



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

DESARROLLO Y EVALUACIÓN DE UN SISTEMA DE COMUNICACIÓN PARA UN
NANOSATÉLITE

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

MATÍAS GABRIEL VIDAL VALLADARES

PROFESOR GUÍA:
MARCOS DÍAZ QUEZADA

MIEMBROS DE LA COMISIÓN:
CLAUDIO ESTÉVEZ MONTERO
TOMÁS OPAZO TORO

SANTIAGO DE CHILE
2016

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: MATÍAS GABRIEL VIDAL VALLADARES
FECHA: 2016
PROF. GUÍA: SR. MARCOS DÍAZ QUEZADA

DESARROLLO Y EVALUACIÓN DE UN SISTEMA DE COMUNICACIÓN PARA UN NANOSATÉLITE

El avance científico y tecnológico en materia aeroespacial, más la experiencia acumulada a lo largo de los años por centros de investigación, ha permitido que se mejoren las técnicas de diseño e integración de satélites y los experimentos de diversa índole que estos transportan al espacio. El paradigma clásico consiste en diseñar la carga útil del satélite y luego el vehículo que la transportará. Esto provoca que cada nuevo satélite tiene una forma personalizada, lo cual implica en mayores costos para el cohete que lo va a enviar al espacio. Los CubeSat rompen ese esquema al establecer un peso, volumen y forma bien definidos, de tal manera que la carga o *payload* del satélite debe ajustarse a esa especificación. Esto implica una drástica disminución del costo de misiones espaciales, ya que se ha transformado en un estándar adoptado por decenas de universidades, centros tecnológicos y organizaciones en el mundo lo que conlleva a que el sistema de acoplado al cohete sea también estándar.

Una de las partes esenciales de un CubeSat es su sistema de comunicación, el cuál a pesar de tener una tendencia a disminuir su costo, sigue teniendo un precio elevado, transformándose en una barrera de entrada para establecimientos educacionales que ven en los nanosatélites una herramienta no sólo de investigación sino que también educativa. Un transceptor de bajo costo permite no solo reducir el impacto económico del nanosatélite, sino que también agregar redundancia al sistema, teniendo la posibilidad de transportar más de uno en un mismo CubeSat.

Esta memoria se enmarca en el SUCHAI 2 y 3, los cuales son la continuación del primer nanosatélite de la Universidad de Chile, el SUCHAI. Se implementó y evaluó un sistema de comunicación que pueda ser utilizado en satélites de diverso tamaño, desde femto a nanosatélites, basado en el módulo RFM22B. Se implementó un protocolo de comunicación, se realizaron múltiples pruebas en terreno y utilizando un globo sonda. Las pruebas en terreno permitieron demostrar que la mejor modulación y tasa de transferencia de datos disponibles considerando la cantidad de paquetes enviados exitosamente es FSK a 2 kb/s. Con radiosonda se probó esta configuración y se recibieron paquetes a 38,9 km con una antena omnidireccional de 3 dBi de ganancia y a 477 km con la antena Yagi de la estación terrena del proyecto SUCHAI que posee 18,9 dBiC de ganancia. Estos resultados demuestran que el RFM22B es una opción viable para nanosatélites en la órbita de la estación espacial internacional (416 km en apogeo) y para constelaciones de femtosatélites.

*Dedicado a Jesús, el Cristo, el Hijo del Dios viviente, a quién le debo mi existencia, mi salvación y mi vida. A Él sea la gloria por los siglos de los siglos, amén.
A mis padres Luis Vidal Lagos y Ximena Valladares Nuñez, a quiénes les debo quién soy.*

Agradecimientos

A todos mis amigos y familiares que me tuvieron paciencia y entendieron cuando debía ausentarme por causa de los estudios.

A mis amigos de plan común, con quienes compartí, estudié y me divertí, a pesar de las dificultades de la carrera, a Jaime Sanz, Daniel Aviv, Felipe Scaramelli, Franco Robbiano, Matías Yañes, Cristian Valdés, César Arriagada y Felipe Garrido. Sin su apoyo, probablemente me habría atrasado mucho en mis estudios.

A mis amigos de eléctrica, a Carolina Mayol, Sebastián Galvez, Felipe Toledo, Armando Cid y Gustavo Moreno. Por formar invaluable grupos de estudio, juntas de esparcimiento y buenas amistades. A todos los miembros del CEIE, fue una experiencia enriquecedora haber podido formar parte de ustedes y poder contribuir un granito de arena a la comunidad de eléctrica.

A mis amigos de CAS, por los proyectos que realizamos juntos y por conocer a través de ustedes todos los beneficios del Open Source, a Victor Díaz, Eileen Guevara, Nicolás Maturana, Javier Pincheira y Joakin Ugalde. Al grupo de amigos que nos hicimos más unidos en la práctica 2, cuya amistad aún perdura, a Franco Curotto, Daniela Mieres y Flor Candia.

Al profesor Marcos Díaz, quien me dió a conocer este proyecto el día en que quería empezar a trabajar en un laboratorio para aplicar los conocimientos aprendidos en clases. A los miembros del laboratorio SPEL, a Tomás Opazo, Carlos González, Alex Becerra, Samuel Gutierrez, Gerardo Peña, Cristóbal Garrido y Joaquín Díaz, por tener siempre la disponibilidad de ayudarme cuando lo necesitaba. A Daniel Valenzuela, quien trabajó codo a codo conmigo en el laboratorio y me ayudó a realizar muchas pruebas para mi memoria. Numerosas fueron las veces en que nos quedamos hasta que cerraran el edificio de eléctrica.

A todos mis hermanos en Cristo que me acompañaron durante la realización de las pruebas en el Parque O'Higgins, a Matías Abarca, Margarita Solimano, Natanael Rodriguez, Jorge Gongora y Mayra Sanchez, quienes a pesar del sol y tener que levantarse temprano, no dudaron en apoyarme.

Tabla de contenido

2. Antecedentes	1
2.1. Fundamentación y objetivos generales	1
2.1.1. Marco general	1
2.1.2. Objetivo general	2
2.1.3. Objetivos específicos	2
2.2. Hipótesis de trabajo y metodología	2
2.2.1. Supuestos	3
2.2.2. Infraestructura disponible	3
2.2.3. Hardware utilizado	3
2.2.4. Software utilizado	4
2.2.5. Plan de Trabajo	5
2.2.6. Elección del terreno de pruebas	5
2.3. Revisión Bibliográfica	6
2.3.1. Características de los satélites	6
2.3.2. Tipos de órbitas terrestres	7
2.3.3. El estándar CubeSat	8
2.3.4. CubeSats en Chile	11
2.3.5. Femtosatélites	12
2.3.6. Motivación	15
2.3.7. Consideraciones técnicas	16
2.3.8. Modulación y demodulación	19
2.3.9. Cálculo de radioenlace	22
3. Implementación	27
3.1. Desarrollo en terreno	27
3.1.1. Programación	29
3.1.2. Diseño de PCB	33
3.1.3. Reciclaje de baterías de litio	35
3.1.4. Montaje final	35
3.2. Desarrollo de la radiosonda	36
3.2.1. Pruebas preliminares	37
3.2.2. Diseño de PCB	39
3.2.3. Código para la radiosonda	39
3.2.4. Código para los receptores	41
3.2.5. Montaje de la radiosonda y los receptores	44

4. Evaluación y análisis de resultados	48
4.1. Pruebas en Parque O'Higgins	48
4.1.1. Resultados a 125 kb/s	50
4.1.2. Resultados a 38,4 kb/s	52
4.1.3. Resultados a 19,2 kb/s	53
4.1.4. Resultados a 2,4 kb/s	54
4.1.5. Resultados a 2 kb/s	55
4.1.6. Resultados generales	56
4.2. Pruebas a mayor distancia	58
4.3. Pruebas con la radiosonda	61
4.4. Cálculo de radioenlace final	64
4.4.1. Comunicación entre transeptores actualizada	67
4.4.2. Comunicación hacia la estación terrena actualizada	68
Conclusión	68
4.5. Posibles usos para el RFM22B	70
4.6. Trabajo futuro	70
Bibliografía	72
5. Anexo	75

Índice de Tablas

2.1.	Características generales del Arduino Due	4
2.2.	Consumo energético de los transceptores	17
2.3.	Tabla para la salida en MSK	21
2.4.	Potencia y sensibilidad de los transceptores	23
2.5.	Especificaciones de las antenas	24
2.6.	Potencias para el cálculo de enlace entre TRX RFM22B	25
2.7.	Potencias para el cálculo de enlace hacia la estación terrena	25
3.1.	Tabla de conexiones RFM22B - Arduino Due	28
3.2.	Consumo de corriente del GPS L80	37
3.3.	Conexiones entre el Arduino Due y el GPS L80	38
4.1.	Radios de zona de Fresnel	50
4.2.	Pruebas a 2.0 kb/s usando un atenuador de 30 dB	58
4.3.	Resultado efectivo considerando la atenuación	58
4.4.	Zonas de Fresnel en cerro La Pirámide (500 m)	60
4.5.	Distancia efectiva en cerro La Pirámide	61
4.6.	Potencias actualizadas para el cálculo de enlace entre TRX RFM22B	67
4.7.	Potencias actualizadas para el cálculo de enlace hacia la estación terrena	68

Índice de Ilustraciones

2.1. Distancias probadas en Parque O'Higgins	5
2.2. La ISS orbita la Tierra una vez cada 90 minutos.	7
2.3. Órbitas polares	8
2.4. 1U CubeSat Design Specification	9
2.5. Transceptor del SUCHAI	12
2.6. Diseño de un Sprite	13
2.7. Diseño conceptual del despliegue de Sprites	13
2.8. Femtosatélites de 1F y 3F	14
2.9. Diseños de los satélites PUCP-Sat-1 y Pocket-PUCP	15
2.10. Transceptor RFM22B	15
2.11. Modulación FSK	20
2.12. Espectros de FSK vs GFSK	21
2.13. Modulación MSK	22
3.1. Transceptor con placa adaptadora	27
3.2. Prototipado del proyecto	28
3.3. Diagrama de flujo general del TX	30
3.4. Diagrama de flujo de la función enviar	31
3.5. Impresión en pantalla de los aciertos	32
3.6. Diseño de PCB en KiCad	33
3.7. Placa de prueba	34
3.8. PCB fabricada para conectar el transceptor al Arduino Due	34
3.9. Baterías de litio	35
3.10. Transceptor más batería unidos al atril	36
3.11. Arduino más GPS L80 en protoboard	38
3.12. Diseño de PCB para incorporar el GPS	39
3.13. Componentes de la radiosonda	45
3.14. Armado de la radiosonda	45
3.15. Reforzamiento de la radiosonda	46
3.16. Antena yagi de la estación terrena	46
3.17. Setup de la estación terrena	47
3.18. Antena omnidireccional y antena Yagi	47
4.1. Zona de Fresnel (usada con permiso de la licencia <i>Creative Commons</i>)	49
4.2. Obtención de datos del transceptor al notebook	49
4.3. Datos recibidos correctamente a 125 kb/s	51

4.4. Tiempo utilizado para enviar 1000 paquetes a 125 kb/s	51
4.5. Datos recibidos correctamente a 38.4 kb/s	52
4.6. Tiempo utilizado para enviar 1000 paquetes a 38.4 kb/s	52
4.7. Datos recibidos correctamente a 19.2 kb/s	53
4.8. Tiempo utilizado para enviar 1000 paquetes a 19.2 kb/s	53
4.9. Datos recibidos correctamente a 2.4 kb/s	54
4.10. Tiempo utilizado para enviar 1000 paquetes a 2.4 kb/s	54
4.11. Datos recibidos correctamente a 2.0 kb/s	55
4.12. Tiempo utilizado para enviar 1000 paquetes a 2.0 kb/s	55
4.13. Datos recibidos correctamente	56
4.14. Tiempo utilizado para enviar 1000 paquetes	57
4.15. Atenuadores de potencia de 30 dB	58
4.16. Prueba en cerro La Pirámide	59
4.17. Línea de vista cerro La Pirámide	60
4.18. Trayectoria de la radiosonda en Google Earth. Los puntos azules representan datos obtenidos por la antena Yagi portátil con 30 dB de atenuación. Los verdes son los datos captados por la antena omnidireccional con 30 dB de atenuación. Morado la antena Yagi portátil sin atenuación. Rojo la estación terrena con la antena Yagi de 6 m y con 40 dB de atenuación. La distancia en tierra entre el inicio y la última posición captada fue de aproximadamente 11 km.	62
4.19. Paquetes de la radiosonda recibidos por la antena omnidireccional	63
4.20. Paquetes de la radiosonda recibidos por la estación terrena	64
4.21. Analizador de espectros	65
4.22. En azul el espectro del transceptor con modulación GFSK a 2 kb/s. En rojo el inicio y término del ancho de banda.	65
4.23. En azul el espectro del transceptor con modulación FSK a 2 kb/s. En rojo el inicio y término del ancho de banda.	66
4.24. En azul el espectro del transceptor con modulación OOK a 2,4 kb/s. En rojo el inicio y término del ancho de banda.	67

Capítulo 2

Antecedentes

2.1. Fundamentación y objetivos generales

2.1.1. Marco general

Existe un estándar que ha facilitado las investigaciones aeroespaciales de forma económica y accesible para empresas y establecimientos educacionales. Dicho estándar es llamado CubeSat y tiene restricciones en tamaño y peso, las cuales limitan la energía que puede generar y consumir. Un CubeSat puede tener un volumen de más de una unidad (cubo de 10 cm de lado), lo cual permite el transporte de múltiples satélites aún más pequeños llamados femtosatélites, los cuales pueden ser desplegados una vez que el CubeSat esté en la órbita deseada. Estos femtosatélites al ser de menor tamaño y de bajo costo, permiten a Universidades y centros de investigación formar una red de satélites abarcando un área mayor para medir y estudiar propiedades que de otro modo no se puede hacer.

Toda misión espacial requiere de un sistema de comunicación robusto ante fallos de transmisión, ya que usualmente las misiones que transportan instrumentos o experimentos generan datos que deben ser enviados a tierra con la mayor fidelidad para no perder o mal interpretar resultados.

Actualmente se está trabajando en el primer satélite tipo CubeSat en ser desarrollado en Chile, el cual será lanzado a fines del año 2016. Dados los costos que conlleva el desarrollo, certificación y lanzamiento de un nanosatélite, se desea buscar alternativas para reducir el impacto económico en las siguientes iteraciones de éste. Debido a esto, se desarrolla un sistema de comunicación, el cual es comparado con las opciones que ofrece el mercado. Luego se evalúa si cumple con los requisitos para ser utilizado en los femtosatélites del CubeSat de la Universidad de Chile en términos de tasa de transferencia de datos, consumo energético, tamaño y masa. En caso de obtenerse resultados satisfactorios, este sistema de comunicaciones será llevado experimentalmente en los femtosatélites del segundo Cu-

beSat hecho por la Universidad de Chile.

2.1.2. Objetivo general

Implementar y evaluar el desempeño de un sistema de comunicación que cumpla con los requisitos para ser utilizado en femtosatélites desarrollados para la siguiente misión de la Universidad de Chile. Los femtosatélites están pensados para ser transportados y lanzados por un CubeSat nodriza con el que deberían comunicarse en el espacio. Además, se estudiará la posibilidad de que dicho sistema de comunicación pueda lograr conexión directa con una estación terrena.

2.1.3. Objetivos específicos

A continuación se enumeran los objetivos específicos de este trabajo:

1. Implementar un protocolo de transmisión de datos en el transceptor RFM22B.
2. Diseñar y realizar pruebas en terreno con distintas combinaciones de modulación, tasa de transmisión de datos y distancias. Durante las pruebas se debe medir la cantidad de paquetes exitosamente enviados versus la cantidad de paquetes perdidos y el tiempo utilizado. Luego de encontrar la mejor configuración para el transceptor, se hace una prueba en un globo sonda para determinar cuál es el alcance máximo del RFM22B utilizando atenuadores de potencia para simular largas distancias entre emisor y receptor.
3. Evaluar y analizar los resultados obtenidos en las pruebas realizadas en terreno. Contrastar el rendimiento del transceptor con las necesidades comunicacionales de un femtosatélite y compararlo con otros sistemas de comunicación existentes.

2.2. Hipótesis de trabajo y metodología

Asumiendo que la electrónica del transceptor utilizado puede soportar las condiciones ambientales extremas del espacio, podría ser utilizado como base de un sistema de comunicación entre satélites (en el espacio) y con tierra. En el caso de tierra se requeriría un cambio de antena en el segmento terrestre, por lo que se asume que antenas comerciales de bajo costo son suficientes para lograr un enlace espacio-tierra desde una órbita LEO.

2.2.1. Supuestos

Se supone que el transceptor escogido logrará desempeñarse adecuadamente en el ambiente hostil del espacio. También se espera que pueda transmitir información desde la estratósfera hasta la estación terrena y que no se verá mayormente afectado por el campo magnético del planeta Tierra. Además, se supone que distintas modulaciones se ven afectadas de igual manera al estar cumpliendo el mismo porcentaje de la zona de Fresnel.

2.2.2. Infraestructura disponible

Toda la electrónica e instrumentación necesaria, tales como microcontroladores, osciloscopios, estación de soldado, transceptores, estación terrena, computadores, entre otros, está disponible gracias al Laboratorio de Exploración Espacial y Planetaria (LEEP) del Departamento de Ingeniería Eléctrica de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile [28].

2.2.3. Hardware utilizado

Durante el trabajo de memoria se utiliza el transceptor RFM22B, tarjetas Arduino Due, antenas monopolo de 3 dBi y la estación terrena ubicada en el Departamento de Ingeniería Eléctrica de la Universidad de Chile, la cual tiene una ganancia de 18,9 dBiC.

A continuación se describen las características principales del transceptor y el microcontrolador utilizados:

Transceptor

El sistema de comunicación que se va a implementar está basado en el transceptor RFM22B, el cual es un dispositivo comercial de bajo costo utilizado en aplicaciones de aeromodelismo. Posee un amplificador de potencia (PA) que le permite aumentar la potencia de la señal transmitida hasta a 20 dBm, opera en la banda de 433 MHz, funciona con un nivel de lógica de 3,3 V e incluye un amplificador de bajo ruido (LNA), con el cual logra una sensibilidad de -121 dBm. Se puede programar la tasa de transferencia de datos hasta un máximo de 125 kb/s, permite cambiar la potencia de salida en pasos de 3 dBm y elegir entre 3 tipos de modulación, FSK, GFSK y OOK.

Microcontrolador

La tarjeta electrónica utilizada para controlar el transceptor es el Arduino DUE, el cual funciona con una lógica de 3,3 V igual que el transceptor RFM22B, esto permite que se puedan conectar directamente sin necesidad de usar un *logic shifter*. Los datos son transmitidos a través de su puerto SPI. Las características generales de ésta tarjeta se pueden ver en la tabla 2.1.

Arduino Due	
Microcontrolador	T91SAM3X8E
Voltaje de operación	3,3 V
Voltaje de entrada recomendado	7 - 12 V
Voltajes límites de entrada	6 - 16 V
Pines I/O digitales	54 (12 PWM)
Pines de entrada analógicos	12
Pines de salida analógicos	2 (DAC)
Corriente DC total de salida en todas la líneas IO	130 mA
Corriente DC para el pin 3,3 V	800 mA
Corriente DC para el pin 5 V	800 mA
Memoria flash	512 kB
SRAM	96 kB (dos bancos: 64 kB y 32 kB)
Velocidad de reloj	84 MHz
Largo	101,52 mm
Ancho	53,3 mm
Peso	36 g

Tabla 2.1: Características generales del Arduino Due

2.2.4. Software utilizado

La programación de los microcontroladores se efectúa principalmente en el IDE de Arduino. Para el diseño de las placas PCB se utilizó KiCad por su apertura y versatilidad respecto a los componentes utilizados. Para graficar los resultados se usa Octave y para almacenar los datos obtenidos por el GPS se usa CuteCom. Todo lo anterior se efectúa en un computador con sistema operativo Ubuntu, de modo que el proyecto será Open Source en todas sus etapas desde el comienzo y no será necesario utilizar software privativo para poder replicarlo en el futuro. Todo el diseño de las placas electrónicas y los códigos utilizados para las pruebas están disponibles en GitHub, por lo que se pueden ver las modificaciones que ha sufrido el proyecto en el tiempo. Para ilustrar la trayectoria de la radiosonda, se utilizó el programa Google Earth.

2.2.5. Plan de Trabajo

Los hitos que marcan el proyecto se encuentran bien definidos, primero se debe lograr implementar el protocolo de comunicación en la tarjeta Arduino Due, para luego poder conectarla al transceptor y hacer pruebas de transmisión de datos. En un comienzo se deben transmitir datos dentro del laboratorio utilizando las diversas combinaciones de modulación, tasa de transmisión de datos y de potencia disponibles en la librería del transceptor RFM22B para Arduino. Posteriormente se debe escoger un lugar con línea de vista y sin obstáculos para realizar pruebas a distancia sin atenuadores de potencia. Durante estas pruebas también se deben ir variando las configuraciones, además de probar distintas distancias entre emisor y receptor. Este proceso es el más largo, pues requiere la realización de una gran cantidad de pruebas probando muchas combinaciones de parámetros, hasta encontrar los que den mejores resultados. Finalmente, se debe hacer una prueba de transmisión con la mejor configuración desde un globo sonda, utilizando atenuadores de potencia en los receptores para simular una comunicación con un rango similar a la órbita de un satélite tipo CubeSat.

2.2.6. Elección del terreno de pruebas

El lugar donde se efectúen las pruebas debe ser grande y estar libre de obstáculos para tener línea de vista entre el emisor y el receptor. Teniendo estos requisitos en consideración, se elige inicialmente el Parque O'Higgins, ya que su cercanía a la facultad y su extensa área pavimentada lo constituyen el lugar ideal para realizar las pruebas de transmisión de datos.

En una primera instancia se decide efectuar pruebas a 75, 150 y 300 m de distancia, tal y como se puede ver en la figura 2.1 (imagen obtenida con Google Maps).

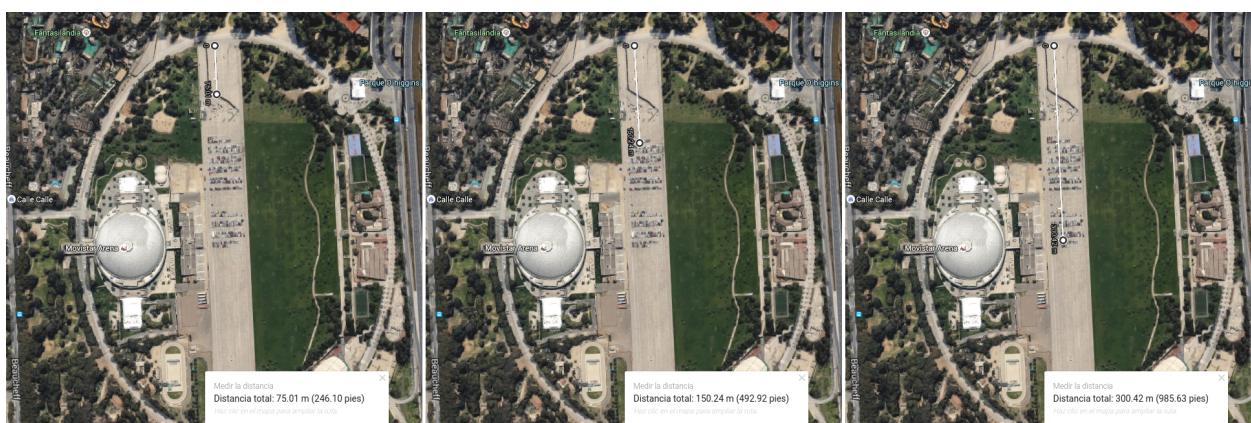


Figura 2.1: Distancias probadas en Parque O'Higgins

Luego de estas pruebas y de analizar los resultados, se deben usar atenuadores de potencia para probar las configuraciones que arrojaron los mejores resultados. Para esto

se requiere de un sitio en una altura como un cerro, tal que no existan obstrucciones de la zona de Fresnel. La configuración que obtenga el mejor desempeño se debe utilizar en un globo sonda para poder determinar cuál es la distancia máxima de transmisión de este dispositivo.

2.3. Revisión Bibliográfica

El objetivo del siguiente capítulo es contextualizar al lector con los avances tecnológicos en materia de femto, pico y nano-satélites, en particular los de tipo CubeSat y en sistemas de comunicación para este tipo de máquinas.

2.3.1. Características de los satélites

Según la Nasa [29], un satélite es una luna, planeta o máquina la cual orbita un planeta o una estrella. Por ejemplo, la Tierra es un satélite porque orbita el sol. Asimismo, la luna es un satélite porque orbita la Tierra. Aunque también en la literatura, la palabra satélite hace referencia a una máquina que es lanzada al espacio y orbita al rededor de la Tierra u otro cuerpo en el espacio.

Existen dos grandes categorías para clasificar a los satélites; por un lado están los satélites naturales como la Tierra o la luna, y por otro lado están los satélites artificiales, los cuales fueron fabricados por el hombre. Entre los satélites artificiales destacan los satélites de observación terrestre, los cuales estudian el medio ambiente, la meteorología y la cartografía, los satélites de comunicaciones encargados de enviar señales telefónicas y de televisión a distintas partes del mundo, satélites de navegación que guían barcos y aviones como el GPS o el GLONASS y satélites científicos de todo tipo, entre otros.

A pesar de la gran variedad de aplicaciones que se le puede dar a un satélite artificial, existen características que trascienden todos estos dispositivos las cuales se muestran a continuación:

- Sistema de energía: Todo satélite artificial requiere de una fuente de poder que alimente su electrónica y equipos, además de baterías para almacenar la energía. Generalmente se utilizan celdas solares para generar energía.
- Sistema de control y monitoreo: Se requiere de un procesador o microcontrolador para ejecutar órdenes hacia el sistema de comunicación, a los actuadores y experimentos a bordo, además de monitorear el resto de los sistemas y el estado de carga de las baterías.
- Sistema de comunicación: Finalmente, la gran mayoría de los satélites poseen un sistema de comunicación que les permite enviar datos sobre todos los sistemas a bordo hacia la Tierra, lo cual se conoce como telemetría, además de poder enviar

información del estado del satélite y de su salud a pedido de la estación terrena. Para ello, es necesario tener un transmisor/receptor de ondas de radio más una antena.

Este trabajo de título se concentra en el área de comunicación satelital. Uno de los factores que influyen en gran manera el dimensionamiento y el tipo de sistema de comunicación de un satélite es su órbita alrededor de la tierra.

2.3.2. Tipos de órbitas terrestres

Básicamente, hay tres tipos de órbitas terrestres: órbita terrestre alta (HEO), órbita terrestre media (MEO) y órbita terrestre baja (LEO) [27]. Muchos de los satélites meteorológicos y de comunicaciones tienen una órbita alta, lejos de la superficie. La órbita media es más usada por satélites de navegación y de ciencia geodésica y ambiente espacial. La mayor parte de los satélites tienen una órbita terrestre baja, siendo la estación espacial internacional (ISS) uno de los más conocidos.



Figura 2.2: La ISS orbita la Tierra una vez cada 90 minutos.

Existe una relación directa entre la altura de la órbita del satélite y qué tan rápido orbita alrededor de la Tierra. En órbitas más bajas, los satélites deben ir más rápido ya que la gravedad es más fuerte, mientras que en órbitas mayores, si los satélites viajan demasiado rápido, corren el riesgo de salir fuera de la órbita hacia el exterior. Los satélites que están en HEO (a 35.780 km aproximadamente) pueden tener una órbita geostacionaria, en donde el satélite rota a una velocidad tal que le permite estar siempre sobre el mismo punto de la Tierra.

Los nanosatélites que están en la órbita terrestre baja dan una vuelta alrededor de la Tierra en aproximadamente 90 minutos [11] a una velocidad de 7,5 km/s [1]. Debido a esto, se tiene una ventana de tiempo muy acotada para obtener información de ellos desde una estación terrena.

El nanosatélite a evaluar es de tipo CubeSat y tendrá una órbita de tipo polar, la cual consiste en una trayectoria con una inclinación cercana a los 90° del ecuador y que le permite pasar por sobre ambos polos. Un satélite en una órbita polar pasa sobre el ecuador en diferentes longitudes en cada una de sus órbitas como se puede ver en la figura 2.3. Esto le permitirá al CubeSat viajar cada cierto tiempo a lo largo de Chile, además de pasar sobre el resto del planeta.

