



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

ESTUDIO DE LA COBERTURA DE LA MODULACIÓN LORA EN LA BANDA DE
915[MHZ] EN UN AMBIENTE URBANO

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

DANIEL ANTONIO JIMÉNEZ BALLART

PROFESOR GUÍA:
CESÁR AZURDIA MEZA

MIEMBROS DE LA COMISIÓN:
ANDRÉS EDUARDO CABA RUTTE
JOSÉ IGNACIO GUERRA GOMEZ

Este trabajo ha sido parcialmente financiado por Telefónica Investigación y Desarrollo Chile

SANTIAGO DE CHILE
2018

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: DANIEL ANTONIO JIMÉNEZ BALLART
FECHA: 2018
PROF. GUÍA: SR. CESÁR AZURDIA MEZA

ESTUDIO DE LA COBERTURA DE LA MODULACIÓN LORA EN LA BANDA DE 915[MHZ] EN UN AMBIENTE URBANO

Grandes empresas relacionadas a la innovación en telecomunicaciones a nivel mundial como IBM, CISCO y Nokia están desarrollando nuevas tecnologías y servicios para el *Internet of Things (IoT)*, entre ellas nuevas técnicas de modulación de señales, como *Chirp Spread Spectrum (CSS)*, que han permitido la creación de un nuevo tipo de redes de bajo consumo y grandes áreas de cobertura. El termino acuñado para estas redes es *Low Power Wide Area Networks (LPWAN)* y ellas se caracterizan por lograr rangos de cobertura del orden de los kilómetros, pero con un consumo de energía mínimo que permite mantener los nodos de la red por años en funcionamiento de manera remota sin un cambio de batería. El desarrollo de este nuevo tipo de redes va de la mano con la aparición de las *Wireless Sensor Network (WSN)*, redes ubicuas y redes *mesh* acercándonos a un futuro en donde se pueda conocer y controlar diversos aspectos de nuestro entorno de forma remota, en tiempo real y a bajo costo.

La tecnología estudiada en este trabajo es LoRa la cual permite enviar información con un bajo costo energético y ser recibida a largas distancias gracias a sus altos niveles de sensibilidad (-130[dBm]). En este trabajo se explicarán en detalle los fundamentos de estas características y se mostrará el análisis gráfico de un radio-enlace de LoRa en un ambiente urbano.

El estudio fue llevado a cabo en la ciudad de Santiago Chile, en la intersección de las comunas de Macul y Ñuñoa. Se integraron distintos *Hardware*s para crear un sistema de comunicación inalámbrico LoRa que opera en la banda ISM de 915[MHz] y permite asociar los parámetros del radio enlace con su respectiva posición georreferenciada. Fue posible obtener rangos de buena cobertura entre los 500[m] y los 700[m], y además se encontraron zonas de cobertura adicional a aproximadamente 1[km] de distancia entre el nodo transmisor y la estación base.

Esta memoria es parcialmente financiada por Telefónica Investigación y Desarrollo Chile, con el objetivo de desarrollar pruebas reales que permitan analizar y validar los resultados teóricos ofrecidos por esta nueva tecnología y además hacerse del *Know How* necesario para utilizar estas tecnologías en desarrollos nacionales.

Dedicado a mis padres, Belinda Ballart y Fernando Jimenez, que me han dado su apoyo y comprensión en estos años estudiando Ingeniería Civil Eléctrica.

Dedicado a mi compañera, Jo Grass, que estuvo a mi lado desde el inicio de la carrera hasta los momentos finales y ha sabido contenerme, apoyarme y entenderme durante las vicisitudes de la vida.

Dedicado a mi canes, Chicle y Perrin, los que me dieron mas amor incondicional del que jamas pudiera haberme imaginado.

Agradecimientos

Agradecimientos a Telefónica I+D y al departamento de Ingeniería Eléctrica de la Universidad de Chile por apoyar este trabajo. Al profesor Cesar Azurdia por guiarme en este proceso y a José Ignacio Guerra por motivarme a tomar este desafío.

A mis amigos Diego A., Francisco A., Antonio C., Sebastian B., David V. y Francisco C. por apoyarme y empujarme a seguir en los momentos de flaqueza.

A mis amigos Matias Navarro (Mato) y Carlos Moreno (Charly), ambos ingenieros destacados, pero por sobre todo, dos referentes de esfuerzo y compromiso que estuvieron atentos hasta el final.

A Jaime Aranda, Jose Caceres y Andres Peñaloza, amigos y compañeros de carrera, quienes me ayudaron a realizar un mejor trabajo, tanto en la obtención como en el análisis de los resultados.

Al equipo docente del DIE por su apoyo y sus enseñanzas durante los duros años de carrera. Además al Proyecto Eolian y el CIL (Centro de Innovación del Litio) donde forje mis principales habilidades como Ingeniero.

Tabla de contenido

Introducción	1
1. Marco Teórico y Estado del Arte	5
1.1. Machine to Machine (M2M) o Comunicación entre máquinas	6
1.1.1. Sistemas de comunicación electrónicos	7
1.1.2. Modelo de comunicación OSI	8
1.1.3. Capa de enlace de datos	9
1.1.4. Capa Física	13
1.1.5. Enlaces Inalámbricos	14
1.2. Redes inalámbricas de área amplia y baja potencia (LPWAN) y LoRa	18
1.2.1. LPWAN	18
1.2.2. LoRaWAN	20
1.2.3. LoRa	21
1.3. SNR, pérdidas y Link Budget	26
1.3.1. SNR	26
1.3.2. Link Budget	30
1.4. Aspectos Regulatorios	31
1.4.1. Regulaciones a nivel mundial	31
1.4.2. Chile	32
1.5. Otros estudios de comunicación y cobertura de LoRa	33

1.5.1.	Cisco PIRL: Laboratorio de Innovación e Investigación de Cisco	33
1.5.2.	Centro de Comunicaciones Inalámbricas, Universidad de Oulu, Finlandia	36
2.	Dispositivos y Herramientas	38
2.1.	Hardware	38
2.1.1.	Módulos de Radio LoRa: DevKit	39
2.1.2.	GPS	44
2.1.3.	<i>Datalogger</i> : MicroSD	46
2.1.4.	Procesador central: Freaduino	46
2.2.	Conexión	48
2.2.1.	Conexión del Sistema Completo	50
2.3.	Entorno de desarrollo	50
2.3.1.	Librerías y código: GPS, microSD y Radio LoRa	50
2.3.2.	<i>Softwares</i>	51
2.3.3.	<i>Firmware</i> Procesamiento Central	54
3.	Implementación y Metodología	57
3.1.	Sistema de Comunicación LoRa	57
3.1.1.	<i>End-device</i> : Nodo Transmisor	57
3.1.2.	Estación Base: Nodo Receptor	59
3.2.	Pruebas	59
3.2.1.	Terreno de Pruebas	60
3.2.2.	<i>End-device</i> móvil	60
3.3.	Mapas de Calor: <i>Heatmaps</i>	61
3.3.1.	Muestras	61
3.3.2.	Ploteo de mapas	62
4.	Análisis de resultados	63

4.1. Parámetros del radio enlace	63
4.1.1. Configuración de Radio y Paquetes de LoRa	63
4.1.2. Resumen de Parámetros	65
4.1.3. Calculo de parámetros del enlace LoRa	65
4.2. Prueba 1: Prueba de decaimiento	66
4.2.1. Análisis gráficos	67
4.2.2. <i>Heatmap</i> Prueba 1	70
4.3. Prueba 2: Prueba de cobertura	71
4.3.1. <i>Heatmap</i> Prueba 2	71
Conclusión	72
Anexos	75
A. Lista de Siglas	76
B. Código Principal del Datalogger LoRa	78
C. Esquema de conexión PCB NZ32-SC151	80
D. Esquema de conexión PCB InAir9B	82
Bibliografía	86

Índice de Tablas

1.	Tecnologías <i>wireless</i> populares en la actualidad.	1
1.1.	Frecuencias de modulación (M=8) 8-MFSK dado: $f_c=250$ [kHz], $f_d=25$ [kHz] y L=3bits.	14
1.2.	Sensibilidad en el receptor en [dBm] para distintos Bandwith y spread factors [16].	24
1.3.	Parámetros Prueba Cisco PIRL [14].	34
1.4.	Resultados de PDR en pruebas de cobertura[14].	35
2.1.	Parámetros mostrados en la pantalla principal del nodo master.	40
2.2.	Características del microcontrolador STM32LC151RC.	42
2.3.	Características del módulo InAir9b [26].	43
2.4.	Tabla de conexión micorSD a microcontrolador.	46
2.5.	Tabla de conexión sistema completo.	49
4.1.	Parámetros de Radio por <i>default</i>	65
4.2.	Parámetros de los paquetes LoRa por <i>default</i>	65
4.3.	Parámetros calculados del enlace LoRa.	65
4.4.	Parámetros ambientales y configuración de radio prueba 1.	66
4.5.	Resumen Zonas prueba 1.	70
4.6.	Parámetros ambientales y configuración de radio prueba 2.	71
4.7.	Beneficios y Upgrades Necesarios	74

Índice de Ilustraciones

1.	Gráfico de cobertura mundial de redes LPWAN [4].	3
1.1.	Diagrama simplificado de la red de internet, sus enlaces y sus componentes [1].	6
1.2.	Diagrama de bloques de un sistema de comunicación electrónico [7].	7
1.3.	Espectro de radio y tecnologías que lo utilizan [8].	8
1.4.	Capas del modelo OSI [7].	9
1.5.	Enlace TCP/IP entre Servidor y PC [8].	10
1.6.	Ejemplo de un adaptador de red y como se entrecruzan los componentes de este con la pila de protocolos [1].	11
1.7.	Diagrama de división de canales de TDM(izq.) y FDM (der.) [8].	12
1.8.	Distintas modulaciones para la misma información digital [8].	13
1.9.	Uso del espectro EM para transmisión 4-FSK [8].	15
1.10.	Ejemplo de Direct Sequence Spread Spectrum (DSSS) [8].	17
1.11.	Ejemplo de esparcimiento SS [10].	17
1.12.	Tasa de conectividad LPWAN por aplicación esperada para el 2023 basado en los datos del 2015 [11].	19
1.13.	Comparación de tecnologías inalámbricas según ancho de banda (BW) y Rango de cobertura[12].	19
1.14.	Ubicación de LoRaWan en el stack de protocolos de comunicación [13].	20
1.15.	Diagrama de conexión de un sistema LoRaWAN[13].	21
1.16.	Pulsos <i>upchirp</i> y <i>downchirp</i> [15].	22
1.17.	LoRa <i>Frame Format</i> [14].	25

1.18. Mensaje de radio LoRa[14].	25
1.19. Forma común de la curva BER vs E_b/N_0 , para distintas técnicas de modulación [8].	27
1.20. Ejemplo de un error de bit, causado por ruido AWG[8].	28
1.21. Ejemplos de <i>multipath propagation</i> [8].	29
1.22. Típico desvanecimiento lento y plano en ambientes urbanos[8].	29
1.23. Herramienta de Semtech para el cálculo de parámetros de radio de LoRa. . .	31
1.24. Resumen de regulaciones a nivel mundial[13], pertinentes a LPWAN y aplicables a LoRa.	32
1.25. Diagrama de la asignación de frecuencias en el territorio nacional, banda 915[MHz].	32
1.26. Resultados de sensibilidad: RSSI medido vs RSSI especificado para distintos SF [14].	34
1.27. Proporción de paquetes recibidos (PDR) en pruebas de cobertura de LoRa [14].	35
1.28. Campo de pruebas para analizar la cobertura de LoRa [14].	36
1.29. Mapa de calor de la zona de Oulu para la cobertura de LoRa[19].	37
2.1. Diagrama de bloques del Nodo Transmisor.	39
2.2. Módulos de desarrollo DevKit.	39
2.3. Ejemplo de pantalla principal del nodo en estado “master”.	40
2.4. Parámetros configurables de los nodos devkit.	40
2.5. Componentes y puertos de la Placa NZ32-SC151 [22].	41
2.6. <i>Pinout</i> Extendido de la placa NZ32 [22].	42
2.7. Imagen del módulo InAir9 con <i>shield</i> RF (izq) y sin <i>shield</i> RF (der) [26]. . .	43
2.8. Diagrama de Bloques del chip SX1276 [16].	44
2.9. Módulo GPS L80 de Quectel [29].	44
2.10. <i>Pinout</i> de placa <i>breakout</i> para el GPS L80 y medidas en [mm].	45
2.11. Diagrama de comunicación Transistor-Transistor Logic (TTL) entre dos puertos UART [30].	45

2.12. Ejemplo de <i>string</i> de datos en formato GPRMC.	46
2.13. Placa microSD <i>breakout</i> de SparkFun [31].	46
2.14. Descripción Freaduino [32].	47
2.15. Diagrama de bloques detallado del nodo transmisor (<i>end-device</i>).	48
2.16. Diagrama de conexión entre Freaduino, GPS, microSD, NZ32-SC y Batería Externa.	49
2.17. Librerías Anexadas.	50
2.18. Repositorio de GitHub[34].	51
2.19. Softwares para el desarrollo.	51
2.20. Dfu File Manager.	52
2.21. DfuSe Demo.	53
2.22. CoolTerm leyendo COM31 a 9600[baud].	54
2.23. Fragmento código principal, definición de pines.	55
2.24. Código agregado a “main.cpp”.	56
3.1. <i>End-device</i> en funcionamiento.	58
3.2. Estación Base en funcionamiento en borde ventana.	58
3.3. Mapa de la zona bajo estudio y el recorrido aproximado del <i>end-device</i>	59
3.4. Foto panorámica hacia el Norte desde la ubicación de la estación base.	60
3.5. <i>End-device</i> montado en bicicleta.	60
3.6. archivo .txt con las muestras.	61
3.7. archivo .csv con las muestras.	62
3.8. ejemplo de <i>heatmap</i> , para densidad de aves de Norteamérica.	62
4.1. Definición del módulo de radio inAir9b y su Freq, BW, SF y Pwr por <i>default</i>	63
4.2. Definición para habilitar CRC.	64
4.3. SF, tamaño preámbulo, CR y habilitación <i>header</i>	64
4.4. Formato del mensaje Ping.	64

4.5. Formato del mensaje Pong.	65
4.6. Decaimiento del RSSI de la estación Base vs distancia.	67
4.7. Decaimiento del RSSI y SNR <i>end-device</i> ida (120 puntos).	68
4.8. SNRed y ECok en función de la longitud.	69
4.9. RSSIeb y su tendencia vs Longitud.	69
4.10. <i>Heatmap</i> de comunicación LoRa Prueba 1.	70
4.11. <i>Heatmap</i> de comunicación LoRa Prueba 2.	72

Introducción

Motivación y Antecedentes

Desde principios del siglo XXI se viene escuchando en el rubro de las Tecnologías de información el término Internet of Things (IoT), en el 2005 la Unión Internacional de Telecomunicaciones (UIT) lanzó una publicación haciendo un resumen de sus principales características [1]. Este término hace referencia a la masiva integración de las tecnologías de comunicación *wireless* y *Machine to Machine (M2M)* a la vida cotidiana.

El Internet es una red de computadores que interconecta cientos de millones de dispositivos terminales, ya en el 2011 la UIT estimó que Internet tenía 2 billones de usuarios [1]. En un futuro cercano los electrodomésticos, terminales móviles o cualquier nuevo dispositivo electrónico estarán conectados a Internet y tendrán la capacidad de censar nuestro ambiente, permitiéndonos tomar mejores decisiones ya que monitorean nuestro entorno en tiempo real almacenando la *data*. La aplicación práctica del IoT y por consiguiente su desarrollo como mercado requiere del avance e integración de nuevas tecnologías por parte de la industria de las Telecomunicaciones. Justamente es en este aspecto que esta investigación pretende ser un aporte estudiando una nueva tecnología *wireless* que permitirá soportar un abanico nuevo de aplicaciones prácticas del IoT para la industria.

Las tecnológicas de telecomunicación predominantes hoy en día, también denominadas *Wireless Area Network (WAN)*, varían según el área de cobertura que pueden ofrecer. Las principales WAN se conocen como; *Personal Area Network (PAN)*, *Local Area Network (LAN)* y *Metropolitan Area Network (MAN)*. Estas redes están pensadas principalmente para permitir un alto flujo de datos desde el *Gateway* (último dispositivo cableado a la red) a los *Endpoints* (dispositivos terminales enlazados de forma inalámbrica). La evolución natural de los enlaces de comunicación *wireless* diseñados para el uso cotidiano de la población fue incrementar el ancho de banda aumentando el *downstream*, priorizando el uso de interfaz gráfica y archivos en alta definición. Algunas tecnologías propietarias relacionadas a cada grupo de WAN se resumen en la Tbl. 1.

Wireless PAN	Wireless LAN	Wireless MAN
Bluetooth	Wifi	WiMax
RFID	ZigBee	2G/3G

Tabla 1: Tecnologías *wireless* populares en la actualidad.

Hace solo un par años, el estado de avance de tecnologías militares de los 50' llevo a la aparición de nuevas patentes y empresas en el rubro de las comunicaciones *wireless*. Empresas como Semtech [2] (impulsados por IBM [3]) están comercializando nuevas tecnologías propietarias de comunicación inalámbrica, las cuales abrieron las puertas a un nuevo tipo de redes, las LPWAN. Estas redes fomentan nuevas líneas de negocios para empresas del rubro de las Telecomunicaciones y Tecnologías de Información en nuestro país. Dicho de otra manera las LPWAN permiten ofrecer nuevos servicios de IoT a distintos sectores productivos de Chile, utilizando y adaptando la infraestructura actual para integrar estas nuevas tecnologías. Las principales características de las LPWAN, según la compañía Machina Research quienes acuñaron el término en el 2013, son[4]:

- *Low Power*: Bajo consumo para permitir que los dispositivos de la red tengan autonomía energética por años, asumiendo un censado de la degradación y auto-descarga de la batería.
- *Wide Area*: Amplios rangos de cobertura, por sobre los 500 metros entre el *gateway* y los *endpoint*, asumiendo condiciones desafiantes en las implementaciones tales como ambientes urbanos o subterráneos.

Además de estas dos características principales, existen otras que se definen como típicas de las LPWAN, tales como: alta densidad de *Endpoints*, costo de hardware reducido, costo de conectividad reducido y baja velocidad de datos o *data rate*. Es importante notar que las LPWAN se caracterizan por transmitir a baja *data rate*. Esto se debe a que están pensadas para comunicación M2M como se ha señalado anteriormente, muy distinto de las WAN, que deben soportar aplicaciones que requieren alto *downstream* como Youtube, Facebook, entre otras. En este caso los dispositivos para el IoT están concebidos desde su diseño para transmitir poca información, como serían algunos parámetros medidos por sus sensores, su estado de batería o funcionamiento y quizás datos del radio enlace.

La empresa Machina Research publicó en [4] que para fines del 2015 la cantidad de dispositivos LPWAN habilitados y activos superaba los 23 millones a nivel mundial. Se puede ver en el gráfico de la Fig. 1 que se estima que la cobertura mundial se sature para el 2023. Este gráfico se estimó utilizando el porcentaje de la población con cobertura de redes LPWAN (todo tipo de tecnologías; Ingenu RPMA[5], Sigfox[6], LoRa, etc.) por sobre el total de la población para cada país estudiado.

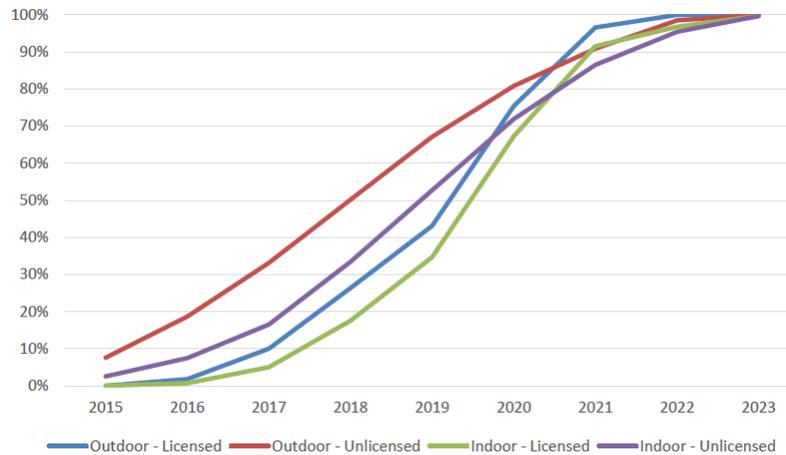


Figura 1: Gráfico de cobertura mundial de redes LPWAN [4].

Las evidencias mencionadas en el párrafo anterior, donde se manifiesta el gran potencial de aplicación de estas nuevas tecnologías, interesó al departamento de investigación y desarrollo de Telefónica junto al departamento de Ingeniería Civil Eléctrica de la Universidad de Chile, quienes apoyaron este trabajo de título. En esta memoria se busca demostrar que la cobertura para enlaces inalámbricos con la modulación LoRa y su desempeño sobre la banda, entre los 902[Mhz] y los 928[Mhz], dentro de un ambiente urbano (como lo es Santiago de Chile) permitiría soportar una red LPWAN.

Hipótesis

La hipótesis de este trabajo es validar que la cobertura de la modulación Long Range (LoRa) es suficiente para sostener una red LPWAN en una zona urbana de Santiago, Chile.

Para esto, se buscó corroborar que la máxima cobertura lograda con los dispositivos utilizados se ajustara a los requerimientos de una LPWAN, esto implica medir dos parámetros: primero, la distancia máxima a la que la intensidad de señal es suficiente para ser escuchada por la estación base, y segundo, la calidad de la comunicación dentro de esa distancia máxima.

Objetivos

Objetivo General

El objetivo de este trabajo es crear un método de medición que registre el área de cobertura alcanzada por la modulación LoRa, permitiendo además, evaluar la calidad de señal dentro

de dicha área.

