecto se va al local host en el puerto 4532.

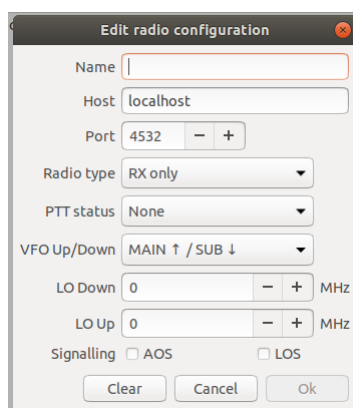


Figura A5: Configuración de radio

Por ultimo se crea el módulo, un módulo es todo el conjunto de satélites y estaciones que se tienen configuradas para que se muestren en un solo layout. Se sale de la ventana de preferencias se entra a “File / Create new module” y se mostrará una ventana como la siguiente:

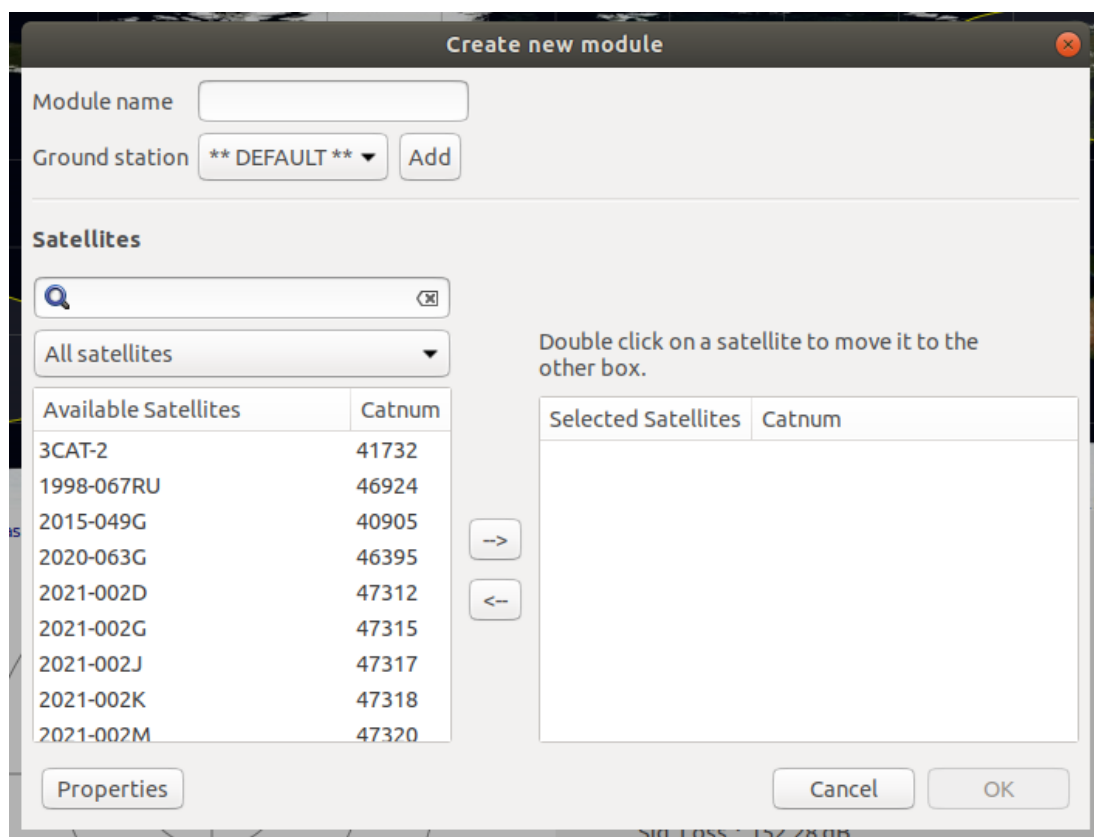


Figura A6: Creación de un nuevo módulo

Se pone algún nombre al nuevo módulo, por ejemplo, “Radio aficionado”, seleccionar la estación terrena que se quiera (previamente configurada) y seleccionar los satélites a los cuales se les quiera hacer seguimiento, en este ejemplo podríamos seleccionar satélites dedicados a radio afición.

Con estos pasos ya se tiene configurado el Gpredict y se puede ver el nuevo módulo como una pestaña en el Gpredict. Para empezar el cálculo y seguimiento de algún satélite, en el módulo que se ha creado, en la Figura A7 se muestra en la esquina superior derecha una flecha, se da clic ahí y se despliega un menú del cual las opciones que interesan en este momento son “radio control” y “antenna control”.