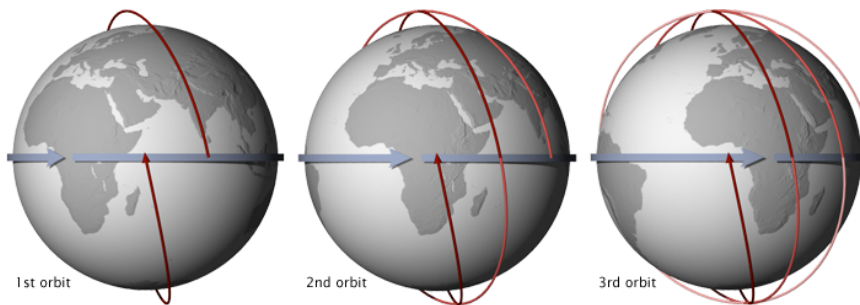


Figura 2.3: Órbitas polares

2.3.3. El estándar CubeSat

El CubeSat es un diseño de referencia propuesto en 1999 por Jordi Puig-Suari de la *California Polytechnic State University*, San Luis Obispo y Bob Twiggs de la *Stanford University's Space Systems Development Laboratory*, con el propósito de crear un estándar que permita a estudiantes universitarios diseñar, construir, probar y operar en el espacio una nave con capacidades similares a la del primer satélite, *Sputnik*¹. Este diseño permite abaratar costos, ya que el sistema para acoplar este tipo de satélites al cohete que los va a enviar, es de un tamaño específico, reduciendo de esta manera los tiempos de coordinación entre las lanzaderas y los fabricantes de satélites.

Actualmente, el Proyecto CubeSat es una colaboración internacional de más de 100 universidades, colegios y firmas privadas desarrollando nanosatélites conteniendo cargas científicas, privadas y gubernamentales.

A continuación se exponen los requisitos que debe cumplir un nanosatélite para poder ser lanzado, además de las características que tiene el sistema de despliegue de este tipo de satélites.

¹Fue el primer satélite artificial hecho en el mundo y se lanzó el 4 de Octubre de 1957. Tenía 58 cm de diámetro, pesaba 83,6 kg y tardó 98 minutos en orbitar la Tierra en su plano elíptico [9].

Requerimientos dimensionales y de masa

Los CubeSats son picosatélites con forma de cubo de un largo nominal de 100 mm por lado para el formato de una unidad (1U), no obstante, los tamaños 1.5 U, 2 U, 3U y 6U también están permitidos. El sistema al cual se acoplan los CubeSats se llama *Poly Picosatellite Orbital Deployer* (P-POD [36]) y sirve para desplegar el CubeSat una vez que se esté en la altura de la órbita deseada. De no cumplirse el diseño y especificación CubeSat (ver figura 2.4), surgirán incompatibilidades con el sistema de despliegue establecido.

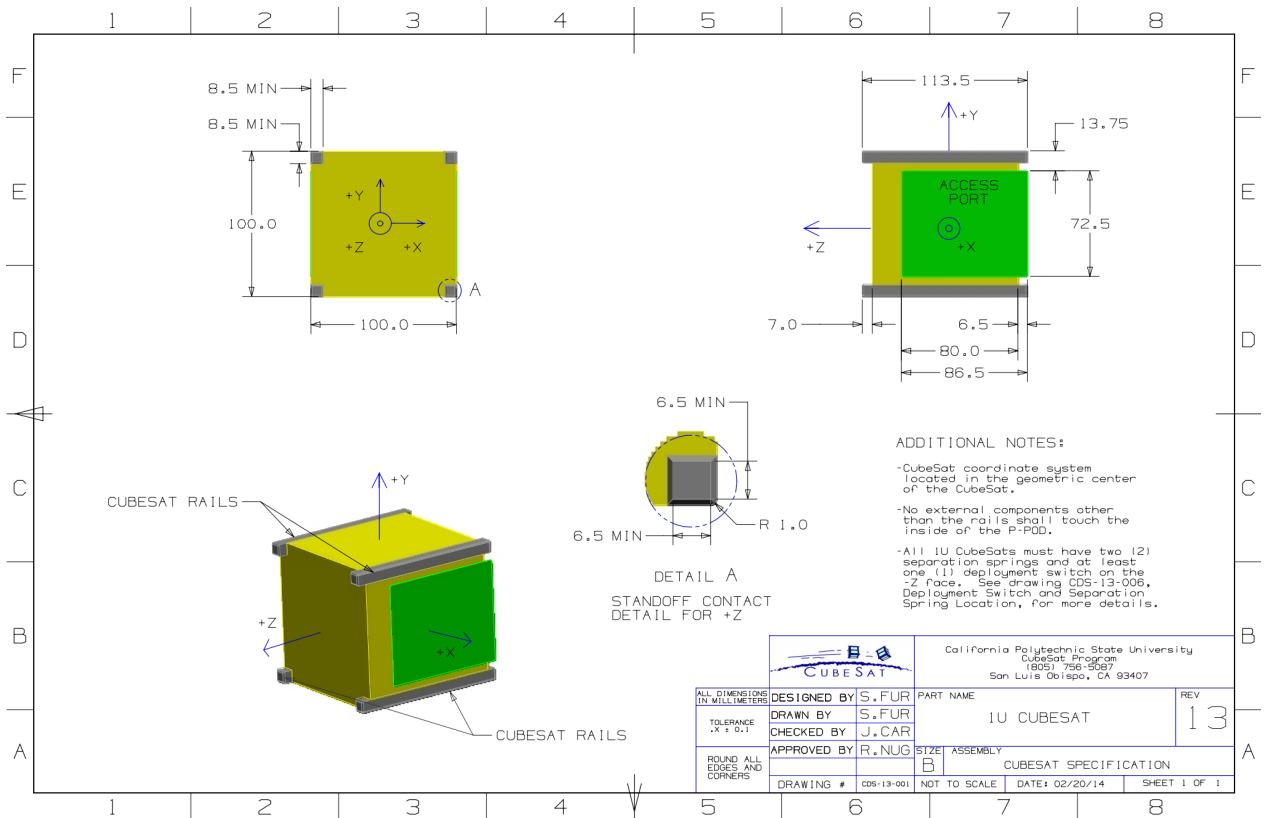


Figura 2.4: 1U CubeSat Design Specification

Los requisitos generales de todos los CubeSats son:

- Aquellos CubeSats que presenten alguna desviación de la especificación de diseño Cubesat (CDS) deben enviar una solicitud de aprobación de exención de desviación (DAR) y adherir al proceso de exención.
- Todas las piezas deben permanecer unidas a los CubeSats durante el lanzamiento, ejecución y operación. No se deben generar nuevos desechos espaciales.
- La pirotecnia no será permitida.
- Todos los sistemas de propulsión deben ser diseñados, integrados y probados de acuerdo con AFSPCMAN 91-710 Volumen 3.
- Los sistemas de propulsión deben tener al menos 3 inhibiciones para la activación.

- La cantidad total de energía almacenada no debe exceder los 100 Watt-Hora.
 - Nota: Capacidades mayores pueden ser permitidas, pero podrían limitar la cantidad de oportunidades de lanzamiento.
- Los materiales peligrosos del CubeSat deben cumplir con AFSPCMAN 91-710, Volumen 3.
- Los materiales del CubeSat deben satisfacer un criterio de baja desgasificación para prevenir la contaminación de otras naves espaciales durante la integración, pruebas y lanzamiento. Una lista de materiales de baja desgasificación se puede encontrar en una página de la NASA [22].
 - Los materiales de los CubeSats deben tener un total de pérdida de masa (TML) ≤ 1.0 %.
 - Los materiales del CubeSat deben tener una colección de material volátil condensable (CVCM) ≤ 0.1 %.
- La última revisión de la especificación de diseño CubeSat [25] será la versión oficial a la cual todos los desarrolladores deberán adherirse.
- Nota: Algunos vehículos de lanzamiento tienen requisitos de intensidad de campo magnético. Además, fuertes imanes pueden interferir con la separación entre el CubeSat y la nave espacial en el mismo P-POD. Como regla general, se recomienda limitar el campo magnético fuera del envoltorio estático del CubeSat a 0.5 Gauss sobre el campo magnético terrestre.
- El CubeSat debe ser diseñado para dar cabida a la ventilación ascendente por volumen/área evitables < 5080 cm.

Requerimientos operacionales

Los CubeSats deben cumplir ciertos requerimientos [34] relativos a la integración y al funcionamiento para cumplir obligaciones legales y asegurar la integridad de otros CubeSats.

- Los operadores obtendrán y proporcionarán los documentos de las licencias apropiadas para el uso de radio frecuencias.
 - Para el uso de frecuencias amateur, se requiere la evidencia de coordinación de frecuencia proveniente de la Unión Internacional de Radio Amateur (IARU) [32].
- Los CubeSats cumplirán con los acuerdos y las restricciones de la licencia de radio de sus países.
- El diseño de la misión CubeSat y el *hardware* deben estar de acuerdo con NPR 8.715,6 para limitar desechos orbitales.
 - Cualquier componente CubeSat deberá volver a entrar con energía inferior a 15 Joules .
 - Los desarrolladores obtendrán y proporcionarán los documentos de la aprobación para el plan de mitigación de desechos orbitales de la FCC (o NOAA si la imagen está presente).

- Nota: Los radioaficionados deben ver la regulación de la FCC [23].
- Nota: El análisis puede llevarse a cabo para satisfacer lo anterior con la NASA *Debris Assessment Software*, disponible en internet [21].
- Todos los desplegables tales como booms, antenas y paneles solares deben esperar mínimo 30 minutos para desplegarse luego que los switches de despliegue del CubeSat son activados desde el sistema de eyección P-POD.
- Ningún CubeSat deberá generar o transmitir señal alguna desde el tiempo de integración dentro del P-POD hasta 45 minutos después del despliegue en órbita desde el P-POD. No obstante, el CubeSat puede ser encendido luego del despliegue desde el P-POD.
- Las entidades privadas (fuera de EEUU) bajo la jurisdicción o control de los Estados Unidos quienes propongan operar un sistema de teledetección espacial (satélite), pueden necesitar una licencia según la ley de EEUU [24].
- Cal Poly llevará a cabo una comprobación de mínimo un ajuste, en donde el *hardware* del desarrollador será inspeccionado e integrado al P-POD o *TestPOD*. Una última comprobación de ajuste se hará previo al lanzamiento. La lista de verificación de aceptación del CubeSat (*CubeSat Acceptance Checklist*) será usada para verificar la conformidad de la especificación.

2.3.4. CubeSats en Chile

Este trabajo de título se enmarca en un proyecto de mayor magnitud, el cual es la segunda iteración de la misión SUCHAI (*Satelite of the University of Chile for Airspace Investigation*) llamada de esa forma ya que se asemeja a la palabra "esperanza" en Mapudungún. El profesor Marcos Díaz es el director del proyecto SUCHAI desde su inicio en el año 2010. Él ha estado involucrado en cada una de las etapas del proyecto y ha adquirido un gran conocimiento acerca de las implicancias de enviar un nanosatélite al espacio, así como los permisos, requisitos y certificados necesarios para esta misión.

Debido a esto, es necesario tener presente lo que se ha hecho hasta ahora, por lo tanto se tienen en consideración los trabajos de memoria efectuados por Tomás Opazo [31] y Carlos González [4], quienes trabajaron en el SUCHAI, para tenerlos como referencia.

El sistema de comunicación utilizado por SUCHAI es el U482C (Figura 2.5) de la empresa *GomSpace*, la cual es reconocida en materia de nanosatélites, por lo que se utilizará este transceptor a modo de referencia para compararlo con el desarrollado en este trabajo de título.

Este transceptor es *half duplex*, es decir, puede enviar información bidireccionalmente pero no de forma simultánea. Opera entre los 435 y 438 MHz con modulación MSK. Su potencia de salida es de 30 dBm y su sensibilidad de recepción es de -126 dBm.

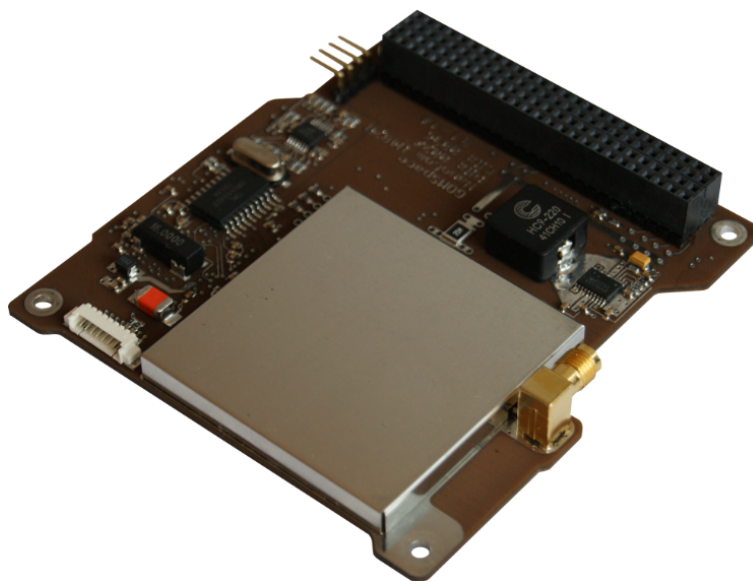


Figura 2.5: Transceptor del SUCHAI

2.3.5. Femtosatélites

Una de las formas para clasificar a los satélites es basandose en su masa. Los CubeSats o nanosatélites están en el rango entre 1 y 10 kg, los picosatélites están entre 0,1 y 1 kg y finalmente los femtosatélites son todos aquellos satélites con una masa inferior a 0,1 kg.

Existen femtosatélites de diversas formas y tamaños, siendo su masa lo único que tienen en común. Algunos requieren de una nave madre para poder comunicarse con las estaciones terrenas. A continuación se muestran ejemplos de femtosatélites ya existentes y en desarrollo:

Sprites

A principios de octubre de 2011 se inició un proyecto CubeSat de 3U llamado KickSat [17], el cual se financió completamente a través del portal *Kickstarter* mediante donaciones. Dentro del KickSat se transportaron más de 100 femtosatélites llamados Sprites, los cuales consisten en un microcontrolador, una radio, un giroscopio, un magnetómetro y celdas solares. El giroscopio se utiliza para determinar la inclinación del dispositivo, el magnetómetro para hacer mediciones del campo magnético terrestre y una combinación de ambos para hacer una *attitude determination*, es decir, obtener la orientación del Sprite con respecto a la Tierra. Cada uno de ellos pesa solamente 5 gramos.

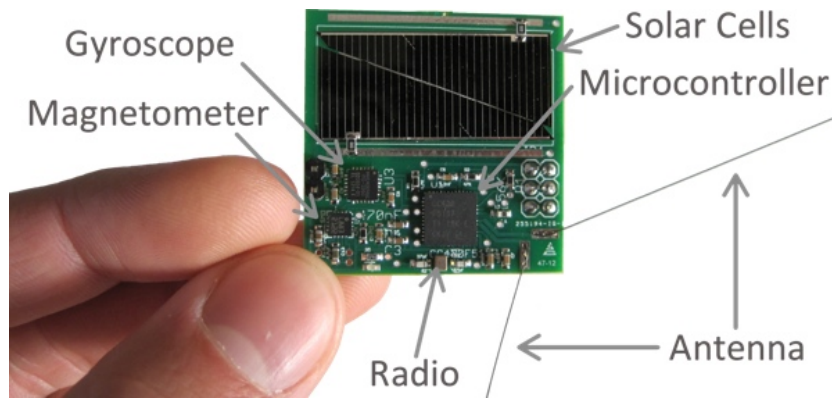


Figura 2.6: Diseño de un Sprite

El sistema de comunicación de los Sprites [18] está basado en el transceptor CC1101 [12]. Cada Sprite transmite a 437,240 MHz, con 10 mW de potencia de transmisión, modulación MSK, a 125 b/s y usando un protocolo de comunicación personalizado.

Dos de las unidades del KickSat se utilizaron para el sistema de despliegue de los Sprites, y la unidad restante para la comunicación, sistema de energía y sistema de control del KickSat. El satélite fue puesto en órbita exitosamente el 18 de Abril de 2014, pero debido a una petición de la Estación Espacial Internacional, el KickSat debió desplegar los Sprites 16 días después del lanzamiento. Lamentablemente, el *watchdog* del KickSat se reseteó el 3 de mayo probablemente debido a la radiación, lo que provocó un retraso del despliegue de los femtosatélites. El 14 de mayo el KickSat entró a la atmósfera con los Sprites todavía dentro de éste.

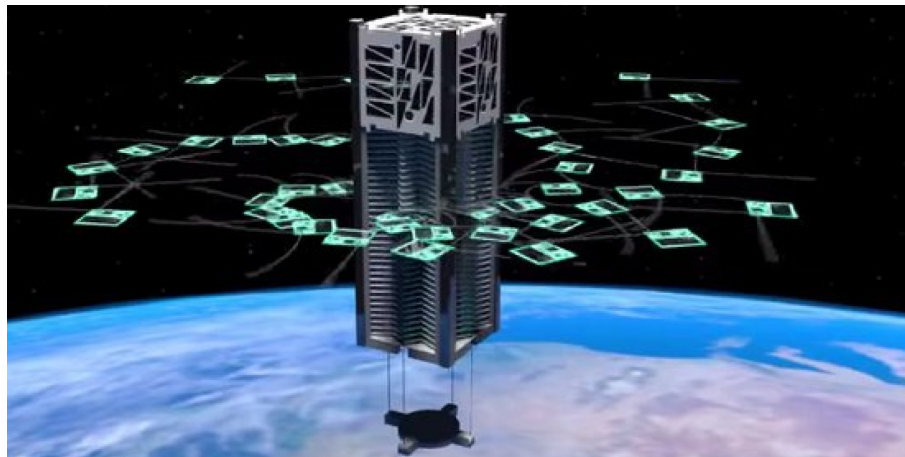


Figura 2.7: Diseño conceptual del despliegue de Sprites

A pesar de lo anterior, los datos enviados por el KickSat fueron captados alrededor de todo el mundo utilizando un equipo con un costo cercano a los 200 dólares. Actualmente se está trabajando en el KickSat-2.

SunCube

En la *Arizona State University*, estudiantes y profesores crearon su propia especificación de diseño para femtosatélites llamados SunCube [33]. Estos satélites tienen una forma de cubo como los CubeSat y existen en dos formatos diferentes: Un SunCube de 1F con un volumen de 3 cm x 3 cm x 3 cm y 35 g de masa y uno de 3F con 9 cm x 3 cm x 3 cm de dimensión y con 100 g de masa.

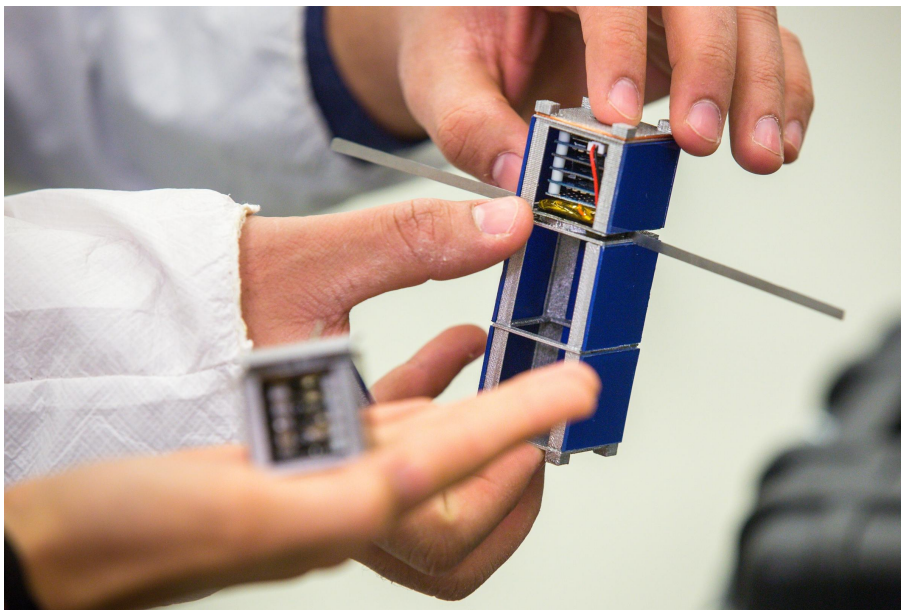


Figura 2.8: Femtosatélites de 1F y 3F

Así como los CubeSat tienen un dispositivo estándar para ser puestos en órbita llamado P-POD, esta especificación crea un nuevo estándar para desplegar femtosatélites llamado F-POD, el cual está diseñado para ser transportado dentro de un CubeSat y por lo tanto, es compatible con la especificación de diseño CubeSat. El equipo espera lanzar un prototipo al espacio dentro del próximo año.

Pocket-PUCP

El Pocket-PUCP es un femtosatélite peruano lanzado junto al CubeSat PUCP-Sat-1 el 21 de noviembre de 2013. Ambos satélites fueron construidos por el Instituto de Radioastronomía de la Pontificia Universidad Católica del Perú. Este femtosatélite fue desplegado exitosamente 15 días después del lanzamiento.

El Pocket-PUCP [5] tiene una masa de 97 g, su volumen es de 8,35 cm x 4,95 cm x 1,55 cm, transmite a 437,20 MHz con una modulación OOK y tiene una potencia de transmisión de 10 mW.

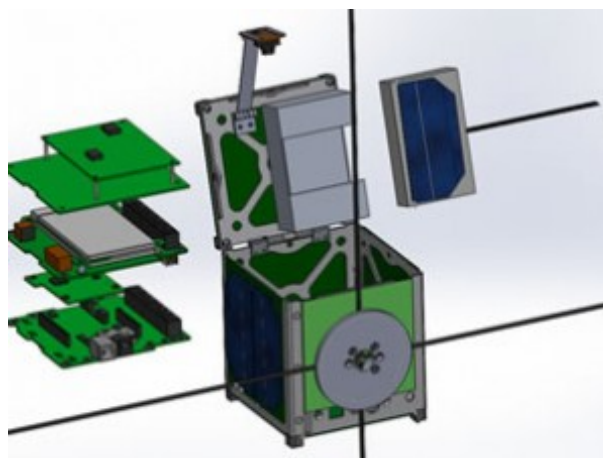


Figura 2.9: Diseños de los satélites PUCP-Sat-1 y Pocket-PUCP

2.3.6. Motivación

El próximo CubeSat será de tres unidades (3U), lo cual permitirá tener más espacio para incluir *payloads* como por ejemplo, sensores, microcontroladores y femtosatélites. Estos últimos serán desplegados en el espacio y deberán comunicarse entre sí y con el CubeSat mediante un transceptor, el cual debe ser pequeño, de bajo costo y de bajo consumo energético.

El transceptor que se pretende utilizar es el RFM22B (figura 2.10) el cual fue desarrollado por la empresa *HopeRF* la cual se especializa en transceptores y en sensores de varios tipos. Este transceptor es de muy bajo costo, funciona en la banda de 433 MHz, es *half duplex* y viene con diversas opciones programables como la modulación, la tasa de transmisión de datos y la potencia de salida. Actualmente es muy utilizado en aeromodelismo a largas distancias debido a su baja sensibilidad de recepción (-121 dBm) y su gran potencia de salida ($+20$ dBm) lo cual lo convierte en un buen candidato para una misión CubeSat.

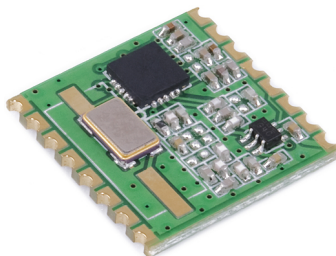


Figura 2.10: Transceptor RFM22B

Dentro de las pruebas, se evaluará si este transceptor se puede comunicar a una distancia similar a la que hay entre la órbita y la estación terrena. Si se logra demostrar que este dispositivo se puede utilizar en misiones espaciales, se hará un gran aporte a la co-

unidad científica y universitaria, ya que se disminuirá la barrera de entrada en términos económicos demostrando que un hardware del orden de los miles de pesos es competitivo con uno del orden de los millones de pesos.

2.3.7. Consideraciones técnicas

Debido a la naturaleza del proyecto, hay varios factores que se deben tomar en cuenta a la hora de implementar un transceptor que va a estar en femtosatélites y en un CubeSat, por lo que se hará un contraste entre las consideraciones técnicas del transceptor RFM22B [7] y las restricciones para estos satélites.

Tamaño

Sin considerar el alto del nanosatélite, cualquier circuito integrado que vaya en éste debe tener un área menor a $10 \times 10 \text{ cm}^2$. El transceptor RFM22B tiene una superficie de $16 \times 16 \text{ mm}^2$, por lo que a pesar de no ser muy grande se debe tener en consideración esta restricción al momento de diseñar una placa que contenga el transceptor con su microcontrolador más todos los componentes asociados. En el caso de los femtosatélites, no existe un tamaño definido, pero se espera reducir al máximo sus medidas para poder transportar la mayor cantidad posible dentro del próximo CubeSat.

Peso

Según la definición de un femtosatélite, éste debe tener una masa inferior a 100 g lo cual restringe en gran medida las características y cantidad de sus componentes. La masa del transceptor RFM22B es de solo 1,27 g, pero a esto se le debe agregar el peso del microcontrolador en donde se implementará el protocolo de transmisión de datos, condensadores, cristal, sistema de energía y sensores. A modo de referencia, el microcontrolador del Arduino Due pesa solamente 14 g [2].

Un CubeSat de tres unidades (3U) no puede pesar mas de 4,00 kg. Como referencia, el transceptor que va a funcionar como el sistema de transmisión de datos primaria de SUCHAI tiene una masa de 75 g.

Temperatura

Los kit de CubeSat que ofrece el mercado soportan un rango de temperaturas que va entre -40 C° y $+85 \text{ C}^\circ$ [13] que es exactamente el mismo rango que soporta el RFM22B. Los femtosatélites tienen las mismas restricciones que los CubeSat en este ámbito, ya que

están sujetos a las mismas condiciones de temperaturas extremas fuera de la atmósfera terrestre.

Energía

El consumo energético es un aspecto muy importante que debe ser considerado a la hora de diseñar un satélite. A modo de comparación se muestran los gastos energéticos en modo de recepción y de transmisión de datos, tanto para el transceptor U482C como para el RFM22B y el CC1101 (transceptor de los Sprites):

transceptor	RX (mA)	TX (mA)
U482C	50,0	300 - 1100
RFM22B	18,5	30 - 85
CC1101	16,0	13,1 - 29,2

Tabla 2.2: Consumo energético de los transceptores

Como se puede ver en la tabla 2.2, el gasto energético del RFM22B es considerablemente menor al del U482C y a su vez, el consumo del CC1101 es menor al del RFM22B sobre todo en modo de transmisión de datos. Estos valores se deben principalmente a que la potencia de transmisión máxima del U482C es diez veces mayor a la del RFM22B, y la de este es diez veces mayor a la del CC1101. Estos datos se mostrarán más adelante en la memoria.

Protocolo de comunicación

Un protocolo de comunicación es un sistema de reglas que permite que dos o más entidades de un sistema de comunicación transmitan información a través de cualquier tipo de variación de una cantidad física. Éstas son las reglas o estándar que define la sintaxis, semántica y sincronización de la comunicación y posibles métodos de recuperación de errores. Los protocolos pueden ser implementados por hardware, software o una combinación de ambos.

Inicialmente se escogió el protocolo de comunicación llamado AX.25 [35] debido a que es ampliamente usado en aplicaciones de radio amateur, es robusto ante fallos, calcula la verificación por redundancia cíclica (CRC) sobre el mensaje, confirma implícitamente la correcta recepción de mensajes utilizando contadores internos en el emisor y el receptor, entre otros factores. Luego de no encontrar una implementación de este protocolo para el hardware utilizado, se opta por programarlo completamente en C++ en el entorno de programación de Arduino.

La programación de AX.25 fue más compleja de lo estimado inicialmente, por lo que después de unos meses, se opta por utilizar una librería existente para el transceptor

RFM22B llamada *Radiohead*, la cual utiliza otro protocolo de comunicación diseñado específicamente para productos *HopeRF*. Este protocolo además de tener las mismas virtudes del AX.25, permite que el receptor confirme explícitamente los mensajes recibidos correctamente mediante un mensaje de confirmación con la identificación del mensaje recibido y tiene incorporada la opción de retransmitir un mensaje una cantidad determinada de veces, hasta un tiempo definido o hasta obtener una confirmación por parte del receptor, por estas razones se considera que este protocolo es mejor que el AX.25 y se escoge para el trabajo de título.

El sistema de comunicaciones desarrollado por *GomSpace* tiene un protocolo de comunicación privativo, el cual no puede ser modificado ni adaptado a las necesidades de cada organización o empresa, mientras que el RFM22B tiene un protocolo y una librería [19] abierta y disponible en internet para que cualquiera pueda alterarla y utilizarla como estime conveniente. Dicha librería está escrita en C++ y es compatible con Arduino, lo cual facilita su uso.