Objetivos específicos

1. Realizar una breve investigación sobre los aspectos teóricos de la comunicación electrónica, las redes inalámbricas y la calidad del radio enlace. Además, presentar un resumen de las características de LoRa.
2. Resumir los resultados de otros trabajos similares y los posibles usos prácticos de esta tecnología.
3. Diseñar y construir un sistema de comunicación inalámbrica LoRa, que permita registrar las mediciones del parámetro *Signal to Noise Ratio (SNR)* y las coordenadas georreferenciadas en cada punto de medición.
4. Desplegar los datos como mapas de calor sobre planos satelitales de la zona estudiada.

Capítulo 1

Marco Teórico y Estado del Arte

En este capítulo se contextualiza al lector en tres puntos importantes a considerar para comprender el tema tratado en esta memoria. Lo primero es conocer algunos aspectos fundamentales de la comunicación entre máquinas. Lo segundo es explicar algunos detalles sobre enlaces inalámbricos, LPWAN y la tecnología LoRa. Lo último será revisar la importancia del SNR, las razones de las pérdidas en la comunicación y la forma de estimar la calidad deseada del sistema de comunicación inalámbrico. Además, se verán algunos resultados obtenidos en otros países en experiencias similares.

Para medir y estudiar la cobertura inalámbrica alcanzada con la tecnología LoRa se deben conocer las capas más bajas de comunicación, la **capa física** y la **capa de enlace**. Como se puede ver en la Fig. 1.1 los paquetes de información, conocidos como **paquetes** en la jerga informática, se envían a través de los **enlaces** de la red hasta el sistema terminal receptor. En la capa más alta, la capa de aplicación, un paquete de información va desde una APP (software) a otra, por ejemplo desde google drive(emisor), hasta el servidor donde se almacena (la nube), atravesando por distintos chips, enlaces físicos, sistemas operativos o *firmwares*, redes internacionales y aplicaciones, hasta finalmente, quedar almacenado en un servidor(receptor) donde los datos pueden ser consultados y requeridos posteriormente. Mientras, en la capa física la información es encapsulada y transmitida a través de un solo medio físico, como por ejemplo lo sería las pistas de la tarjeta madre dentro de un computador.

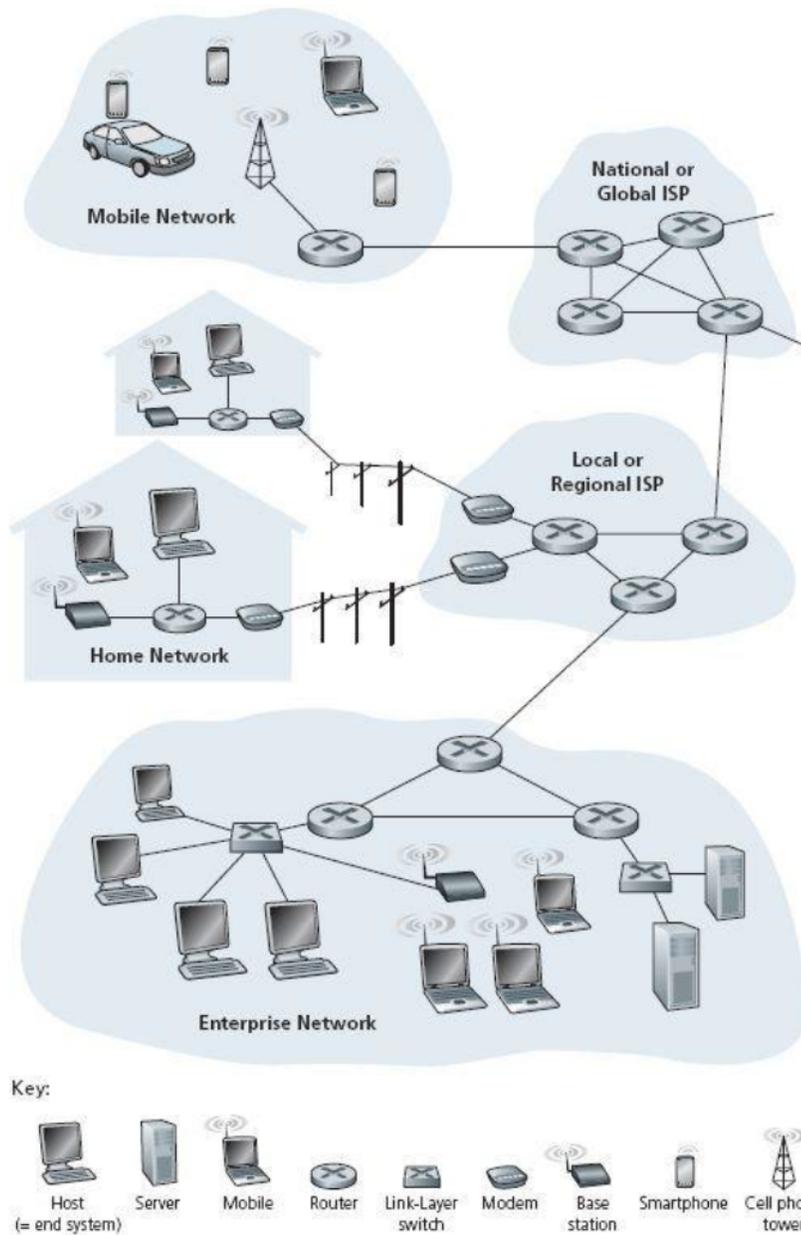


Figura 1.1: Diagrama simplificado de la red de internet, sus enlaces y sus componentes [1].

1.1. M2M o Comunicación entre máquinas

Internet, las redes de computadoras y los enlaces de comunicación entre máquinas en general, hacen un uso extensivo de los protocolos. Los distintos protocolos se utilizan para llevar a cabo las distintas tareas de comunicación. Algunos protocolos son simples y directos, mientras que otros son complejos e intelectualmente profundos. Dominar el campo de las redes de computadoras es equivalente a entender el qué, el por qué y el cómo de los protocolos de red.

1.1.1. Sistemas de comunicación electrónicos

Tal como se describe en [7] (cap 1) los sistemas de comunicación electrónicos son aquellos que comunican máquinas y su objetivo principal es la transmisión, recepción y procesamiento de información mediante el uso de circuitos electrónicos. Toda información, analógica o digital se debe convertir en energía electromagnética para ser propagada por algún medio (cable, fibra óptica o aire) entre el transmisor y receptor. Los sistemas de comunicación electrónicos están compuestos por tres partes como se puede ver en la Fig. 1.2 estas son: transmisor, receptor y medio.

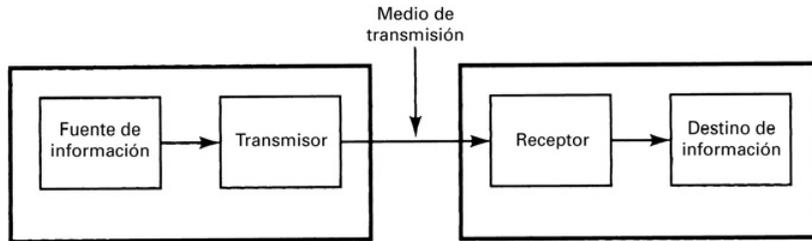


Figura 1.2: Diagrama de bloques de un sistema de comunicación electrónico [7].

En el caso particular de este trabajo, el medio de transmisión o **canal** fue el aire y se utilizó el espectro de radio entre los 902[MHz] y los 928[MHz] para comunicar al transmisor con el receptor. Las ondas moduladas con LoRa permitirán la transmisión de **paquetes de información** estableciendo un **enlace de comunicación** entre ambos nodos, nodo transmisor y nodo receptor. En la Fig. 1.3, se aprecian los nombres asociados a diferentes rangos del espectro de radio (bandas) y el desarrollo de tecnologías a lo largo del tiempo que utilizan estas bandas. Los canales de radio transportan los paquetes en estas bandas del espectro electromagnético (EM) y son bastante atractivos porque no requieren de la instalación de un medio físico. Las ondas EM se pueden propagar por distintos medios (vacío, aire o guías de onda) y tienen un comportamiento complejo al propagarse mientras cambian de medio, rebotan y/o son absorbidas. En la sección 1.3 se ahondará más en este tema.

Para comprender mejor los parámetros con los que se desea evaluar el radio enlace de LoRa, se hará una revisión del modelo de comunicación OSI (aplicado en más de 130 países) y sus primeras capas.

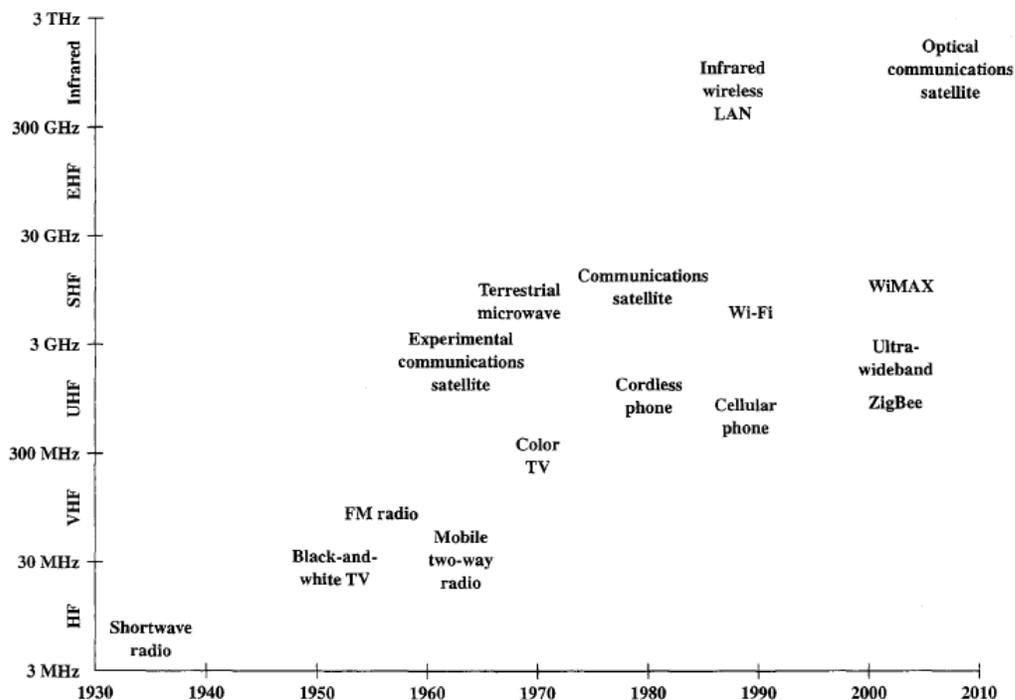


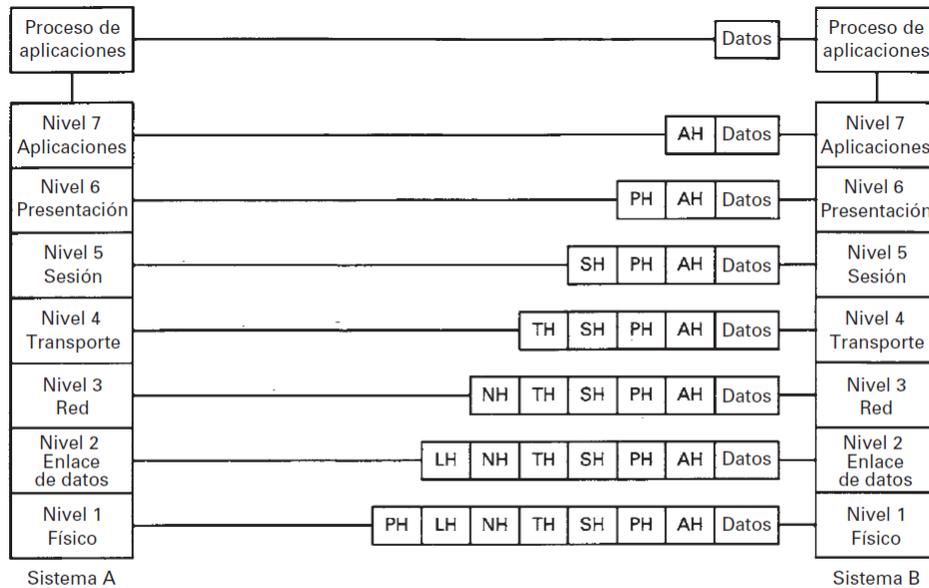
Figura 1.3: Espectro de radio y tecnologías que lo utilizan [8].

1.1.2. Modelo de comunicación OSI

El modelo de referencia Open System Interconnection (OSI) [7] (cap 14) es un conjunto de normas que guían la comunicación entre máquinas (computadoras, terminales y redes). El OSI fue desarrollado a inicios del '77 por la *International Organization for Standardization (ISO)* y por el Comité Consultivo Internacional Telegráfico y Telefónico (CCITT). Una vez terminado fue publicado en el '84 y el '83 respectivamente. El modelo OSI corresponde a un modelo de comunicación de 7 capas, en la Fig. 1.4 se pueden observar cómo se ordenan y su posición jerárquica dentro del protocolo. También se puede apreciar que cada capa agrega encabezados a la información, generalmente menos del 15% del mensaje es de la fuente.

Hay que destacar que, cada capa consta de un protocolo específico de comunicación, por ejemplo, en el estándar 802 del *Institute of Electrical and Electronics Engineers (IEEE)*, basado en las referencias que hace el modelo OSI, la sección 802.1 define la relación entre los dos estándares 802 y OSI, en la sección 802.2 se define el protocolo de control del enlace lógico (LLC) que forma parte de la capa de enlace de datos. El protocolo 802 en su sección 802.11, una de las más conocidas, se encarga de definir el funcionamiento de Wifi(*wireless*).

Los niveles superiores (3, 4, 5, 6 y 7) permiten la comunicación directa entre computadoras, mientras que las capas inferiores (1 y 2) tienen que ver con la mecánica real de transmitir datos (a nivel de bits y tramas). Los parámetros utilizados para evaluar el enlace de LoRa corresponden a las dos primeras capas que se verán a continuación.



Jerarquías del protocolo internacional ISO. AH = encabezado de aplicaciones; PH = encabezado de presentación; SH = encabezado de sesión; TH = encabezado de transporte; NH = encabezado de red; LH = encabezado de enlace; PH = encabezado físico

Figura 1.4: Capas del modelo OSI [7].

1.1.3. Capa de enlace de datos

En la tercera capa, la capa de red, se define el camino que debe tomar la información a través de la red para llegar desde el nodo emisor al nodo receptor, para esto los paquetes viajarán por la red por distintos tipos de enlaces, quizás algunos cableados y otros inalámbricos. La segunda capa, **la capa de enlace de datos**, se encarga del traspaso de los paquetes entre cada enlace individual de la red hasta que llegan al receptor. La capa de enlace toma los datagramas de la capa de red, los encapsula en tramas y se encarga de transmitir cada trama de manera fidedigna entre cada enlace individual. La capa de enlace está conformada por dos subcapas, la de control del enlace lógico (LLC) y la *Medium Access Control (MAC)*.

Aunque existen diversos tipos de enlaces individuales en la red, algunos tienen canales de difusión punto a punto y otros canales de difusión compartidos. Para el caso de un enlace punto a punto PPP (*point to point protocol*), coordinar la comunicación para evitar la colisión de las tramas es más trivial, pero en el caso de las comunicaciones inalámbricas, ya sea una WAN o una red satelital, muchos dispositivos comparten un mismo canal de comunicaciones volviendo muchísimo más complejo e importante la coordinación entre ellos. La capa de enlace proporciona de un protocolo de acceso al medio para coordinar las transmisiones y evitar la colisión de las tramas, además provee de un medio para activar, mantener y desactivar el enlace de datos preocupándose de la detección y corrección de errores.

El router es un dispositivo especial de la red, pues ayuda en la dirección de paquetes dentro de las redes. En la Fig. 1.5 se puede ver un enlace TCP/IP (configuración regida por IEEE 802) que conecta un servidor con un PC mediante dos tipos de protocolos distintos de

la capa de enlace, a la izquierda ATM y a la derecha IEEE 802 LAN. La capa TCP hace el trabajo de las capas de transporte, sesión y presentación (4, 5 y 6) y la capa IP hace el trabajo de la capa de Red (3).

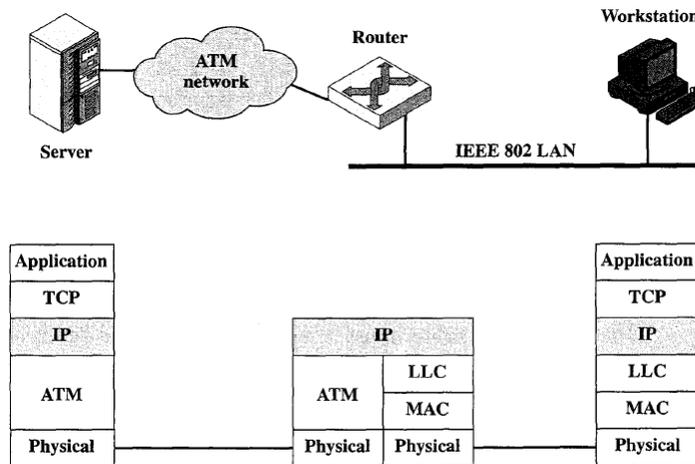


Figura 1.5: Enlace TCP/IP entre Servidor y PC [8].

El dispositivo que hace provecho de la capa 2, sin necesitar de la capa 3, es el *switch* (*link-layer switch* en la Fig. 1.1) que se encarga de redireccionar paquetes a los respectivos nodos de la red mediante la utilización de la información *MAC address* que es un parámetro único para cada dispositivo físico.

Los servicios principales que ofrecen los protocolos de la capa de enlace son:

1. **Entramado:** Se encapsulan los datagramas de la capa de red en una **trama** de la capa de enlace antes de transmitirla por el enlace. La estructura de la trama varía según el protocolo de la capa de enlace utilizado, pero en general contienen un campo de datos, donde se inserta el datagrama de la capa de red, y de una serie de campos de cabecera o *header* (incluidos los de cola).
2. **Acceso al enlace:** Un protocolo MAC, especifica las reglas para transmitir una trama a través del enlace. Cuando se tiene un enlace de difusión compartido por varios nodos es cuando se presenta el principal problema de acceso al medio. En este caso el protocolo MAC es primordial para coordinar la transmisión múltiple.
3. **Entrega fiable:** En el caso de los medios cableados que tienen baja tasa de error de bit generalmente no se implementa, sin embargo, en los medios inalámbricos y de difusión compartida es esencial. Para asegurar el envío de la información de manera fidedigna se utiliza reconocimiento y retransmisión de los paquetes en caso de detectarse errores en la trama.
4. **Detección y corrección de errores:** Siempre es posible que el hardware de detección de los pulsos electromagnéticos confunda un bit que fue transmitido como 0 como un 1 o viceversa. Estos errores se deben principalmente a la atenuación de señal y al ruido electromagnético. Para evitar el reenvío completo de la trama, el protocolo de enlace exige que el nodo transmisor incorpore ciertos bits de detección de errores en el *header*,

para que el nodo receptor realice una comprobación de errores en la trama e inclusive corrija los errores. Aunque depende del protocolo que tipo de detección y corrección de errores se usa, cabe destacar que existen tres métodos:

- (a) Comprobación de paridad (uni y bi dimensional)
- (b) Suma de comprobaciones
- (c) Códigos de redundancia cíclica (CRC)

La capa de enlace es una combinación de software y hardware, como se puede ver en la Fig. 1.6, parte del protocolo es ejecutado en la CPU y parte en el Chip del adaptador de red. La capa de enlace es el lugar de la pila de protocolos donde estos dos mundos se cruzan y se pasa del código a las señales que van a la antena o al medio de enlace de red.

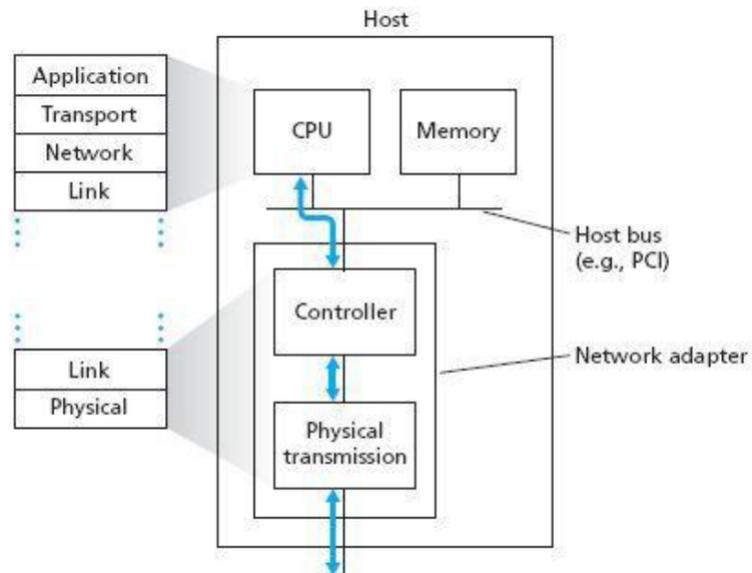


Figura 1.6: Ejemplo de un adaptador de red y como se entrecruzan los componentes de este con la pila de protocolos [1].

Protocolos de acceso múltiple

En el caso de los medios *wireless* de difusión compartida, existen una serie de protocolos de acceso múltiple al medio de transmisión que utilizan distintas técnicas para coordinar la transmisión simultánea de los nodos (decenas, cientos y hasta miles). A la fecha se han implementado docenas de protocolos de acceso múltiple al medio [1], no obstante, se pueden clasificar casi todos dentro de tres categorías:

- **Protocolos de particionamiento de canal**

- **Multiplexación por división de Tiempo (TDM):** A cada nodo se le asigna una trama de tiempo para transmitir, de manera cíclica cada nodo espera su turno para hablar.
- **Multiplexación por división de Frecuencia (FDM):** A cada nodo se le asigna una sección del ancho de banda total del canal, pueden transmitir simultáneamente

pero en una sección del espectro EM que se reduce a medida que incrementa la cantidad de nodos en la red.