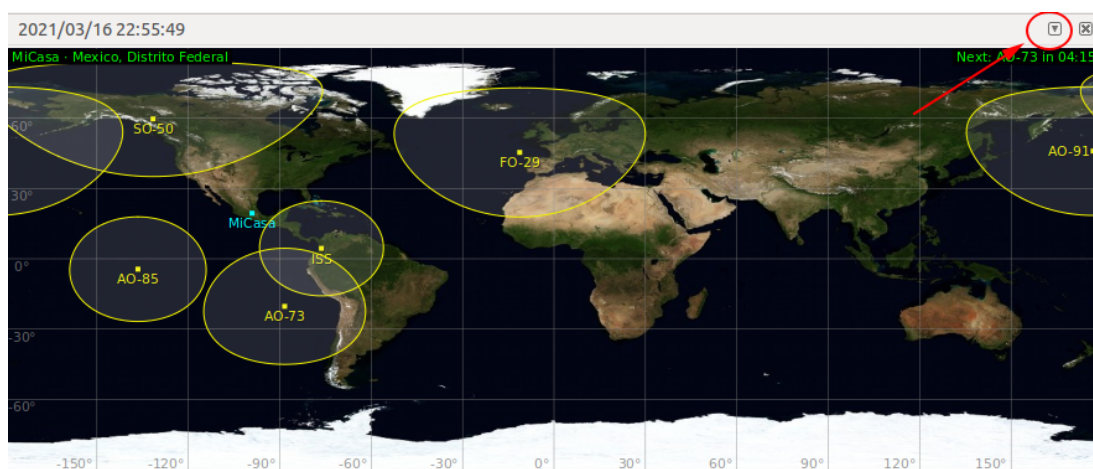


Figura A7: Seguimiento del satélite

“Antenna control” sirve como un módulo auxiliar para el control de los rotores para que se muevan las antenas y “radio control” calcula la frecuencia Doppler y le avisa al radio del cambio en frecuencia. Si se hace clic en el “radio control” nos aparece una ventana como en la figura A8.

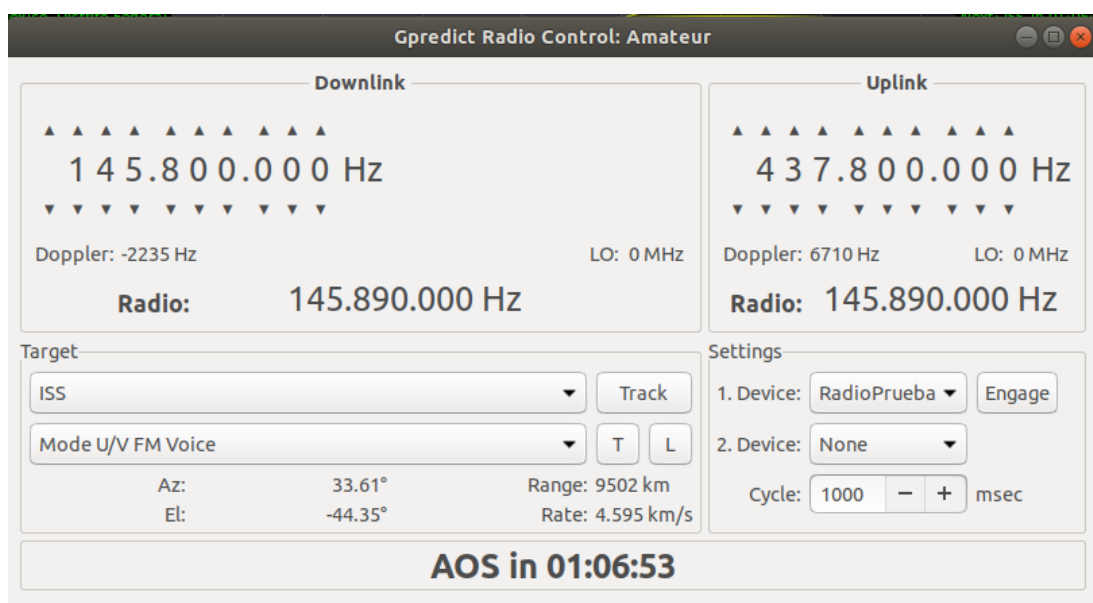


Figura A8: Radio control

Se pueden distinguir cuatro secciones:

1. Downlink: Muestra la frecuencia central y la corregida por el efecto Doppler del enlace de bajada

2. Uplink: De manera similar que en downlink pero ahora para el enlace de subida.
3. Target: Aquí se selecciona el satélite que se requiere seguir y el tipo de comunicación que habrá, esto depende del satélite, por ejemplo, la ISS¹ tiene varios tipos de comunicación, como voz por FM o un repetidor de mensaje también en FM, entre otros.
4. Settings: Aquí se selecciona el o los radios al que se les mandará la información

En la sección de “Target” se encuentra el botón “Track”, en el momento que se presiona este botón el Gpredict empezará a calcular la frecuencia Doppler para el satélite seleccionado, pero no es hasta que se presiona el botón “Engage” de la sección “Settings” que se mandará la información al radio. Si cuando se selecciona el botón “engage”, el radio no se logra comunicar con el Gpredict, el botón se desselecciona automáticamente y hay que volver a intentarlo.

¹Estación Espacial Internacional

Anexo III. Hamlib

Hamlib es una biblioteca que provee a los distintos programas una API para controlar gran variedad de rotores y radios para comunicaciones satelitales. El uso de esta biblioteca está recomendada y documentada por el mismo manual de usuario de Gpredict en el capítulo 7 del mismo.