Cada paquete de datos del protocolo del transceptor RFM22B consta de la siguiente estructura:

- **Preámbulo:** Es de 4 octetos de largo y sirve para indicarle al receptor que viene un nuevo mensaje.
- **Sincronización:** Contiene dos octetos (0x2D, 0xD4). Si el receptor no detecta la sincronización, se reinicia la búsqueda del preámbulo, de lo contrario, se da comienzo al *Header*.
- **Header:** Es un marco de 4 octetos, donde se especifica hacia quién va dirigido el mensaje, quién es el emisor, la identificación del mensaje (ID) y el *flag* que indica el comienzo a la información sobre los datos enviados.
- **Largo:** Especifica la cantidad de octetos para los datos. Es de un octeto de largo.
- **Datos:** Aquí van los datos que se desea enviar, cada mensaje contiene de 0 a 255 bytes de datos.
- **Verificación por redundancia cíclica (CRC):** Es una forma de verificar si los datos enviados sufrieron alguna alteración durante la transmisión. Tiene un tamaño de 2 octetos y es computado sobre el *Header*, el Largo y sobre los Datos.

El cálculo del CRC consiste en dividir los datos por un polinomio haciendo una serie de operaciones XOR hasta que el cociente sea cero, luego el resto es el CRC, a menos que se decida hacer un postprocesamiento. A continuación se muestra un ejemplo en donde se desea calcular un CRC de 3 bits con el dato binario 1011010111 y el polinomio $x^3 + x + 1 = 1011$:


```

1011010111 000 ← Se agregan 3 ceros binarios pues se requiere un CRC de 3 bits.
1011           ← Polinomio divisor.
-----
0000010111 000 ← Resultado de la división, solo son afectados los primeros 4 bits.
   1011         ← El polinomio se mueve los espacios necesarios hasta el siguiente 1.
0000000001 000
      1 011
-----
0000000000 011 ← Termina el algoritmo pues el cociente es igual a cero.

```

Luego, el CRC es el resto, el cual resulta ser 011. Para verificar la validez de un mensaje, se debe realizar el mismo cálculo, pero incorporando el CRC en vez de los ceros agregados, por lo que el resto debiera ser 0:

```

1011010111 011
1011
-----
0000010111 011
   1011
0000000001 011
      1 011
-----
0000000000 000

```

2.3.8. Modulación y demodulación

La modulación corresponde al proceso que permite transmitir una señal con información por medio de una señal portadora, llevándola de banda base a una frecuencia mayor mediante el uso de un oscilador. Para modular se utilizan diferentes técnicas que permiten alterar propiedades de la portadora, tales como su amplitud, frecuencia o fase. Debido a esto, existen varias formas de transmitir inalámbricamente señales digitales, como por ejemplo FSK, GFSK, MSK y OOK.

La demodulación es el proceso inverso a la modulación, en donde se extrae la señal que contiene información desde la onda portadora. Dado que se está trabajando con comunicaciones digitales, el resultado de la demodulación son datos binarios.

Frequency Shift Keying

Es un tipo de modulación para transmisiones digitales, el cual utiliza dos frecuencias distintas para representar los dos símbolos que en este caso son el 0 y el 1. Debido a que en esta modulación cada símbolo corresponde a un bit, la tasa de bits por segundo es igual a la de baudios (símbolos) por segundo.

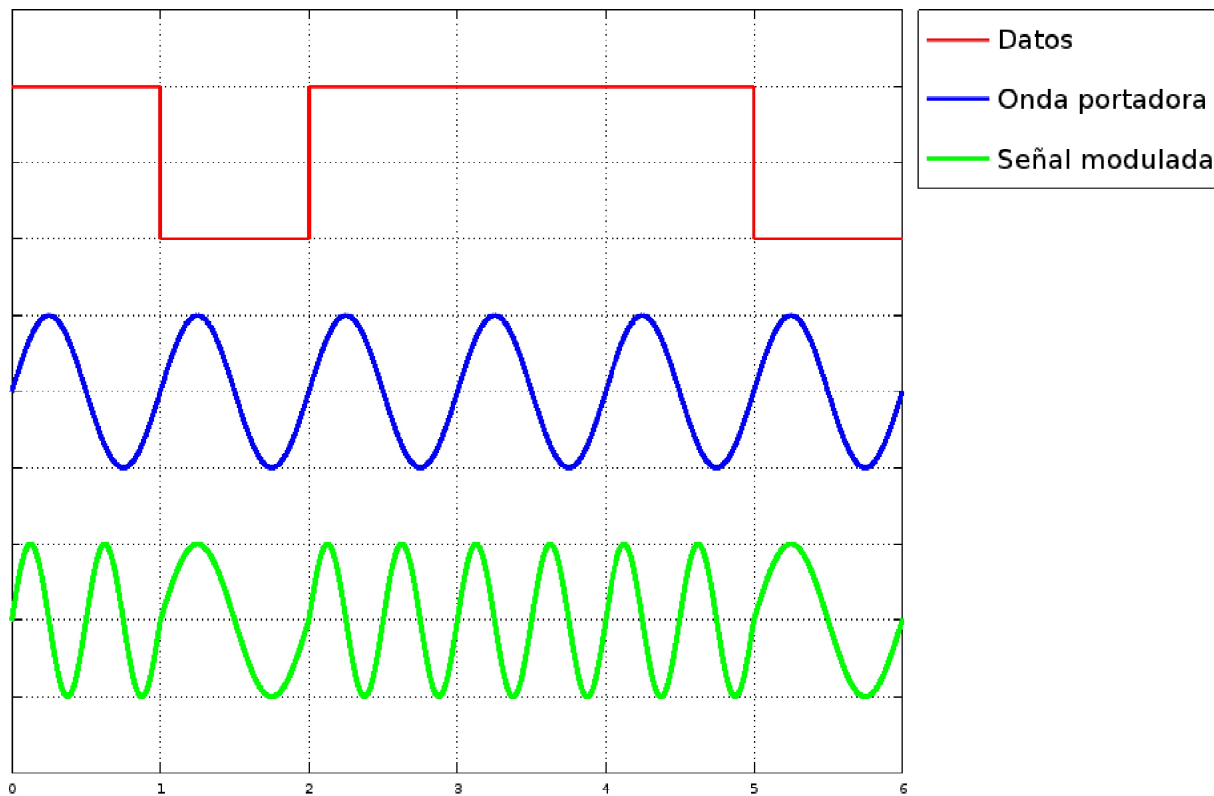


Figura 2.11: Modulación FSK

La señal modulada con FSK tiene una forma sinusoidal cuya función es de la siguiente forma:

$$A * \sin(2\pi(f \pm \Delta f)t) \quad (2.1)$$

Donde A es la amplitud de la señal, f es la frecuencia central y t es el tiempo. En esta modulación, la frecuencia que representa el 1 binario es $f + \Delta f$ y la que representa el 0 binario es $f - \Delta f$.

La diferencia en frecuencia Δf en torno a la frecuencia central f se denomina "desviación de frecuencia" y tiene una directa relación con el ancho de banda.

Gaussian Frequency Shift Keying

Es similar a la modulación FSK, pero antes que la señal en banda base entre a un modulador FSK, pasa por un filtro gaussiano para suavizar las transiciones y de esta manera limitar el ancho del espectro.

Es utilizado en dispositivos Bluetooth, Cypress WirelessUSB, Nordic Semiconductor, Texas Instruments LPRF, Z-Wave y Wavenis.

En la izquierda de la figura 2.12 tomada del datasheet del RFM22B, se puede apreciar como el filtro gaussiano suaviza los bordes de los datos digitales, mientras que en la derecha se ve en rojo que la modulación GFSK utiliza un espectro mucho más angosto que FSK, ya que no posee los lóbulos laterales.

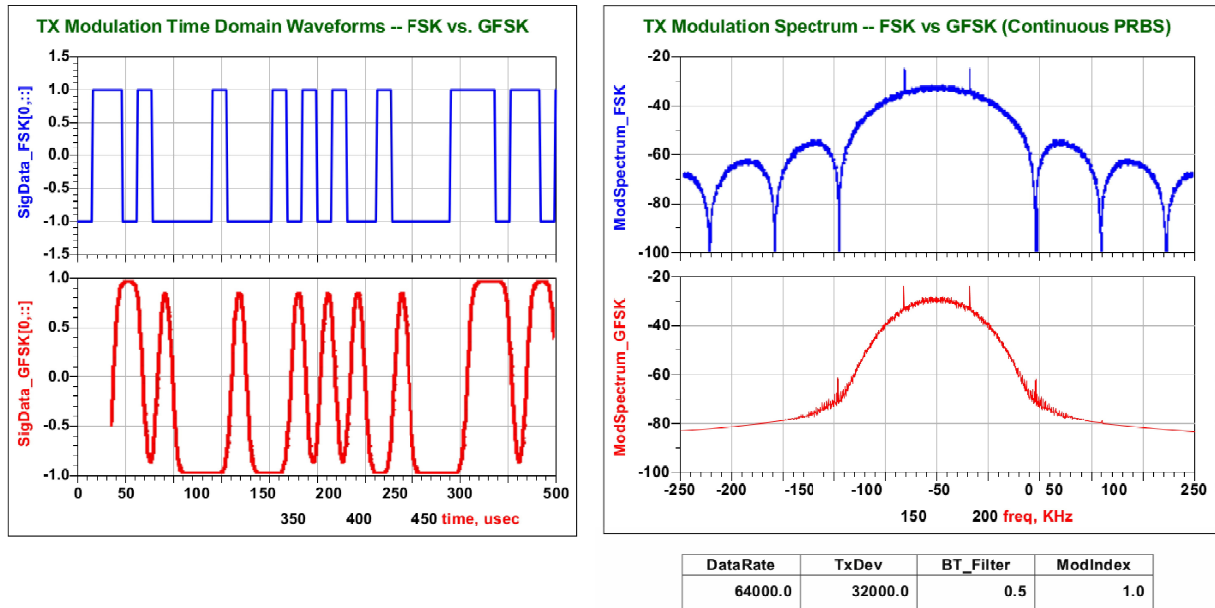


Figura 2.12: Espectros de FSK vs GFSK

Minimum Shift Keying

Esta modulación consiste en dos componentes de cuadratura llamados I y Q, en donde el componente Q está retrasado en la mitad del período de símbolo. Para transmitir datos usando MSK, estos son separados en dos series de bits en donde cada bit dura el doble de la señal original. Una de las series contiene los bits impares y la otra los pares con un offset de la duración de un bit. Además, se utilizan dos señales sinusoidales, una con el doble de frecuencia de la otra y dependiendo de los valores de las series pares e impares, se selecciona la señal más rápida o la más lenta y se escoge una fase de 0° o de 90° .

Valor del bit		Salida MSK	
Impar	Par	Frec.	Sentido
1	1	Alta	+
-1	1	Baja	-
1	-1	Baja	+
-1	-1	Alta	-

Tabla 2.3: Tabla para la salida en MSK

En la figura 2.13 se muestra un ejemplo para 10 bits de datos (1011010011), luego, los bits impares son 11001, los cuales al duplicar su duración de cada bit son 1111000011. Usando

el mismo procedimiento para los bits impares y agregándoles el *offset* de la duración de un bit, queda de la forma 011110011. Luego, se usa la tabla anterior para decidir si usar la alta o la baja frecuencia y en qué sentido. Notar que ambas tienen un desfase de 90 grados.

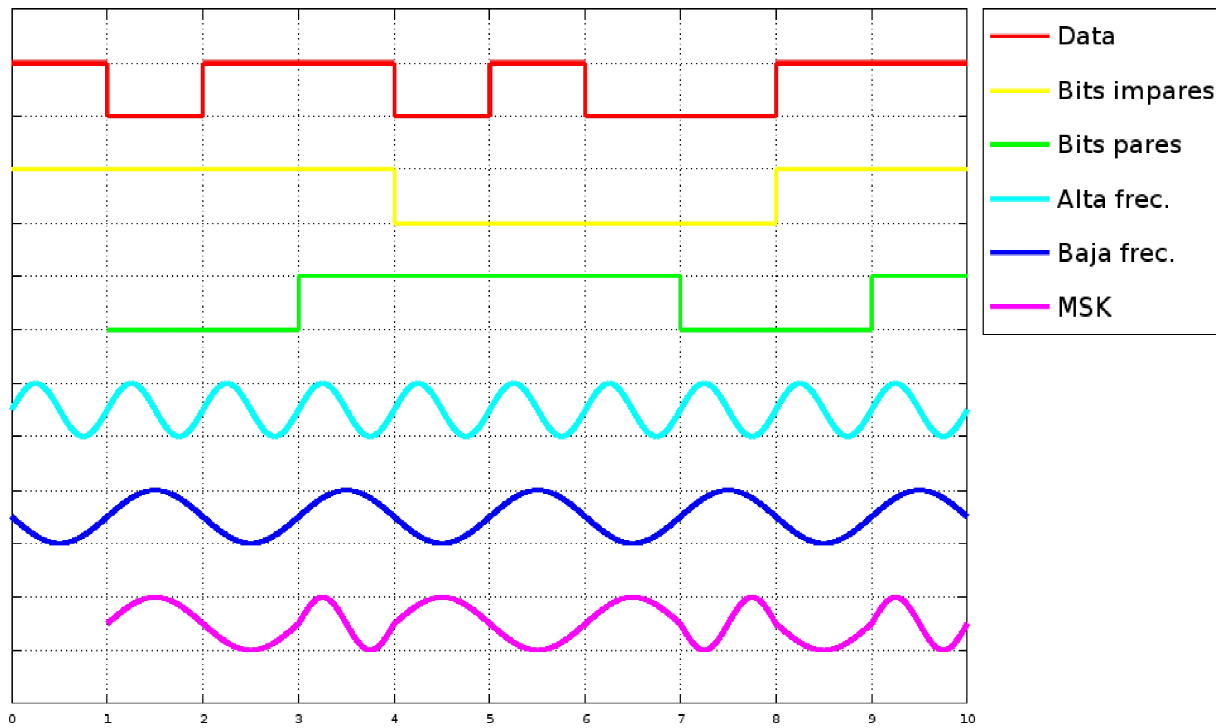


Figura 2.13: Modulación MSK

On Off Keying

Tal como dice su nombre, OOK es una modulación que representa datos con la presencia o la ausencia de la onda portadora. En su forma más simple, la presencia de una onda portadora durante cierta duración es un uno binario, mientras que su ausencia por la misma duración representa un cero binario. Es muy utilizada para transmitir código morse a través de radiofrecuencias.

2.3.9. Cálculo de radioenlace

Para calcular cuál es la distancia máxima teórica del sistema de comunicación que se va a implementar, se debe tener en consideración una serie de características que influyen en su desempeño. Entre los factores que afectan un enlace de comunicación están los siguientes:

- La sensibilidad del receptor: Se relaciona con cuál es el mínimo valor de potencia que debe llegar al receptor para que éste pueda decodificar los datos transmitidos.

- Ruido en la banda: Cuánta es la mínima tasa de señal a ruido que le permite al receptor extraer una señal a pesar de las demás señales que están ocupando la misma banda en frecuencia.
- Potencia de salida: El nivel de potencia que le puede entregar el transmisor a la antena.
- Patrón de radiación: Representa gráficamente la forma, dirección e intensidad de la energía radiada al espacio.
- Pérdidas de transmisión: Muestra la disipación de la señal a medida que viaja a través de la atmósfera o espacio libre, desde el emisor al receptor.

Potencia y sensibilidad

Los niveles de potencia y sensibilidad de los transceptores U482C, RFM22B y CC1101 se pueden ver en la tabla 2.4. En este tópico en particular el transceptor U482C es muchas veces superior ya que su potencia de salida es equivalente a 1 W, mientras que el RFM22B es de 0,1 W y el de CC1101 es de solo 0,01 W. En el caso de la sensibilidad, la diferencia no es tan grande entre el U482C y el RFM22B. De estos datos se desprende que el RFM22B es diez veces inferior al transceptor de alta gama llamado U482C en términos de potencia de transmisión, sin embargo, es de un precio mucho menor y es diez veces más potente que un transceptor que fue enviado en una misión espacial en femtosatélites, lo cual es considerable.

transceptor	Potencia de salida	Sensibilidad
U482C	30 dBm	-126 dBm
RFM22B	20 dBm	-121 dBm
CC1101	10 dBm	-116 dBm

Tabla 2.4: Potencia y sensibilidad de los transceptores

Antenas

El tipo de antena afecta considerablemente el rango máximo de un transceptor. Una forma de ver qué tan buena es una antena es viendo su directividad ². La directividad es una medida logarítmica de la relación entre qué tan intensa es la radiación de la antena en la dirección del máximo y la intensidad de radiación de una antena isotrópica que radia con la misma potencia total. Mientras más directiva sea una antena, mejor concentra la radiación en una dirección, y por lo tanto, la potencia de la señal. La directividad no tiene unidades y se expresa en unidades logarítmicas (dBi) como se ve en la ecuación 2.2.

$$D = 10 \cdot \log \left(\frac{U(max)}{U(iso)} \right) \text{ dBi} \quad (2.2)$$

²Pueden haber casos en que se desee transmitir omnidireccionalmente, pero para ésta aplicación en particular, es mejor que sea más directiva

El transceptor RFM22B se puede adquirir con una *whip antenna* con conector SMA , la cual es de un largo de 100 mm y genera una radiación omnidireccional. En la tabla 2.5 se muestran las características de esta antena [3] y la de la estación terrena [30] ubicada sobre el Departamento de Ingeniería Eléctrica. No están las especificaciones de la antena del SUCHAI debido a que fue fabricada en el laboratorio de SPEL del Departamento de Ingeniería Eléctrica de la Universidad de Chile, por lo que no se cuenta con una hoja de datos de donde sacar información.

Transmisor	Directividad	VSWR	Rango de frecuencia
RFM22B	3 dBi	$\leq 1,5$	413 - 453 MHz
Estación terrena	18,9 dBiC	$\leq 1,5$	430 - 438 MHz

Tabla 2.5: Especificaciones de las antenas

El parámetro VSWR (*Voltage Standing Wave Ratio*) describe la potencia reflejada desde la antena y está en función del coeficiente de reflexión (Γ). La fórmula del VSWR se encuentra en la ecuación 2.3.

$$VSWR = \frac{1 + |\Gamma|}{1 - |\Gamma|} \quad (2.3)$$

En un caso ideal, el valor del VSWR sería 1, lo cual indicaría que el coeficiente de reflexión es 0 y por lo tanto no hay potencia reflejada desde la antena. En este caso específico el coeficiente es suficientemente bajo, por lo que la mayor parte de la potencia llega a la antena.

Free-space path loss

Las pérdidas de propagación en el espacio libre (FSPL) son la pérdida de potencia de la señal que resulta de una trayectoria en línea de vista sin obstáculos que causen reflexiones ni difracciones. Para poder determinar de qué magnitud es la atenuación en decibelios de una distancia entre emisor y receptor, se utiliza la ecuación de Friis 2.4 en donde d es la distancia entre emisor y receptor y f es la frecuencia a la que se transmite la señal.

$$FSPL(dB) = 20 \log_{10} \left(\frac{4\pi d f}{c} \right) \quad (2.4)$$

Distancia máxima teórica

Utilizando la información anterior, se calcula la distancia máxima teórica para transmisiones entre dos transceptores RFM22B y para transmisiones entre el mismo transceptor y la estación terrena. El primer caso es para obtener la distancia a la que se podrían comunicar dos femtosatélites si utilizaran este transceptor, y el segundo es para verificar si

un CubeSat se podría comunicar con una estación terrena usando el RFM22B. En ambos casos se considera una frecuencia de transmisión de 437,225 MHz

Comunicación entre transceptores

Considerando los datos técnicos mostrados anteriormente, se suman las potencias de el transmisor, las antenas, se restan las pérdidas por transmisión de espacio libre y se iguala a la sensibilidad del transceptor. Las pérdidas de transmisión por cable son despreciadas, ya que la antena se conecta directamente al transceptor, además se agrega un margen de desvanecimiento [6] para incluir en el cálculo de radioenlace la influencia de condiciones climáticas extremas.

Transmisor	Antena TX	Antena RX	Sensibilidad RX	Margen de desvanecimiento
+20 dBm	+3 dBi	+3 dBi	-121 dBm	+20 dB

Tabla 2.6: Potencias para el cálculo de enlace entre TRX RFM22B

Sumando los términos e igualándolos a la sensibilidad del transceptor más el margen de desvanecimiento, se obtiene la ecuación siguiente:

$$\begin{array}{rccccccccc}
 \text{TX} & + & \text{Ant. TX} & - & \text{FSPL} & + & \text{Ant. RX} & = & - & \text{Sens. RX} & + & \text{Margen} \\
 20 \text{ dBm} & + & 3 \text{ dBi} & - & \text{FSPL} & + & 3 \text{ dBi} & = & - & 121 \text{ dBm} & + & 20 \text{ dB}
 \end{array}$$

El resultado de este cálculo es de una atenuación máxima de 127 dB. Utilizando la ecuación de pérdidas de espacio libre, se obtiene una distancia máxima de 122,2 km, lo cual permitiría abarcar un área considerable utilizando una constelación de femtosatélites.

Comunicación hacia la estación terrena

Análogamente al caso anterior, se crea una tabla con los valores de potencias para un enlace de comunicación entre el transceptor RFM22B y la estación terrena. En este caso se agregan pérdidas por transmisión por los cables y conectores de la estación terrena.

Transmisor	Antena TX	Antena RX	Sensibilidad RX	Pérdidas cables	Margen
+20 dBm	+3 dBi	+18.9 dBiC	-121 dBm	-2 dB	+20 dB

Tabla 2.7: Potencias para el cálculo de enlace hacia la estación terrena

Se calcula el radioenlace considerando las pérdidas mencionadas anteriormente, como se puede ver en la siguiente expresión:

$$\begin{array}{rcccccccccccc}
 \text{TX} & + & \text{Ant. TX} & - & \text{FSPL} & + & \text{Ant. RX} & - & \text{Cables} & = & - & \text{Sens. RX} & + & \text{Margen} \\
 20 \text{ dBm} & + & 3 \text{ dBi} & - & \text{FSPL} & + & 18,9 \text{ dBiC} & - & 2 \text{ dB} & = & - & 121 \text{ dBm} & + & 20 \text{ dB}
 \end{array}$$

Las pérdidas permitidas en una comunicación entre el transceptor RFM22B y la estación terrena son de 140,9 dB, lo cual equivale a una distancia máxima teórica de aproximadamente 605,6 km gracias a la gran directividad que posee la antena de la estación terrena.

Estos cálculos demuestran que desde el punto de vista de la potencia del transceptor, es posible establecer un enlace de comunicación con la estación terrena, considerando que el SUCHAI 1 va a estar en una órbita entre 400 y 700 km de altura.

Capítulo 3

Implementación

Para evaluar el transceptor se establecen dos tipos de pruebas. Por un lado están las realizadas en terreno, en las cuales se prueban distintos tipos de modulación y tasa de transferencia de datos del RFM22B para determinar cual es la configuración que entrega mejores resultados. Luego, con la configuración de mejor desempeño se hace una prueba en un globo sonda para cuantificar cuál es el alcance máximo que puede tener este transceptor. Esta última prueba es necesaria ya que es la mejor forma de simular un lugar libre de obstáculos como los que hay en el espacio.

3.1. Desarrollo en terreno

Para hacer un prototipado rápido del proyecto, se compra una versión del transceptor fabricada por Modtronics [20] que viene con una placa de adaptación para poder conectarlo a una protoboard (figura 3.1). El adaptador viene con un conector SMA para ponerle una antena al transceptor, la cual está incluida junto con los pines *header* macho, los cuales se deben soldar en el laboratorio.

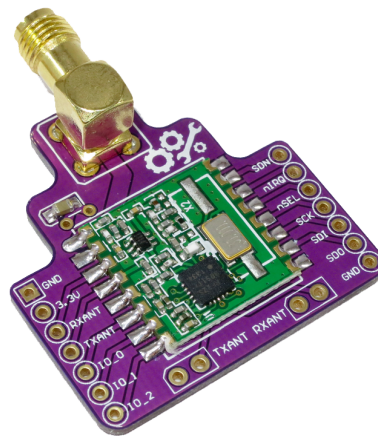


Figura 3.1: Transceptor con placa adaptadora

Las conexiones del transceptor al Arduino Due se pueden ver en la tabla 3.1. Es necesario tener en consideración que son específicas para ese modelo de transceptor y para ese modelo de Arduino. En los casos en que salen en color, es porque las conexiones se hacen entre pines del transceptor del mismo color y no con el Arduino, salvo por el caso de GPIO_2, el cual no se utiliza. La conexión a la entrada de voltaje del transceptor (3,3 V) es opcional, pues el usuario puede escoger entre usar el regulador de voltaje del arduino o uno externo.

RFM22B	Arduino Due
GND	GND
3,3 V	3,3 V
RXANT	-
TXANT	-
GPIO_0	-
GPIO_1	-
GPIO_2	-
SDN	Pin D10
NIRQ	Pin D2
NSEL	Pin D4
SCK	SCK
SDI	MOSI
SDO	MISO

Tabla 3.1: Tabla de conexiones RFM22B - Arduino Due

Luego, el Arduino se conecta a la protoboard mediante cables (figura 3.2) y se alimenta a través de un puerto USB del computador. El transceptor se alimenta utilizando una fuente de poder, sin embargo, es posible alimentarlo a través del Arduino Due, pues éste es capaz de entregar 800 mA en su salida de corriente continua de 3,3 V. A su vez, un puerto USB 2.0 puede entregar una corriente máxima de 500 mA, lo cual es suficiente para alimentar al transceptor cuando está transmitiendo a máxima potencia (80 mA - 85 mA).

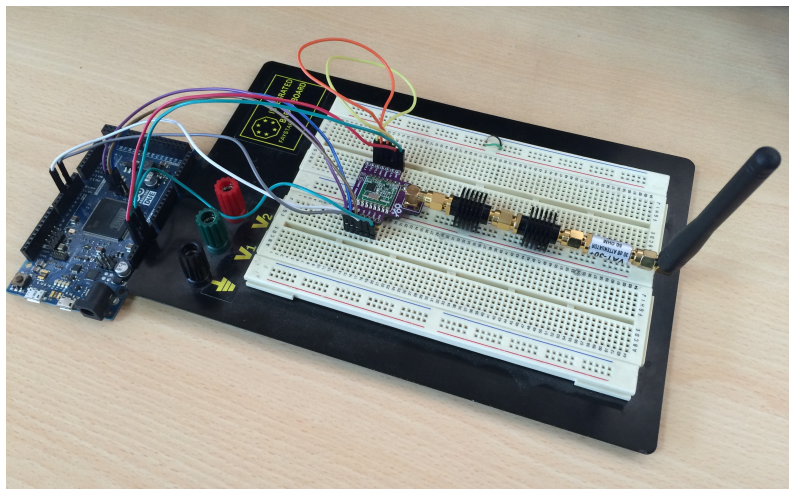


Figura 3.2: Prototipado del proyecto

3.1.1. Programación

Inicialmente se hace un programa tipo *chat*, en donde el usuario puede escribir un mensaje a través del monitor serial del IDE de Arduino. El tamaño del mensaje se limita por el carácter nulo o por el largo máximo de un mensaje (255 bytes). Dicho mensaje se envía al transceptor a través del cable USB conectado al *programming port* del Arduino Due y luego se transmite a otro transceptor que a su vez está conectado a otro computador. Finalmente, el mensaje se imprime en la pantalla del otro usuario, el cual también puede responder. De esta forma se comprueba que efectivamente se puede establecer un enlace de comunicación en ambos sentidos utilizando estos dispositivos.

Código 3.1: Parte del código para el chat

```
73 void chat() {
74     msn_size = 0;
75     while (Serial.available()) {
76         letra = Serial.read();
77         if (letra == '\0' || msn_size == 255) {
78             break;
79         }
80         array[msn_size] = letra;
81         msn_size++;
82         delay(3);
83     }
84     uint8_t datos[msn_size];
85     for (i = 0; i < msn_size; i++) {
86         datos[i] = array[i];
87         Serial.print((char)array[i]);
88     }
89     enviar(datos, msn_size);
90 }
```

En la siguiente iteración se opta por hacer un código en que un transceptor se comporta como el encargado de generar y enviar los mensajes, mientras que el otro los recibe y los reenvía como si fuera un eco. De esta manera, el emisor puede luego comparar el mensaje enviado inicialmente con el recibido posteriormente y ver si son iguales, de ser así se tiene un acierto y se cuenta internamente. Cada 10 mensajes enviados se imprime en la pantalla los aciertos parciales hasta ese momento y cuántos de los 10 mensajes enviados llegaron correctamente. Esto permite que solo se necesite un computador conectado al emisor para ver los resultados, ya que el receptor solamente va a reenviar el mensaje.

El diagrama de flujo del programa del emisor en forma general se puede ver en la figura 3.3. Al inicializar el transceptor, se escoge la frecuencia a la que se va a transmitir, cual va a ser el *header* del mensaje y el tipo de CRC que se va a utilizar, además de otras variables que son ajustadas por la librería.