- **Acceso múltiple por división de Código (CDMA):** A cada nodo emisor se le asigna un **código** conocido por el nodo receptor, los bits de información son codificados con este código y luego transmitidos en el medio. Si los códigos se seleccionan bien, varios nodos pueden transmitir y ser escuchados al unísono.
- **Protocolos de acceso aleatorio:** En este tipo de protocolos, cada nodo trata de transmitir cuando lo necesita y con el máximo ancho de banda del canal, pero si ocurre colisión de las tramas los emisores que no pudieron transmitir seleccionan un retardo aleatorio antes de transmitir otra vez.
 - **Protocolo ALOHA con ranuras de tiempo:** Protocolo muy descentralizado ya que cada nodo decide cuando retransmitir, lo que da una probabilidad de error muy alta a medida que aumenta la cantidad de nodos.
 - **Protocolo de acceso múltiple con sondeo de portadora (CSMA):** Se caracteriza porque cada nodo “escucha antes de hablar” (*Carrier Sense*) es decir que si otro nodo está transmitiendo en el mismo intervalo de tiempo, el nodo dejará de transmitir y esperará otra trama de tiempo para intentarlo nuevamente.
- **Protocolos de toma de turnos**
 - **Protocolo de sondeo (*Polling*):** En este caso, un nodo se torna el nodo maestro y sondea al resto de los nodos para asignarles cuando transmitir. El problema principal es que, si el nodo maestro deja de funcionar la comunicación completa se viene abajo. (ej: Bluetooth y 802.15)
 - **Protocolo de paso de testigo (*Token*):** En este caso, se utiliza un campo del *header* como testigo (*Token*) que es pasado entre los nodos junto a la trama. El nodo con el *token* tiene el turno para transmitir o entregar el *testigo* si es que no tiene nada que transmitir. (ej: 802.5 también llamado *Token Ring*)

En la Fig. 1.7 se puede apreciar cómo se divide la frecuencia (FDM) o el tiempo (TDM) para asignar los canales para cada emisor. Si bien, se dió una revisión general de los tipos de protocolo de acceso al medio y algunos ejemplos de cada uno, cabe destacar que el más importante para la comprensión de LoRa es CDMA, en el punto 1.2 se revisa esta relación.

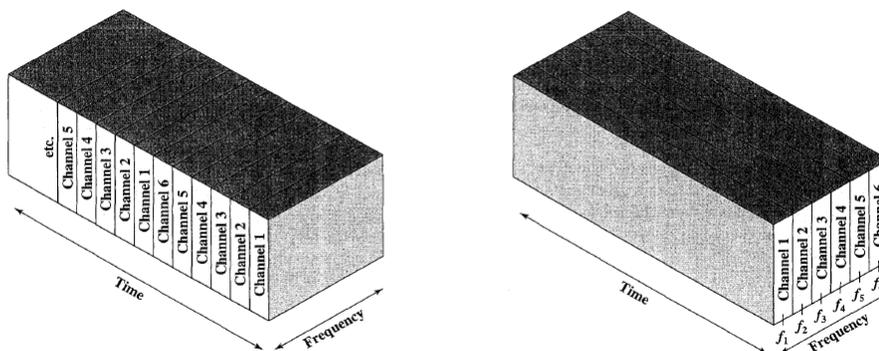


Figura 1.7: Diagrama de división de canales de TDM(izq.) y FDM (der.) [8].

1.1.4. Capa Física

En la capa física la unidad básica de información son los bits, los *frames* o tramas utilizadas en la capa de enlace son básicamente la unión de bits en paquetes. La capa física es el primer nivel, y especifica las normas físicas, eléctricas, funcionales y de procedimiento. En esta capa se definen valores máximos y mínimos para aspectos como el voltaje, frecuencia o la duración de los pulsos. Mientras, el trabajo de la capa de enlace es mover las tramas completas de un nodo de la red hasta el nodo adyacente, el trabajo de la capa física es el de mover los bits individuales de la trama de un nodo al siguiente. Como se puede ver en la Fig. 1.8 existen distintos protocolos de modulación de señales para transmitir la información por el canal EM. En cada caso, los bits se desplazan a través del enlace de forma diferente.

Esta capa abarca mucha teoría física y matemática compleja; el diseño de antenas, el procesamiento de señales, la propagación electromagnética, entre otros. Se sugiere el libro [9] para ver estos aspectos en profundidad, mientras que en esta sección se explican algunos detalles puntuales y necesarios para comprender LoRa.

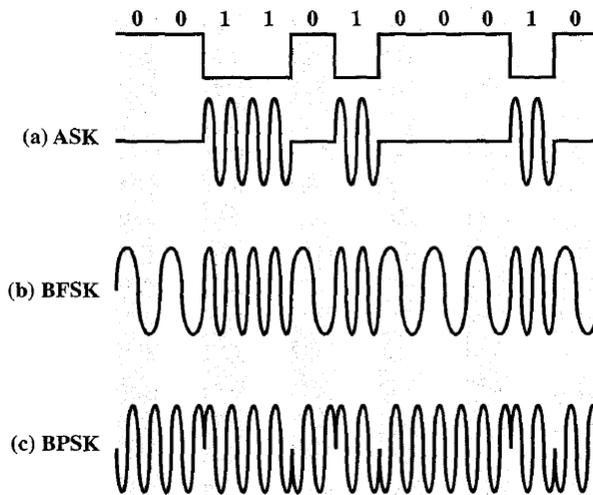


Figura 1.8: Distintas modulaciones para la misma información digital [8].

Es importante tener en consideración que existe una gran variedad de protocolos de capa física, por ejemplo, para Ethernet (protocolo de capa de enlace) existe un protocolo de capa física para cada medio de transmisión que se utilice al enlazar los elementos de la red; existe uno para el cable de par trenzado de cobre, otro para el cable coaxial y otro si el enlace es por fibra óptica.

Los objetivos principales de esta capa son:

1. Definir el medio físico por donde viaja la comunicación
2. Definir las características materiales y eléctricas que se usaran en la comunicación
3. Definir características funcionales de la interfaz

Gracias a las definiciones anteriores el *hardware* puede ejecutar las funciones principales de esta capa, las cuales son:

1. Transmitir el flujo de bits a través del medio
2. Manejar las señales eléctricas del medio de transmisión
3. Garantizar la conexión (aunque no la fiabilidad de dicha conexión)

1.1.5. Enlaces Inalámbricos

A continuación, se presentan algunos aspectos importantes para comprender los enlaces *wireless*.

Modulación

Un ejemplo de modulación similar a LoRa es *Multiple Frequency Shift Keying (MFSK)*, donde múltiples(M) frecuencias son usadas para diferenciar los símbolos (secuencias de bits) transmitidos [8].

$$\text{MFSK} : S_i(t) = A \cos 2\pi f_i t \quad ; \quad 1 \leq i \leq M. \quad (1.1)$$

Donde:

$$f_i = f_c + (2i - 1 - M)f_d$$

f_c = Frecuencia de portadora

f_d = Delta entre frecuencias i

M = Número de símbolos (tramas de bits) distintos = 2^L

L = Número de bits del símbolo

En la Tbl. 1.1 se puede ver el valor de las 8 frecuencias de transmisión para los 8 símbolos que proporciona la modulación 8-FSK en esta caso. El ancho de banda requerido del canal para transmitir un mensaje usando esta modulación es igual a $W_d = 2Mf_d = 400[kHz]$ (entre los 50[kHz] y los 450[kHz]) y el *data rate* es igual a $2Lf_d = 150kbps$.

f_i	kHz	trama	f_i	kHz	trama
f_1	75	000	f_5	275	100
f_2	125	001	f_6	325	101
f_3	175	010	f_7	375	110
f_4	225	011	f_8	425	111

Tabla 1.1: Frecuencias de modulación (M=8) 8-MFSK dado: $f_c=250[kHz]$, $f_d=25[kHz]$ y L=3bits.

En la Fig. 1.9 se puede ver la distribución de los símbolos enviados para una modulación 4-FSK, los casilleros ennegrecidos representan un símbolo enviado en una ventana de tiempo T_s y en un ancho de banda centrado en f_i . El Ancho de banda total requerido para este canal es de W_d .

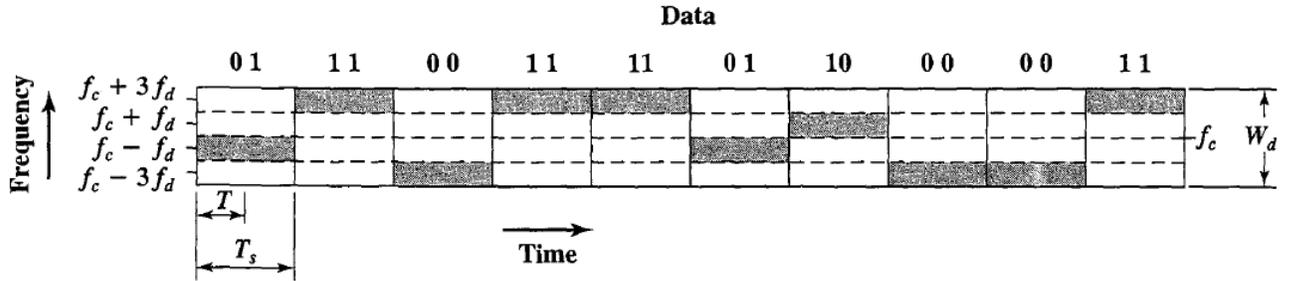


Figura 1.9: Uso del espectro EM para transmisión 4-FSK [8].

Capacidad del Canal

La capacidad del canal se define como la máxima velocidad de transmisión de información (*data rate* [bps]) que soporta determinado canal, lo que para efectos del enlace de comunicación entre ambos nodos es la velocidad de transmisión de bits de cada trama y depende de manera intrínseca del protocolo de transmisión de la capa de enlace y la capa física; que tanta información se envía y cada cuanto tiempo. Hay 4 conceptos fundamentales:

1. **Data Rate:** La velocidad de transmisión de información medido en bit por segundos [bps]
2. **Bandwidth:** El ancho de banda de la señal transmitida, rango de frecuencia en que la señal es emitida, medido en ciclos por segundo o Hertz[Hz]
3. **Noise:** En este caso referido al nivel de ruido promedio sobre el canal de comunicación
4. **Error rate:** La tasa de fallo o error de bits transmitidos (confusión de 0 y 1).

En general, se desea obtener un uso eficiente del *bandwidth*, tanto como sea posible. En el caso de la transmisión digital se describe como: obtener un *data rate* tan alto como sea posible para un límite particular de *error rate* en un *bandwidth* dado. El principal problema para obtener tal eficiencia es el *Noise*.

El parámetro clave para evaluar esta relación o eficiencia es el *SNR*, que más detalladamente es la relación entre la potencia de una señal (señal emitida) y la potencia contenida en el ruido, la señal recibida es la suma de la señal emitida más el ruido. Típicamente se mide en el receptor, pues es aquí donde se requiere procesar la señal y eliminar el ruido no deseado. Por conveniencia se mide en *decibels* [dB]. Un alto SNR significa una alta calidad de señal.

Los decibelios o [dB], son unidades logarítmicas que son usadas comúnmente para representar la potencia de las RF. Los [dBm] son una unidad de medida de potencia expresada en [dB] relativa a un miliWatt [mW]. Para convertir de miliWatts [mW] a [dBm] se utiliza la siguiente conversión:

1. (mW a dBm) $P_{dBm} = 10 * \log_{10} P_{mW}$
2. (dBm a mW) $P_{mW} = 10^{P_{dBm}/10}$

Mejora de sensibilidad con SS

En teoría de la información, el teorema de Shannon-Hartley define el máximo *data rate* al que la información puede ser transmitida por un canal de comunicación AWGN y en un *bandwidth* definido. El teorema, en la Ecu. 1.2 establece la capacidad del canal de Shannon y define el máximo *data rate* al que la información puede ser transmitida por el canal [10].

$$C = B * \log_2\left(1 + \frac{S}{N}\right) \quad (1.2)$$

Donde:

C = Capacidad del canal (bps)

B = *Bandwidth* del canal (Hz)

S = Potencia promedio de la señal recibida (Watts)

N = Potencia promedio del ruido (e interferencia) (Watts)

$\frac{S}{N}$ = Proporción de señal a ruido (SNR).

Reescribiendo la Ecu. 1.2 con las propiedades del logaritmo natural, se obtiene la siguiente ecuación:

$$\frac{C}{B} = 1,433 * \frac{S}{N} \quad (1.3)$$

Para las aplicaciones de SS el SNR suele ser pequeño, ya que la potencia de la señal puede ser menor que el piso de ruido [10]. Asumiendo un nivel de ruido tal que $\frac{S}{N} \ll 1$ la Ecu. 1.3 puede ser reescrita de la siguiente forma:

$$\frac{C}{B} \approx \frac{S}{N} \quad o \quad \frac{N}{S} \approx \frac{B}{C} \quad (1.4)$$

De la Ecu. 1.4 podemos ver que para transmitir información libre de error en un canal de SNR fijo, solo es necesario aumentar el *Bandwidth* de la señal transmitida (B) para aumentar la sensibilidad del receptor.

CDMA mediante Spread Spectrum

Esta es una importante técnica de comunicación y se puede utilizar para transmitir información digital y analógica, mediante el uso de señales analógicas. La idea principal es esparcir la señal de información sobre un ancho de banda más amplio para dificultar el bloqueo y la interceptación.

Existe un tipo particular de *Spread Spectrum* (SS) llamado DSSS, que es similar al proceso de modulación de LoRa. En la Fig. 1.10 se puede ver un ejemplo de la multiplexación de la información por un *spreading code* (*Locally generated PN bit stream*) en el emisor y como es recuperada mediante la misma técnica en el receptor. La frecuencia de la información ($1/T$)

aumenta cuatro veces al ser multiplexada con el código ($1/T_c$), un ejemplo de este proceso se puede apreciar en la Fig. 1.11.

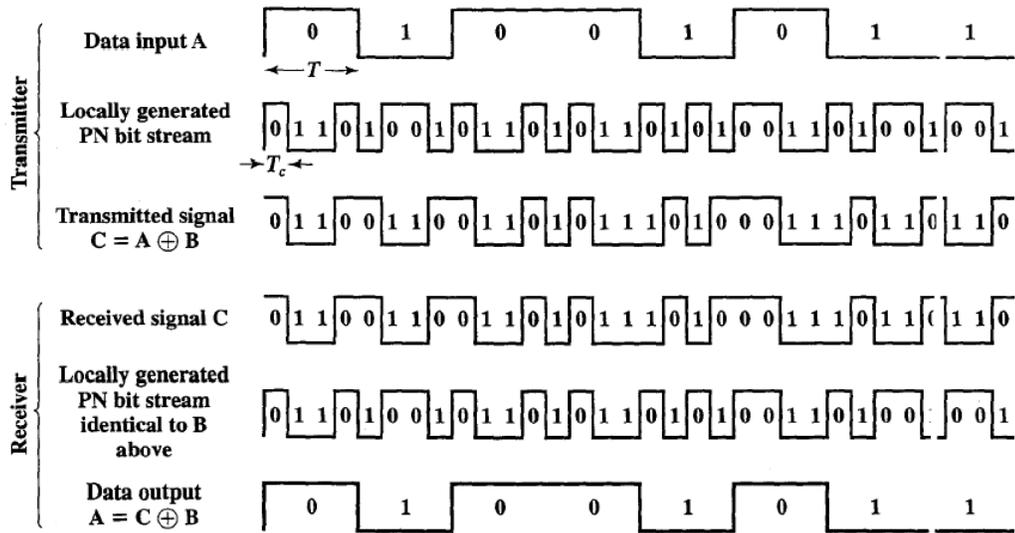


Figura 1.10: Ejemplo de DSSS [8].

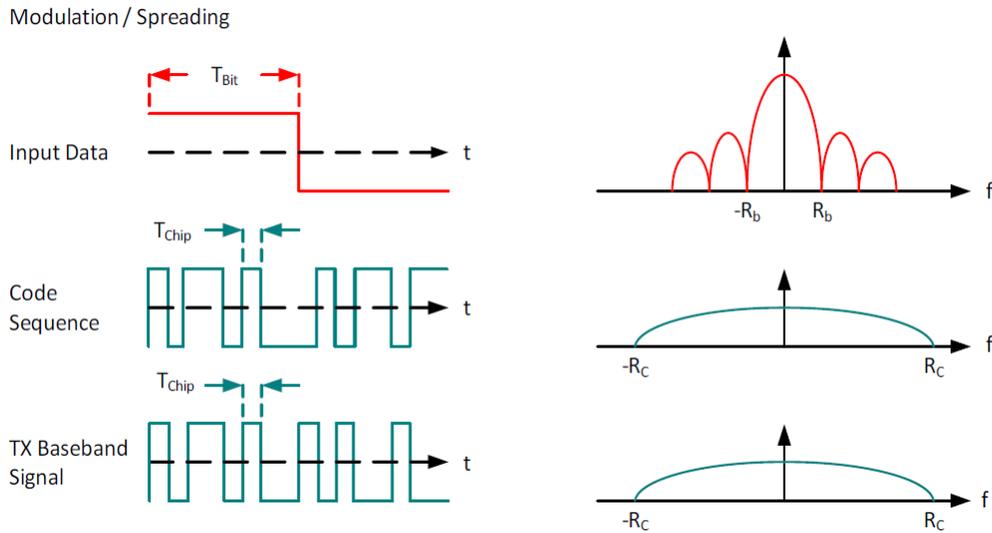


Figura 1.11: Ejemplo de esparcimiento SS [10].

La característica principal de un sistema SS es que la señal de información alimenta un codificador de canal que produce una señal analógica con ancho de banda relativamente estrecho. Esta señal es modulada usando una secuencia de dígitos conocidos como código de esparcimiento (*spreading code*) o *spreading sequence*. El efecto de esta modulación es incrementar significativamente el ancho de banda (de ahí *spread spectrum*) de la señal transmitida. En el receptor, la misma *spreading sequence* es utilizada para demodular la señal de *spread spectrum*. SS ofrece tres mejoras importantes [8]:

1. La señal gana cierto grado de inmunidad a varios tipos de ruido y distorsión *multipath*
2. También es usada para esconder y encriptar la señal. Solo el receptor que tiene el *spreading code* puede recuperar la información codificada.
3. Varios usuarios pueden usar independientemente todo el ancho de banda del canal, con muy poca interferencia, de aquí su relación con CDMA.

1.2. Redes inalámbricas de área amplia y baja potencia (LPWAN) y LoRa

1.2.1. LPWAN

Aunque el término LPWAN fue acuñado el 2013 para hacer referencia a un pequeño y definido grupo de tecnologías emergentes, terminó por ser ampliamente aceptado por la industria. Como se especificó antes, los aspectos fundamentales son *Low Power*, *Wide Area*, bajos costos de *hardware*, alta densidad de nodos y baja tasa de transmisión. La penetración de las tecnologías para LPWAN ya es alta y una realidad en ciertas partes del mundo, mientras que en otras aún es una oportunidad, por ejemplo [11] :

1. **Francia:** Cuenta con una cobertura nacional
2. **España, UK y USA:** Tiene un nivel de cobertura significativa y creciente.
3. **China y varios países de Europa del Oeste:** Tienen varios proyectos avanzados, aunque sin una cobertura significativa a nivel Nacional y se espera que en pocos años ya tengan un acceso más masivo.
4. **África y América Latina:** Tienen algunos desarrollos, sin embargo, la integración de las tecnologías LPWAN aún es lenta debido al desarrollo de estos mercados y restricciones legales.

La principal razón de la rápida penetración de las tecnologías LPWAN a nivel mundial (una lista exhaustiva de las tecnologías y desarrollos de redes LPWAN en cada país se puede ver en [4]) se debe a que muchas aplicaciones (en diversos mercados) requieren de necesidades técnicas que se ajustan muy bien a las características que estas tecnologías ofrecen. Como podemos ver en la Fig. 1.12 existe un abanico bastante grande de aplicaciones, desarrolladas y en desarrollo, que usarán este tipo de conectividad para el 2023.

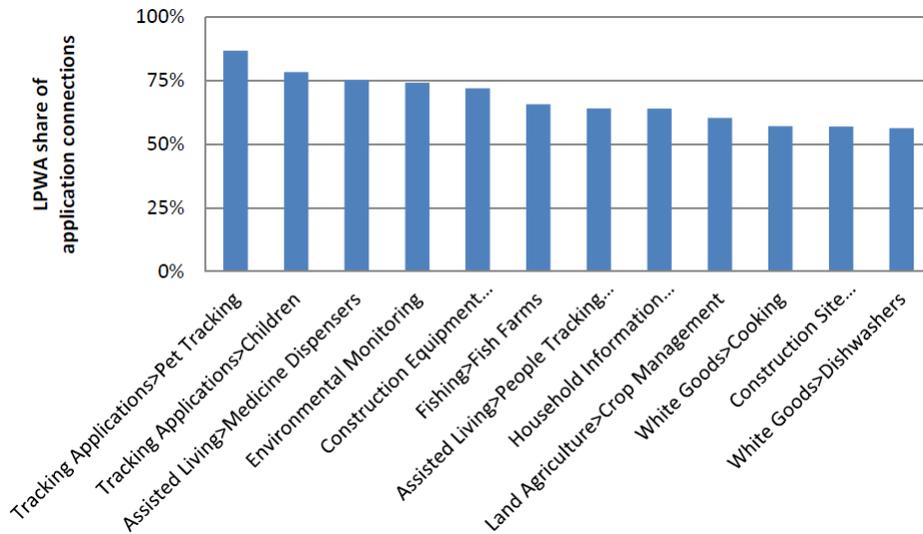


Figura 1.12: Tasa de conectividad LPWAN por aplicación esperada para el 2023 basado en los datos del 2015 [11].