Instalación

Primeramente para instalar hamlib se acude al repositorio de github¹ donde se encuentra la versión estable más actual y descargar el tar y compilarlo o alternativamente podemos correr los siguientes comandos.

```
1 sudo apt-get update -y
   sudo apt-get install -y hamlib-utils
```

Configuración

Una vez instalado se puede hacer uso de varios comandos.

- **rigctl** – Permite mandar comandos a nuestro radio por la línea de comandos (si el radio está conectado vía USB)
- **rotctl** – Permite mandar comandos a los rotores de la antena por la línea de comandos (si el rotor está conectado vía USB)
- **rigctld** – Permite mandar comandos a nuestro radio vía protocolo TCP/IP
- **rotctld** – Permite mandar comandos a los rotores de la antena vía protocolo TCP/IP

Es muy importante especificar y configurar bien la comunicación con el radio, ya que de hacerlo mal se podría dañar el equipo. Si se tiene el equipo físicamente se puede buscar si es compatible y el número con el que está registrado en la base de datos de Hamlib con los comandos siguientes, para el radio y los rotores respectivamente.

```
1 rigctl --list
2 rotctl --list
```

¹<https://hamlib.github.io/>

Hamlib provee un radio ficticio o de prueba para aprender a usar el driver, este tiene el modelo = 1. Otras opciones importantes de los comandos son especificar la dirección IP y el puerto.

Existen otras opciones para estos comandos, si es de interés del lector las puede revisar con el comando. - rigctld -help -

Referencias

- [1] CENAPRED, *Desastres en México: Impacto social y económico*. 2019. www.cenapred.unam.mx/es/Publicaciones/archivos/318-INFOGRAFADIASAS_TRESENMXICO-IMPACTOSOCIALYECONMICO.PDF Consultado el 30/06/2021. ix, 3, 5, 10, 11
- [2] hy gain, *Instruction Manual UB-7030SAT*. https://www.radioworld.ca/related/product_manuals/amateur_accessories/antennas/hygain/hyg-ub7030sat_manual.pdf Consultado el 20/06/2021. ix, 26
- [3] cap.org, *CAP Alert Editor*. <https://cap.alert-hub.org/login.php> Consultado el 25/06/2021. ix, 30
- [4] G. Radio, *GNU Radio Manual and C++ API Reference*. https://www.gnuradio.org/doc/doxygen/classgr_1_1uhd_1_1usrp__sink.html Consultado el 05/07/2021. x, 54
- [5] A. Zentella, *Metodología para la evaluación en campo de redes SFN con ATSC en la ciudad de México*. Posgrado UNAM, 2017. x, 70
- [6] SINAPROC, *Informe Anual 2020*. 2020. www.preparados.cenapred.unam.mx/documentos/20201230_Informe%20anual%20CNPC.pdf Consultado el 30/06/2021. 5
- [7] CENAPRED, *Sistemas de Alerta Temprana*. 2019. www.cenapred.gob.mx/es/Publicaciones/archivos/298-INFOGRAFASISTEMASDEALERTATEMPRANA.PDF Consultado el 30/06/2021. 9, 10
- [8] P. Carrizo, *Crean un sistema satelital de alerta y respuesta temprana contra incendios forestales*. 2017. <https://unciencia.unc.edu.ar/medioambiente> Consultado el 20/07/2021. 9
- [9] E. de Armas Sardiñas, *An Integral Emergency Alert System for Mexico*. IEEE, 2021. 11

REFERENCIAS

- [10] E. Jones and J. Westfall, *Common Alerting Protocol Version 1.2*. 2017. <http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2.html> Consultado el 30/06/2021. 11
- [11] Google, *What is Google Public Alerts?* https://support.google.com/publicalerts/answer/1753666?hl=en&ref_topic=1753665 Consultado el 30/06/2021. 11
- [12] ITU-T, *Rec. X1303 Bis. Common alerting protocol (CAP 1.2)*. 2014. <http://handle.itu.int/11.1002/1000/12150> Consultado el 30/06/2021. 11
- [13] W. I. Forum, *What is Software Defined Radio*. <https://www.wirelessinnovation.org/assets/documents/SoftwareDefinedRadio.pdf> Consultado el 30/06/2021. 14
- [14] V. K. Garg, *Wireless Communications and Networking*. 500 Sansome Street, Suite 400, San Francisco, CA 94111: Morgan Kaufmann, 2007. 14
- [15] G. Radio, *What is GNU Radio?* https://wiki.gnuradio.org/index.php/What_is_GNU_Radio%3F Consultado el 30/06/2021. 15
- [16] G. Radio, *GNU Radio Main Page*. https://wiki.gnuradio.org/index.php/Main_Page Consultado el 30/06/2021. 15
- [17] Kantronics, *Guia de Uso Kantronics Packet Communicator 9612+*. 2015. <https://kantronics.com/wp-content/uploads/2018/10/KPC-9612Plus-Manual-RevG.pdf> Consultado el 30/06/2021. 25
- [18] R. Neri, *Líneas de Transmisión*. México: Univeridad Veracruzana, 2013. 26
- [19] S. L. A. Rodolfo Neri Vela, *Comunicaciones por satelite*. Universidad Veracruzana, 2015. 46