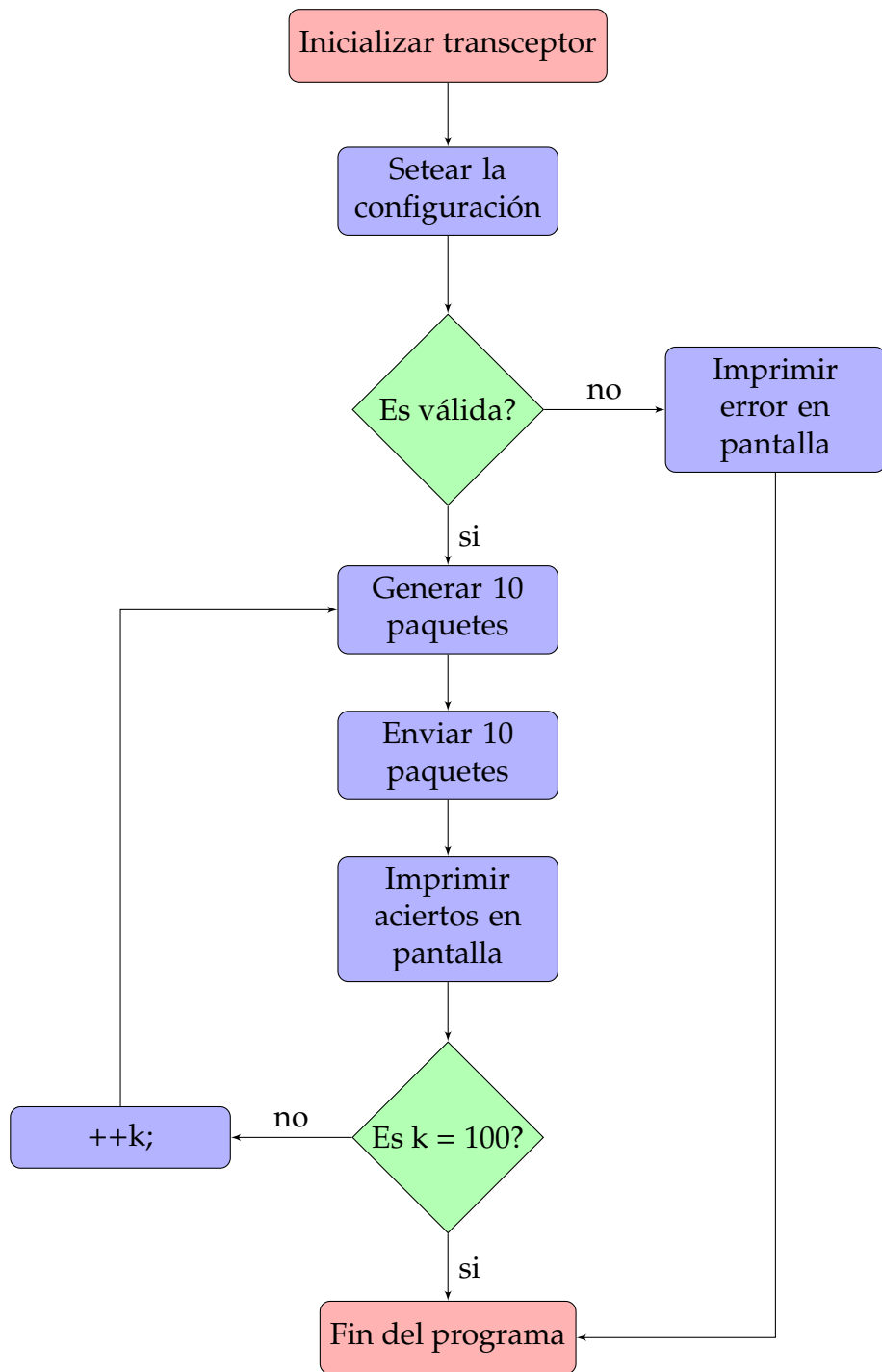


Figura 3.3: Diagrama de flujo general del TX

Para transmitir los paquetes se crea una función llamada *sendPacket*, la cual recibe como parámetros el mensaje y su tamaño. La mayor parte de sus funcionalidades son dadas por la librería de RFM22B para Arduino. Su comportamiento está descrito en el diagrama de flujo de la figura 3.4. Cada vez que ocurre un error, se imprime en pantalla un mensaje indicando cual fue el fallo. Al momento de comparar el mensaje original con el recibido, se suma en una variable global el nuevo acierto del sistema de comunicación.

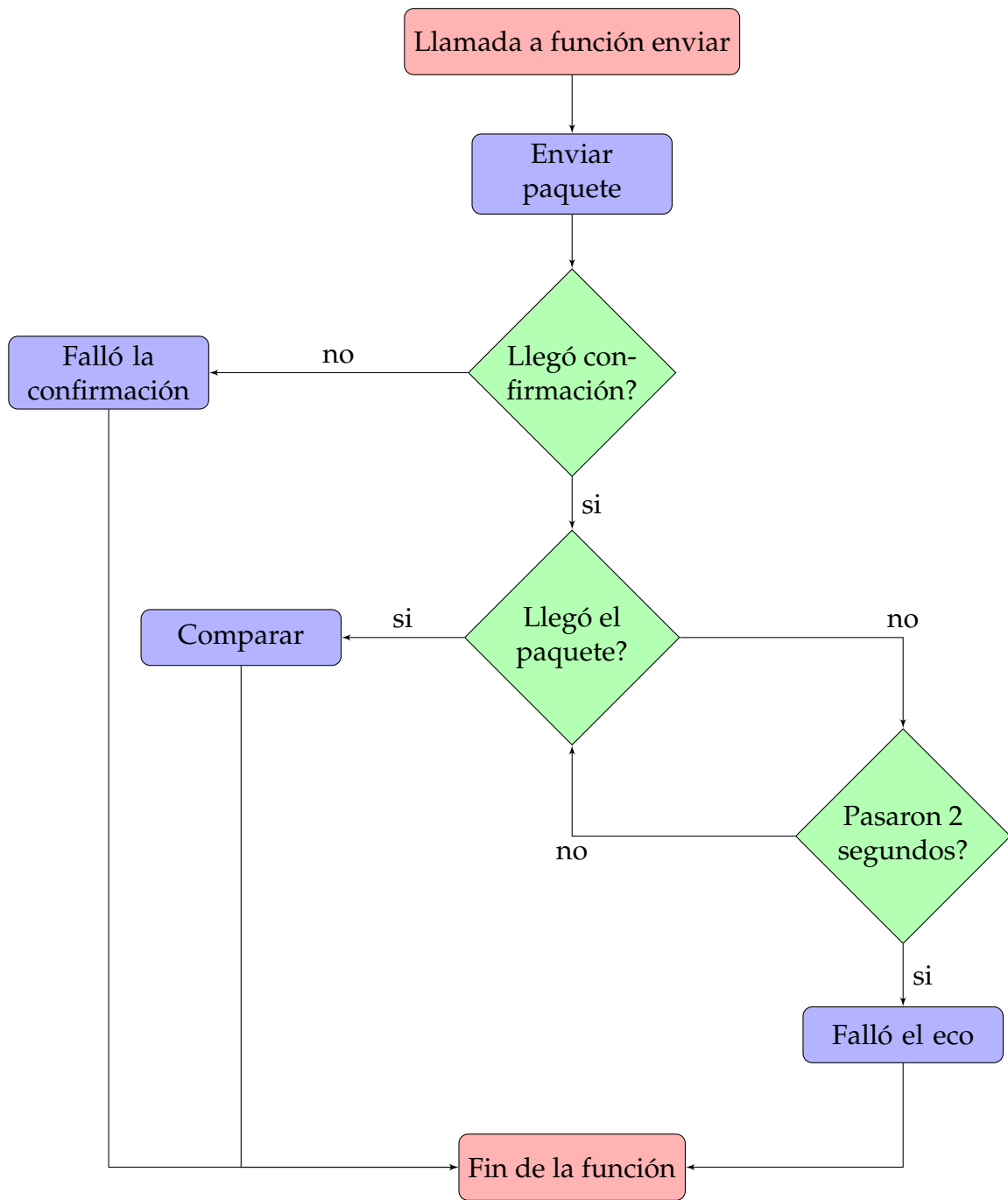


Figura 3.4: Diagrama de flujo de la función enviar

La función utilizada para recibir los mensajes es similar a la que los envía. Primero se verifica si un mensaje recibido está disponible. Si el mensaje es válido, se guarda en un buffer en el Arduino. Luego el mensaje se reenvía y se espera una confirmación, si ésta no llega, se imprime un error en pantalla (para el caso en que se estén utilizando dos computadores). Para que la recepción del mensaje sea efectiva, esta función se debe llamar reiteradamente en el `loop` principal.

Debido a la simpleza de esta función, se opta por mostrar el código en vez de un diagrama de flujo. La función parte verificando si hay un mensaje disponible, de ser válido, el método *recvfromAck* envía una confirmación a la dirección de la fuente de donde provino el paquete y luego devuelve *true*, de lo contrario, devuelve *false*. Los paquetes se guardan en *buf*, su tamaño en *len* y la dirección de la fuente en *from*. Luego, el método *sendtoWait* en este caso se utiliza para enviar el mismo mensaje almacenado en *buf* a la fuente *from*. Si no hay una confirmación de la recepción del mensaje, se imprime un error en el monitor. La totalidad del código de RX y TX se puede encontrar en Anexos.

Código 3.2: Función *receivePacket*

```

46 void receivePacket() {
47     if (rf22.available()) {
48         uint8_t len = sizeof(buf);
49         uint8_t from;
50         if (rf22.recvfromAck(buf, &len, &from)) {
51             Serial.print("\n");
52             if (!rf22.sendtoWait(buf, len, from)) {
53                 Serial.println("Confirmation failed");
54             }
55         }
56     }
57 }

```

El computador conectado al transceptor encargado de generar los mensajes puede ver la cantidad de aciertos en tiempo real. Esto permite saber cuál es el desempeño de la actual configuración en cierta distancia. Para ello, como se dijo anteriormente, se imprime en pantalla la cantidad de aciertos en cada 10 paquetes enviados y los aciertos parciales hasta el momento. Para ilustrar esto, se muestra una imagen tomada de una prueba en laboratorio (figura 3.5), en donde el receptor se deja deliberadamente apagado un tiempo al principio, para mostrar que efectivamente se generan mensajes indicando que hubo un fallo en la confirmación de que el mensaje llegó a destino.

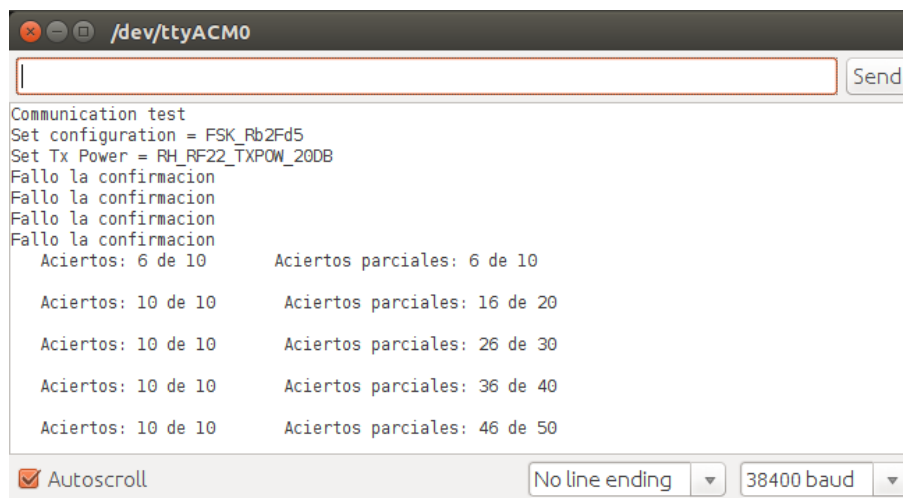


Figura 3.5: Impresión en pantalla de los aciertos

3.1.2. Diseño de PCB

Debido a que es necesario hacer pruebas en terreno y se desea evitar tener problemas como cables sueltos o que se han cortado internamente, además de la complicación de llevar la protoboard con todos los cables conectados al arduino, se decide diseñar una PCB. El diseño debe ser de tal manera que sea fácil de conectar el transceptor al Arduino, teniendo cuidado de no hacer un corto circuito con los múltiples conectores que el Arduino posee. Debido a esto, se opta por hacer un *shield* que es una placa diseñada para conectarse directamente a las entradas y salidas del Arduino.

El desarrollo de la PCB se hace en el *software* KiCad, el cual es de código abierto y tiene múltiples componentes disponibles para acelerar el proceso de diseño. Además, existe una librería para hacer un shield de Arduino, la cual está a libre disposición en GitHub [8]. Esta librería viene con la mayor parte de las entradas y salidas del Arduino dispuestas en la posición y distancias que corresponde, facilitando el diseño de la placa. Los pines que aparecen son los digitales, analógicos, las alimentaciones, la tierra (referencia) y los puertos SPI, que son esenciales para comunicarse con el transceptor RFM22B.

En la figura 3.6 se ve un diseño definitivo para la placa de adaptación transceptor-Arduino. En el lado izquierdo está la parte trasera de la placa y en el derecho la frontal. Fue necesario utilizar vías para poder hacer conexiones entre las dos capas de la PCB.

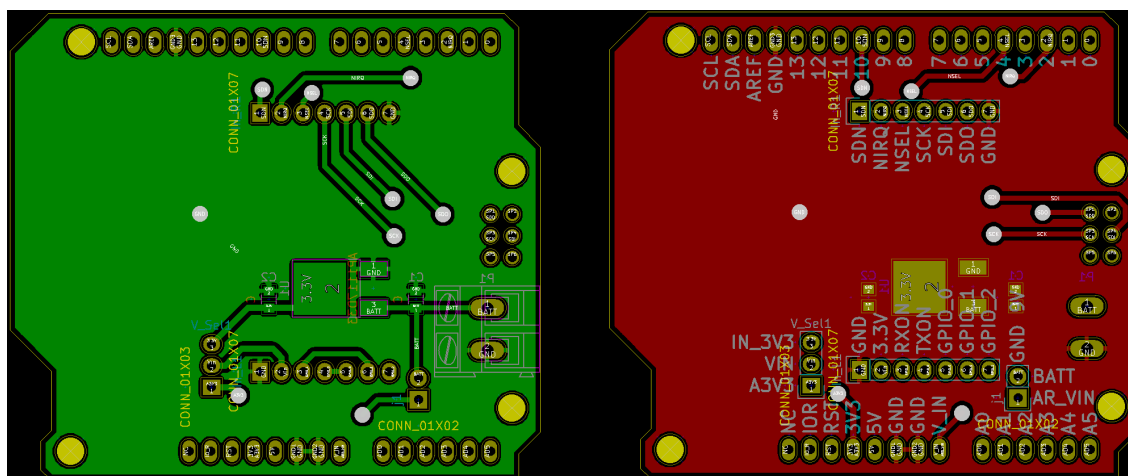


Figura 3.6: Diseño de PCB en KiCad

Luego, gracias a la gentileza de Miguel Patiño, encargado del Laboratorio de Electrónica del DIE, se fabricaron placas (figura 3.7) con el diseño realizado en KiCad. Para ello se usó una máquina ruteadora CNC, la cual hace perforaciones en una placa de cobre con brocas de distintos tamaños para poder insertar los componentes electrónicos. Éstas fueron utilizadas para hacer pruebas en el laboratorio y fuera de él hasta que llegaron las placas definitivas.

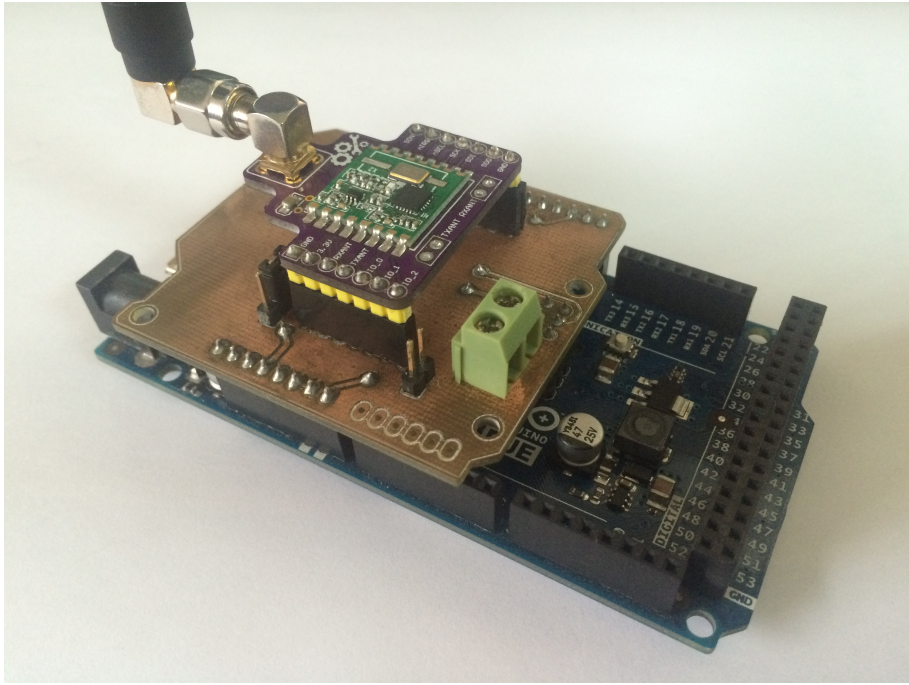


Figura 3.7: Placa de prueba

Posteriormente llegaron las placas fabricadas con un alto nivel de calidad, las cuales al estar protegidas con una máscara de soldadura, facilitan el proceso de soldar los componentes, además de no tener áreas de cobre expuestas que pueden conducir a tener un corto-circuito. Estas placas vienen con la opción descrita anteriormente de escoger entre alimentar el transceptor utilizando el regulador de voltaje incorporado en el Arduino, o a través de un regulador externo soldado en la misma placa, haciendo el sistema más robusto ante fallos inesperados. Además, viene con etiquetas de cada pin, para saber cual es la orientación correcta en que se debe conectar el transceptor a la placa.

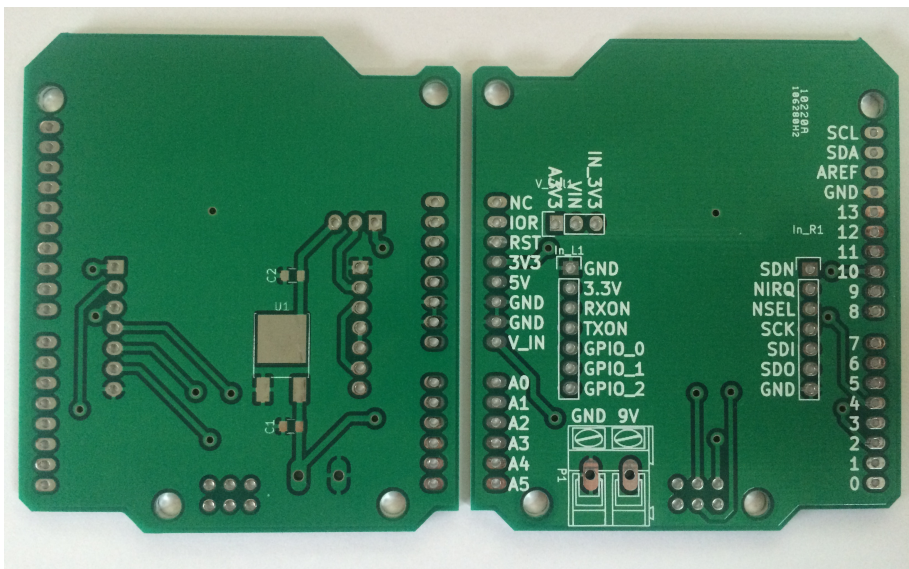


Figura 3.8: PCB fabricada para conectar el transceptor al Arduino Due

3.1.3. Reciclaje de baterías de litio

Para hacer pruebas fuera del laboratorio, es necesario tener baterías, pues no se puede llevar una fuente de poder del laboratorio a terreno. Debido a esto se buscan baterías que sirvan para alimentar el transceptor y el Arduino, los cuales requieren de 3,3 V y 7 – 12 V respectivamente. Para minimizar el impacto medioambiental del proyecto, se opta por reciclar baterías de *notebooks* viejos, las cuales al estar completamente cargadas, pueden entregar 4,2 V cada una y suficiente corriente para todo el sistema.

Para poder alimentar el Arduino Due, se conectan dos baterías en serie, y para el transceptor se utiliza un regulador de voltaje ya contemplado anteriormente. Además, se le agrega un switch al polo positivo de las baterías, para poder encender y apagar cómodamente todo el sistema. También se utilizan cables *jumper* debido a su comodidad de uso pues ya tienen los bordes listos para conectarlos a un terminal, sin gastarse. Todas las partes expuestas de las baterías y de su unión con los cables se cubre con huincha aislante y las uniones entre los cables y los switch se protegen con tubos termoretráctiles para evitar posibles corto-circuitos. El resultado se puede apreciar en la figura 3.9.

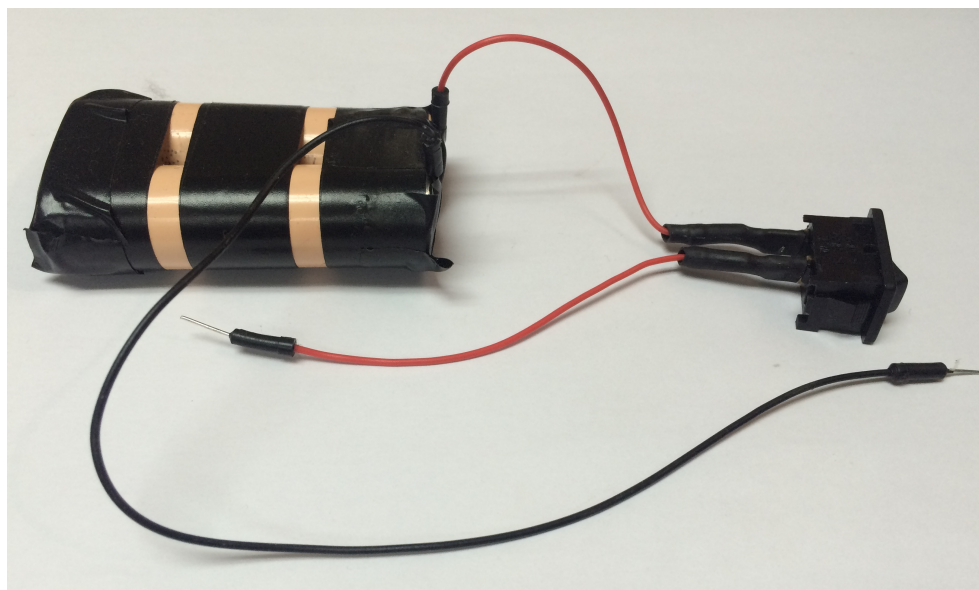


Figura 3.9: Baterías de litio

3.1.4. Montaje final

Luego de soldar los componentes necesarios tales como el regulador de voltaje que alimenta el transceiver, condensadores, pines *header*, entre otros, se procede a conectar el transceptor a la placa adaptadora, y ésta al Arduino Due. Después de conectar la batería al sistema, se fija toda la electrónica a un atril de música, pues es necesario tener las antenas en altura para poder lograr un buen enlace de comunicación. Un atril mide 3,15 m mientras que el otro 2,15 m, lo cual es suficiente para hacer pruebas iniciales.

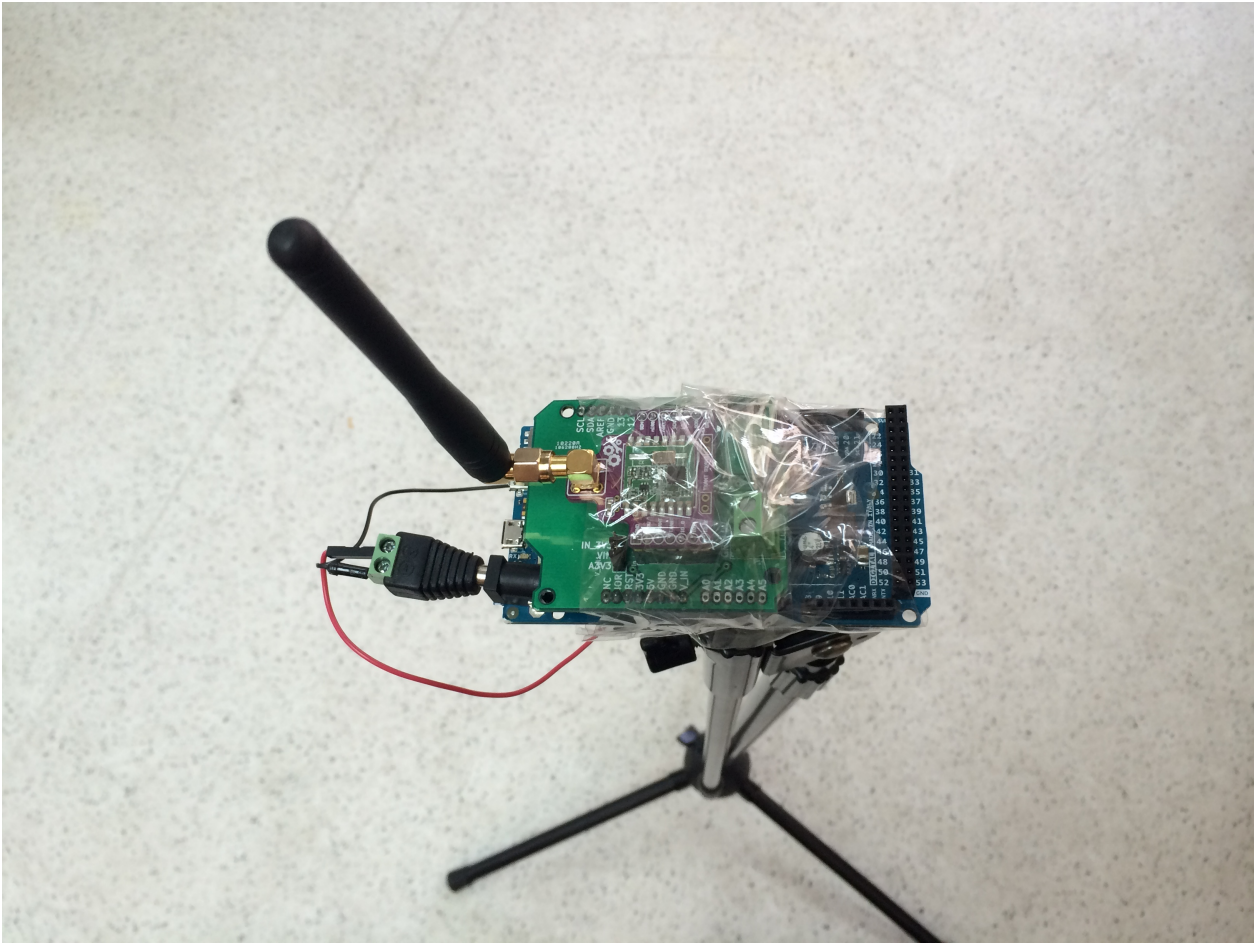


Figura 3.10: Transceptor más batería unidos al atril

Este montaje fue utilizado en todas las pruebas realizadas fuera del laboratorio, incluyendo el parque O'Higgins, el cerro La Pirámide, el Cerro San Cristóbal y la azotea del edificio de Electrotecnologías.

3.2. Desarrollo de la radiosonda

Para determinar el alcance máximo del transceptor, se hace una prueba utilizando un globo con helio facilitado por el Departamento de Geofísica de la Universidad de Chile para llevar el transmisor. Esta prueba permite tener la mejor aproximación posible a las condiciones de línea de vista que se pueden tener en el espacio.

A diferencia de las pruebas anteriores, se incluye dentro de los materiales un GPS para poder determinar la posición y la altura de la radiosonda. De esta manera se puede calcular cual es la distancia que hay entre la radiosonda y los receptores. También se transmite la

hora, la velocidad, el curso y la cantidad de satélites a los que se logra conectar el GPS.

El GPS utilizado es el L80 [26], el cual requiere un voltaje de entrada de 3,3 V y un consumo de corriente que se puede ver en la tabla 3.2. Cuando el GPS entra en modo *backup*, se debe apagar la entrada VCC y solo se alimenta por V_BCKP. El L80 transmite sus datos por UART, pero cuando está en el modo *backup*, no puede adquirir datos, buscar satélites ni acceder al UART. A pesar de lo anterior, mantiene una memoria de respaldo para poder tener un inicio rápido. Este GPS recibe señales en la banda L1 desde los satélites GPS a una frecuencia nominal de 1575,42 MHz. Además, tiene embebidos un amplificador de bajo ruido y una *patch antenna*.