Es importante recordar que los operadores de redes celulares deben participar con los proveedores de tecnologías LPWAN para establecer estrategias de adaptación viables en cada país. Como se puede ver en la Fig. 1.13 las tecnologías para el LPWAN están diseñadas para comunicación en ambientes de redes M2M, con bajo ancho de banda y amplia cobertura, comparado a las redes celulares donde se apostó por privilegiar aumentar el ancho de banda como se ve en 4G y 5G. Las LPWAN impulsan la penetración y expansión del M2M y el IoT.

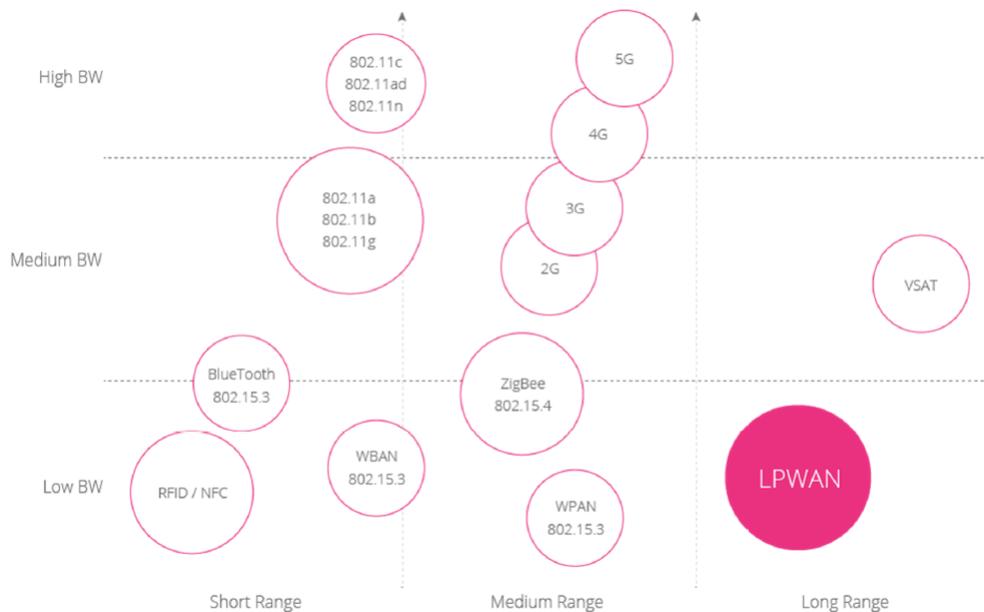


Figura 1.13: Comparación de tecnologías inalámbricas según ancho de banda (BW) y Rango de cobertura[12].

1.2.2. LoRaWAN

LoRaWAN™, como se puede ver en azul en la Fig. 1.14, es un protocolo de comunicación diseñado para optimizar el uso del protocolo de capa física LoRa, dando soporte en las capas superiores como la capa de enlace, de red y de transporte [13]. Además, en la Fig. 1.14 se ve en rojo la zona y frecuencia sobre la que se trabajó en este estudio. El uso de LoRaWAN como protocolo y arquitectura de red tiene mucha influencia en parámetros fundamentales como el tiempo de vida de la batería de los nodos, la capacidad de la red, la *Quality of Service (QoS)* y la seguridad de los datos.

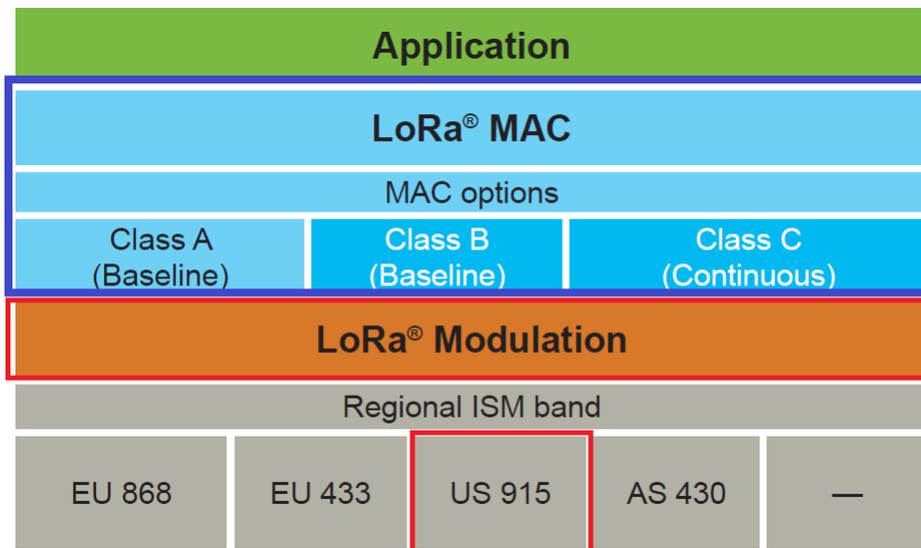


Figura 1.14: Ubicación de LoRaWan en el stack de protocolos de comunicación [13].

Como se puede ver en la Fig. 1.15, LoRaWAN utiliza una arquitectura de red distribuida, en donde varios terminales (también llamados *end-points* o *end-nodes*) se conectan a uno o más *gateway*, LoRaWAN logra optimizar los parámetros antes nombrados mediante la gestión eficiente de la comunicación entre los nodos con su respectivo *gateway* y desde este con los servidores de red de Internet (paso a TCP/IP). Aunque LoRa está pensado para funcionar como topología estrella, LoRaWAN define que la información enviada por los *end-nodes* puede ser escuchada por más de un *gateway* (lo que simplifica los protocolos de enlace para nodos móviles), luego en las capas superiores (servidores de red) se agrega la inteligencia y complejidad de filtrar paquetes redundantes, ordenar las tablas de conexión optimas y definir la tasa de datos adaptativa (*Adaptive Data Rate (ADR)*).

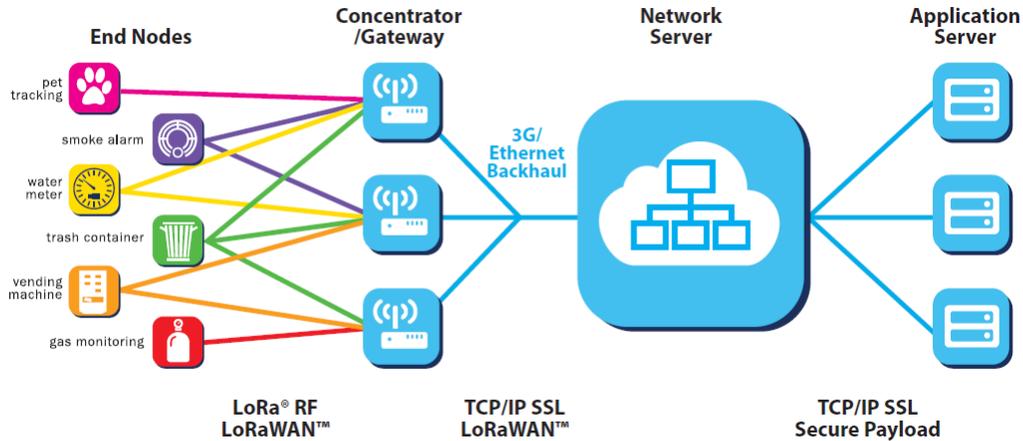


Figura 1.15: Diagrama de conexión de un sistema LoRaWAN[13].

En LoRaWAN se define que los nodos son asíncronos y se comunican cuando tiene datos para ser enviados, similar a ALOHA, lo que crea un sistema de coordinación descentralizado. Esto permite un ahorro considerable de energía ya que los nodos no necesitan estar coordinándose constantemente con el nodo maestro o el resto de los nodos.

1.2.3. LoRa

LoRa™ es un esquema de modulación de espectro ensanchado (SS) y como tecnología comercial es implementado en un circuito integrado (*Chip transceivers*) como un protocolo de modulación de capa física, es utilizado para crear enlaces de comunicación inalámbricos de largo alcance [13]. El atractivo comercial de LoRa se debe a que un solo *gateway* permite generar una red de topología estrella y de largo alcance, capaz de sostener miles de nodos conectados transmitiendo con bajo *duty cycle*.

LoRa está basado y utiliza una técnica de modulación llamada *CSS*, en este caso el *spread code* es conocido como *chirp sequence* o pulso *chirp*. El *chirp sequence* sucede a una frecuencia mucho mayor que la señal de información, esto produce el esparcimiento del *BandWidth* (BW) de la señal transmitida. CSS ha sido utilizado en comunicaciones espaciales y militares por décadas para lograr largas distancias de comunicación y robustez frente a la interferencia, pero LoRa es la primera implementación de bajo costo para uso comercial [13].

En LoRa predomina el tráfico del *uplink* sobre el del *downlink*, esto se debe a que su uso esta pensado para que los *end-device* manden información al *Gateway*. Según la región y sus regulaciones LoRa puede funcionar en las bandas ISM de 433[MHz], 868[MHz] o 915[MHz].

CSS

CSS es la técnica de modulación de radio que ocupa LoRa en su capa física [14]. La modulación CSS usa pulsos *chirp* con una variación lineal de la frecuencia en el tiempo para codificar la información. Debido a la linealidad de los pulsos *chirp*, los desfases de frecuencia entre el receptor y el transmisor son equivalentes a los desfases en el tiempo, lo que hace que esta modulación tenga inmunidad al efecto Doppler [14]. Otras de sus ventajas son el bajo requerimiento de potencia de transmisión, y su inherente robustez a las degradaciones del canal por *multipath*, *fading*, *doppler* e *Inter Symbol Interference (ISI)*.

Pulsos *Chirp*

CSS es una aplicación derivada de CDMA y similar a DSSS, se podría definir como un tipo modulación de espectro ensanchado con acceso múltiple por división de códigos. Como se explicó antes su funcionamiento se basa en la utilización de pulsos *chirp* [15], que incrementan o disminuyen su frecuencia durante un periodo de tiempo, en la Fig. 1.16 se pueden apreciar los dos tipos de pulsos utilizados, *upchirp* y *downchirp*.

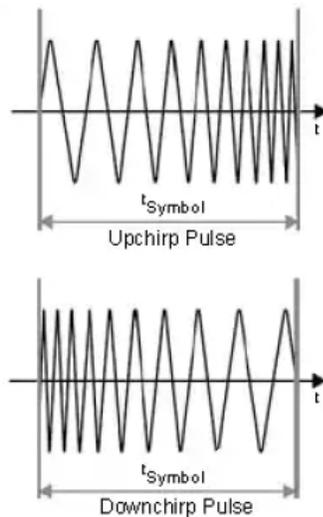


Figura 1.16: Pulsos *upchirp* y *downchirp* [15].

Un pulso *chirp* es medido usando el *Chirp Rate* que define la cantidad de [chirps/segundo]. Un dato interesante es que los pulsos *chirp* son usados en la naturaleza desde hace miles de años por delfines y murciélagos para la ecolocalización y recién fueron patentados durante 1940 para su aplicación en radares.

Processing Gain G_p

LoRa mejora la sensibilidad del receptor mediante la respectiva ganancia de procesamiento G_p que le da el método CSS. La cantidad de ensanchamiento es dependiente de la relación de

“chirps por bit”, en otras palabras, la relación del *chirp sequence* para un *data rate* deseado. La ganancia G_p es comúnmente expresada en [dB], en la Ecu. 1.5 se puede ver como se calcula el:

$$G_p = 10 * \log_{10}\left(\frac{R_c}{R_b}\right)[dB] \quad (1.5)$$

Donde:

$R_c = \text{Chirp-Rate}(\text{chirps/sec})$

$R_b = \text{Data-Rate} (\text{bits/sec})$

Gracias a esta ganancia LoRa es capaz de recuperar correctamente la señal transmitida aún cuando el SNR es negativo. El *BandWidth* de frecuencia del *chirp sequence* es equivalente al *BandWidth* espectral de la señal transmitida. Esto quiere decir que el valor del *Chirp-Rate* es igual que el BW programado (se define en el datashet[16] como chirp-per-second-per-hertz) y puede tomar valores de 125[kHz], 250[kHz] y hasta 500[kHz].

Parámetros de interés en LoRa

La modulación LoRa permite cambiar tres parámetros fundamentales que definen el desempeño del enlace de comunicación, estos son *BandWidth* (BW), *Spread Factor* (SF) y *Code Rate* (CR).

Un símbolo en LoRa está compuesto por 2^{SF} *chirps* que cubren el ancho de banda completo, debido a esto un símbolo puede efectivamente codificar SF bits de información y por ende SF bits de información son transmitidos por símbolo. Definiendo el periodo por símbolo (T_s) y el *Symbol Rate* (R_s) como:

$$\frac{1}{R_s} = T_s = \frac{2^{SF}}{BW}[\text{sec/symbol}]$$

Se puede definir el *Chirp-Rate* R_c como:

$$R_c = R_s[\text{symbol/sec}] * 2^{SF}[\text{chirps/symbol}] = BW[\text{chirps/sec}]$$

Dado que los paquetes de LoRa también incluyen el parámetro CR con (n=1,2,3,4), definido como:

$$CR = \frac{4}{(4 + n)}$$

Finalmente, el *Data-Rate* (R_b) deseado puede ser expresado de la siguiente forma:

$$R_b = SF * \frac{CR}{\frac{2^{SF}}{BW}}(\text{bits/sec}) \quad (1.6)$$

Donde:

$SF = \text{Spread Factor}$ (de 7 a 12)

$BW = \text{BandWidth}$ de la modulación (Hz)

Considerando la ecuación anterior, se puede decir que para una configuración de LoRa de $BW=125[\text{kHz}]$, $SF=7$ y $CR=4/5$, se tiene un *Data-Rate* $R_b=5.5[\text{kbps}]$.

Relación entre parámetros

El *Chirp-Rate* depende solo del BW, esto tiene serias consecuencias en la modulación, por ejemplo, el incremento en una unidad de SF dividirá el rango de frecuencia de un *chirp* en dos y multiplicará la duración del símbolo por dos. Básicamente el *Symbol-Rate* y el *Data-Rate* para un SF dado, son proporcionales al ancho de banda de la frecuencia, por lo tanto, duplicar el BW duplicará la tasa de transmisión.

Estos parámetros también afectan la sensibilidad del receptor, se debe recordar que la sensibilidad se define como la potencia mínima a la que la señal emitida debe llegar al receptor para ser escuchada. Por ejemplo:

- Aumentar el BW disminuye la sensibilidad del receptor (empeorándola).
- Aumentar el SF incrementa la sensibilidad del receptor (mejorándola).
- Disminuir el CR ayuda a reducir el PER. Para paquetes con CR de 4/8 se tendrá más tolerancia a la interferencia que un paquete con CR de 4/5 ($4/8 < 4/5$).

En la Tbl. 1.2 se puede ver la sensibilidad esperada en el receptor para distintas configuración de SF y BW, y como se cumplen las relaciones antes descritas.

BW	Spread Factor					
	7	8	9	10	11	12
125[kHz]	-123	-126	-129	-132	-133	-136
250[kHz]	-120	-123	-125	-128	-130	-133
500[kHz]	-116	-119	-122	-125	-128	-130

Tabla 1.2: Sensibilidad en el receptor en [dBm] para distintos Bandwith y spread factors [16].

Frame Format para el paquete de LoRa

El formato definido para los paquetes de LoRa se puede observar en la Fig. 1.17, las partes fijas son el **Preámbulo** y el **Payload**, mientras que de forma opcional se pueden incluir el **Header** y el **CRC Payload**.

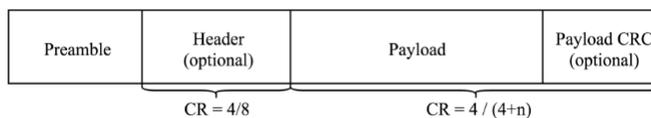


Figura 1.17: LoRa *Frame Format*[14].

- **Preámbulo:** El preámbulo se encarga de sincronizar al emisor y el receptor, comienza con una secuencia de *upchirps* que cubren el ancho de banda completo. Los últimos dos *upchirps* codifican la palabra de sincronización (*sync word*), esta es un valor de 1 [byte] utilizado para diferenciar más de una red de LoRa en la misma banda de frecuencia, si la *sync word* no corresponde el receptor deja de escuchar el mensaje. Finalmente, luego de la *sync word* vienen dos y cuarto *downchirp* con una duración de 2,25 símbolos. El preámbulo completo puede durar entre 10,25 y 65.539,25 símbolos.
- **Header:** Este campo opcional tiene un CR de 4/8 e indica el tamaño del campo *Payload* en 1 [byte], el valor de CR usado en el campo *CRC Payload* en 1 [byte] y la existencia de este último campo en 1[byte]. El campo tiene un ancho variable para ajustar el SF pero mantiene su relación de CR 4/8.
- **Payload:** Campo de datos transmitidos, que va de 2 a 255 bytes.
- **CRC Payload:** Campo de *Forward error correction code* que utiliza la corrección de errores CRC de 16 bits para el Payload.

En la Fig. 1.18 se puede apreciar en detalle los *upchirps* y *downchirps* enviados en el preámbulo y los símbolos enviados en el *payload*. El tamaño mínimo de un mensaje es de 8 símbolos por paquete.

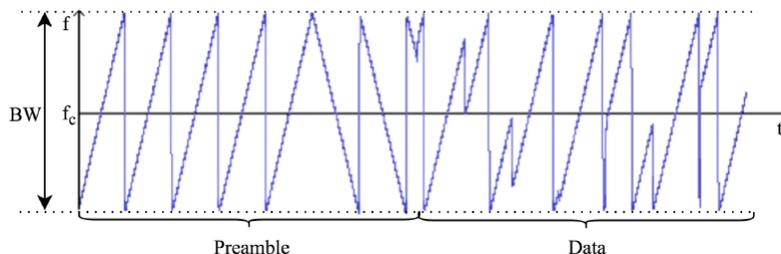


Figura 1.18: Mensaje de radio LoRa[14].

Códigos ortogonales

Los códigos ortogonales aseguran que para todo par de códigos utilizados para dar acceso al medio y ensanchar el espectro del mensaje, se cumple que la correlación cruzada es cero, esto se hace algebraicamente para lograr que el solapamiento de las señales cumple con cierta independencia lineal permitiendo separar entre la transmisión de un nodo y otro. Asegurando esta característica en las *chirp sequence* de CSS se pueden escuchar y demodular transmisiones al unísono en el receptor[7].

1.3. SNR, pérdidas y Link Budget

La señal inalámbrica es modulada y transformada por la antena del emisor en una onda EM, viaja por el aire y llega hasta la antena del receptor donde se transforma nuevamente en una señal eléctrica. En el receptor la señal es demodulada y transformada en pulsos de información. El objetivo principal del demodulador es recuperar el pulso de información (banda base) con el mejor SNR posible y libre de ISI.

1.3.1. SNR

La razón de señal a ruido SNR en un sistema de comunicaciones inalámbrico, hace referencia a la razón entre la potencia de la señal emitida por el transmisor (recibida en el receptor) con respecto a la potencia de las señales de ruido recibidas en el receptor. En la ecuación del SNR [17], que se puede ver en la Ec.1.7, el término P_{SR} es la Potencia de la señal recibida y P_R la potencia del ruido.

$$SNR = \frac{P_{SR}}{P_R} \quad (1.7)$$

En las comunicaciones digitales, es relevante replantear la ecuación del SNR en términos de energía por bit con respecto a la densidad espectral de P_R que se puede expresar como $\frac{E_b}{N_0}$. De esta forma se replantea la Ecu. 1.7 para que quede como la Ecu. 1.8, en términos de la tasa de velocidad de datos R_b [$\frac{bits}{s}$] y el ancho de banda BW del canal en [Hz].

$$SNR = \frac{R_b}{BW} * \frac{E_b}{N_0} \quad (1.8)$$

En la Fig. 1.19 se muestra la tasa de error de bits BER, que dicho de forma simple es la probabilidad de que un bit llegue de forma errónea al receptor, en función del SNR para tres técnicas distintas de modulación. A pesar de que se podría incrementar el SNR aumentando la potencia de transmisión para disminuir el BER esto es algo poco práctico en redes LPWAN donde se busca disminuir el costo de energía de las transmisiones, es por esto que LoRa utiliza otro tipo de métodos (ej:ADR) para alcanzar un desempeño óptimo.

Generalmente para diseñar un enlace inalámbrico se define un BER tolerable que permita definir el umbral mínimo de tasa de error que sea aceptable para considerar la comunicación exitosa, y se espera que el desempeño en la práctica del radioenlace sea al menos mayor a ese BER.

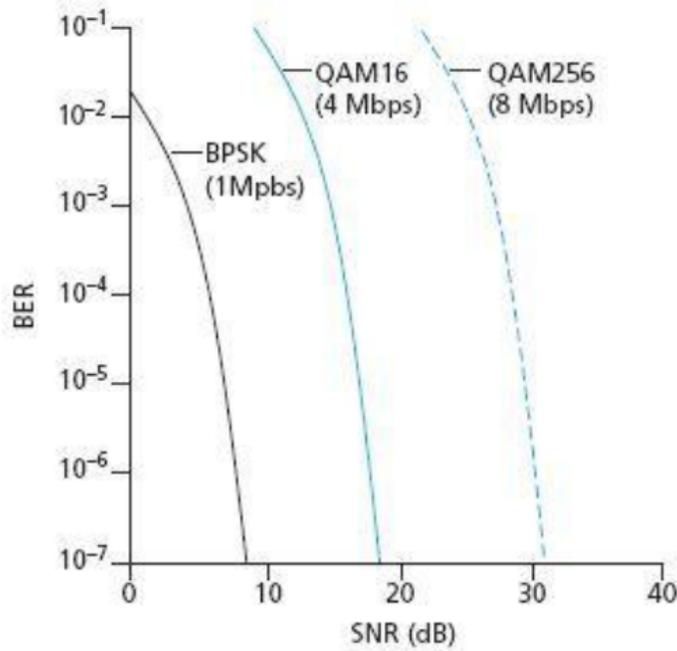


Figura 1.19: Forma común de la curva BER vs E_b/N_0 , para distintas técnicas de modulación [8].