Modo	Condición	Típico	Unidad
Adquisición de datos	VCC = V_BCKP = 3,3 V	25	mA
Buscando satélites	VCC = V_BCKP = 3,3 V	20	mA
En espera	VCC = V_BCKP = 3,3 V	1,0	mA
<i>Backup</i>	V_BCKP = 3,3 V	7	nA

Tabla 3.2: Consumo de corriente del GPS L80

Para este proyecto se utiliza una versión hecha por MCI Olimex con una placa adaptadora para poder conectarlo fácilmente a una protoboard. Esta versión viene con la opción de utilizar una batería CR1225 para guardar la configuración del GPS e inicializarse rápidamente una vez encendido. Una de las ventajas de esta tarjeta es que puede conectarse al computador utilizando un conector FTDI, lo cual permite utilizar otros programas aparte del IDE de Arduino para poder ver e incluso almacenar los datos obtenidos del GPS. También tiene un conector UFL para agregar una antena externa si así se desea.

3.2.1. Pruebas preliminares

Inicialmente se hacen pruebas en protoboard para ver la compatibilidad del GPS L80 con el Arduino Due. Los GPS reciben los datos con la especificación NMEA (*National Marine Electronics Association*). Para facilitar la lectura de los datos para el usuario, se hace necesario interpretarlos con un programa, es por ello que se utiliza una librería compatible con Arduino llamada *TinyGPS++* [10]. Esta librería tiene licencia GNU LGPL, por lo que se puede utilizar, distribuir y modificar si es que se desea.

En la figura 3.11 se puede apreciar el prototipo de la unión del Arduino y el transceptor con el GPS. El GPS funciona correctamente, solo que es recomendable no utilizarlo dentro de un edificio, ya que no siempre el GPS logra conectarse a suficientes satélites como para tener datos suficientemente precisos de la posición y la altura. El primer dato que puede ser obtenido sin mayores errores es la hora, la cual está en UTC. También se utilizó el transceptor RFM22B para transmitir los datos sin problemas de compatibilidad, por lo que las pruebas fueron un éxito.

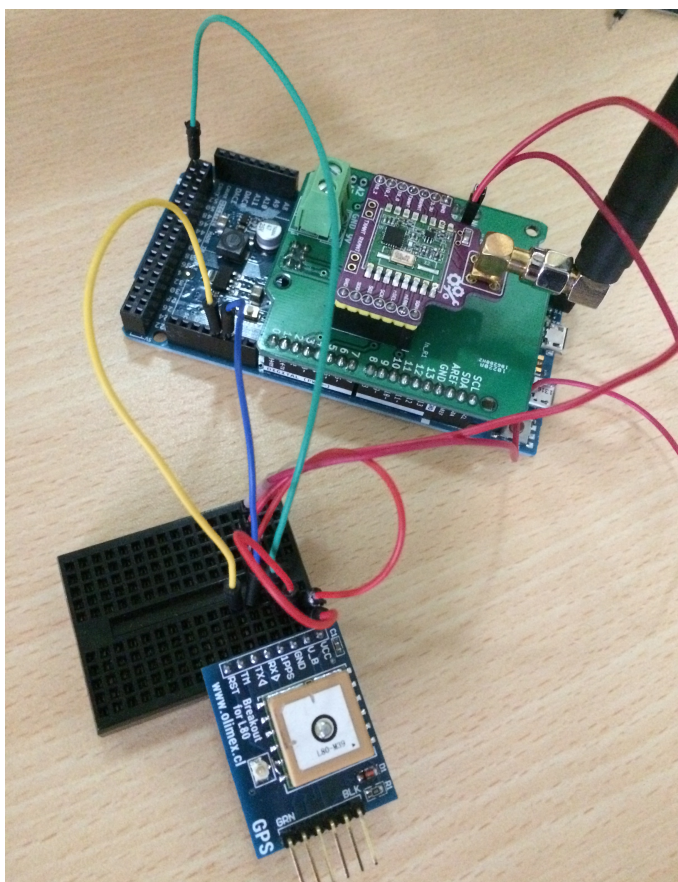


Figura 3.11: Arduino más GPS L80 en protoboard

Para que el arduino se pueda comunicar con el GPS, debe iniciar la comunicación serial con una tasa de baudios de 9600. El Arduino Due posee múltiples puertos de comunicación serial, por lo que no hubo problemas al utilizar el GPS para obtener los datos y el computador para imprimirlos en pantalla en las pruebas preliminares.

Las conexiones que se hicieron entre el Arduino Due y el GPS L80 se pueden ver en la tabla 3.3. Éstas sólo se utilizaron en pruebas en protoboard, ya que posteriormente se usó una batería externa para la entrada de voltaje de respaldo V_B .

Arduino Due	GPS L80
3,3 V	VCC
3,3 V	V_B
GND	GND
TX2 (Pin 16)	RX
RX2 (Pin 17)	TX

Tabla 3.3: Conexiones entre el Arduino Due y el GPS L80

3.2.2. Diseño de PCB

Se diseña una tarjeta electrónica utilizando el *software* KiCad para unir el transceptor y el GPS al Arduino, y de esta manera minimizar lo más posible los cables en la radiosonda. Para minimizar los tiempos de diseño y fabricación, se decide no incluir un regulador de voltaje aparte y utilizar la salida de 3,3 V del Arduino Due. La entrada de voltaje V_BCKP (llamada V_B en la placa de Olimex), se deja sin conectar para utilizar una batería CR1225.

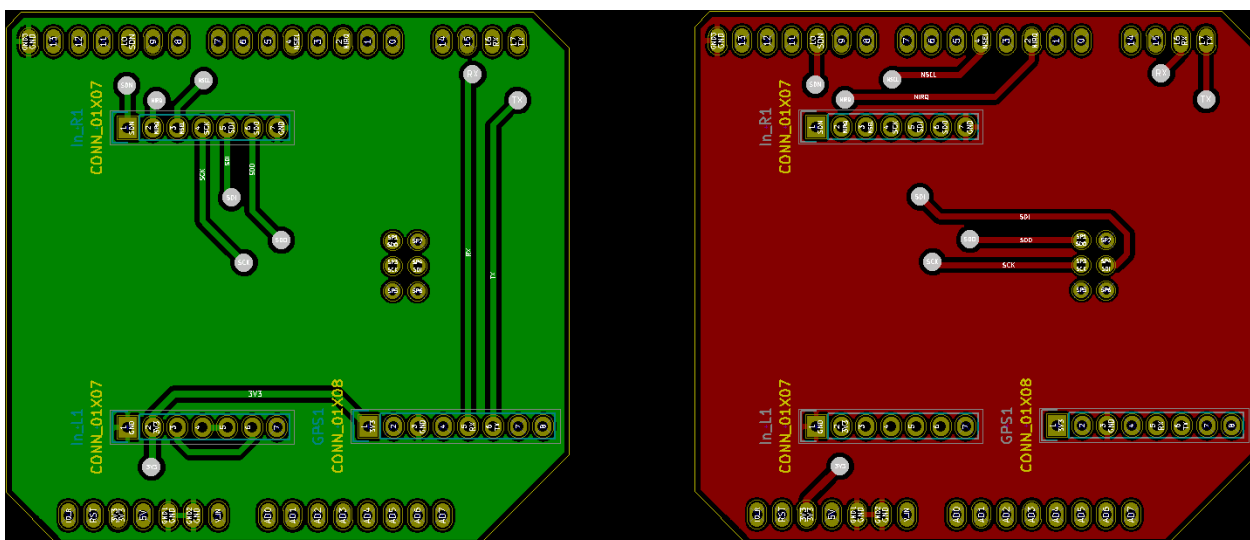


Figura 3.12: Diseño de PCB para incorporar el GPS

3.2.3. Código para la radiosonda

A diferencia de las pruebas anteriores, en este caso se transmiten los datos a tres receptores. Para lograr esto, primero se le asigna un número a cada uno de los receptores como se puede ver en el código 3.3.

Código 3.3: Definición de los receptores

```
7 #define OMNI_ARDUINO_ADDRESS 2
8 #define YAGI_ARDUINO_ADDRESS 3
9 #define EARTH_STATION_ADDRESS 4
```

Luego, se utiliza el método *sendtoWait* de la librería del transceptor y se especifica para qué receptor está destinado el mensaje. El programa está diseñado para intentar enviar el mismo mensaje a los tres receptores antes de pasar al siguiente mensaje. Por motivos de pruebas iniciales en las que el GPS está conectado a una pantalla, si un mensaje no es recibido, se imprime un error en el monitor serial indicando cuál fue el receptor que no confirmó la recepción del paquete.

Código 3.4: Función *sendPacket*

```
72 void sendPacket(uint8_t data[], int data_size) {
73     if (!rf22.sendtoWait(data, data_size, OMNI_ARDUINO_ADDRESS))
74         Serial.println(F("sendtoWait failed (Omnidirectional arduino)"));
75     if (!rf22.sendtoWait(data, data_size, YAGI_ARDUINO_ADDRESS))
76         Serial.println(F("sendtoWait failed (Yagi arduino)"));
77     if (!rf22.sendtoWait(data, data_size, EARTH_STATION_ADDRESS))
78         Serial.println(F("sendtoWait failed (Earth station)"));
79 }
```

La clase TinyGPS++ almacena los datos en estructuras. Por ejemplo, los datos de la velocidad se guardan en la estructura *TinyGPSSpeed*, declarada con el nombre *speed*, la cual contiene métodos que devuelven la velocidad en nudos (*knots()*), millas por hora (*mph()*), metros por segundo (*mps()*) y kilómetros por hora (*kmph()*).

Los datos obtenidos por el GPS muchas veces consisten en números decimales de tipo *double*, pero el transceptor RFM22B solo transmite datos tipo *uint8_t*, por lo que fue necesario crear una función que codifique la información en un formato aceptable para el RFM22B. Para lograr esto sin perder información, se opta por multiplicar el dato por un número múltiplo de 10, tal que los decimales importantes pasen a ser enteros. Después, el resultado se almacena en una variable tipo *int* (de 32 bits) y se separa en cuatro octetos tipo *uint8_t* por medio de la operación lógica *shift*.

En el código 3.5, se ve parte de la función encargada de transformar los datos adquiridos por el GPS al tipo de dato que acepta el transceptor. En cada caso, se verifica si el nuevo dato es válido. Si no lo es, se envía el último dato válido de esa variable, en este caso la altura. Si lo es, entonces como es un decimal y como la precisión es más allá de tres decimales, se multiplica por mil, se guarda en el entero *alt*, se separa en cuatro partes de 8 bits cada una y se guarda en el paquete a enviar.

Código 3.5: Extracto de función *encodePacket* (altura)

```
100     if (gps.altitude.isValid()) {
101         alt = gps.altitude.meters() * 1000;
102
103         packet[8] = alt >> 24;
104         packet[9] = alt >> 16;
105         packet[10] = alt >> 8;
106         packet[11] = alt;
107     }
```

Para el dato de la cantidad de satélites a los que se conectó el GPS, no es necesario multiplicarlo por un número, pues es un entero de tipo *uint32_t* (*unsigned int* de 32 bits), por lo que sólo se separa en cuatro partes y se guarda en el mensaje. La hora, el minuto y el segundo son de tipo *uint8_t*, por lo que se pueden guardar directamente en el paquete, como se ve en el código 3.6.

Código 3.6: Extracto de función encodePacket (tiempo)

```
124     if (gps.time.isValid()) {
125         packet[20] = gps.time.hour();
126         packet[21] = gps.time.minute();
127         packet[22] = gps.time.second();
128     }
```

Como todo programa en Arduino, la parte principal que se repite continuamente, se escribe en el *loop* (Código 3.7). En la línea 57 se muestra que mientras hayan datos disponibles en el *Serial2*, que es donde está conectado el GPS, va a estar la posibilidad de transmitir datos. Para que esto ocurra, se tiene que cumplir la condición de que el objeto *gps* esté obteniendo datos en formato NMEA para tenerlos actualizados, los cuales deben pasar una prueba de *checksum*. Si esto ocurre, los datos son transformados a *uint8_t* y luego son transmitidos. Si se reciben menos de 10 caracteres en más de 5 segundos, se imprime un error en pantalla diciendo que no se ha detectado un GPS.

Código 3.7: Loop de la radiosonda

```
56 void loop() {
57     while (Serial2.available() > 0) {
58         if (gps.encode(Serial2.read())) {
59             encodePacket();
60             sendPacket(packet, sizeof(packet));
61         }
62     }
63     if (millis() > 5000 && gps.charsProcessed() < 10) {
64         Serial.println(F("No GPS detected: check wiring."));
65     }
66 }
```

3.2.4. Código para los receptores

Para que cada receptor reciba el mensaje que le corresponda, se le debe asignar un número al momento de hacer una instancia de la clase *RHReliableDatagram*. El número se define de la misma forma que en el caso del transmisor y la instanciación se puede ver en el código 3.8. Primero se hace una instancia de la clase *RH_RF22* que es la versión más básica de la librería, la cual solo puede enviar y recibir mensajes, sin esperar confirmaciones. Luego, se hace la instancia de la clase *RHReliableDatagram*, que es la que tiene todas las ventajas de reenviar mensajes y confirmarlos.

Código 3.8: Instancias de las clases RH_RF22 y RHReliableDatagram

```
11 // Singleton instance of the radio driver
12 RH_RF22 driver(4,2);
13
14 // Class to manage message delivery and receipt, using the driver declared above
15 RHReliableDatagram rf22(driver, EARTH_STATION_ADDRESS);
```

En la subsección anterior se explicó cómo el transmisor transformó los datos a *uint8_t* para poder enviarlos con el transceptor. En este caso, el receptor debe hacer el proceso inverso, es por esto que se crea una función llamada *decodePacket*. Esta función guarda los cuatro octetos en una variable temporal de tipo *int* y luego esta variable la multiplica por un múltiplo de 10 correspondiente a la división que se hizo en el transmisor y se guarda el dato en otra variable. En el código 3.9 se ve un ejemplo de la decodificación del dato de la altura de la radiosonda, como se puede apreciar, es exactamente el proceso inverso al descrito anteriormente.

Código 3.9: Extracto de función *decodePacket* (altura)

```

83     tmp = packet[8];
84     tmp = (tmp << 8) + packet[9];
85     tmp = (tmp << 8) + packet[10];
86     tmp = (tmp << 8) + packet[11];
87     alt_dbl = tmp/1000.0;

```

Todos los datos transmitidos por la radiosonda deben ser mostrados en el monitor, para que el usuario sepa en donde se encuentra la radiosonda en tiempo real. Esto se hace con una función llamada *displayInfo*, la cual imprime los datos diciendo a qué corresponden y con qué unidad de medida están. Con el objetivo de determinar a qué distancia se encuentra la radiosonda y cuál es su azimut y su elevación con respecto al receptor, se crean funciones adicionales que hacen estos cálculos.

En el código 3.10 se muestra un ejemplo real de 3 datos obtenidos por la radiosonda. En las pruebas los datos fueron desplegados en pantalla completa, por lo que cada set de datos alcanzaba a estar en una sola línea. Estos consisten en la ubicación (latitud, longitud), la altura en metros, la distancia con respecto al receptor en metros, el curso en grados, la velocidad en metros por segundo, el tiempo, la cantidad de satélites de los que obtiene la información el GPS y por último, el azimut y la elevación en grados.

Código 3.10: Ejemplo de los datos impresos en el monitor

```

1  Loc: -33.379680, -70.593876   Alt: 6656.3 [m]   Dist: 12373.18 [m]   Course: 48.50
   Speed: 15.33 [mps]   Time: 16:54:55   Sat: 10   Az/El: 40.9/29.5
2  Loc: -33.376725, -70.589653   Alt: 6656.3 [m]   Dist: 12807.60 [m]   Course: 53.56
   Speed: 15.75 [mps]   Time: 16:55:28   Sat: 10   Az/El: 41.6/28.4
3  Loc: -33.375436, -70.587825   Alt: 6775.8 [m]   Dist: 13054.69 [m]   Course: 45.14
   Speed: 16.07 [mps]   Time: 16:55:42   Sat: 10   Az/El: 41.8/28.5

```

Esta parte de la prueba fue muy importante en la memoria, ya que permitió obtener una buena aproximación de como trabajar con la transmisión de la telemetría de futuros femtosatélites. El código de esta prueba está disponible en anexos y puede ser de utilidad para los próximos satélites desarrollados en la Universidad de Chile.

El azimut se obtuvo con el arcotangente de la división entre las diferencias de longitudes y las diferencias de latitudes del transmisor y del receptor. Este cálculo no es muy

exacto, pero el error máximo que se obtuvo fue menor a $4,7^\circ$. El arcotangente común no puede diferenciar el caso cuando el numerador y el denominador son ambos positivos o negativos, es por esto que se utilizó la función *atan2*, la cual recibe dos argumentos y por lo tanto considera los signos para entregar un resultado correcto entre $-\pi$ y π . En arduino se crea la función *azimuth*, la cual realiza esta operación y transforma el resultados de radianes a grados en un rango entre 0° y 360° .

Código 3.11: Función *azimuth*

```

167 double azimuth(double lat2, double lon2) {
168     dlat = (lat2 - stationsLat) * deg2rad;
169     dlon = (lon2 - stationsLon) * deg2rad;
170     az = atan2(dlon, dlat) * rad2deg;
171     if(az < 0) {
172         az = 360 + az;
173     }
174     return az;
175 }
```

La elevación en grados se obtiene del arcotangente de la división entre la diferencia de altitud entre el transmisor y el receptor, y la distancia en la superficie entre emisor y receptor (código 3.12).

Código 3.12: Función *elevation*

```

185 double elevation(double lat2, double lon2, double alt) {
186     gnd_dist = coord_dist(lat2, lon2);
187     return atan((alt - stationsAlt) / gnd_dist) * rad2deg;
188 }
```

Para calcular la distancia en la superficie de la Tierra entre la radiosonda y los receptores, se utilizó la fórmula de haversine, la cual permite obtener la distancia entre dos puntos en una esfera mediante las coordenadas de latitud y longitud. Es por esto que se hace una aproximación, ya que la tierra en realidad tiene una forma achatada, pero para las distancias que se están manejando, los errores son despreciables.

La fórmula de haversine se puede ver en la ecuación 3.1, donde d es la distancia que se desea calcular, R es el radio de la esfera, φ_1 es la latitud del punto 1, φ_2 la latitud del punto 2 y $\Delta\lambda$ es la diferencia de longitudes. En este caso se utiliza el radio promedio de la Tierra (6.372 km).

$$\text{haversin}\left(\frac{d}{R}\right) = \text{haversin}(\varphi_1 - \varphi_2) + \cos(\varphi_1) \cos(\varphi_2) \text{haversin}(\Delta\lambda) \quad (3.1)$$

Luego, los receptores calculan esta distancia utilizando las coordenadas recibidas de la radiosonda y las coordenadas del receptor fijadas al inicio del programa. Este cálculo es realizado por cada receptor individualmente utilizando una función llamada *coord_dist* la cual se puede apreciar en el código 3.13. Para ahorrar tiempo de cómputo, se guarda en

variables algunas operaciones recurrentes como el coseno de la latitud del receptor, el cual se calcula una sola vez en el *setup*. Los datos del receptor como la latitud, longitud y la altura son definidos utilizando *seters*.

Código 3.13: Función *coord_dist*

```
198 double coord_dist(double lat2, double lon2) {
199     dlat = (lat2 - stationsLat) * deg2rad;
200     dlon = (lon2 - stationsLon) * deg2rad;
201     haversin_dlat = sin(dlat / 2.0);
202     haversin_dlon = sin(dlon / 2.0);
203     a = haversin_dlat * haversin_dlat + cos_stationsLat *
cos(lat2 * deg2rad) * haversin_dlon * haversin_dlon;
204     c = 2 * atan2(sqrt(a), sqrt(1 - a));
205     d = 6371 * c;
206     return d * 1000;
207 }
```

Para calcular la distancia considerando la altura, se crea una función llamada *realDistance* (Código 3.14). En este caso se utiliza pitágoras y se considera la altura a la que están los receptores como la referencia para obtener la altura del globo sonda.

Código 3.14: Función *realDistance*

```
217 double realDistance(double lat2, double lon2, double alt) {
218     gnd_dist = coord_dist(lat2, lon2);
219     z_dist = alt - stationsAlt;
220     return sqrt((z_dist * z_dist) + (gnd_dist * gnd_dist));
221 }
```

Los receptores tienen muchas funciones en común, por lo que se crea una librería para evitar el código duplicado y así tener un programa más ordenado y fácil de entender para futuros usuarios. Esta librería se llama *RX_GPS* y se puede encontrar en anexos, junto con el resto del código utilizado en todas las pruebas.

3.2.5. Montaje de la radiosonda y los receptores

Existieron dos lanzamientos de radiosondas. En el primero se perdió la comunicación al momento de lanzar el globo, probablemente debido a que los componentes no estaban suficientemente firmes en su posición y se desconectó la batería. Esto fue considerado en el diseño de la segunda radiosonda, la cual se describe en esta subsección.

La estructura de la radiosonda está compuesta de un material de espuma utilizado para proteger componentes dentro de las cajas. A este armazón se le hicieron dos orificios, uno para la antena y otro para que el GPS entre justo y no se pueda mover de su posición. La antena del transceptor va en la cara inferior de la radiosonda para mejorar la transmisión de datos, debido a esto el GPS va en la misma posición, pero no se afecta mayormente.

Entre el GPS y la placa adaptadora se inserta un cubo del material de espuma para fijarlo en su posición y para presionarlo hacia la placa utilizando huincha aislante. Además de toda la electrónica, se agrega una bolsa de un producto que absorbe la humedad, pues el rápido ascenso de un globo sonda produce cambios de temperatura que generan condensación, la cual es perjudicial para los componentes electrónicos.

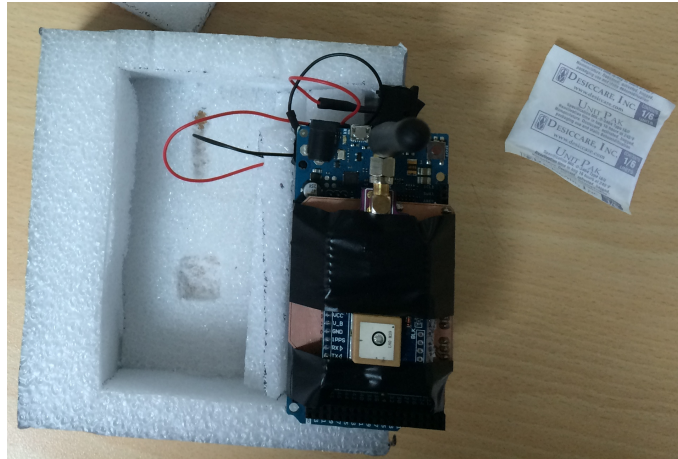


Figura 3.13: Componentes de la radiosonda

Los cables de la batería se sueldan directamente a la placa del Arduino para evitar que se desconecten en medio del vuelo. Una vez insertados el Arduino con sus componentes y la batería, se agregan piezas del material de espuma de una forma tal que impida el movimiento de los componentes al interior de la radiosonda. El proceso de armado se encuentra en la figura 3.14, en donde se puede apreciar en la izquierda que la bolsa anti-humedad se encuentra justo frente al Arduino. En la imagen de al medio se ve que se inserta material de espuma sobre el Arduino para evitar el movimiento en el sentido vertical y en la imagen de la derecha se ve como queda cubierta completamente la radiosonda.

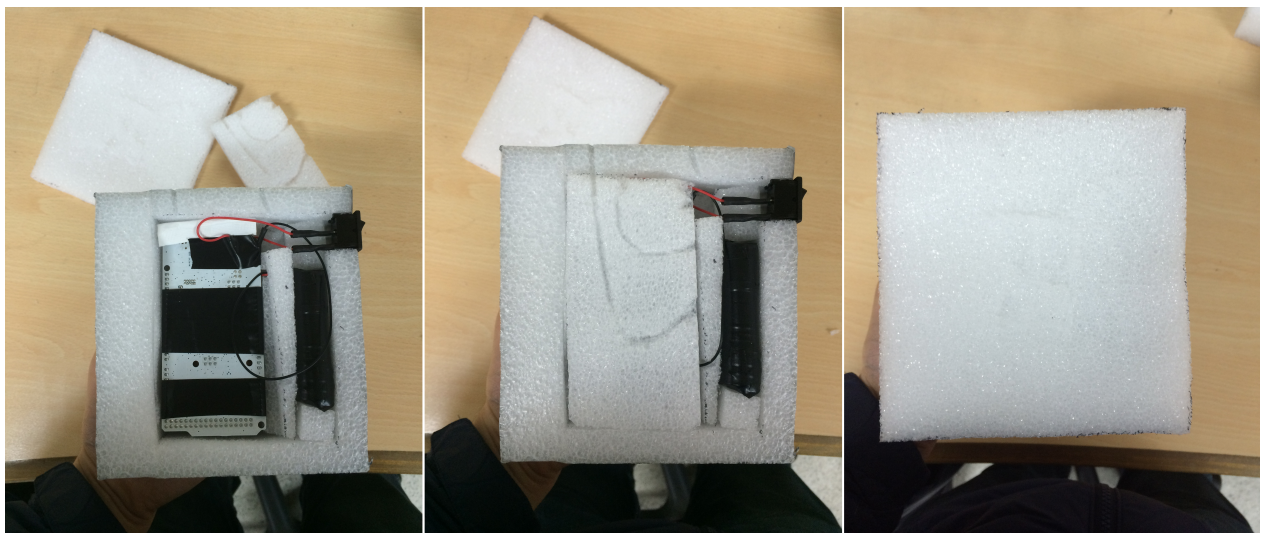


Figura 3.14: Armado de la radiosonda

Luego de finalizar el armado, se procedió a reforzar la estructura utilizando *duct tape*, la cual es más fuerte y resistente a la tracción y a la presión que otras cintas. En la figura 3.15 se ve un instante en medio del proceso de reforzamiento de la estructura. Los únicos materiales expuestos fuera del armazón son la antena y el *switch* de encendido/apagado.

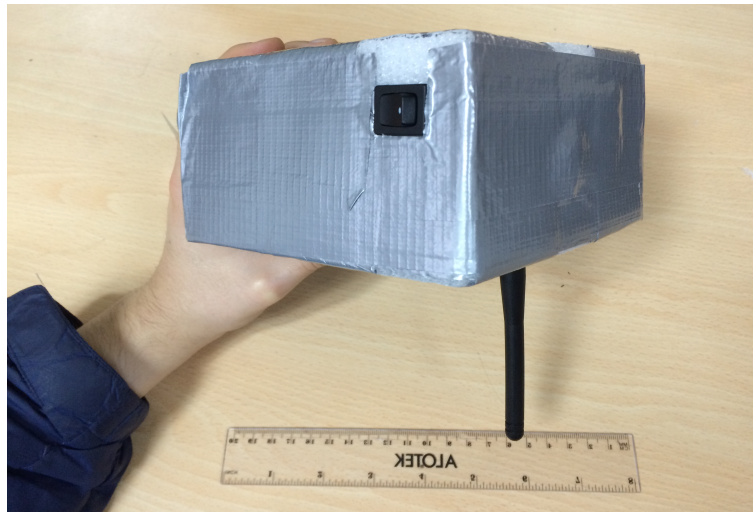


Figura 3.15: Reforzamiento de la radiosonda

Finalmente, se agregó una asa formada por *duct tape* en la cara superior de la radiosonda, para luego poder amarrar el globo a la estructura.

Para el desarrollo de esta prueba se dispusieron tres antenas receptoras: La antena Yagi de la estación terrena, la antena omnidireccional (monopolo) que se utilizó en el montaje para las pruebas en terreno y una antena Yagi portátil. En la figura 3.16 se encuentra la antena de la estación terrena encerrada en un rectángulo rojo.



Figura 3.16: Antena yagi de la estación terrena

La antena tiene un cable que llega hasta el quinto piso del Departamento de Ingeniería Eléctrica. En ese lugar se conecta el transceptor en reemplazo de la radio con pantalla verde que se ve en la figura 3.17 utilizando un adaptador de N a SMA.



Figura 3.17: Setup de la estación terrena

En la figura 3.18 está la antena omnidireccional encerrada en un ciculo rojo, montada sobre uno de los atriles y la antena Yagi portátil encerrada en un rectángulo rojo. Ambas antenas conectadas a transceptores RFM22B, los cuales a su vez se conectan a dos *notebooks*. Estas dos antenas se ubican en la azotea del Departamento de Ingeniería eléctrica a 550 metros de altura.



Figura 3.18: Antena omnidireccional y antena Yagi

Capítulo 4

Evaluación y análisis de resultados

4.1. Pruebas en Parque O'Higgins

Las pruebas realizadas en Parque O'Higgins consisten en enviar 1000 paquetes de 255 bytes cada uno, desde un emisor a un receptor. Luego el receptor reenvía el mensaje y el emisor lo compara con el paquete enviado inicialmente. Si el mensaje recibido es idéntico al enviado, se considera una transmisión exitosa. El emisor puede enviar el mensaje hasta 3 veces antes de pasar al siguiente mensaje en caso de no recibir la confirmación de parte del receptor.