Ruido y Pérdidas

Existen tres factores fundamentales en la degradación del SNR, por un lado, las pérdidas que hacen referencia a la energía de la señal transmitida que no llega al receptor por efectos de absorción, reflexión, difracción y scattering, por otro lado, el ruido que hace referencia a la energía de ondas que generan interferencia junto a la señal transmitida y finalmente el ruido térmico. Estos factores afectan la inteligibilidad de la señal original al llegar al receptor.

En este caso la razón $\frac{E_b}{N_0}$ es una medida del SNR normalizado donde el N_0 se calcula como el ruido térmico, el cual es modelado como un ruido blanco Gaussiano (AWGN) y su fórmula se muestra en la Ecu. 1.9 donde κ es la constante de Boltzman (-228.6 [dB*Watts/(K*Hz)]) y T° la temperatura en grados kelvin. En la Fig. 1.20 se puede apreciar la pérdida de información transmitida debido al ruido térmico N_0 .

$$N_0 = \frac{N}{W} = \frac{\kappa * T^\circ * W}{W} = \kappa * T^\circ [\text{watts/Hz}] \quad (1.9)$$

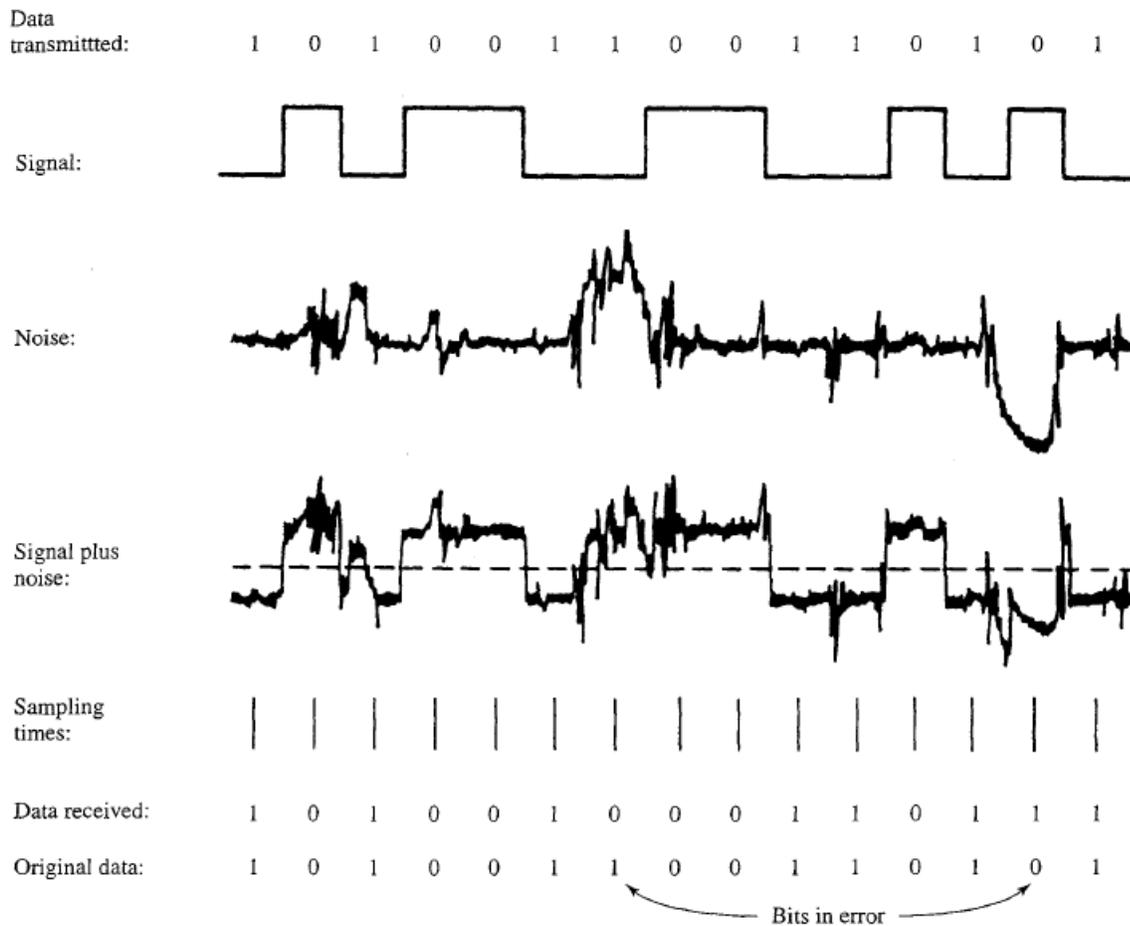


Figura 1.20: Ejemplo de un error de bit, causado por ruido AWG[8].

Como ya se esclareció anteriormente, las transmisiones inalámbricas tienen mayor tasa de error de bits que los medios cableados, es por esto que los protocolos de enlaces inalámbricos emplean potentes códigos de CRC y transferencia fiable, con el objetivo de retransmitir tramas erradas. Además, los protocolos de capa física se diseñan para tratar de resolver las problemáticas intrínsecas de los canales de enlace *wireless*, algunas de éstas complejidades son [1]:

- **Intensidad Decreciente de la señal (Fading):** La radiación electromagnética se atenúa a medida que atraviesa el canal, inclusive en el espacio vacío debido a que se dispersa, como resultado la intensidad de señal decrece siempre a medida que se incrementa la distancia entre el emisor y el receptor. Esto también es conocido como **pérdida de propagación** o *path loss* o *free space loss*.
- **Interferencia de otras fuentes:** Un receptor que recibe la intensidad de señal en cierta banda de frecuencia, escuchará todas las fuentes de radio que transmiten en esa banda, por lo tanto, todas las fuentes que no sean el emisor como tal generarán interferencia. Además de otras fuentes de transmisión, existe el ruido electromagnético (EM) presente en el entorno que también genera interferencia pues está distribuido en todas las frecuencias del espectro EM.

- **Pérdida de propagación multicamino:** Este efecto, también conocido como *multipath fading*, tiene lugar cuando partes de la onda EM se reflejan en los objetos y en el suelo, tomando caminos de diferentes longitudes entre el emisor y el receptor. Como resultado la señal del emisor llega desde distintos ángulos y con diferentes retardos. Se pueden ver tres ejemplos importantes de *multipath* en la Fig. 1.21; Reflexion (R), Scattering (S) y Difraccion (D).

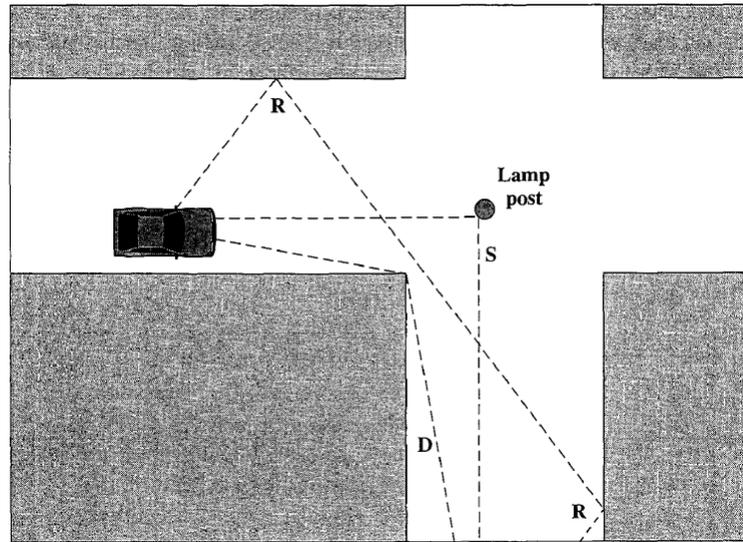


Figura 1.21: Ejemplos de *multipath propagation*[8].

Existe un modelo de desvanecimiento del canal llamado Rayleigh que se utiliza para ambiente *outdoors*. Este *fading model* considera el peor caso para el escenario en que no existe una trayectoria definida como Line of Sight (LOS), si no que múltiples trayectorias entre el transmisor y Receptor. El *Rayleigh Fading Model* es utilizado en ambientes densamente urbanos, para calcular analíticamente el modelo de desvanecimiento. Como se ve en la Fig. 1.22 este desvanecimiento es típicamente lento y plano a medida que se aumenta el radio de cobertura (en caso de cobertura omnidireccional).

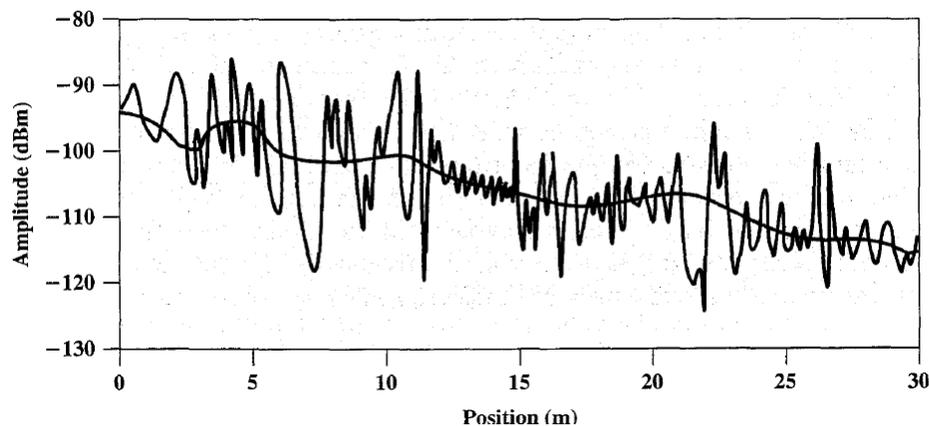


Figura 1.22: Típico desvanecimiento lento y plano en ambientes urbanos[8].

Si volcamos nuestra atención en el nodo receptor de la señal EM y consideramos las problemáticas del medio antes descritas, nos damos cuenta que la señal recibida en la antena receptora es una combinación de la señal original emitida por el emisor, degradada por el medio, y del ruido de fondo. Es debido a esto que la relación de señal a ruido (*SNR*) en unidades de decibelios (dB), es la medida más utilizada para evaluar el desempeño de un protocolo de comunicación *wireless*, mientras mayor sea el SNR más fácil será para el receptor extraer la trama transmitida del ruido de fondo.

1.3.2. Link Budget

El link budget (típicamente dado en [dB]) es el factor primario para determinar el alcance del enlace inalámbrico en ambientes dados, algunos de los parámetros del *link budget* son estadísticos como el margen de desvanecimiento de la señal. El cálculo del *linkbudget* es una técnica de estimación para evaluar el rendimiento del sistema de comunicación ante el error.

El *linkbudget* es un balance de las ganancias y pérdidas, describe detalladamente el aporte de cada parte del transmisor y del receptor, fuentes de ruido, atenuación de señal y efecto de atravesar el canal.

Semtech ofrece una herramienta para el cálculo del *link budget* y otros aspectos del radio-enlace tales como *Data-Rate*, consumo de energía y sensibilidad esperada en el receptor.

En la Fig. 1.23 se puede ver que para los parámetros de radio SF=12, BW=500[kHz] y CR=4/5, transmitiendo paquetes con un payload de 8 bytes, en la frecuencia de 918.8[MHz] y con un Tx *power* de 14[dBm] se obtiene un Link Budget de 151[dB], suficiente para entablar un enlace inalámbrico de largo alcance (en condiciones ideales), considerando la sensibilidad calculada del receptor en -137[dBm].

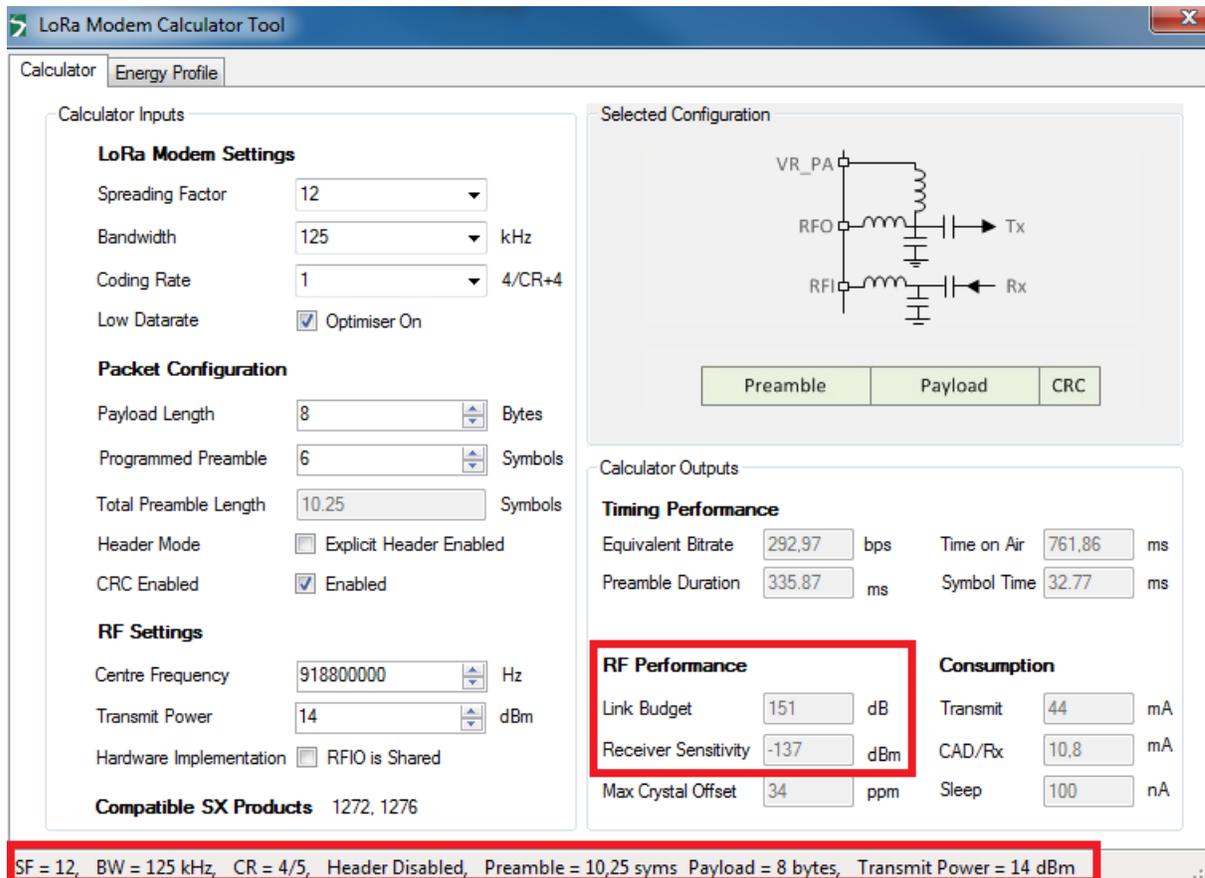


Figura 1.23: Herramienta de Semtech para el cálculo de parámetros de radio de LoRa.

1.4. Aspectos Regulatorios

En el mundo y en ley chilena, existen algunos aspectos regulatorios que competen a las bandas de RF utilizadas por los sistemas M2M con base en LoRa, de interés particular para la banda de 915[MHz], pero igualmente para las bandas de 868[MHz] y 433[MHz]. A continuación, se especifica un poco más al respecto.

1.4.1. Regulaciones a nivel mundial

A nivel Mundial, la *Federal Communications Commission (FCC)* y otras entidades como la UIT, han definido una serie de regulaciones para establecer los niveles de emisión y el uso para varios rangos de frecuencias EM. En particular se definió ciertas bandas como *Industrial, Scientific and Medical (ISM) bands* las que pueden ser utilizadas de forma libre para la investigación. Entre las bandas ISM las más utilizadas en Estados Unidos, México, Canadá, Israel y Australia son las de 915[MHz] y la de 2.4[GHz], el rango completo de estas bandas es de 902[MHz] a 928[MHz] y 2.4000[GHz] a 2.4835[GHz].

En la Fig. 1.24 se puede ver un resumen de otras regulaciones a nivel mundial, aunque la que más compete es la de *North America*. Es interesante ver que la potencia de salida (*Tx Power Up*) está permitida hasta los +30[dBm], o 1000[mW], muy por encima de lo que se encuentra en las referencias de la regulación nacional.

	Europe	North America	China	Korea	Japan	India
Frequency band	867-869MHz	902-928MHz	470-510MHz	920-925MHz	920-925MHz	865-867MHz
Channels	10	64 + 8 +8	In definition by Technical Committee			
Channel BW Up	125/250kHz	125/500kHz				
Channel BW Dn	125kHz	500kHz				
TX Power Up	+14dBm	+20dBm typ (+30dBm allowed)				
TX Power Dn	+14dBm	+27dBm				
SF Up	7-12	7-10				
Data rate	250bps- 50kbps	980bps-21.9kbps				
Link Budget Up	155dB	154dB				
Link Budget Dn	155dB	157dB				

Figura 1.24: Resumen de regulaciones a nivel mundial[13], pertinentes a LPWAN y aplicables a LoRa.

1.4.2. Chile

En el caso de Chile, el espectro EM está regido en primera instancia por el “Plan general de uso de espectro Radioeléctrico” ¹, en la Fig. 1.25² se puede apreciar un resumen de las características de la banda de 915[MHz] definidas por este documento.

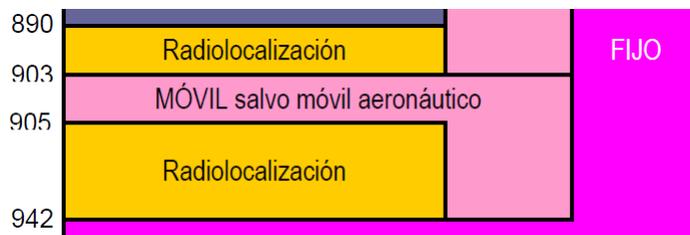


Figura 1.25: Diagrama de la asignación de frecuencias en el territorio nacional, banda 915[MHz].

Además de la descripción anterior, existen tres normativas que pueden ser consideradas:

¹http://www.subtel.gob.cl/images/stories/articles/subtel/asocfile/6_espectro_vol2.pdf

²<http://loco.lab.asu.edu/loco-memos/loco040.pdf>

1. La mayoría de los dispositivos que utilizan la banda de 915[MHz] están regidos por la Resolución Exenta 755[18].
2. Las Bandas de 800, 900, 1900 y 2600 [MHz] están concesionadas con las empresas Entel, Movistar, Claro, VTR y Nextel para proveer servicios móviles 2G, 3G y 4G.
3. Se espera que los dispositivos implementen alguna técnica de CDMA o que permita múltiples transmisores, como Saltos de Frecuencia (*Frequency Hopping*), DSSS o CSMA. Esto es necesario para que puedan gestionar las interferencias con otros dispositivos en bandas compartidas.

La Res. 755 lleva por nombre “Norma Técnica de Equipos de alcance reducido”, fue elaborada por la Subsecretaria de Telecomunicaciones el año 2005 y regula la mayoría de equipos en las bandas VHF y UHF (definidas en 1.1). Esta normativa está actualizada a la fecha año 2017 y es aplicable a transceptores portátiles, controles remotos, micrófonos inalámbricos, tecnologías RFID, radares de vehículos, entre otros dispositivos donde se requiere regular la intensidad de campo ($[mV/m^2]$) o la potencia máxima ($[mW]$) irradiada.

Sin embargo, esta regulación no responde muy bien a las nuevas tecnologías para LP-WAN, se debe recordar que está concebida para “... Equipos de alcance reducido”. De hecho, en el inciso J.2 del artículo 1, se define que los equipos generales que operen en la banda de 902[MHz] a 928[MHz] deben tener una potencia de salida máxima de 7[mW] lo que es equivalente a 8.45[dBm]. Sin embargo, en este trabajo se harán pruebas de hasta 17[dBm] (50[mW]) ateniéndose a la normativa de las bandas ISM para investigación y debido a la falta de una actualización de las normativas nacionales para equipos de largo alcance y bajo ancho de banda.

Cabe destacar que lo que no queda regulado por las normativas chilenas, está regulado por las normativas UIT, principal acuerdo internacional suscrito por Chile en materia de radiocomunicaciones.

1.5. Otros estudios de comunicación y cobertura de LoRa

En este punto se revisaran otros datos obtenidos experimentalmente, en diversas partes del mundo.

1.5.1. Cisco PIRL: Laboratorio de Innovación e Investigación de Cisco

En París, Francia, el año 2016 el equipo del Cisco PIRL elaboró el documento [14], en éste plasmaron los resultados obtenidos en pruebas de sensibilidad y cobertura. El objetivo era verificar si los datos especificados por LoRa en su *datasheet* eran obtenidos en la práctica. Los equipos/servicios utilizados fueron:

1. *End-device*: La placa de desarrollo Freescale KRDM-KL25Z con el chip SemtechSX1276

2. *Gateway*: El router Industrial Cisco910.
3. *Network Server*: Proveedor Thingpark³ a través de Ethernet

Los paquetes recibidos eran monitoreados en el lado del servidor, se enviaron alrededor de 10.000 paquetes y se guardó el valor del *Received Signal Strength Indicator (RSSI)* para cada paquete recibido mientras el *end-device* se movía. La prueba consideraba el *gateway* puesto en *indoor* y el *end-device outdoor* en ambiente urbano (no se dan más características sobre el entorno). Los parámetros del radioenlace se definen en la Tbl. 1.3, donde se puede ver que configuraron el *end-device* para la menor potencia posible, con lo que limitaron el rango de alcance hasta los 100[m]. No se define la frecuencia de transmisión, pero, se sabe que es en la banda de 915[MHz].