Estas mismas pruebas se efectúan cambiando el tipo de modulación, la tasa de transmisión de datos y la distancia emisor-receptor.

Para tener un buen enlace de comunicación se debe tener en consideración la zona de Fresnel, la cual indica una serie de regiones circulares, las cuales si tienen un obstáculo entremedio, puede provocar desfases en la señal con la consecuente generación de interferencias destructivas. Como regla general, la obstrucción de la primera zona de Fresnel no debe exceder el 40 % para que el enlace se comporte como una limpia trayectoria en espacio libre.

Como se puede apreciar en la figura 4.1, se puede calcular la distancia r que corresponde a la región elipsoidal que debe estar libre de obstáculos. Para ello, se necesita conocer las distancias d_1 , d_2 y la longitud de onda de la señal que se está transmitiendo.

$$r = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}} \quad (4.1)$$

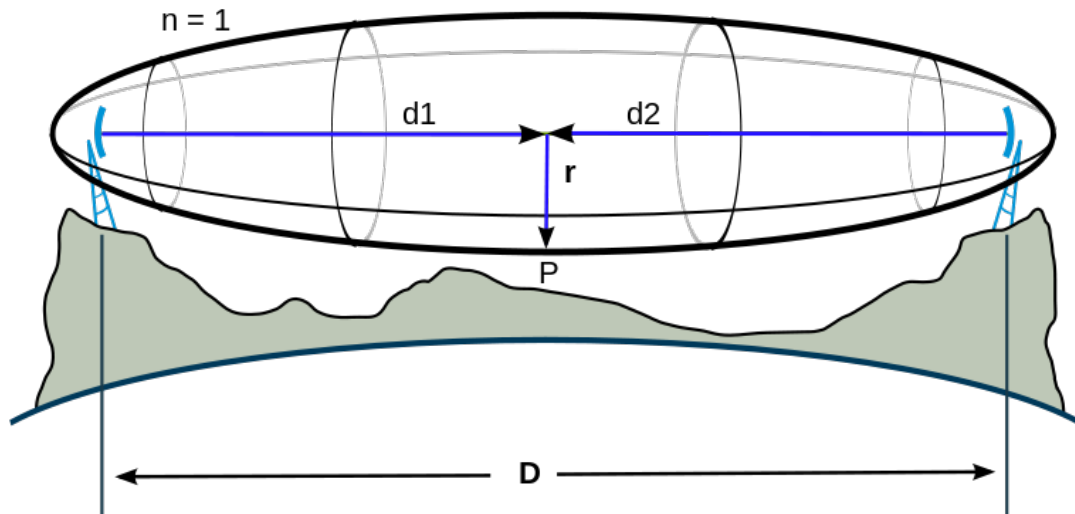


Figura 4.1: Zona de Fresnel (usada con permiso de la licencia *Creative Commons*)

En las pruebas efectuadas en el parque O'Higgins, se utilizan dos atriles de música para elevar del nivel del suelo al transmisor y al receptor. Uno mide 3,15 m y el otro 2,15 m, lo cual permite una altura a distancia media de 2,65 m. Esta altura no será suficiente para todas las pruebas, sin embargo permitirá establecer cuales son las mejores configuraciones para luego poder hacer una prueba final con las que den mejores resultados.



Figura 4.2: Obtención de datos del transceptor al notebook

A continuación, se muestra una tabla con los distintos radios de la primera zona de Fresnel a mitad de rango en distintas distancias y el radio mínimo recomendado. Se puede ver que en las distancias de 150 y 300 m, la obstrucción de la primera zona de Fresnel es mayor a la recomendada. A pesar de esto, se espera seleccionar cuál es la mejor configuración para luego hacer pruebas a mayores distancias, utilizando para ello atenuadores de potencia, y cumpliendo con el requisito dado por la zona de Fresnel.

Distancia (m)	Zona de Fresnel (m)	Radio recomendado (m)	Porcentaje logrado
75	3,63	2,18	76,06%
150	5,10	3,06	52,01%
300	7,21	4,32	36,78%

Tabla 4.1: Radios de zona de Fresnel

En las siguientes subsecciones se comparan los resultados obtenidos con distintas configuraciones en las pruebas realizadas fuera del laboratorio.

Para las modulaciones FSK y GFSK, se muestra en los gráficos la desviación en frecuencia (Fd) que es la distancia en hertz entre la frecuencia central y las frecuencias de las señales que representan el 1 y el 0 binarios, mientras que para OOK, se muestra el ancho de banda del receptor (Bw). En cada subsección se muestran dos gráficos, uno mostrando la cantidad de mensajes válidos recibidos, es decir, que pasaron la verificación por CRC, y otro mostrando el tiempo empleado para enviar 1000 mensajes en total, independiente de si son recibidos o no. Este tiempo incluye la transmisión del TX y la retransmisión del RX. Además, en los gráficos del tiempo se incluye el porcentaje de paquetes válidos recibidos. Estos resultados se muestran a una misma tasa de transmisión de datos, para las distintas modulaciones disponibles¹.

4.1.1. Resultados a 125 kb/s

En la figura 4.3 se ven los resultados obtenidos con la mayor tasa de bits por segundo en las modulaciones FSK y GFSK. Como se puede apreciar, todos los datos se reciben exitosamente a 75 m de distancia, sin embargo, a mayores distancias se pierden todos los paquetes enviados. Esto se puede deber a que no se está cumpliendo con el requisito mínimo de la zona de Fresnel. A pesar de ello, hay mejores resultados a menores tasas de transmisión, lo cual advierte no ser una buena opción para aplicaciones en que se requiere trabajar a grandes distancias.

¹En algunos casos no están los tres tipos de modulaciones implementadas en el transceptor a una misma tasa de kb/s.

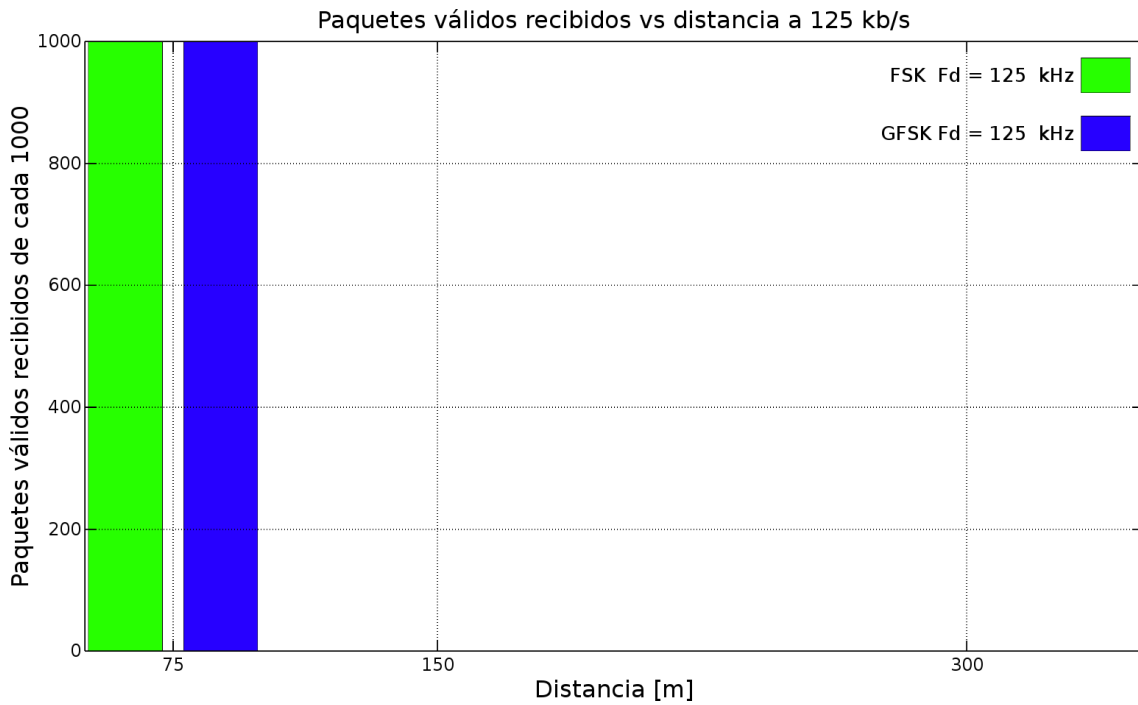


Figura 4.3: Datos recibidos correctamente a 125 kb/s

En la figura 4.4 se ve que el tiempo requerido para enviar 1000 paquetes usando GFSK es de 1 minuto y 9 segundos, mientras que con FSK tarda solamente 43 segundos en enviar y recibir exitosamente 1000 mensajes de 1000. En las distancias de 75 y 150 m no se reciben paquetes, por lo que el tiempo no aporta información.

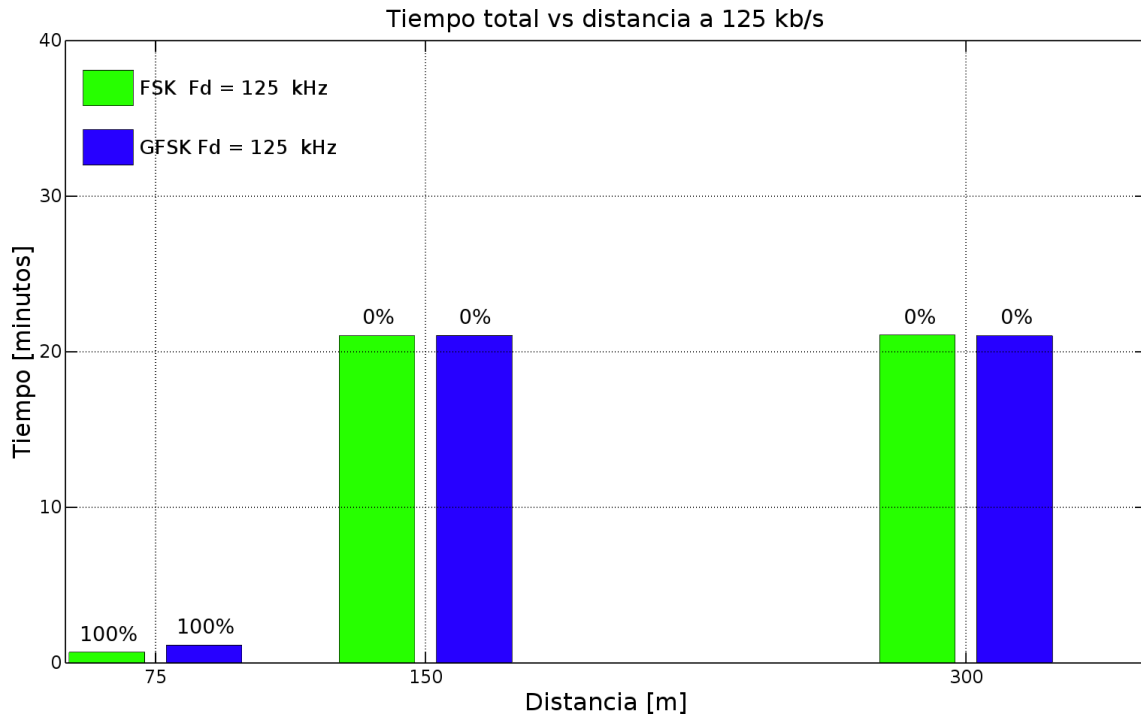


Figura 4.4: Tiempo utilizado para enviar 1000 paquetes a 125 kb/s

4.1.2. Resultados a 38,4 kb/s

Al disminuir la tasa de transmisión de datos a 38.4 kb/s, se logra obtener la totalidad de los paquetes a 150 m de distancia con FSK y GFSK e incluso 57 paquetes a 300 m en el caso de la modulación FSK. Cuando se cambia a una modulación OOK, llegan solamente 99 paquetes de 1000 a 75 m y 0 datos a mayores distancias.

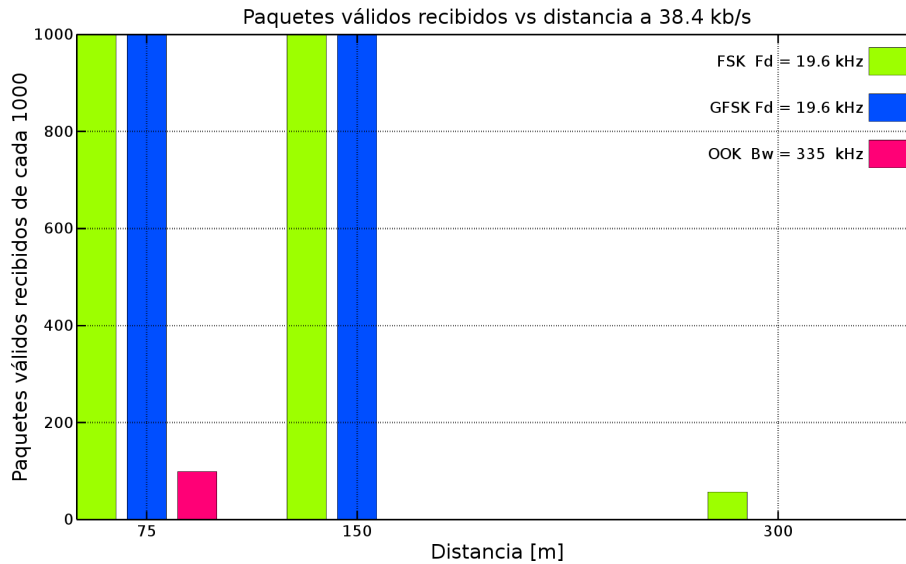


Figura 4.5: Datos recibidos correctamente a 38.4 kb/s

Las modulaciones FSK y GFSK tardan alrededor de 2 minutos en enviar 1000 paquetes a 75 y 150 m, pero al fallar a 300 m, se demora más de 22 minutos en intentar enviar datos, esto debido a que aumenta la cantidad de reintentos. Lo mismo ocurre con OOK al pasar de 75 a 150 m, como se puede ver en la figura 4.6.

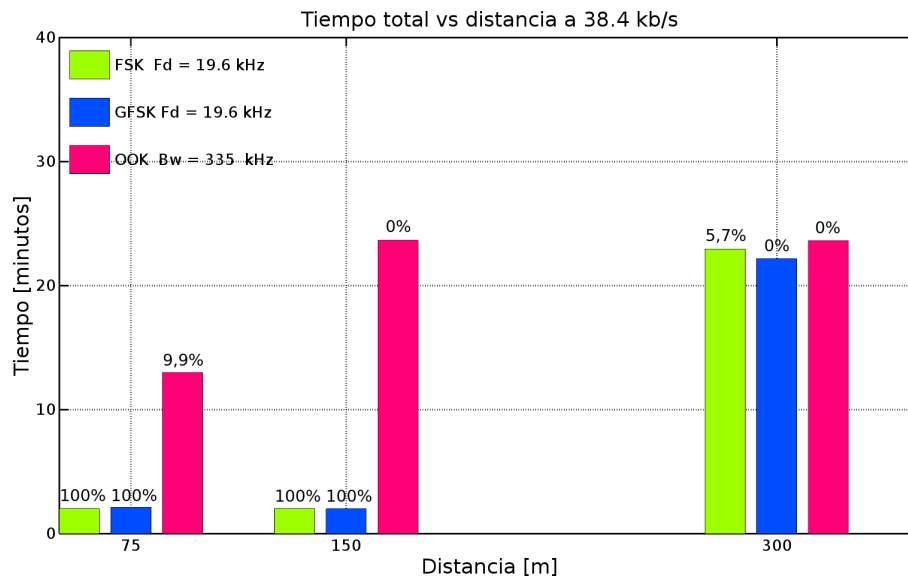


Figura 4.6: Tiempo utilizado para enviar 1000 paquetes a 38.4 kb/s

4.1.3. Resultados a 19,2 kb/s

Al disminuir a la mitad la tasa de transmisión de datos anterior, mejora el desempeño de OOK a 75 m y de FSK a 150 y 300 m, recibiendo 1000 y 448 paquetes respectivamente.

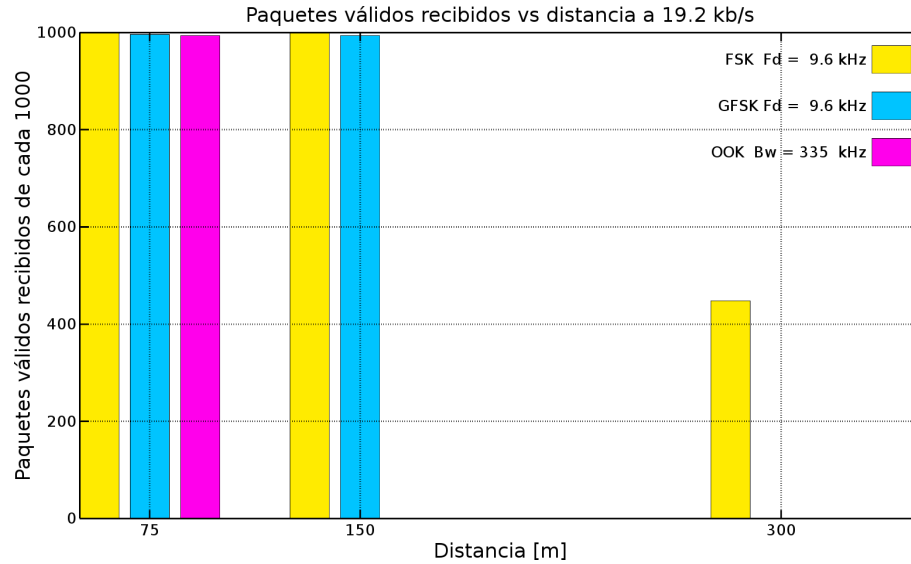


Figura 4.7: Datos recibidos correctamente a 19.2 kb/s

En la figura 4.8 se puede apreciar que el tiempo para OOK a 75 m disminuyó con respecto al gráfico anterior, esto debido a que se recibieron más paquetes que a 38.4 kb/s, lo cual disminuye la cantidad de retransmisiones necesarias para enviar 1000 paquetes. En el caso de FSK a 300 m, el tiempo disminuye en 57 segundos debido a la misma razón.

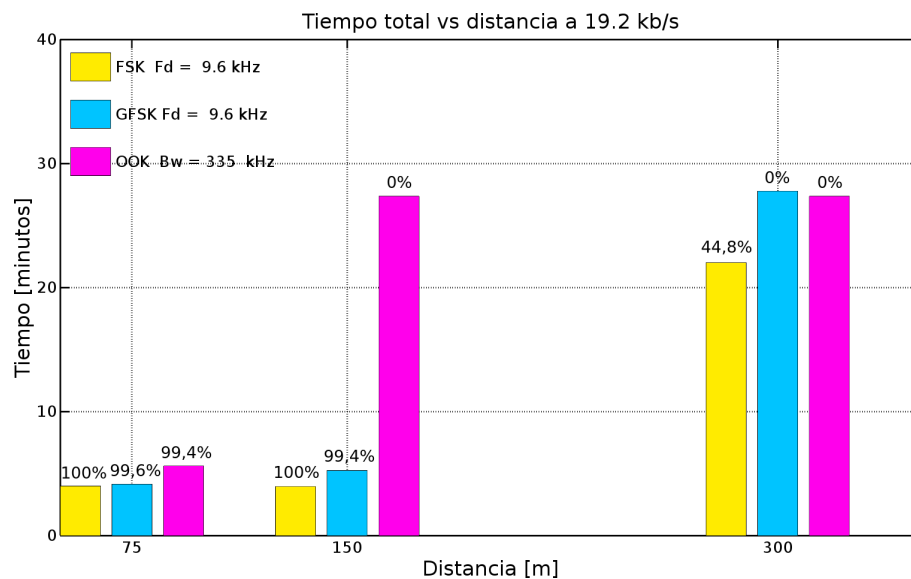


Figura 4.8: Tiempo utilizado para enviar 1000 paquetes a 19.2 kb/s

4.1.4. Resultados a 2,4 kb/s

A esta tasa de transmisión de datos se obtiene una mejora considerable a 300 m de distancia, es más, para los 3 tipos de modulación disponibles se obtiene sobre el 94 % de los mensajes enviados. Llama la atención que en este caso la modulación OOK obtuvo el mejor desempeño recibiendo un total de 978 de los 1000 mensajes enviados. También GFSK tuvo un rendimiento superior a FSK recibiendo 964 paquetes, 16 más que FSK.

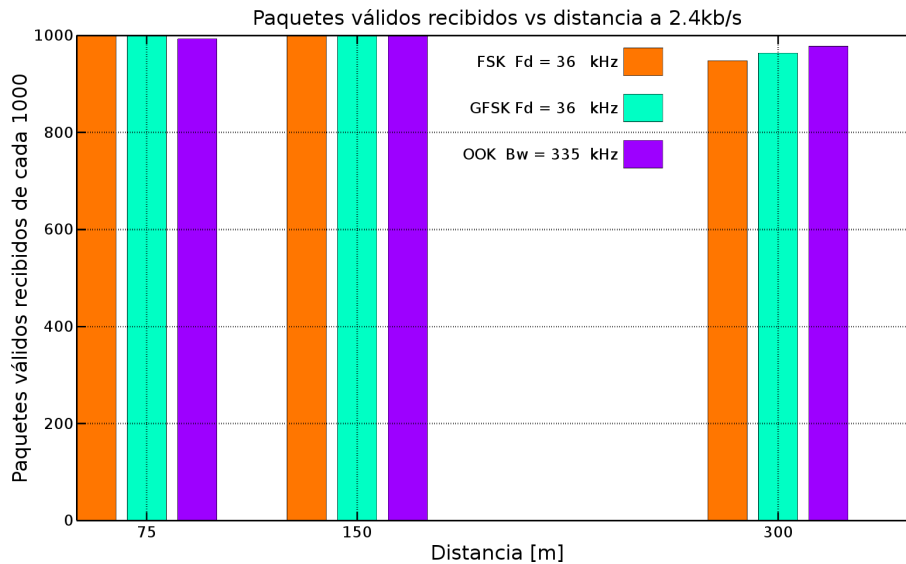


Figura 4.9: Datos recibidos correctamente a 2.4 kb/s

Como era de esperar, el tiempo necesario para enviar los 1000 paquetes de datos aumentó también. Para la distancia de 300 m, todas las modulaciones tardaron más de 32 minutos en hacer las pruebas.

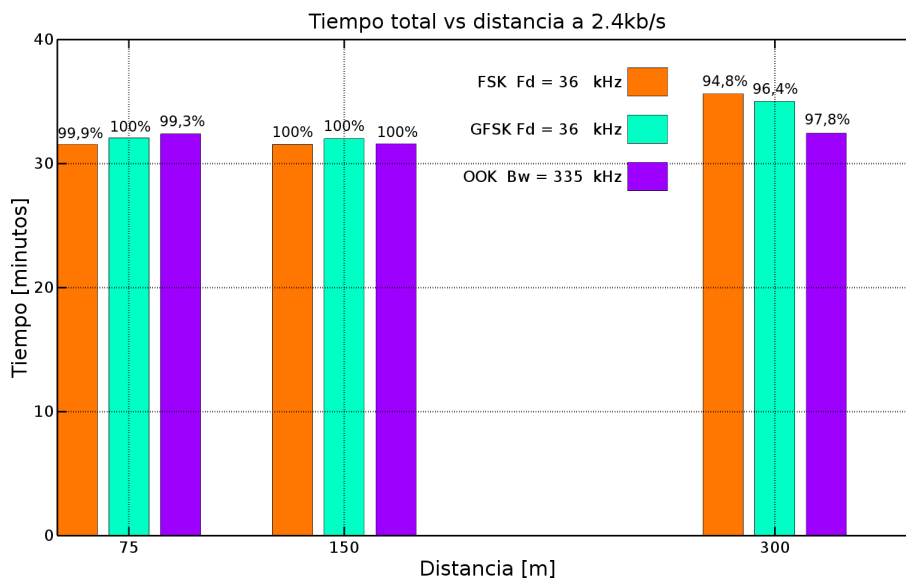


Figura 4.10: Tiempo utilizado para enviar 1000 paquetes a 2.4 kb/s

4.1.5. Resultados a 2 kb/s

Al disminuir aún más la tasa de transmisión de datos hasta el mínimo disponible, se observa que a 300 m se recibe la totalidad de los mensajes enviados tanto para FSK como para GFSK. No se hace una prueba con la modulación OOK debido a que no está la opción disponible en la librería utilizada.

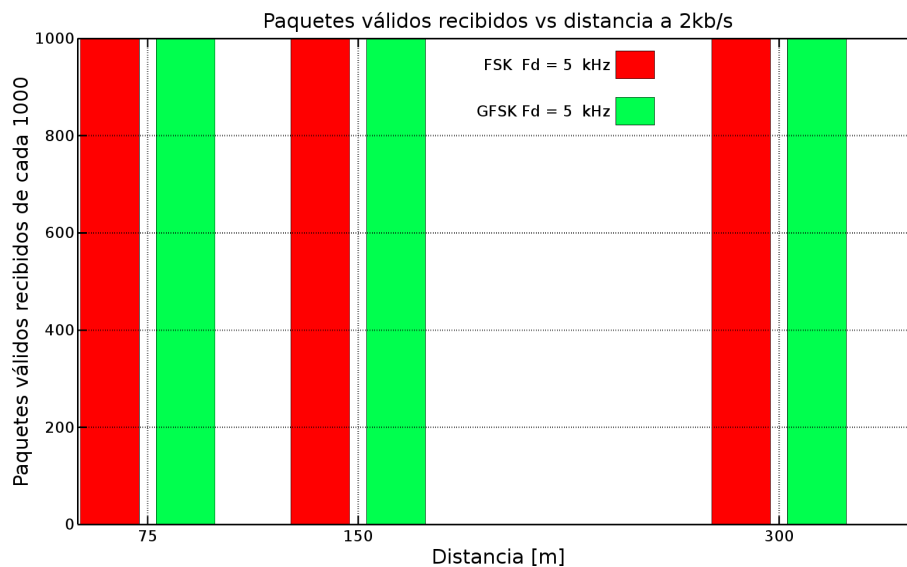


Figura 4.11: Datos recibidos correctamente a 2.0 kb/s

Esta mejora en el desempeño de ambas modulaciones viene con la desventaja de un aumento en el tiempo total de las pruebas, siendo de 37 minutos con 50 segundos de duración para la modulación FSK y de 37 minutos con 53 segundos para GFSK.

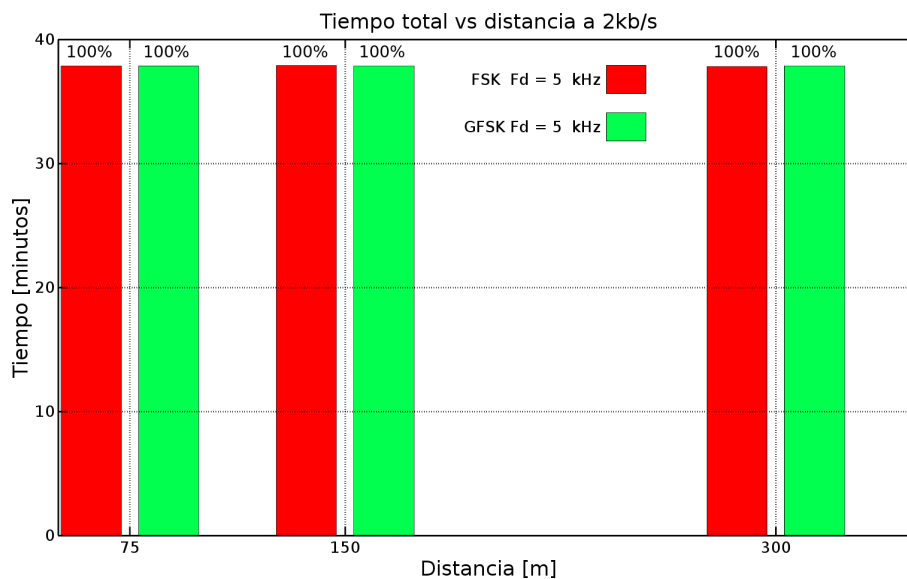


Figura 4.12: Tiempo utilizado para enviar 1000 paquetes a 2.0 kb/s

4.1.6. Resultados generales

En general los resultados fueron precarios a 300 m, siendo que es una distancia mínima en comparación con los cientos de kilómetros a los que orbitan los CubeSats, sin embargo, se debe considerar que no se está cumpliendo el requisito mínimo de la zona de Fresnel, ya que hay más de un 60 % del radio de Fresnel obstruido por el suelo. Esto podría provocar que hayan señales que cambien su fase al rebotar con los obstáculos y generen interferencias destructivas, disminuyendo la potencia de la señal recibida. A pesar de ello, se toma como supuesto que todas las configuraciones se ven afectadas de igual manera por este hecho, y se decide por escoger las que tienen un mejor desempeño para hacer nuevas pruebas cumpliendo el mínimo requerido.

Viendo la figura 4.13, la tendencia observada en todos los resultados obtenidos indica claramente que al disminuir la tasa de transmisión de datos, se obtiene un mejor rendimiento a mayores distancias. Esto se debe a que al minimizar la tasa de datos transmitidos, el ancho de banda se hace cada vez más angosto, lo cual mejora la sensibilidad del receptor [14][15], aumentando considerablemente la cantidad de paquetes recibidos de manera correcta.

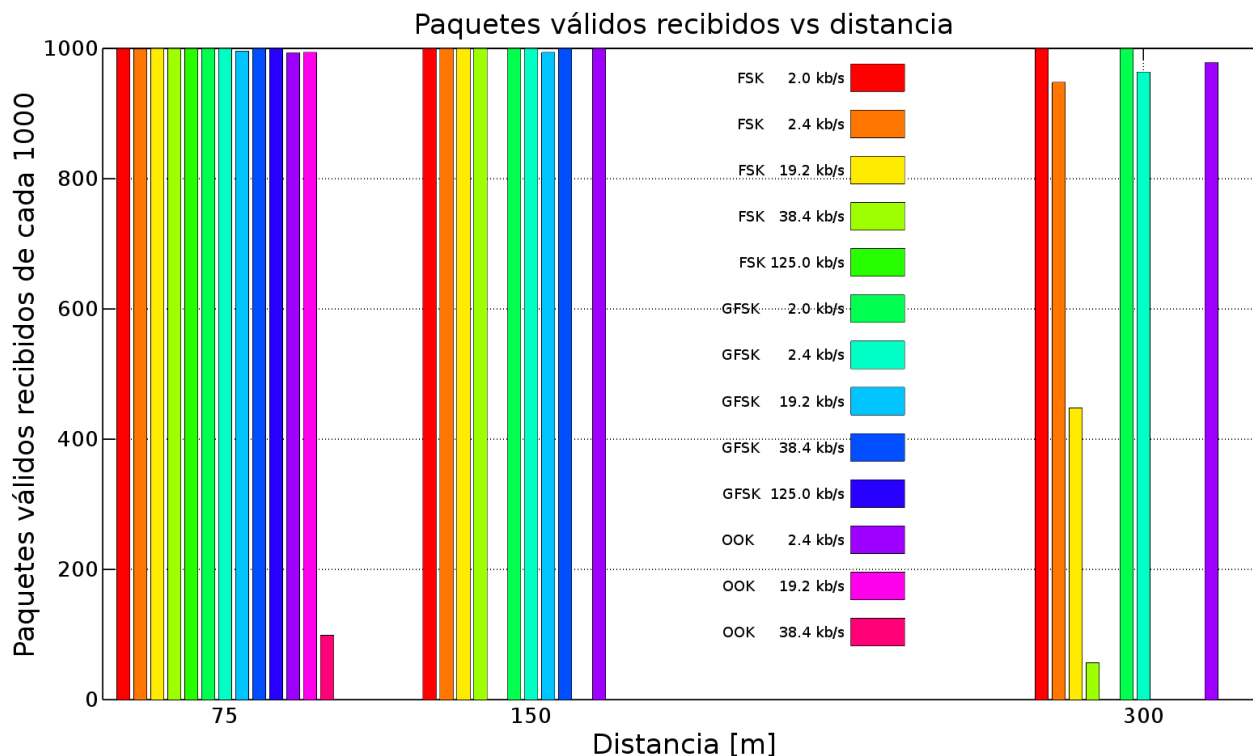


Figura 4.13: Datos recibidos correctamente

No es muy extraño disminuir la tasa de transmisión de datos para lograr mayores rangos. La sonda espacial Voyager 1 fue lanzada en 1977 con la misión de visitar Júpiter y Saturno, pero luego de cumplirla, se le ha dado el objetivo de explorar las regiones y los

límites de las afueras de la heliósfera. El satélite ha seguido funcionando hasta el día de hoy y es actualmente el primer objeto hecho por el ser humano en entrar al espacio interestelar y estudiar el medio interestelar. Debido a la ley del cuadrado inverso en comunicaciones por radio, la tasa de datos digitales del Voyager ha disminuido continuamente a medida que se va alejando de la Tierra. Por ejemplo, la tasa de datos usada en Júpiter era de 115,2 kb/s y la usada en Saturno habría sido de aproximadamente 29 kb/s, si no fuera porque se mejoraron las antenas de la estación terrena. Aún con la mejora, la tasa de datos fue de 44,8 kb/s [16].

La mejora que se obtiene en rango, se ve comprometida con el tiempo requerido para transmitir la misma cantidad de paquetes, lo cual se puede apreciar en la figura 4.14, en donde se ve el claro aumento de tiempo. No obstante, se debe tomar en cuenta que todas estas pruebas han sido efectuadas a modo de eco, ya que el receptor reenvía el mensaje al transmisor, y luego éste lo compara con el que había enviado. Debido a esto, el tiempo real en transmitir 1000 paquetes es aproximadamente la mitad del tiempo que se ve en los gráficos aquí presentes.

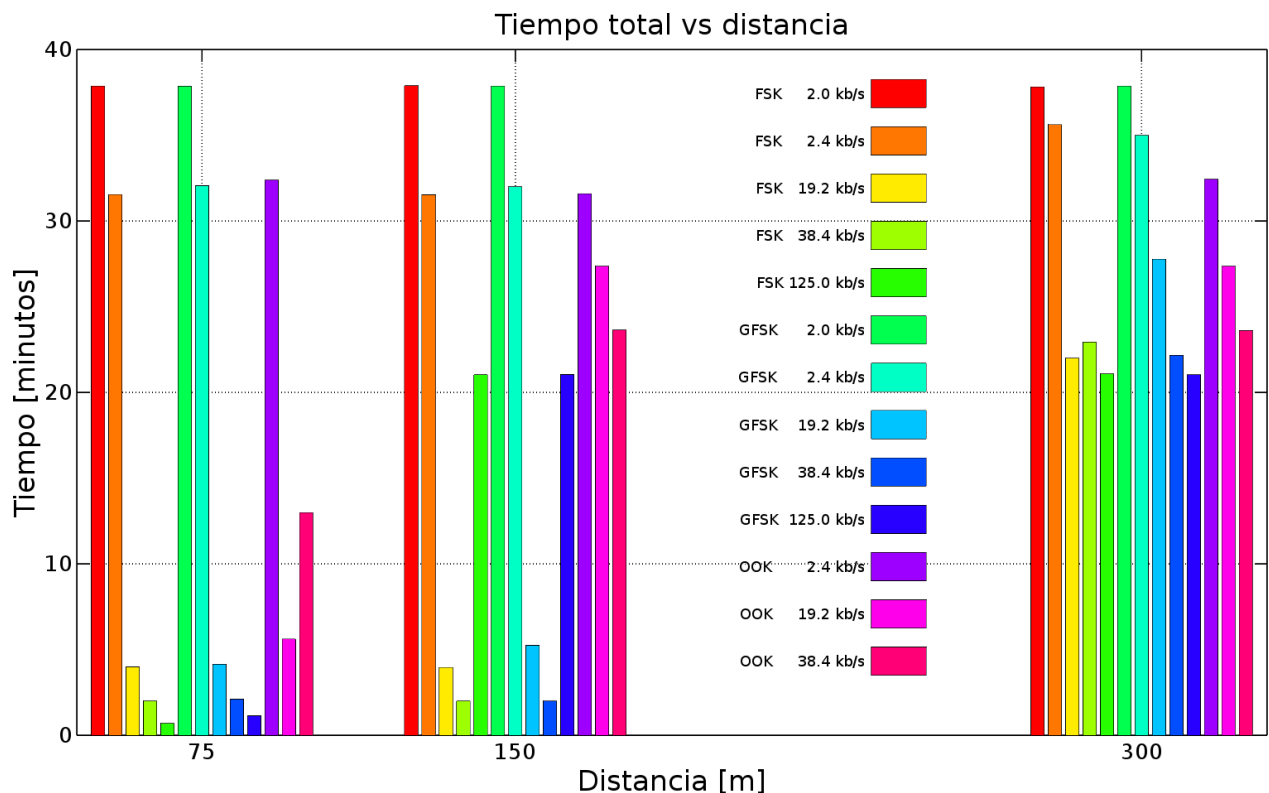


Figura 4.14: Tiempo utilizado para enviar 1000 paquetes

En general, las modulaciones FSK y GFSK tuvieron un mejor desempeño que OOK, siendo capaces de transmitir la mayor parte de los paquetes a una distancia en la cual OOK no lograba obtener un solo acierto. Además, la librería no permite disminuir la tasa de transmisión de datos a 2 kb/s usando OOK. Debido a esto y a su bajo desempeño en este dispositivo, se descarta este tipo de modulación para futuras pruebas.

4.2. Pruebas a mayor distancia

Debido a los resultados anteriores, se decide continuar con las pruebas utilizando las dos mejores configuraciones a mayores distancias. A causa de la limitación en espacio, se utiliza un atenuador de potencia de 30 dB para poder simular un rango mayor. Para la modulación FSK se logra transmitir exitosamente a una distancia real de 150 m entre emisor y receptor, mientras que para la modulación GFSK se obtienen 75 m de distancia máxima antes de empezar a perder paquetes, ambas a una tasa de transferencia de 2 kb/s.



Figura 4.15: Atenuadores de potencia de 30 dB

Modulación	Paquetes recibidos	Tiempo (MM:SS.00)	Distancia (m)
FSK	1000	40:27.09	150
GFSK	1000	40:31.35	75

Tabla 4.2: Pruebas a 2.0 kb/s usando un atenuador de 30 dB

Para calcular a qué distancia corresponde esta prueba, se utiliza la fórmula de pérdidas de espacio libre 2.4, para transformar la distancia en la equivalente atenuación en decibeles. Reemplazando $f = 4,33 \cdot 10^8$ Hz, $c = 3 \cdot 10^8$ m/s y la distancia d de acuerdo al caso, se pueden obtener los valores de la atenuación en decibeles de la distancia real entre emisor y receptor. En el caso de FSK a 150 m, se tiene una atenuación de 68,69 dB, mientras que GFSK a 75 m, tiene una atenuación de 62,67 dB. Luego, sumando estos resultados a los 30 dB del atenuador de potencia, se puede calcular la distancia equivalente utilizando la fórmula ya mencionada.

Modulación	Atenuación total (dB)	Distancia efectiva (km)
FSK	98,69	4,74
GFSK	92,67	2,37

Tabla 4.3: Resultado efectivo considerando la atenuación

Viendo los resultados anteriores, se decide hacer una prueba con la modulación FSK, ya que fue la que dio un mejor desempeño en comparación a GFSK. Para justificar la utilidad de una radiosonda para ver el rango máximo del transceptor, se intentó establecer un enlace de comunicación entre el Cerro San Cristóbal y la estación terrena del Departamento de Ingeniería Eléctrica, pero esto no fue posible debido a la existencia de edificios que obstruían la línea de vista entre ambos puntos. Debido a esto, se hizo una búsqueda de cerros cercanos a la facultad que no tengan el mismo problema que el Cerro San Cristóbal.

El lugar escogido es el cerro La Pirámide, ubicado en el Parque Metropolitano de Santiago. Este cerro, de tamaño medio, permite optimizar el tiempo de ascenso y descenso, pero sigue teniendo una altura que logra cumplir las restricciones de la zona de Fresnel. En la figura 4.16 se puede ver la distancia a la que se pudo hacer la prueba. Hay varios puntos desde donde mandar datos, sin embargo existen múltiples obstáculos como torres de alta tensión, árboles muy grandes que no se ven en los mapas y los mismos cables de transmisión de energía, por lo que se utiliza un punto a una distancia de 500 m del receptor.



Figura 4.16: Prueba en cerro La Pirámide

El terreno escogido tiene línea de vista hacia el receptor ubicado sobre el tunel, aún así hay varios arbustos presentes y lugares donde la señal podría reflejarse. El emisor está situado a 760 m sobre el nivel del mar y se puede ver en la imagen 4.17, mientras que el receptor está a 751 m de altura. El obstáculo más evidente sin considerar los arbustos, es la autopista nororiente, la cuál está a 706 m de altura, por lo cual existe por lo menos

una distancia de 45 m desde la línea de vista hasta la autopista. Con estos datos se puede calcular si se cumple el criterio de no obstaculizar la zona de Fresnel a mitad de la distancia entre el emisor y el receptor (tabla 4.4).

Zona de Fresnel	Radio (m)	Porcentaje logrado
n = 1	9.3	100 %
n = 2	13.2	100 %
n = 3	16.1	100 %

Tabla 4.4: Zonas de Fresnel en cerro La Pirámide (500 m)

En la tabla solo se muestran los 3 primeros radios de la zona de Fresnel, sin embargo es claro que se cumple el criterio para valores mayores. Los resultados de este sitio en cuanto a la zona de Fresnel superan con creces a los vistos en Parque O'Higgins, debido a la altura alcanzada al estar los transeptores sobre los cerros.



Figura 4.17: Línea de vista cerro La Pirámide

Se efectúan pruebas con atenuadores de 30 dB a 500 m de distancia entre emisor y receptor, ya que era el lugar con menos obstáculos. Al usar un total de 60 dB de atenuación

no se logró establecer un enlace de comunicación, por lo que se usó solamente un atenuador de 30 dB recibiendo la totalidad de los paquetes. Los resultados de esta prueba se pueden ver en la tabla 4.5.

Modulación	Distancia real (m)	Atenuación total (dB)	Distancia efectiva (km)
FSK	500	109.15	15.8

Tabla 4.5: Distancia efectiva en cerro La Pirámide

Finalmente, se debe tener en consideración que a solo 200 m del lugar de pruebas había una torre con múltiples antenas de varios tipos. Esto pudo generar interferencias y alterar los resultados de las pruebas. La mayor parte de los cerros en Santiago son utilizados por antenas de telecomunicaciones, por lo que es difícil encontrar un lugar ideal para realizar pruebas de este tipo. Es por esto que se hace necesario utilizar un globo sonda para encontrar el alcance máximo del transceptor RFM22B.

4.3. Pruebas con la radiosonda

A continuación se evalúan los resultados de la prueba de la radiosonda con los tres receptores:

La antena yagi portátil utilizada en esta prueba estaba optimizada para la frecuencia de 890 MHz a 960 MHz, por lo que no estaba operando en un óptimo, sin embargo, permitió corroborar algunos cálculos teóricos. A esta antena se le conectó un atenuador de potencia de 30 dB y recibió datos hasta a 409 m, lo cual equivale a una distancia de 12,9 km aproximadamente. Luego de quitar el atenuador de potencia, el rango máximo obtenido fue de 13 km, lo cual demuestra que los cálculos realizados con respecto a las pérdidas de espacio libre son correctos.

La antena omnidireccional también tenía un atenuador de potencia de 30 dB. En esta condición logró captar paquetes hasta a 1231 m de distancia del transmisor, lo cual equivale a 38,9 km de distancia. Esta distancia sería muy útil para hacer experimentos de constelaciones de femtosatélites.

En el caso de la estación terrena, se utilizó una atenuación de 40 dB y logró captar paquetes a una distancia de 4773 m. Utilizando la ecuación de Friis, la distancia equivalente es de 477,3 km, lo cual permitiría establecer un enlace de comunicación con un satélite en la órbita de la estación espacial internacional (416 km en apogeo). La atenuación se disminuyó a menos de 5 dB y se logró captar un paquete a 14,6 km de distancia, pero luego no se lograron obtener más mensajes debido a la directividad de la antena y al movimiento impredecible del globo sonda.

La figura 4.18 muestra la trayectoria que tomó la radiosonda. Inicialmente el globo tuvo un rápido ascenso, pero después se movió más hacia el noreste debido al viento que viene del suroeste. Los colores de los puntos graficados muestran cuál de los receptores obtuvo un paquete a esta distancia. La mayor parte de los datos, sin considerar la antena Yagi sin atenuación, fueron recibidos por la estación terrena, a pesar de los 40 dB de atenuación debido a su alta ganancia. La última posición obtenida de la radiosonda fue en las cercanías del cerro Manquehue a más de 7 km de altura.

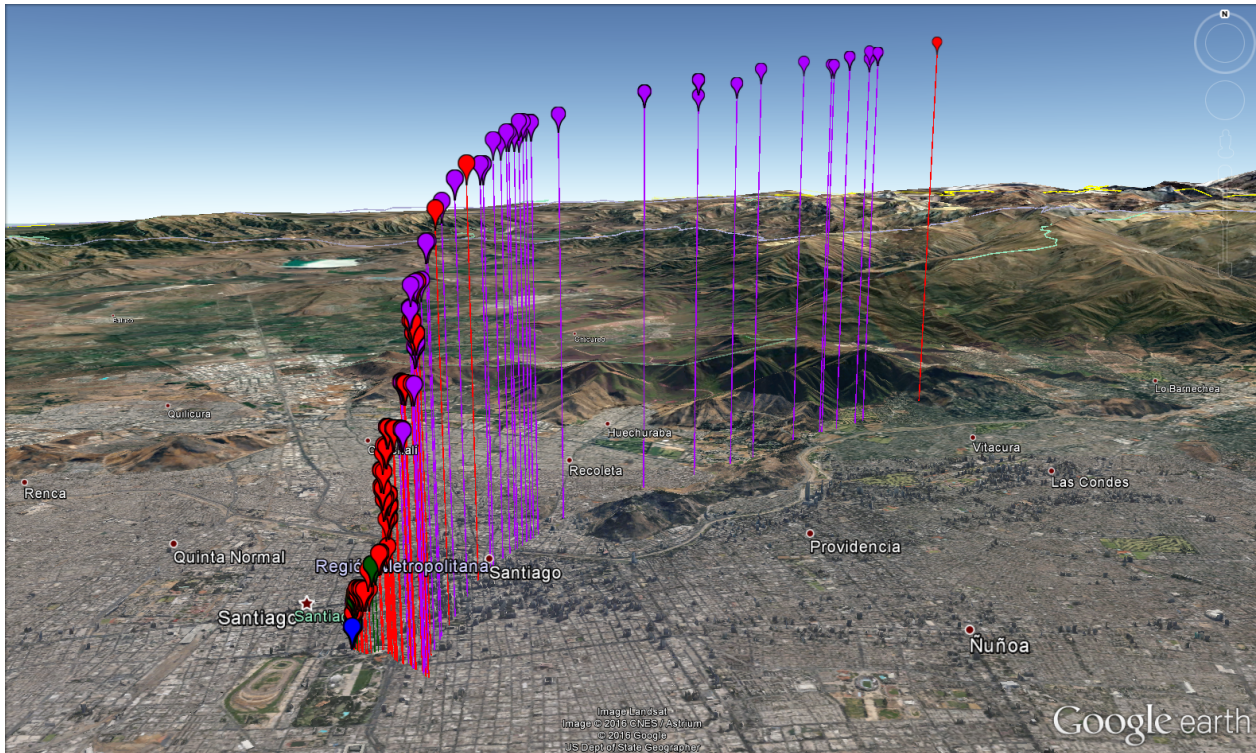


Figura 4.18: Trayectoria de la radiosonda en Google Earth. Los puntos azules representan datos obtenidos por la antena Yagi portátil con 30 dB de atenuación. Los verdes son los datos captados por la antena omnidireccional con 30 dB de atenuación. Morado la antena Yagi portátil sin atenuación. Rojo la estación terrena con la antena Yagi de 6 m y con 40 dB de atenuación. La distancia en tierra entre el inicio y la última posición captada fue de aproximadamente 11 km.

Una vez verificado que se logra un rango satisfactorio con este transeptor, se pasa a la etapa de ver cuántos paquetes válidos se recibieron del total. Para tiempos equivalentes en la recepción de datos con atenuación, el sistema de la antena Yagi de la estación terrena fue la que obtuvo la mejor tasa de recepción de paquetes. En la figura 4.19, se encuentra un gráfico con el porcentaje de los paquetes recibidos por la antena omnidireccional en rangos de 200 m reales, es decir, sin considerar la atenuación de 30 dB. La cantidad de paquetes recibidos disminuye con la distancia debido a que a mayor altura, el viento desplaza al globo a mayor velocidad, por ejemplo, en el primer minuto la radiosonda avanzó 67 m, mientras que en el último minuto dentro del rango captado por la antena omnidireccional, el globo avanzó más de 120 m.

En el gráfico se observa que en los primeros 200 m, el RFM22B recibe todos los paquetes. Entre 200 y 400 m recibe aproximadamente el 80 % de los mensajes transmitidos, lo cual equivale a una distancia entre 6,3 km y 12,6 km. Entre 400 y 600 m obtiene cerca del 18 % de los paquetes, probablemente debido a interferencias circunstanciales. Luego de eso, se recibe más del 45 % de los paquetes entre 600 y 800 m (19 a 25,3 km equivalentes). Entre 800 y 1000 m no se recibe un solo paquete de los 10 enviados, pero luego, entre 1000 y 1200 m se obtienen 2 de 5 paquetes, con una distancia equivalente de 31,6 km a 37,9 km. Finalmente, se recibe un último paquete a 1231 m, equivalentes a 38,9 km.

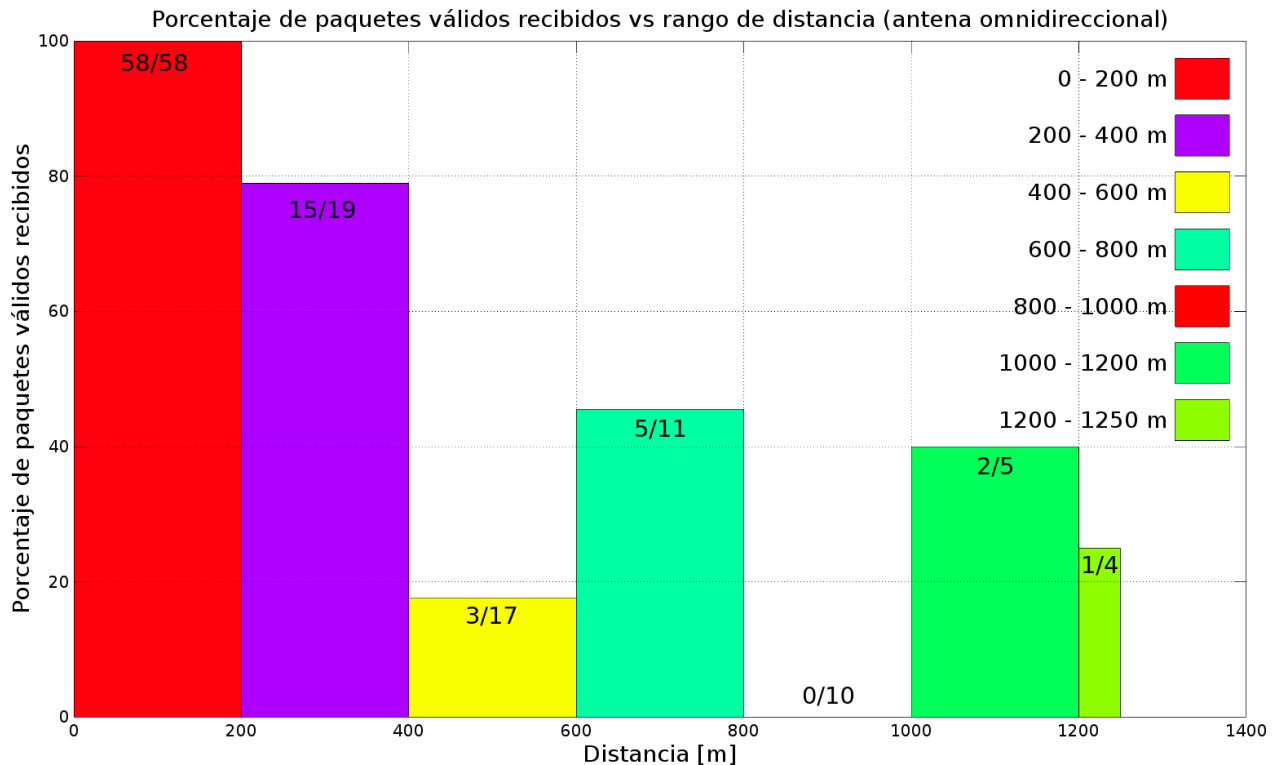


Figura 4.19: Paquetes de la radiosonda recibidos por la antena omnidireccional

Los resultados anteriores muestran que si bien es cierto, se puede llegar a una distancia cercana a los 40 km, hay una gran pérdida de paquetes, por lo que es más seguro limitarse a un rango menor a 25 km, lo cual es considerable tomando en cuenta las características de este tranceptor.

Durante la evaluación del desempeño de la estación terrena, se notó que existieron paquetes recibidos por ésta y no por la antena Yagi sin atenuación y viceversa. Esto se debe principalmente a la dificultad de apuntar al globo sonda. La figura 4.20 muestra el porcentaje de datos válidos recibidos por la estación terrena. A pesar de la complejidad en la estimación de la posición del globo sonda, se logró recibir apropiadamente el 50 % de los paquetes entre 4 y 4,4 km. Esto equivale a una distancia entre 400 y 440 km. En este

rango se encuentra la órbita de la ISS.

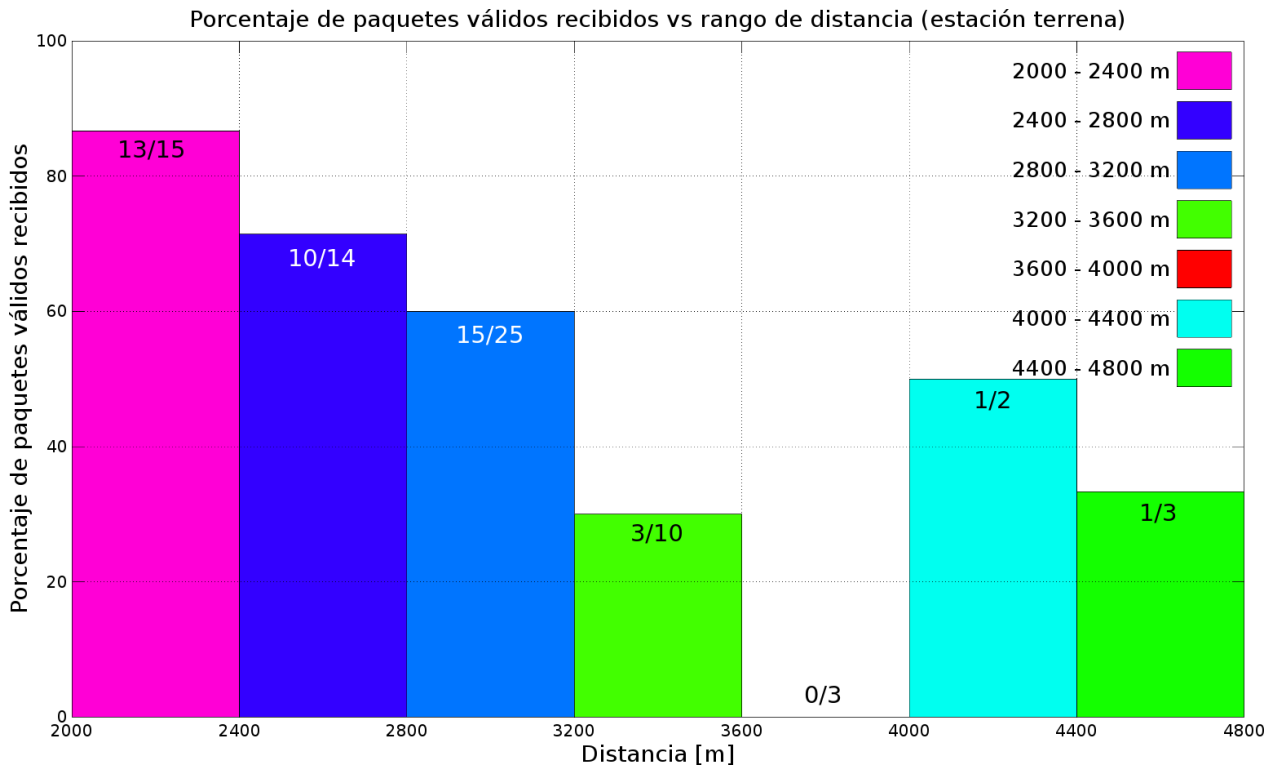


Figura 4.20: Paquetes de la radiosonda recibidos por la estación terrena

4.4. Cálculo de radioenlace final

Anteriormente se hizo un cálculo teórico de cuál sería el alcance máximo del transceptor RFM22B. Pero ahora se desea hacer un cálculo con datos empíricos y contrastarlo con el desempeño obtenido en la prueba de la radiosonda. Con este objetivo, se utiliza un analizador de espectros de donde se obtienen los datos que permiten calcular la potencia de salida en las modulaciones GFSK, FSK y OOK. Para conectar el analizador de espectros a la radio, se utiliza un adaptador de tipo N a SMA y otro conector SMA macho a SMA macho, como se puede apreciar en la figura 4.21.