BW	125[kHz]	code rate	4/5
Tx Power	2[dBm]	Antenna end-device	3[dBi]

Tabla 1.3: Parámetros Prueba Cisco PIRL [14].

Los resultados obtenidos referentes al mínimo RSSI observado, tal que se recibían paquetes (entorno a los 100[m]), se pueden observar en la Fig. 1.26. En este gráfico se aprecia que el RSSI medido en el límite de la cobertura (naranja), es mayor que el mínimo especificado por la Tbl.1.2 (azul). Esto quiere decir que no se logra la Sensibilidad esperada en el receptor, de hecho no se logró ir más allá de los -125[dBm], aunque en las especificaciones se daba un rango aproximado de -123[dBm] a -136[dBm].

Al aumentar el Spread Factor se obtuvo un ligero aumento en la sensibilidad del receptor, con respecto a lo esperado. A pesar de todo, los paquetes recibidos con menor RSSI también eran recibidos con un alto SNR, cercano a 20[dBm]. Dado que se tiene un buen nivel de señal sobre ruido, es posible que el *gap* en la sensibilidad esperada con respecto a la obtenida se deba al *shadowing* extra que debe atravesar la señal para salir del ambiente *indoor* en que se encuentra el *gateway*.

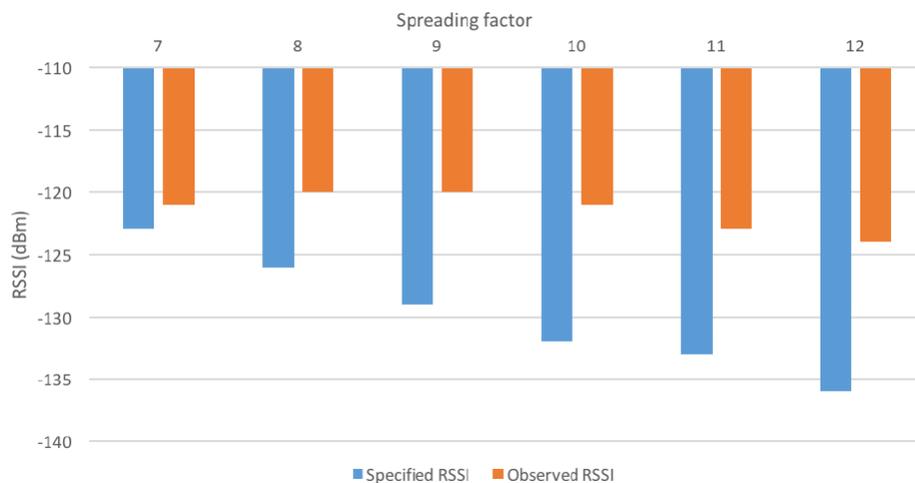


Figura 1.26: Resultados de sensibilidad: RSSI medido vs RSSI especificado para distintos SF [14].

³<https://actility.thingpark.com>

Además, se realizaron pruebas de cobertura, donde se obtuvieron los resultados observados en la Fig. 1.27. Durante la prueba la $T^{\circ}C$ era de $15^{\circ}C$ y la %HR era 55%. El *gateway* estaba colocado en el segundo piso de una casa a unos 5[m] de altura y el *end-device* estaba en un auto, la potencia de transmisión (Tx Power) del *end-device* era de 14[dBm]. Se utilizó un SF de 7, 9 y 12 cambiado de forma manual, no por ADR.

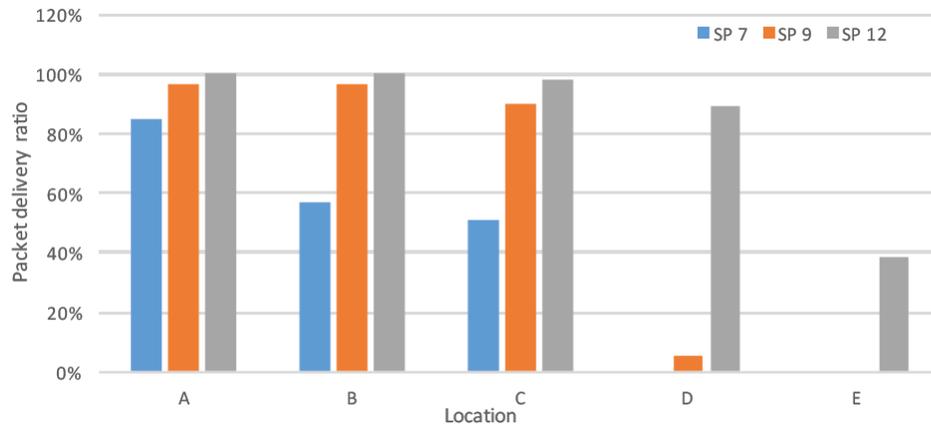


Figura 1.27: Proporción de paquetes recibidos (PDR) en pruebas de cobertura de LoRa [14].

El campo de pruebas de la cobertura de LoRa se puede apreciar en la Fig. 1.28 y los resultados aproximados del *Packet Delivery Ratio* (PDR) se resumen en la Tbl. 1.4. Se puede apreciar un fuerte decaimiento en el punto D, lo que se debe a que estaba ubicado justo detrás de un edificio de 7 pisos, y en el punto E, donde ya se daba por superado el límite de buena cobertura (70% PDR). También se puede ver que para un SF de 7 la cobertura es considerablemente menor que para 9 y 12.

SF	Distancia [m]				
	A=650	B=1400	C=2300	D=2800	E=3400
7	80 %	60 %	50 %	0 %	0 %
9	98 %	98 %	90 %	5 %	0 %
12	100 %	100 %	99 %	90 %	38 %

Tabla 1.4: Resultados de PDR en pruebas de cobertura[14].

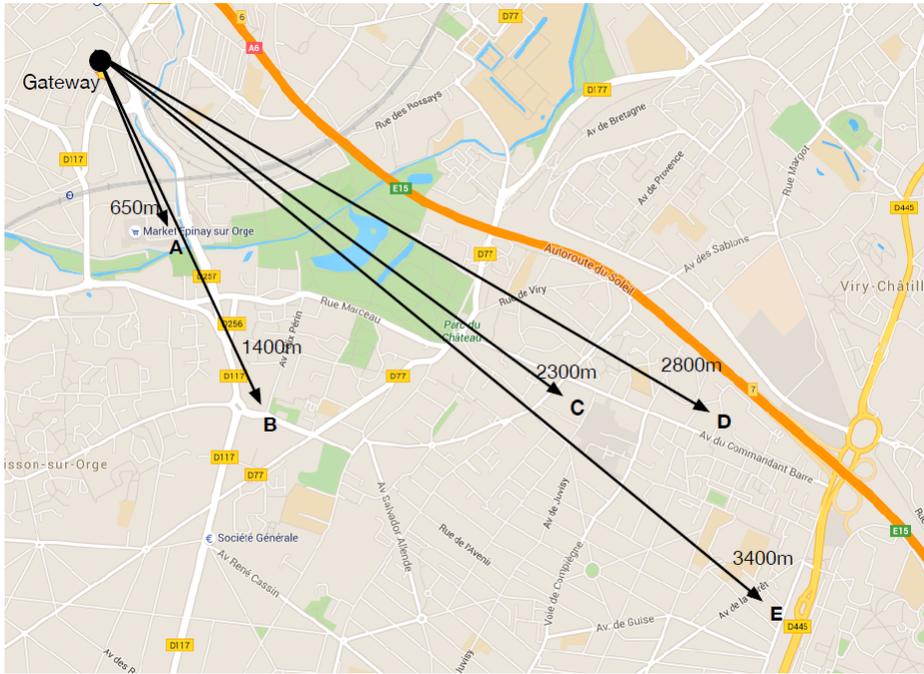


Figura 1.28: Campo de pruebas para analizar la cobertura de LoRa [14].

1.5.2. Centro de Comunicaciones Inalámbricas, Universidad de Oulu, Finlandia

Investigadores de la universidad de Oulu, Finlandia, con el apoyo de *Nokia Technologies* desarrollaron el documento [19] donde exponen los resultados obtenidos para pruebas de cobertura y plantean un modelo de atenuación de canal para LoRa. Los dispositivos utilizados para las pruebas se definen a continuación:

1. *End-device*: LoRaMote con chip Semtech SX1272 y antena PCB tipo Planar-F. Además incluye GPS.
2. *Gateway*: Estación Kerlink's LoRa IoT con antena Aerial D100-1000 biconica de 2[dBi] a 24[m] de altura.

Las pruebas fueron realizadas el 2015 en la ciudad de Oulu, Finlandia, para diferentes condiciones climáticas, durante finales de primavera e inicios del verano. Los edificios más altos en la zona llegan a los 12 pisos de altura y la ciudad está situada sobre un terreno plano, sin grandes diferencias geográficas. La ciudad se encuentra en la costa por lo que parte de las mediciones fueron tomadas sobre el agua.

El *end-device* fue posicionado sobre un auto para las mediciones en tierra y sobre un bote para las mediciones sobre el agua. La velocidad del *end-device* varió entre los 40[km/h] y los 100[km/h] y se seleccionó un SF de 12 con el objetivo de encontrar el máximo rango de cobertura para un BW de 125[kHz]. La potencia de transmisión del *end-device* fue de 14[dBm] (25[mW]) y se utilizaron 6 canales de radio en la banda de 868[mHz].

Los resultados obtenidos demostraron que los rangos de cobertura aceptables, es decir con aproximadamente 30 % de *Packet loss ratio* (*PLR*), son de 10[km] en tierra y 15[km] sobre el agua. Además, se elaboró un mapa de calor sobre la zona de Oulu, utilizando el RSSI recibido en el *gateway* para las distintas posiciones del *end-device*, en la Fig. 1.29 se puede apreciar el resultado.

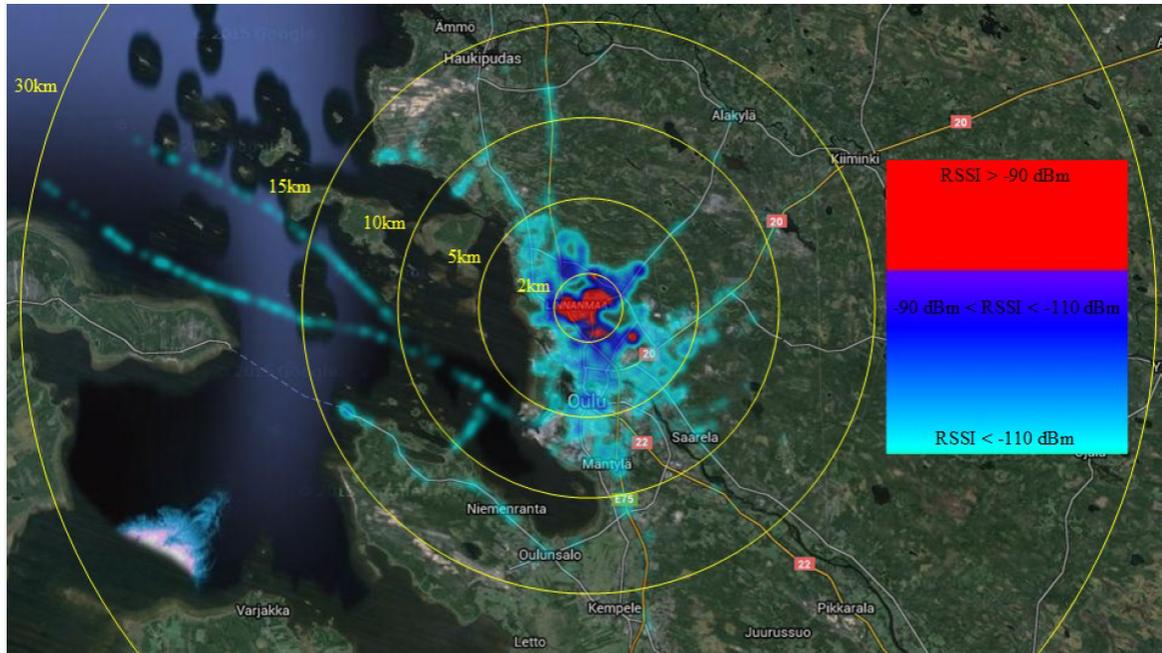


Figura 1.29: Mapa de calor de la zona de Oulu para la cobertura de LoRa[19].

Capítulo 2

Dispositivos y Herramientas

En este capítulo se dan a conocer los equipos y herramientas que se utilizaron en la creación del sistema de comunicación inalámbrica para el estudio de cobertura de LoRa. A continuación, se explicarán el *Hardware*, la conexión, el entorno de desarrollo y el *Firmware* utilizados para elaborar el nodo transmisor. Todas las instrucciones necesarias para ejecutar el proyecto pueden encontrarse en las referencias correspondientes.

2.1. Hardware

El Nodo Transmisor se diseñó pensando en que debía presentar las siguientes funcionalidades:

1. **Módulo de Radio LoRa:** Para obtener los datos del radioenlace LoRa.
2. **GPS:** Para obtener los datos de posición en cada punto de medición
3. **Datalogger:** Para almacenar correctamente los datos de GPS y LoRa.
4. **Procesamiento Central:** Para gestionar la comunicación entre los distintos dispositivos.
5. **Fuente de Alimentación:** Para alimentar de energía el sistema de forma remota mientras se mueve.

Un diagrama de bloques del sistema diseñado se puede apreciar en la Fig. 2.1. Este Nodo es el encargado de obtener y guardar tanto la posición como otros parámetros del radio enlace formado entre el nodo transmisor y el nodo receptor.

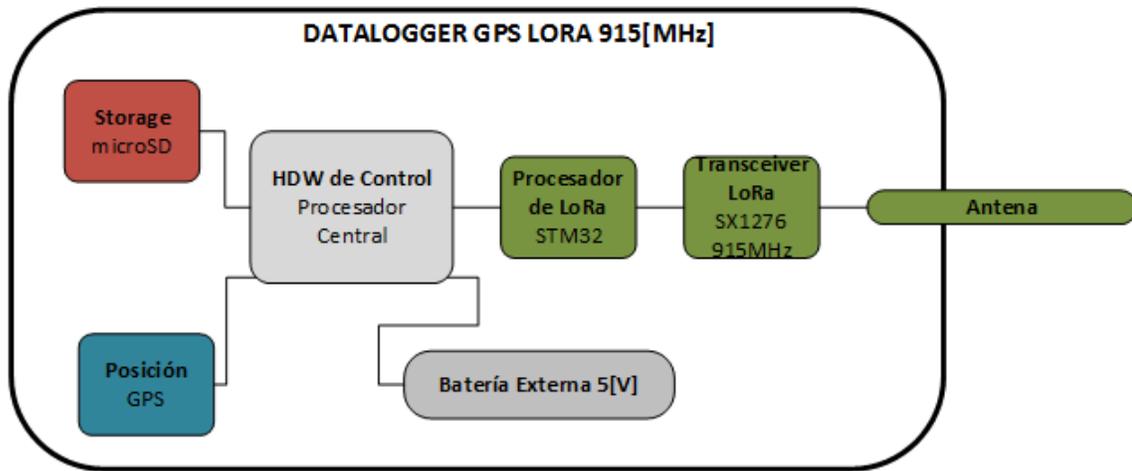


Figura 2.1: Diagrama de bloques del Nodo Transmisor.

2.1.1. Módulos de Radio LoRa: DevKit

Los módulos de desarrollo DevKit (*Development Kit*) fueron proporcionados por Telefónica Investigación y Desarrollo, como se puede ver en la Fig. 2.2 ambos nodos, fabricados por Modtronix Engineering[20], conforman un kit de desarrollo para realizar pruebas con el chip SX1276 (*transceiver* LoRa) de Semtech [21].



Figura 2.2: Módulos de desarrollo DevKit.

Cada Módulo, o Nodo, cuenta con la siguiente lista de componentes:

1. **Placa NZ32-SC151 [22]:** con el microcontrolador STM32L151RC de STMicroelectronics [23].
2. **Placa InAir9b:** con *Transceiver* SX1276 y antena omnidireccional de 2[dbi]
3. **Batería:** Batería de litio de 3.7[V] recargable de 1000[mAh]
4. **Case Metálico:** con PCB *breakout* para la placa NZ32-SC151

5. **Interfaz Usuario:** con Botones, Leds, Buzzer y una pantalla OLED de 128x64[px]

Cada módulo posee tres estados (“Mode”) posibles “stopped”, “master” y “slave”. Para realizar las pruebas se debe configurar uno de los nodos como “master” y el otro como “slave”, esto causará que el nodo “master” comience a enviar mensajes “Ping” al nodo “slave” y que este último responda con un “Pong” más otros parámetros, entablando un radio enlace. En la Fig. 2.3 se puede apreciar la pantalla principal del nodo configurado “master” el cual está comunicándose con el nodo “slave 1”, además, se muestran una serie de parámetros del radio enlace que se explican más en detalle en la Tbl. 2.1.



Figura 2.3: Ejemplo de pantalla principal del nodo en estado “master”.

Otra característica importante de los nodos, es que mediante su interfaz de botones se pueden configurar algunos parámetros del radio enlace, como se ve en la Fig. 2.4 se puede cambiar:

1. **"Board"**: que hace referencia al módulo de *radiowireless* que se está usando, en el caso de la imagen el InAir9.
2. **"Freq"**: que es la frecuencia de la portadora pudiendo ser seleccionada en la banda de frecuencia de 915[MHz].
3. **"BW"** : *Bandwidth* BW de la modulación LoRa.
4. **"SF"**: *Spread Factor* SF de la modulación LoRa.

L-016	Local RSSI	89 %	Estado de carga de la batería
R-004	Remote RSSI	F=918.8	Frecuencia de portadora en [MHz]
SF=12	Spread Factor	Err=0000	Mensajes Errados
RXed OK=0011	Mensajes Recibidos OK	–	–

Tabla 2.1: Parámetros mostrados en la pantalla principal del nodo master.

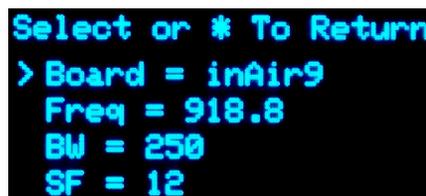


Figura 2.4: Parámetros configurables de los nodos devkit.

Placa de desarrollo NZ32-SC151

Esta placa está construida para soportar el microcontrolador STM32LC151RC (STM32 de ahora en adelante) junto a los componentes pasivos necesarios para su funcionamiento como resistencias, capacitores, botones, leds, cristales y conectores, además, cuenta con una entrada para batería de litio de 3.7[V], un cargador para baterías de ion litio o LiPo y un regulador de 3.3 [V] para alimentar el sistema completo. En la imagen 2.5 se pueden apreciar en detalle la mayoría de los componentes antes descritos.

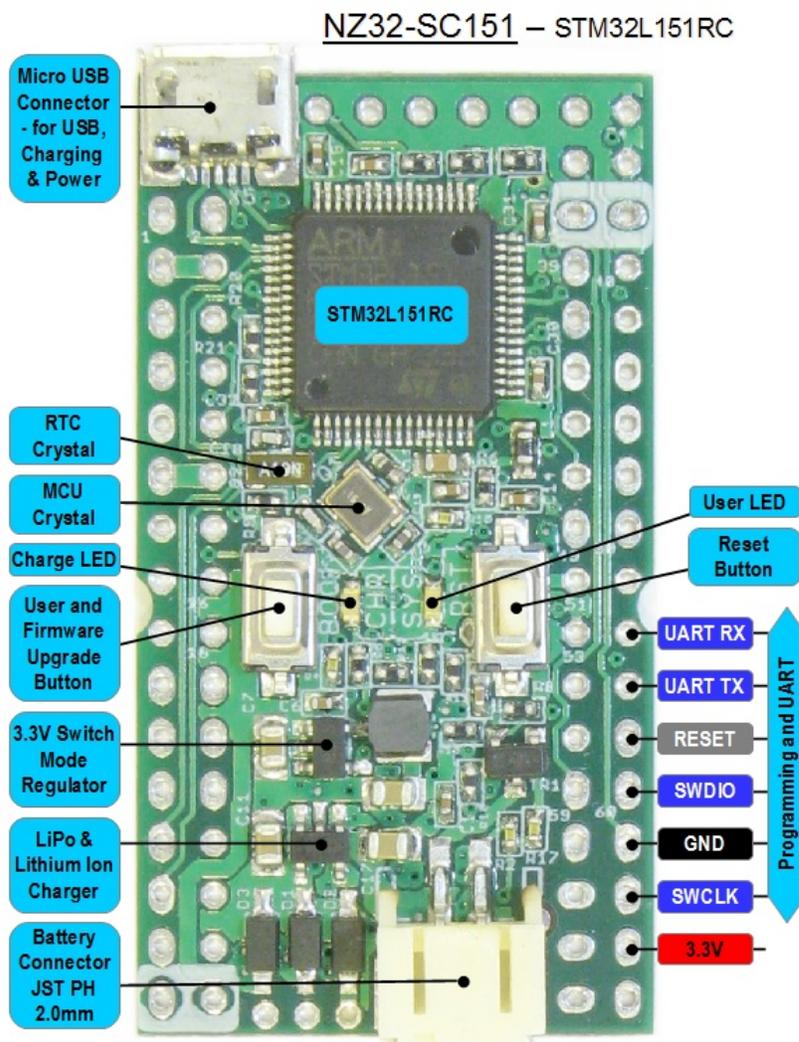


Figura 2.5: Componentes y puertos de la Placa NZ32-SC151 [22].

Una característica interesante de la familia de microcontroladores STM32LC es que pueden ser programados directamente desde el puerto USB sin la necesidad de pasos intermedios, ni de un conector JTAG, esto gracias a que trae el *bootloader* incorporado de fábrica. En particular el STM32 es una CPU de ultra bajo consumo, menor a $10[\mu\text{A}]$ en su estado de funcionamiento normal, esto lo hace un chip ideal para dispositivos LPWAN en comparación con el típico ATMEGA 328p del Arduino cuyo consumo mínimo ronda los $100[\mu\text{A}]$ [24]. En

la Tbl. 2.2 se resumen más características importantes [25] de esta familia de microcontroladores.

CORE	ARM Cortex M3 32[bit]	Consumo	<10 [μ A]
memoria Flash	256[kB]	memoria RAM	32[kB]
memoria EEPROM	8[kB]	velocidad máxima	32[MHz]
Periféricos	1xUSB + 3xUSART	Periféricos	3xSPI + 2xI2C

Tabla 2.2: Características del microcontrolador STM32LC151RC.

Para comunicar los datos del enlace LoRa se utilizó el puerto de comunicación UART que se ve a la derecha en la Fig. 2.5. Se puede apreciar en azul el pin UART TX (o pin56 en Fig. 2.6), este será conectado a un puerto UART RX del procesador central para recibir los datos.

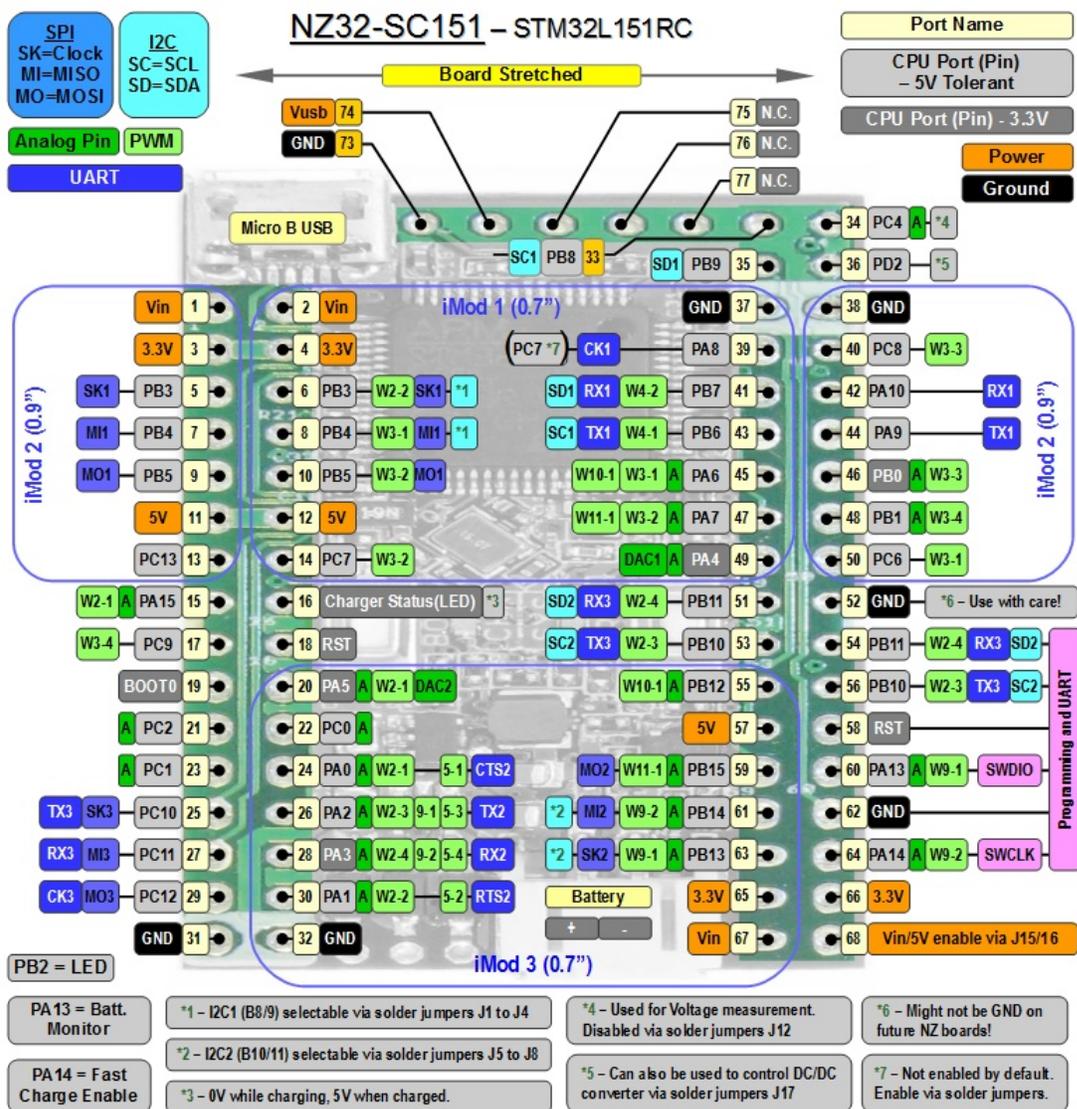


Figura 2.6: Pinout Extendido de la placa NZ32 [22].

En la Fig. 2.6 se puede ver el *pinout* detallado de la placa NZ32-SC con sus conexiones SPI, I2C, UART, etc. Para la conexión del Inair9b se utiliza el bloque de pines iMod2 (viene montado de fábrica) que se puede ver entre los pines 1 a 13 y 38 a 50. El esquemático se puede ver en el Anexo C.

InAir9b con *Transceiver* SX1276

La placa Inair9b [26], diseñada y vendida por Modtronix, es un módulo de comunicación inalámbrica que utiliza el chip SX1276 de Semtech y funciona para las bandas ISM de 863[MHz] a 870[MHz] y de 902[MHz] a 928[MHz]. En la Fig. 2.7 se puede apreciar la placa con y sin *shield* RF, al ver el módulo sin su *shield* se puede apreciar el Chip SX1276.



Figura 2.7: Imagen del módulo InAir9 con *shield* RF (izq) y sin *shield* RF (der) [26].

La *Printed Circuit Board (PCB)* del módulo InAir9b es una placa de alta calidad con 4 *layers*, optimizada para RF y basada en las referencias de diseño que se pueden encontrar en los documentos SX1276RF1JAS [27] y SX1276MB1LAS [28] de Semtech. En la Tbl. 2.3 se resumen las principales características técnicas de este módulo basado en LoRa™ y el esquemático se encuentra en el anexo A al final de este documento.

Voltaje operación	2.4[V] a 3.7[V]	Potencia de Salida max	+20 [dBm]
FSK max data rate	300 [kbps]	Sensibilidad max	-148 [dBm]
LoRa max data rate	37.5 [kbps]	Modulaciones	FSK, GFSK, LoRa™

Tabla 2.3: Características del módulo InAir9b [26].

El chip SX1276 [16] es un *transceiver* que cuenta con el módulo de largo alcance LoRa™, y provee de comunicación de espectro ensanchado (SS) de ultra largo alcance, alta inmunidad a interferencia y mínimo consumo de corriente. En la Fig. 2.8 se puede apreciar la estructura de bloques funcionales del chip, en particular al lado derecho del diagrama se ve como la interfaz de comunicación SPI se conecta a la pila *First Input First Output (FIFO)* LoRa/FSK para recibir o enviar los paquetes entre el microcontrolador STM32 y la antena.

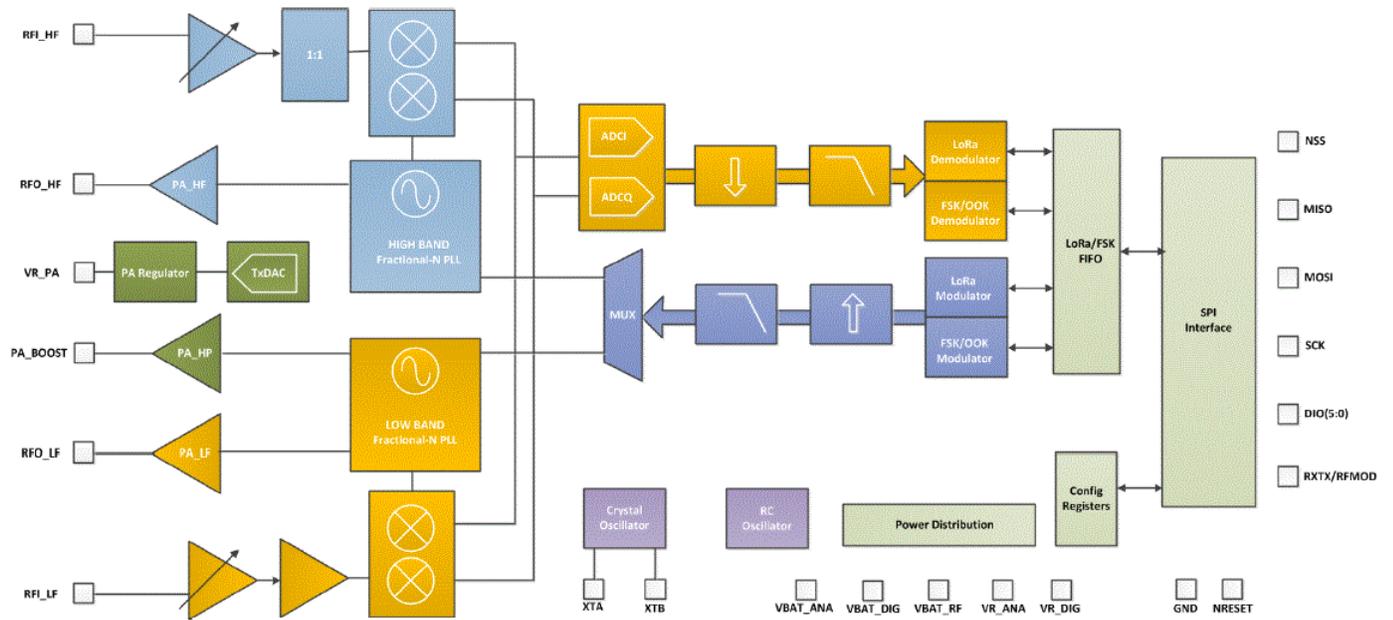


Figura 2.8: Diagrama de Bloques del chip SX1276 [16].

2.1.2. GPS

Para la obtención de la posición se utilizará un módulo GPS L80 [29] fabricado por Quectel y montado en una placa *breakout* fabricada por Olimex, este *hardware* es facilitado por Telefónica I+D. En la Fig. 2.9 se puede observar el módulo y se aprecia la antena interna incorporada (mercada en un círculo negro) para la recepción de los datos satelitales, se requiere captar al menos 3 satélites para calcular la posición del nodo transmisor.

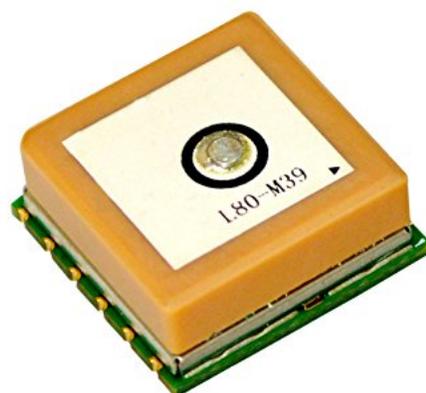


Figura 2.9: Módulo GPS L80 de Quectel [29].

En la Fig. 2.10 se pueden ver en detalle los pines de conexión hacia el módulo L80, esto es gracias a la placa *breakout* que permite acceder a ellos de manera fácil. El módulo de *Global Positioning System (GPS)* será conectado al procesador central mediante un puerto

UART para poder obtener los datos de geolocalización, los cuales, llegan al microcontrolador a través de la salida Tx del GPS. En ésta comunicación entre microcontrolador y el GPS, que se ilustra en la Fig. 2.11, la capa física está parametrizada por el protocolo de comunicación TTL a 3.3[V].

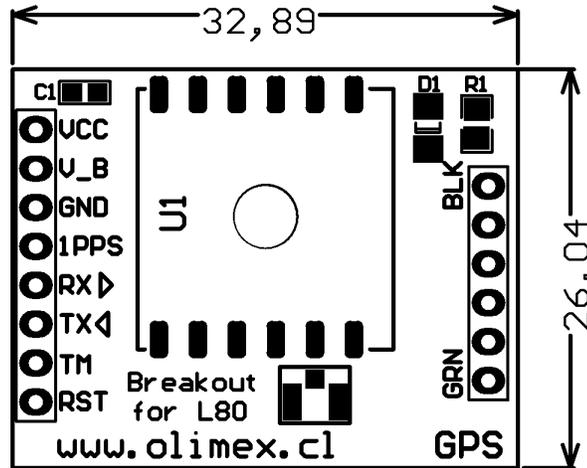


Figura 2.10: Pinout de placa breakout para el GPS L80 y medidas en [mm].

El rango del voltaje de alimentación del GPS es de 3[V] hasta 4,3[V], y será alimentado con 3,3[V]. Además se utilizará una batería tipo moneda de 12mm de 3[V] para mantener el GPS con *backup* y así poder guardar los datos de los satélites, para que al reiniciarlo no parta en “frío”, esto debido a que toma unos 8 minutos ajustar la posición inicial si no se guardan los datos anteriores.

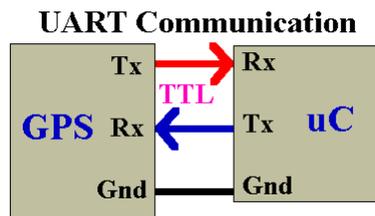


Figura 2.11: Diagrama de comunicación TTL entre dos puertos UART [30].

Los datos de la geolocalización están en el *standard National Marine Electronics Association (NMEA)* y son desplegados en el *bus serial* en varios formatos, no obstante, se utilizara el formato GPRMC. Como se ve en la Fig. 2.12, luego del *string* “\$GPRMC,” se puede ver la hora UTC, indicador de datos habilitados(A/V), Latitud, indicador Norte/Sur, Longitud, indicador Este/Oeste y velocidad entre otros. Con esta información será posible emparejar la posición con la intensidad de señal obtenida con el módulo InAir9b.

Example:

```
$GPRMC,013732.000,A,3150.7238,N,11711.7278,E,0.00,0.00,220413,,,A*68<CR><LF>
```

Figura 2.12: Ejemplo de *string* de datos en formato GPRMC.

2.1.3. Datalogger: MicroSD

Para almacenar los datos obtenidos del radioenlace y la posición, se utilizará como *storage* una microSD, para esto se requiere un *socket* para microSD montado en una placa *breakout* que permita conectarlo fácilmente. Se utilizó la placa microSD *breakout* de Sparkfun [31], que se puede ver en la Fig. 2.13, la cual se alimenta con 3.3[V].

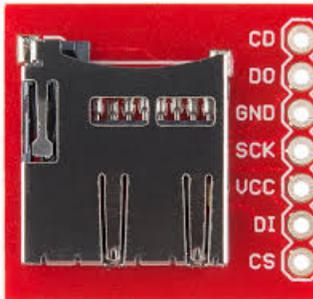


Figura 2.13: Placa microSD *breakout* de SparkFun [31].

La microSD se conecta al procesador central a través de la placa *breakout* según se muestra en la Tbl. 2.4, este protocolo de comunicación de capas inferiores es llamado *Serial Peripheral Interface (SPI)*.

microSD	procesador central
D0	MISO=pin12
GND	GND
SCK	SCK=pin13
VCC	3.3V
DI	MOSI=pin11
CS	CS/S=pin4

Tabla 2.4: Tabla de conexión micorSD a microcontrolador.

2.1.4. Procesador central: Freaduino

El procesamiento central se realiza en una placa de desarrollo Freaduino UNO rev1.8, donde se recibe la información de los buses UART del GPS y de la placa NZ-SC151. La placa

Freaduino [32] tiene un procesador ATMEGA 328, y un *pinout* equivalente al Arduino UNO rev3. Además, es compatible con el IDE de Arduino y tiene un *switch* selector de voltaje para (independiente de la entrada) trabajar a 5[V] o 3,3[V]. Dado que la microSD, el GPS y los buses de comunicación de la NZ-SC151 funcionan a 3,3[V] el *switch* es seleccionado a 3,3[V].

En la imagen 2.14 se pueden ver el *switch* de voltaje, el procesador Mega328, la misma configuración del Arduino UNO y además, la D0-D13 *pinrow* que da una conexión extra a cada pin digital más sus respectivos pines de voltaje 3.3 y GND.

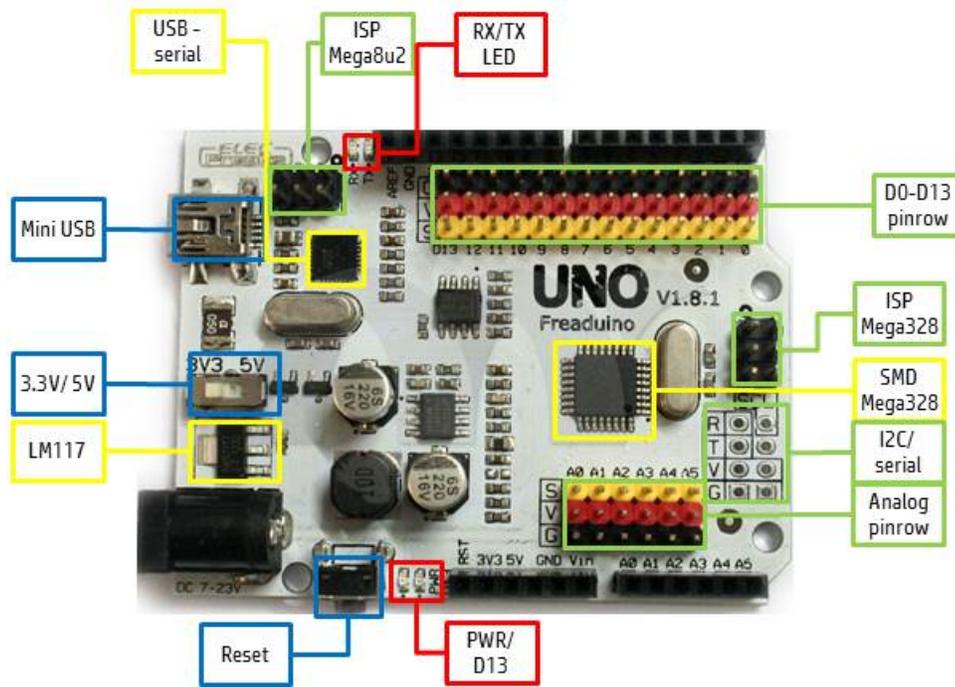


Figura 2.14: Descripción Freaduino [32].

A pesar de que existía la opción de integrar GPS y microSD en el procesador STM32, al hacer pruebas de integración sobre éste existieron problemas con la coordinación de los buses de comunicación UART. Debido a esta descordinación, el Nodo Transmisor funcionaba a una velocidad distinta de su par, el Nodo Receptor, dado que este aspecto es fundamental para la correcta coordinación del enlace, se decidió integrar el Datalogger microSD, el GPS y la salida de comunicación del NZ32-151 en la placa Freaduino. Finalmente, el Freaduino recibe la información de posición y radio enlace LoRa para guardarlos en la microSD.

2.2. Conexión

El Nodo Transmisor, detallado en la Fig. 2.15, es el sistema completo formado por los dispositivos descritos en la sección 2.1, gracias a la integración de estos *hardwares* el nodo es capaz de:

1. Crear un radioenlace de comunicación con la tecnología de modulación LoRaTM dentro de la banda 902[MHz] a 928[MHz].
2. Obtener el RSSI en ambos nodos, el SNR, el número de paquetes errados y el número de paquetes recibidos del radioenlace.
3. Conectarse al sistema GPS y obtener la posición georreferenciada en formato NMEA.
4. Recopilar, todos los datos antes descrito, en un archivo .txt y almacenarlo en una microSD.

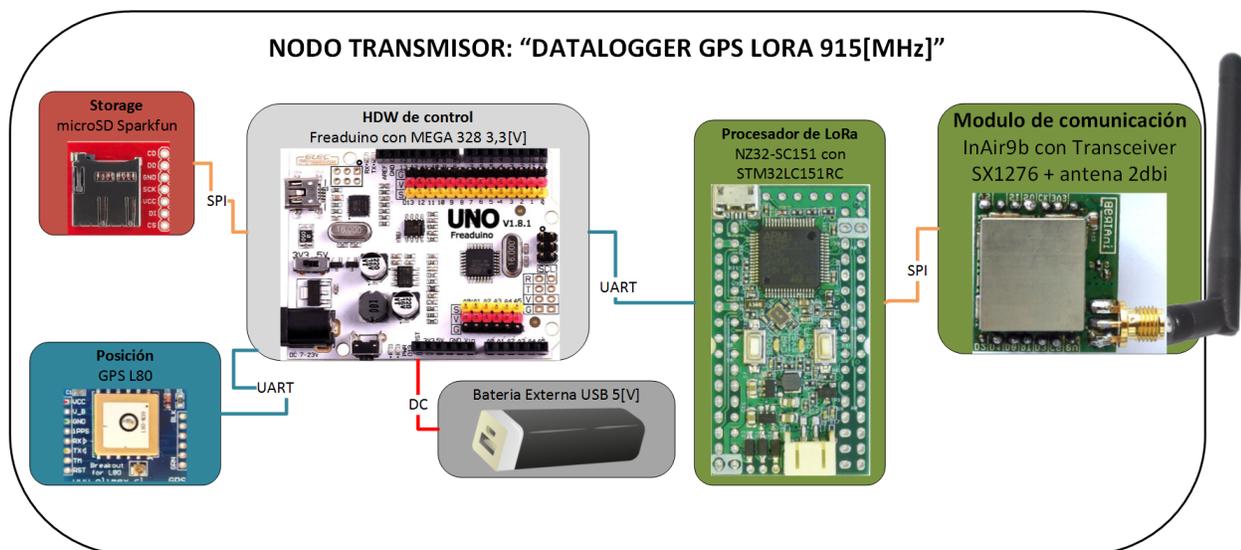


Figura 2.15: Diagrama de bloques detallado del nodo transmisor (*end-device*).