El ancho de banda de las señales se obtiene viendo en que punto la señal decae en 3 dB con respecto al máximo. Para calcular la potencia, los datos (dentro del ancho de banda) se convierten de escala logarítmica (dBm) a lineal (mW), luego se suman entre sí y finalmente se transforma el resultado a dBm.

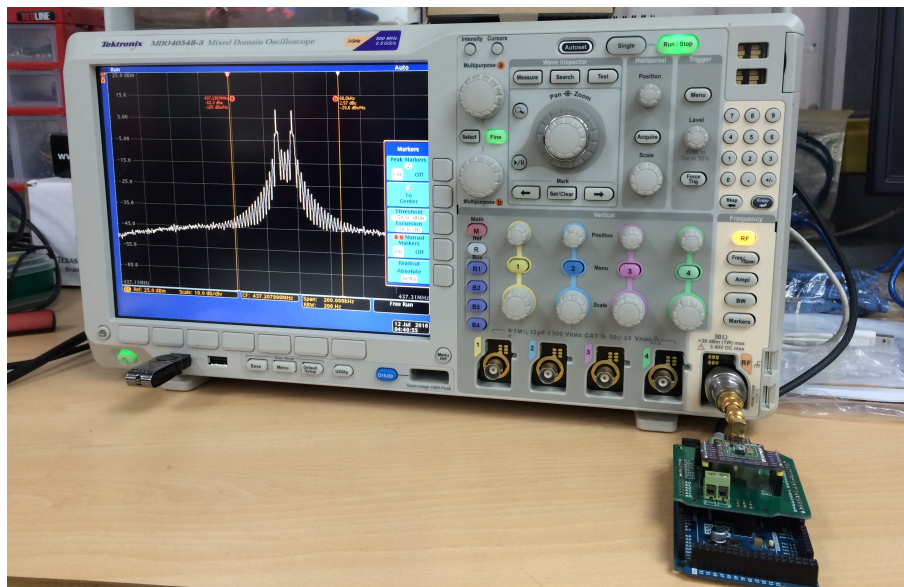


Figura 4.21: Analizador de espectros

En la figura 4.22 se ve en azul la potencia promedio de salida de la radio para cada frecuencia. Las líneas rojas muestran el inicio y término del ancho de banda de la señal. En modulación GFSK a 2 kb/s se obtiene una potencia máxima promedio de 18,146 dBm, lo cual es aproximadamente 2 dBm menos de lo especificado en el *datasheet*.

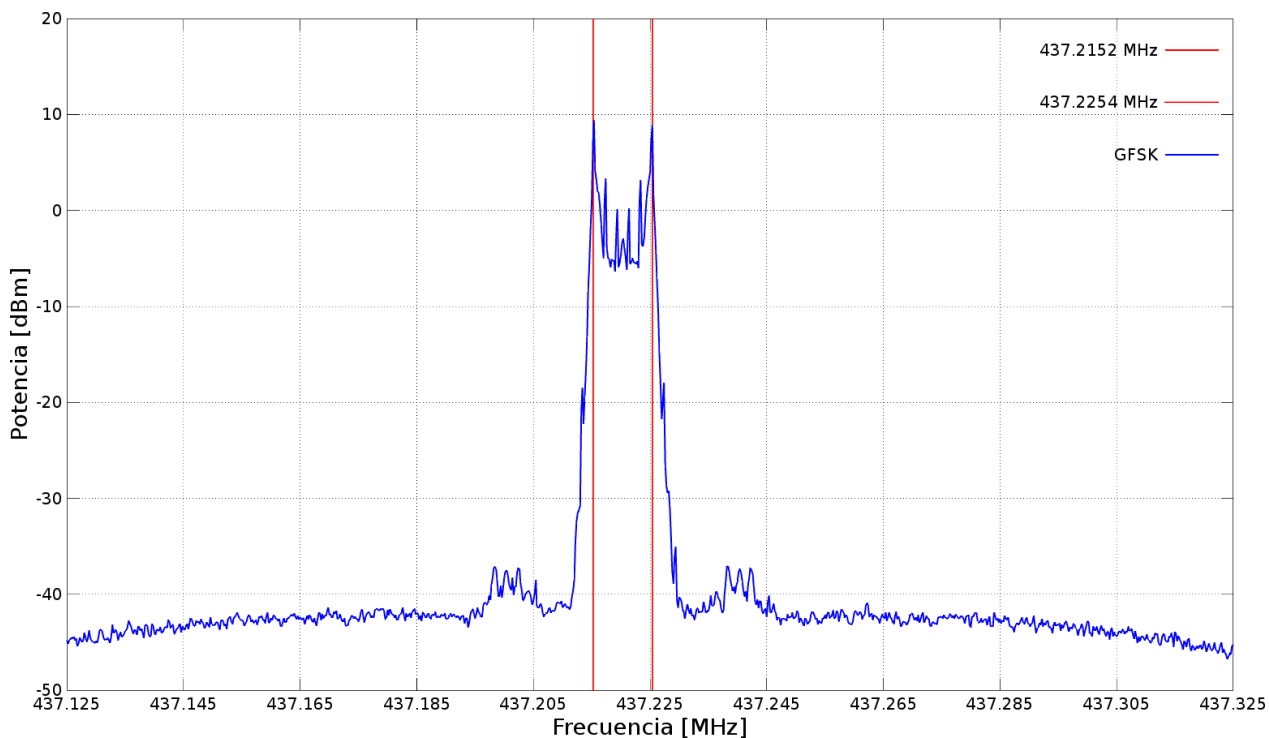


Figura 4.22: En azul el espectro del transmisor con modulación GFSK a 2 kb/s. En rojo el inicio y término del ancho de banda.

Por otro lado, la modulación FSK a 2 kb/s obtiene una potencia máxima promedio de 18,036 dBm. Este resultado es menor al esperado por las especificaciones de la radio. El ancho de banda con la modulación FSK es el mismo al utilizado por GFSK. Sin embargo, los lóbulos laterales con FSK contaminan el resto del espectro.

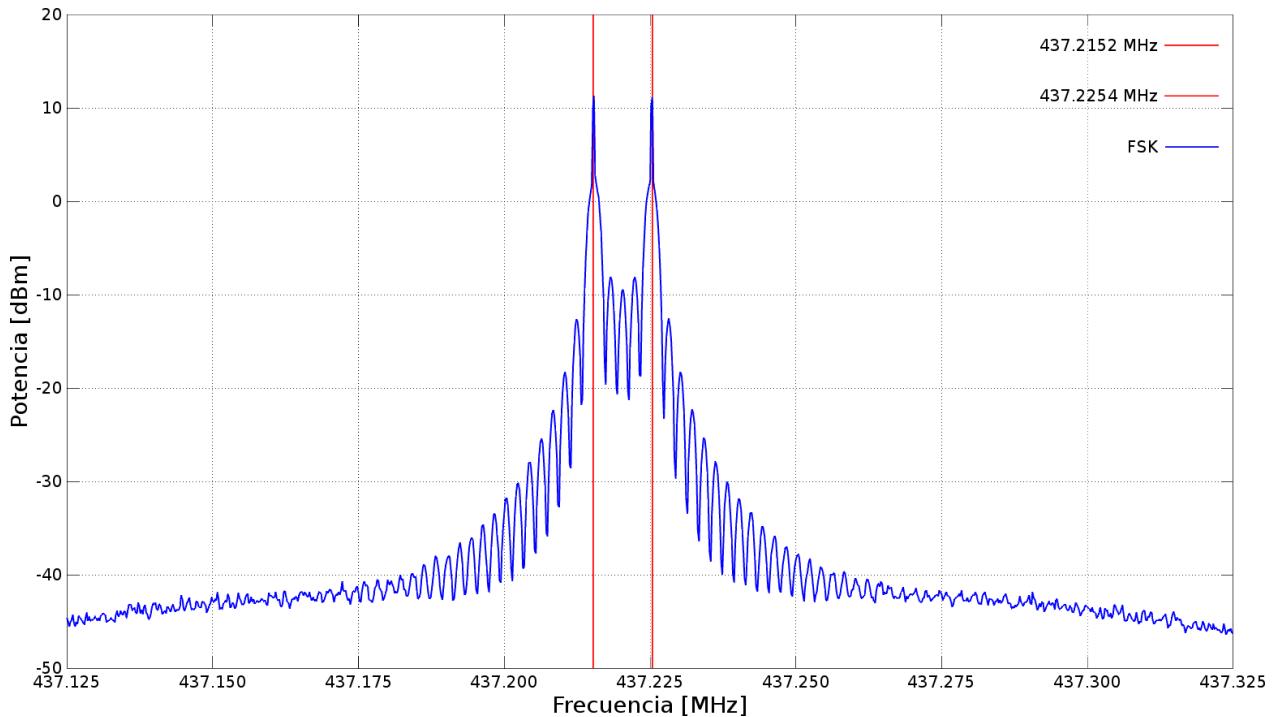


Figura 4.23: En azul el espectro del transmisor con modulación FSK a 2 kb/s. En rojo el inicio y término del ancho de banda.

El espectro de la modulación OOK a 2,4 kb/s se encuentra en la figura 4.24. El ancho de banda calculado es de sólo 200 Hz, pero los lóbulos laterales decaen en potencia más lentamente que en FSK. La potencia calculada es de 12,663 dBm. Esta potencia es aproximadamente 6 dBm menor a las modulaciones anteriores, lo cual explica su bajo desempeño en las pruebas realizadas en Parque O'Higgins.

También se debe considerar que según la hoja de datos, solamente GFSK logra obtener una sensibilidad de -121 dBm, mientras que la sensibilidad con FSK es cercana a -114 dBm y la de OOK de -110 dBm. La mejora de FSK por sobre OOK se explica por la potencia de salida que es mayor en FSK que en OOK. En el caso de GFSK, solo se puede explicar por como fue diseñada la radio, ya que la diferencia a favor de FSK se ve en la práctica y no en la teoría.

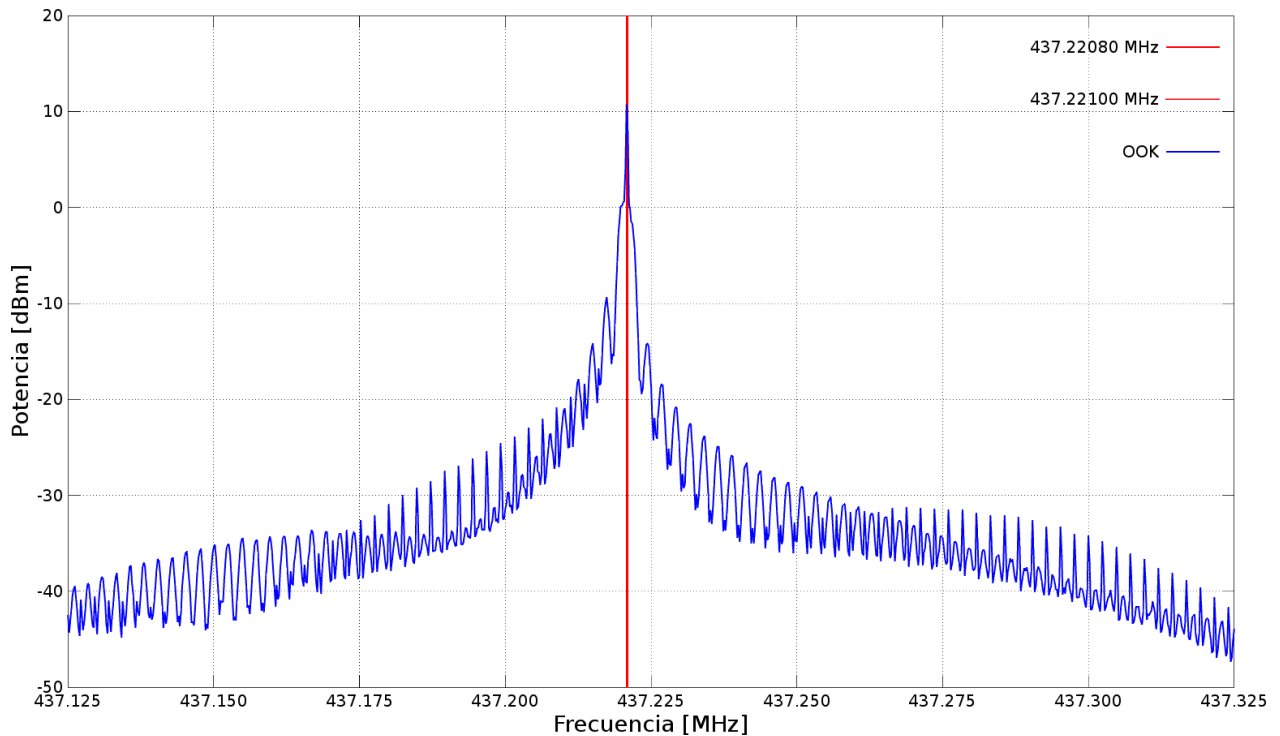


Figura 4.24: En azul el espectro del transceptor con modulación OOK a 2,4 kb/s. En rojo el inicio y término del ancho de banda.

Con este nuevo resultado, se hace un último cálculo de radioenlace entre dos radios usando antenas omnidireccionales y entre una radio con antena omnidireccional (mono-polo) y otra con la antena Yagi de la estación terrena.

4.4.1. Comunicación entre transceptores actualizada

Luego de ver el resultado obtenido por el analizador de espectros y la hoja de datos, se actualiza la tabla con los datos necesarios para hacer el cálculo de radioenlace entre dos radios RFM22B.

Transmisor	Antena TX	Antena RX	Sensibilidad RX	Margen de desvanecimiento
+18 dBm	+3 dBi	+3 dBi	-114 dBm	+20 dB

Tabla 4.6: Potencias actualizadas para el cálculo de enlace entre TRX RFM22B

A continuación, se disponen los términos en la ecuación que permite calcular la máxima pérdida de potencia en espacio libre (FSPL) que se puede permitir con esta configuración.

$$\begin{array}{rcccccccc}
 \text{TX} & + & \text{Ant. TX} & - & \text{FSPL} & + & \text{Ant. RX} & = & - & \text{Sens. RX} & + & \text{Margen} \\
 18 \text{ dBm} & + & 3 \text{ dBi} & - & \text{FSPL} & + & 3 \text{ dBi} & = & - & 114 \text{ dBm} & + & 20 \text{ dB}
 \end{array}$$

Esto da un total de 118 dB, lo cual en la frecuencia de 437,225 MHz equivale a una distancia máxima de 43,4 km. Este resultado es mayor al obtenido en la prueba de la radiosonda (38,9 km). Esto demuestra que la limitación del transceptor no está en la potencia de salida, sino que se debe a un problema con su sensibilidad o con las antenas.

4.4.2. Comunicación hacia la estación terrena actualizada

Análogamente al caso anterior, se hace una nueva tabla con los datos necesarios para realizar el cálculo de enlace desde una radio RFM22B con antena omnidireccional, hacia una radio igual conectada con la antena Yagi de la estación terrena del edificio de ingeniería eléctrica.

Transmisor	Antena TX	Antena RX	Sensibilidad RX	Pérdidas cables	Margen
+18 dBm	+3 dBi	+18.9 dBiC	-114 dBm	-2 dB	+20 dB

Tabla 4.7: Potencias actualizadas para el cálculo de enlace hacia la estación terrena

Los datos de la tabla 4.7 se ingresan a la misma ecuación anterior para calcular las pérdidas de espacio libre, como se puede ver a continuación.

$$\begin{array}{r r}
 \text{TX} & + & \text{Ant. TX} & - & \text{FSPL} & + & \text{Ant. RX} & - & \text{Cables} & = & - & \text{Sens. RX} & + & \text{Margen} \\
 18 \text{ dBm} & + & 3 \text{ dBi} & - & \text{FSPL} & + & 18,9 \text{ dBiC} & - & 2 \text{ dB} & = & - & 114 \text{ dBm} & + & 20 \text{ dB}
 \end{array}$$

El resultado para la FSPL es de 131,9 dB, lo cual también en la frecuencia de 437,225 MHz equivale a 214,9 km de distancia máxima aproximadamente. Este resultado es menor al obtenido en la prueba de la radiosonda, donde se obtuvo paquetes a una distancia equivalente de 477,3 km.

Los cálculos teóricos contemplan un margen de 20 dBm debido a condiciones climáticas extremas. Durante la realización de la prueba con la radiosonda, el cielo estaba completamente despejado, por lo que no existía una atenuación considerable debido al clima. Debido a esto, los resultados teóricos fueron peores a los prácticos en el caso de la estación terrena, ya que consideran este margen. En el cálculo entre transceptores con antenas monopolo, el valor teórico fue mayor al práctico. Esto se puede deber a que la ganancia de estas antenas no es realmente 3 dBi. A pesar de esto, en condiciones climáticas normales, es posible la transmisión y recepción de datos a una distancia considerable.

Conclusión

En este trabajo de título se evaluó el dispositivo RFM22B como posible transceptor para femto y nano-satélites. Se escogió el RFM22B debido a su pequeño tamaño (16 x 16 mm²), por ser liviano (1,26 g), de bajo consumo energético (30 - 85 mA), bajo costo (menos de 10 dólares) y por su flexibilidad (modulación, tasa de transferencia de datos y potencia programables). La evaluación fue en laboratorio y en terreno. En terreno se utilizó un globo sonda como transmisor y tres receptores con distintos tipos de antenas.

En la radio RFM22B la modulación OOK tiene un desempeño bajo en comparación con las otras modulaciones debido a su baja potencia de salida y por no tener la opción de disminuir su tasa de transmisión de datos a 2 kb/s. La modulación FSK ocupa el mismo ancho de banda que GFSK, pero tiene lóbulos laterales que contaminan el espectro electromagnético. En las pruebas en terreno, la modulación FSK fue consistentemente mejor a las otras modulaciones, a pesar de que la potencia de salida es similar a la de GFSK. Por estas razones se escoge FSK por sobre las otras modulaciones en este dispositivo en particular.

En pruebas en terreno (estaciones en tierra y transmisor en radiosonda), el transceptor RFM22B demostró ser capaz de recibir paquetes a una distancia de 38,9 km utilizando una antena monopolo de 3 dBi tanto en el emisor como en el receptor. Sin embargo, no es un enlace confiable ya que muchos de los paquetes no son recibidos o no pasan el cálculo CRC para ser considerados mensajes válidos. Es por esto que se recomienda hacer constelaciones de femtosatélites con una distancia de espaciado no mayor a 25 km o implementar sistemas de corrección de error o reenviar los paquetes hasta recibir una confirmación de que fue recibido correctamente.

Al analizar los resultados obtenidos por la estación terrena, se ve que se obtuvo un 60% de los paquetes a una distancia equivalente de 320 km y el 50% a 440 km, por lo que es posible recibir datos desde una órbita baja utilizando esa antena. Se recomienda buscar maneras de aumentar el rango, la sensibilidad del transceptor y mecanismos de reenvío con confirmación (que pueden bajar la tasa de datos efectiva), para no perder datos durante la transmisión.

4.5. Posibles usos para el RFM22B

Considerando lo anterior, este transceptor es un experimento viable para ser llevado como *payload* en un futuro CubeSat desarrollado por la universidad o en femtosatélites, pero se recomienda mejorarlo con las propuestas hechas en la sección "Trabajo futuro".

Otro posible uso es el de instrumentación para geofísica, agronomía, minería, o cualquier área de la ciencia en donde se necesiten múltiples instrumentos con sistemas embebidos obteniendo datos de sensores de humedad, presión, temperatura, contaminación, entre otros. Para que los datos de estos instrumentos sean transmitidos a un punto donde se pueda procesar la información.

El RFM22B es un transmisor/receptor de bajo costo y bajo consumo energético. Es ideal para aplicaciones en las que haya una limitación en la cantidad de energía disponible. Su tamaño pequeño y su bajo peso lo convierten en un dispositivo versátil y adaptable a las necesidades del usuario. Además, tiene un sensor de temperatura y cuenta con la capacidad de generar un reloj para un microcontrolador. Estas últimas prestaciones no fueron utilizadas durante la realización de esta memoria.

4.6. Trabajo futuro

Para aumentar el alcance de este transceptor y mejorar el *bit error rate*, se proponen las siguientes soluciones que pueden ser implementadas simultáneamente:

- **Cambiar la antena:** Se recomienda cambiar la antena omnidireccional por una antena más directiva. Esto se puede lograr con un arreglo de antenas omnidireccionales, las cuales al regular la amplitud y fase de sus corrientes, se puede modificar el patrón de radiación hacia la dirección deseada a expensas de otras direcciones.
- **Implementar un LNA:** Al incluir un amplificador de bajo ruido (LNA) al diseño de la PCB, se puede mejorar considerablemente la sensibilidad del satélite, sin saturar el espectro electromagnético. Existen reportes de usuarios de esta radio que han podido incluso cuadruplicar el rango de distancia utilizando un buen diseño de LNA. Esta solución permitiría tener femtosatélites separados a una mayor distancia, cubriendo una superficie aún mayor a la posible actualmente. Si además se incluye un LNA en la estación terrena, se podrá lograr un enlace confiable con un CubeSat o femtosatélite a pesar de estar a más de 500 km de distancia, utilizando esta radio.
- **Mejorar el sistema de reenvío:** Actualmente el software del transceptor permite reenviar un paquete una determinada cantidad de veces. Una vez alcanzada esa cantidad, se pasa al siguiente paquete. Una mejora necesaria es programar que sea una condición recibir la confirmación por parte del receptor de que el paquete fue

recibido correctamente, antes de transmitir uno nuevo. Esta mejora disminuye la tasa de transferencia de datos, pero asegura que mientras el receptor esté dentro del alcance del transmisor, no se pierdan paquetes.

- **Evaluar un nuevo modelo:** Existe otra radio llamada RFM23BP, la cual tiene una potencia de 30 dBm, pero un consumo de 550 mA. El RFM23BP es compatible con la misma librería utilizada en esta memoria. Se recomienda evaluar la posibilidad de utilizarlo en un nanosatélite. En anexos se encuentra un diseño de placa adaptadora para este transceptor.

Bibliografía

- [1] Australian Space Academy. Orbital parameters. <http://www.spaceacademy.net.au/watch/track/leopars.htm>, jun 2015.
- [2] Atmel. Sam3x/sam3a series datasheet. http://www.atmel.com/Images/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A_Datasheet.pdf, mar 2015.
- [3] Modtronics Australia. 433mhz rubber stub antenna - sma male. <http://modtronicsaustralia.com/shop/433mhz-rubber-stub-antenna-sma-male/>, mar 2016.
- [4] Carlos Eduardo González Cortés. Diseño e implementación del software de vuelo para un nano-satélite tipo cubesat. Master's thesis, Universidad de Chile, 2013.
- [5] Instituto de radioastronomía PUCP. Satélites: Pucp-sat-1. <http://inras.pucp.edu.pe/proyectos/pucp-sat-1/especificaciones-del-satelite/?lang=en>, mar 2014.
- [6] Wood & Douglas. What is 'fade margin'? <http://ultra-woodanddouglas.com/FAQ/what-is-fade-margin>, jul 2011.
- [7] HopeRF Electronic. Rfm22b/23b ism transceiver module v1.1. <https://www.sparkfun.com/datasheets/Wireless/General/RFM22B.pdf>, jan 2006.
- [8] Freetronics. Freetronics kicad schematic symbol & footprint library. https://github.com/freetronics/freetronics_kicad_library, nov 2015.
- [9] Steve Garber. Sputnik and the dawn of the space age. <http://history.nasa.gov/sputnik/>, oct 2007.
- [10] Mikal Hart. Tinygps++. <http://arduiniiana.org/libraries/tinygpsplus/>, sep 2014.
- [11] David Hitt. What is orbit? <http://www.nasa.gov/audience/forstudents/5-8/features/nasa-knows/what-is-orbit-58.html>, jul 2015.
- [12] Texas Instruments. Low-power sub-1ghz rf transceiver. <http://www.ti.com/lit/ds/symlink/cc1101.pdf>, mar 2016.

- [13] CubeSat Kit. Frequently asked questions. <http://www.cubesatkit.com/content/faq.html>, jun 2015.
- [14] Silicon Laboratories. Maximizing range and battery life in low-cost wireless networks. <http://www.silabs.com/Support%20Documents/TechnicalDocs/Maximize-Wireless-Network-Range-and-Battery-Life.pdf>, feb 2014.
- [15] Terje Lassen. Long-range rf communication: Why narrowband is the facto standard. <http://www.ti.com/lit/wp/swry006/swry006.pdf?DCMP=longrange&HQS=ep-wcs-lprf-longrange-contrib-whip-narrowband-wwe>, mar 2014.
- [16] Roger Ludwig and Jim Taylor. Yoyager telecommunitacions. http://descanso.jpl.nasa.gov/DPSummary/Descanso4--Voyager_new.pdf, mar 2002.
- [17] Zachary Manchester. Kicksat, your personal spacecraft in space! <https://www.kickstarter.com/projects/zacinaction/kicksat-your-personal-spacecraft-in-space/updates>, oct 2011.
- [18] Zachary Manchester. Radio info. <https://github.com/zacinaction/kicksat/wiki/Radio-Info>, apr 2014.
- [19] Mike McCauley. Rf22 library for arduino. <http://www.airspayce.com/mikem/arduino/RF22/>, feb 2016.
- [20] Modtronics. Rfm22b breakout board rf transceiver. <http://modtronicsaustralia.com/shop/rfm22b-s2-rf-transceiver-breakout-board-v2/>, mar 2016.
- [21] NASA. Nasa debris assessment software. <http://orbitaldebris.jsc.nasa.gov/mitigate/das.html>, dec 2015.
- [22] NASA. Outgassing data for selecting spacecraft materials online. <http://outgassing.nasa.gov>, jan 2016.
- [23] The national association for Amateur Radio. Part 97-amateur radio service. <http://www.arrl.org/part-97-amateur-radio>, mar 2016.
- [24] National Oceanic and Atmospheric Administration. About the licensing of private remote sensing space systems. <http://www.nesdis.noaa.gov/CRSRA/licenseHome.html>, mar 2016.
- [25] CubeSat org. Developers. <http://www.cubesat.org/>, mar 2016.
- [26] Quectel. L80 hardware design. http://www.quectel.com/uploadimage/downlad/180_hardware_design_v1.1.pdf, aug 2013.
- [27] Holli Riebeek. Catalog of earth satellite orbits. <http://earthobservatory.nasa.gov/Features/OrbitsCatalog/>, sep 2009.
- [28] SPEL. Spel is dedicated to research and development of instrumentation for the

geosciences. <http://spel.ing.uchile.cl/>, mar 2014.

- [29] Dan Stillman. What is a satellite? <http://www.nasa.gov/audience/forstudents/5-8/features/nasa-knows/what-is-a-satellite-58.html>, sep 2015.
- [30] M2 Antenna Systems. M2 antenna systems, inc. model no: 436cp42ug. <http://www.m2inc.com/content/PDF%20MANUALS/70CMANTS/436CP42MAN03-W.pdf>, jul 2015.
- [31] Tomás Ignacio Opazo Toro. Requerimientos, implementación y verificación del nanosatélite suchai. Master's thesis, Universidad de Chile, aug 2013.
- [32] International Amateur Radio Union. Worldwide voice of radio amateurs. <http://www.iaru.org/>, mar 2016.
- [33] Arizona State University. Suncube femtosat design specifications (sfd). http://femtosat.asu.edu/suncube_femtosat_design_specifications_v1.pdf, apr 2016.
- [34] California Polytechnic State University. Cubesat design specification rev 13. http://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/56e9b62337013b6c063a655a/1458157095454/cds_rev13_final2.pdf, feb 2014.
- [35] Douglas E. Nielsen y Jack Taylor William A. Beech. Ax.25 link access protocol for amateur packet radio. <http://www.ax25.net/AX25.2.2-Jul%2098-2.pdf>, jul 1998.
- [36] California Polytechnic State University y San Luis Obispo. Poly picosatellite orbital deployer mk. iii rev. e user guide. http://129.65.180.72/images/LaunchProviders/p-pod_mkiiiireve_userguide_cp-ppodug-1.0-1_rev1.pdf, mar 2014.

Capítulo 5

Anexo

Todos los programas, diseños y hojas de datos de elementos utilizados en las pruebas de este trabajo de título se encuentran en GitHub en la siguiente dirección URL:

https://github.com/matvidal/RFM22B_GPS

Además, se encuentra un respaldo de toda esa información en un CDROM adjunto a este documento. Los programas y librerías se encuentran en la carpeta *Code*, las hojas de datos del GPS, del transceiver RFM22B y del RFM23BP (otra radio de mayor potencia pero mayor consumo que podría servir en el futuro) están en la carpeta *Doc* y los diseños de las placas electrónicas en la carpeta *Hardware*.