El objetivo de esta sección, es detallar la conexión de todos los dispositivos al procesador central, como se ilustra en la Tbl. 2.5. Se puede ver que el GPS y la microSD se alimentan desde el Freaduino, mientras que la placa NZ32SC tiene alimentación propia gracias al Devkit, además todo los GND están conectados para dar el mismo piso de referencia en las comunicaciones. La batería externa se conecta directamente a la placa Freaduino a través de un cable USB, luego esta placa se encarga de regular el voltaje a 3,3[V] para alimentar el resto de los sistemas.

FREADUINO	NZ32-SC	GPS	microSD
pinD11	–	–	DI (mosi)
pinD12	–	–	D0 (miso)
pinD13	–	–	SCK (sck)
pinD4	–	–	CS
3.3V	–	VCC	VCC
GND	GND	GND	GND
pinD2	–	TX	–
pinD5	TX3(pin56)	–	–

Tabla 2.5: Tabla de conexión sistema completo.

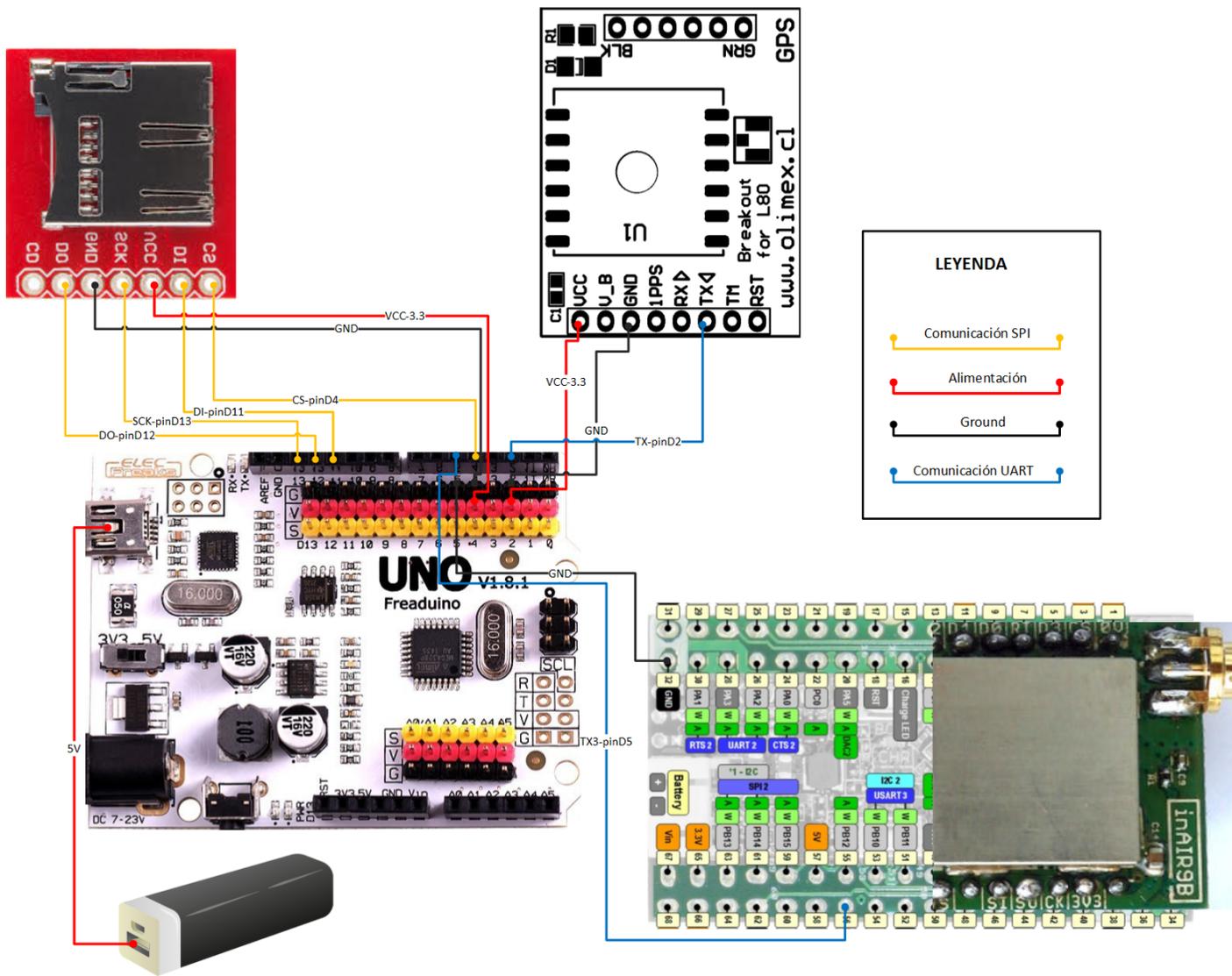


Figura 2.16: Diagrama de conexión entre Freaduino, GPS, microSD, NZ32-SC y Batería Externa.

2.2.1. Conexión del Sistema Completo

El diagrama de conexión de todos los dispositivos se observa en la Fig. 2.16, aquí se ve como se conecta pin a pin el sistema completo del Nodo Transmisor. Se debe recordar que el sistema de alimentación del Freaduino, GPS y microSD es de 3,3[V] y es independiente del sistema de alimentación de la NZ32-SC, el cual también es de 3,3[V]. Debido a esta conexión múltiple, se conectan todos los GND para ofrecer el mismo piso de referencia a ambos sistemas, aspecto clave para la correcta comunicación de los buses.

2.3. Entorno de desarrollo

En este punto se muestran los códigos base y programas de computador utilizados para desarrollar *firmware* del Nodo Transmisor.

2.3.1. Librerías y código: GPS, microSD y Radio LoRa

Se utilizaron las librerías y códigos descritos a continuación para procesar los datos recibidos desde el GPS y el STM32, y los datos enviados a *Storage* en la microSD. En la Fig. 2.17 se ve la inclusión de estas en el *firmware* del Nodo Transmisor. Además, se presenta el código base con el que funcionan los DevKit el cual es de código abierto.

Librerías

- **GPS:** Se utilizó la librería “TinyGPS++”[33] de Mikal Hart con fecha 2010-2013.
- **microSD:** Se utilizaron las librerías “SD.h” y “SPI.h” incluidas por *default* en el IDE de Arduino.
- **Comunicación UART con STM32:** Se utilizó la librería “SoftwareSerial.h” incluida por *default* en el IDE de Arduino.

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <SPI.h>
#include <SD.h>
```

Figura 2.17: Librerías Anexadas.

Firmware Dev Kit: STM32

El *firmware* puesto en el STM32 es utilizado para coordinar el enlace de radio LoRa del nodo transmisor. Para este se utilizó el *firmware* por defecto que Modtronix ofrece y viene

instalado en ambos módulos. Este código se haya albergado en GitHub [34], como se puede ver en la Fig. 2.18 fue creado por Ingenieros de Modtronix. Este es el código que será alterado en ciertas líneas para obtener los valores de LoRa por la salida TX3 de la placa NZ32-SC.

modtronix-com Added YAT Serial Terminal files		Latest commit bc54a76 on 22 Sep 2016
SW4STM32	Changes for release candidate	a year ago
Src	Changes for release candidate	a year ago
bin	Changes for release candidate	a year ago
mbed_nz32sc151	Initial version	a year ago
modtronix_NZ32S	Changes for release candidate	a year ago
modtronix_im4OLED	Changes for release candidate	a year ago
modtronix_inAir	Changes for release candidate	a year ago
tools/yat	Added YAT Serial Terminal files	a year ago
usbcdc-cube	Initial version	a year ago
.gitattributes	Initial version	a year ago
.gitignore	Added YAT Serial Terminal files	a year ago
DFU_Readme.txt	Initial version	a year ago
README.md	Initial version	a year ago
Readme.txt	Initial version	a year ago
lib_checkout.bat	Initial version	a year ago
lib_checkout_LATEST.bat	Initial version	a year ago
lib_versions.txt	Updated library versions	a year ago

Figura 2.18: Repositorio de GitHub[34].

2.3.2. Softwares

Los *softwares* utilizados se pueden ver en la imagen 2.19, de izquierda a derecha y son: Arduino IDE, CoolTerm, DfuSe Demo, SWSTM32 y Dfu File Manager. Estos fueron los programas utilizados para desarrollar el *firmware* del nodo transmisor, correspondiente al código central y al *hack* en el código del Dev-Kit.



Figura 2.19: Softwares para el desarrollo.

SWSTM32

Para poder alterar el *firmware* del Dev-Kit se utilizó el *software* "System Workbench for STM32"[35] que es un *Integrated Development Environment (IDE)* del programa Eclipse™. Este compilador permite manejar el código de "C++" proporcionado por Modtronix para operar el STM32 y las librerías correspondientes a la pantalla LCD y el módulo de radio inAir9b. Al compilar el código en el SWSTM32 es generado el archivo .hex que luego se modificó y cargo en el procesador STM32.

Dfu File Manager

Este programa es capaz de cambiar el formato de un archivo .hex a uno binario .bin y agregar un *header* con el ID del procesador. En la Fig. 2.20 se puede ver en rojo los campos que deben ser rellenados con el ID del procesador (DF11 y 2200) y en verde el botón para seleccionar el archivo .hex que se desea cambiar. Finalmente, al presionar "Generate" se crea el archivo .bin para cargar en el procesador.

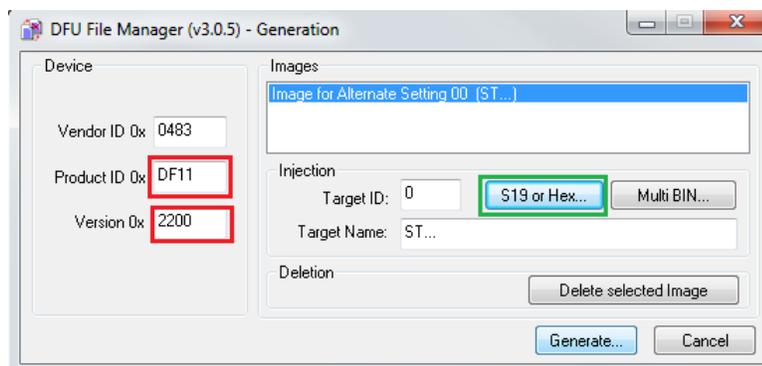


Figura 2.20: Dfu File Manager.

DfuSe Demo

Este programa permite cargar el *firmware* en el microcontrolador. Para ellos se selecciona el Dev-Kit en modalidad *bootloader*, presionando el botón "Upgrade Button" de manera constante y paralelamente haciendo un click en el "Reset Button", una vez que el Dfu Demo lee la información del procesador y se empareja, se debe seleccionar el archivo .bin para cargar. En la Fig. 2.21 se aprecia en verde el botón "Choose" para seleccionar el archivo y en azul el botón "Upgrade" para cargarlo en el STM32 del Dev-Kit.

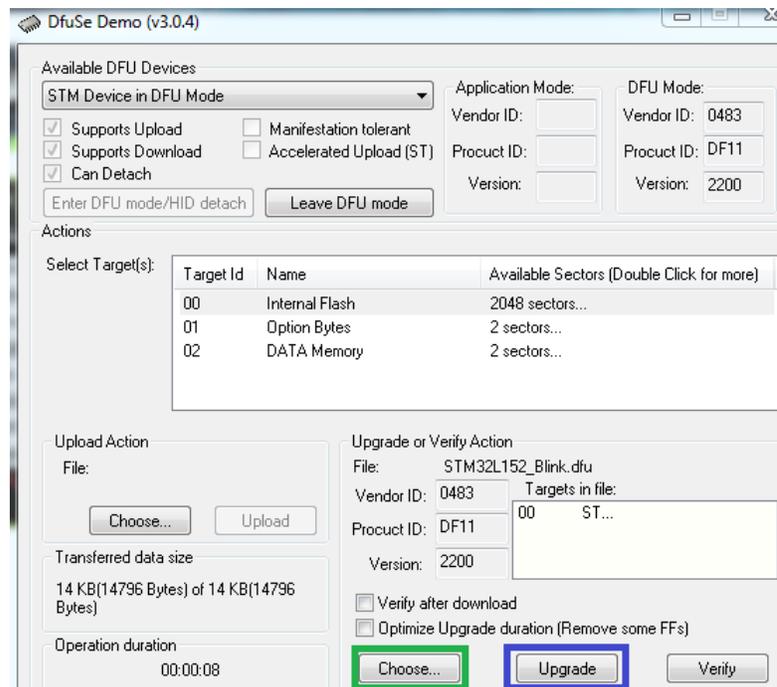


Figura 2.21: DfuSe Demo.

IDE Arduino

Se utilizó el IDE Arduino ver1.6.9 que se puede descargar directamente de la página de Arduino (www.arduino.cc). En este programa se creó el código de procesamiento central en formato .ino, este código será cargado y ejecutado en el Freduino.

CoolTerm

Finalmente, para poder observar el resultado de los cambios implementados en el código mientras se hacían las pruebas se utilizó un terminal USB del computador y el programa CoolTerm para mostrar en pantalla lo que entregaban los puertos UART TX. En la Fig. 2.22 se puede ver los datos entregados por el GPS y en rojo un *string* GPRMC.

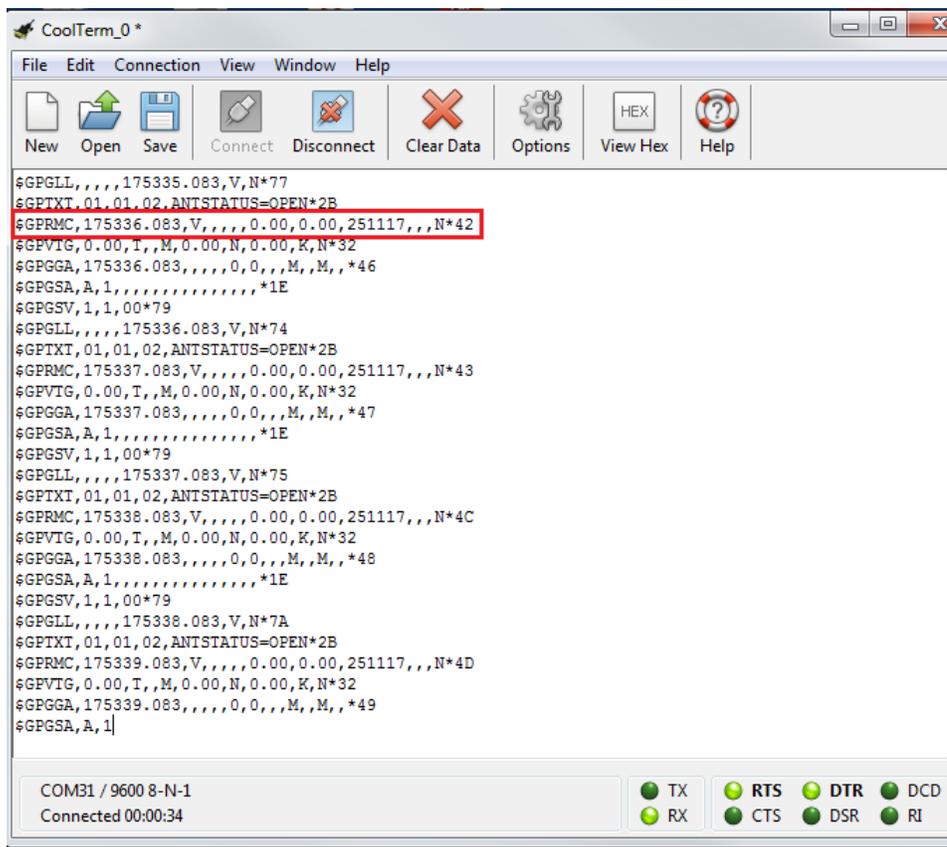


Figura 2.22: CoolTerm leyendo COM31 a 9600[baud].

2.3.3. Firmware Procesamiento Central

En el Frearduino se realiza el procesamiento central correspondiente a la coordinación de los canales de comunicación, la recepción de los datos GPS y Lora, y el almacenamiento de la información emparejada en la microSD.

Canales de comunicación

En la Fig. 2.23 se puede ver la asignación de pines de los canales de comunicación: SerialGPS, SerialLoRa y chipSelect. Se debe recordar que la librería SD.h trae los pines D11, D12 y D13 asignados por *default*. Los canales de comunicación utilizados presentan tres formatos:

1. **SerialGPS**: UART por puerto Serial TTL a 9600[baud] (pinD2)
2. **SerialLora**: UART por canal SoftwareSerial a 19200[baud] (pinD5)
3. **microSD**: SPI por pines digitales 13, 12 y 11, y el CS “chipSelect” en el pinD4.

```

SoftwareSerial SerialGPS(2, 3);
SoftwareSerial SerialLora(5, 6);
TinyGPSPlus gps;
unsigned long time_base;
unsigned long time_now;
char buffer[50];
char sd_msg[100];
char lat[15];
char lng[15];
const int chipSelect = 4;

bool isTimeout(unsigned long t_start, unsigned long t_current, unsigned long timeout, bool print_time)

void setup()
{
  Serial.begin(9600);
  if (!SD.begin(chipSelect)) { //Por default son los pines 13,12,11 mas el ChipSelect
    return;
  }
}

```

Figura 2.23: Fragmento código principal, definición de pines.

Código Principal

Dentro del *Loop* principal se ejecuta, en primera instancia, la lectura del canal SerialGPS y se almacena la longitud y latitud en la memoria del MEGA328. Luego se ejecuta la lectura del SerialLora y se almacena en la memoria del MEGA328 toda la información contenida entre los caracteres “(” y “)” donde viene la información de LoRa. Finalmente se concatena la latitud, la longitud y la información LoRa, y se guarda en la microSD como un archivo .txt en forma de columnas. El código completo está incluido al final de este documento en el anexo B.

Hack Devkit

Para obtener los datos LoRa desde el procesador STM32, encargado del procesamiento del módulo de radio, se agregó una sección de código en el archivo “main.cpp” del *firmware* descargado de Github. Como se puede ver en la Fig. 2.24 se agregó en la línea 88 la variable “char lomo0” definida como un arreglo de [50] caracteres, en la línea 149 se definió la salida del bus de comunicación “streamDebug”, referente al puerto Tx3, como 19200[baud] y entre las líneas 447 y 460 la concatenación de las variables “SnrValue”, “RssiValue”, “RssiValueSlave”, “rxCountPingPong” y “rxErrCountPingPong”, en el arreglo “lomo0” que finalmente es enviado por el puerto Tx3 con el comando “streamDebug.printf(lomo0)”. Aquí, se puede apreciar que al concatenar las variables con el comando “sprintf” se agregan los identificadores “(” y “)” para reconocer el inicio y el final del paquete de información, esto da cierto margen para

incluir más parámetros de radio o procesamiento para otro tipo de pruebas.

```
87 //lomo things
88 char lomo0[50];

148 //streamDebug.baud(230400); //Set UART speed used for debugging
149 streamDebug.baud(19200); //Set UART speed used for debugging
150 //streamDebug.baud(57600); //Set UART speed used for debugging
151 //streamDebug.baud(9600);

447 sprintf(lomo0,"%d,%d,%d,%d,%d)/n",
448 pRadios[0]->settings.LoRaPacketHandler.SnrValue, // int8
449 pRadios[0]->settings.LoRaPacketHandler.RssiValue, // int16
450 radioData[0].RssiValueSlave, // int16
451 appData.rxCountPingPong, //uint16 OK
452 appData.rxErrCountPingPong //uint16 OK
453 //pRadios[0]->settings.LoRa.Bandwidth, // uint32
454 //pRadios[0]->settings.LoRa.Datarate, // uint32
455 //radioData[0].SnrValue, // int8
456 //radioData[0].rxCount, //uint16
457 //radioData[0].rxErrCnt //uint16
458 );
459 //Ejemplo de String: (39,-15,17,-4,39,25,7,0)0
460 streamDebug.printf(lomo0);
```

Figura 2.24: Código agregado a “main.cpp”.

Capítulo 3

Implementación y Metodología

En este capítulo se explica como se utilizó el sistema de comunicación descrito en el capítulo 2 y las características del terreno seleccionado para las pruebas. También se mostrará el formato de las muestras obtenidas y como se utilizaron para crear los mapas de calor.

3.1. Sistema de Comunicación LoRa

El sistema de comunicación está compuesto por el Nodo Transmisor y la Estación Base. Estos dispositivos se comunican mediante el protocolo inalámbrico LoRa en la banda de 915[MHz], mientras el Nodo Transmisor se desplaza por las calles cercanas a la Estación Base tomando datos. Cada muestra generada por el Nodo Transmisor es guardada en un archivo .txt como un *string* de datos correspondiente a la posición y sus respectivos parámetros de radio.

3.1.1. *End-device*: Nodo Transmisor

El Nodo Transmisor (*end-device* de ahora en adelante) se puede observar en la Fig. 3.1, este fue montado dentro de un case plástico para fijar los diferentes *hardwares* con pegamento. Se dejó un cable USB conectado al Freaduino para permitir conectar al PC, en caso de necesitar hacer cambios, y para conectar la batería externa fijada sobre la tapa verde del case plástico. Además, se dejó un espacio sobre la tapa para tener acceso a la interfaz del Dev-Kit y la microSD. Se debe recordar que desde la interfaz de botones se seleccionan los parámetros de radio BW, SF, Frecuencia y “Mode” master/slave.

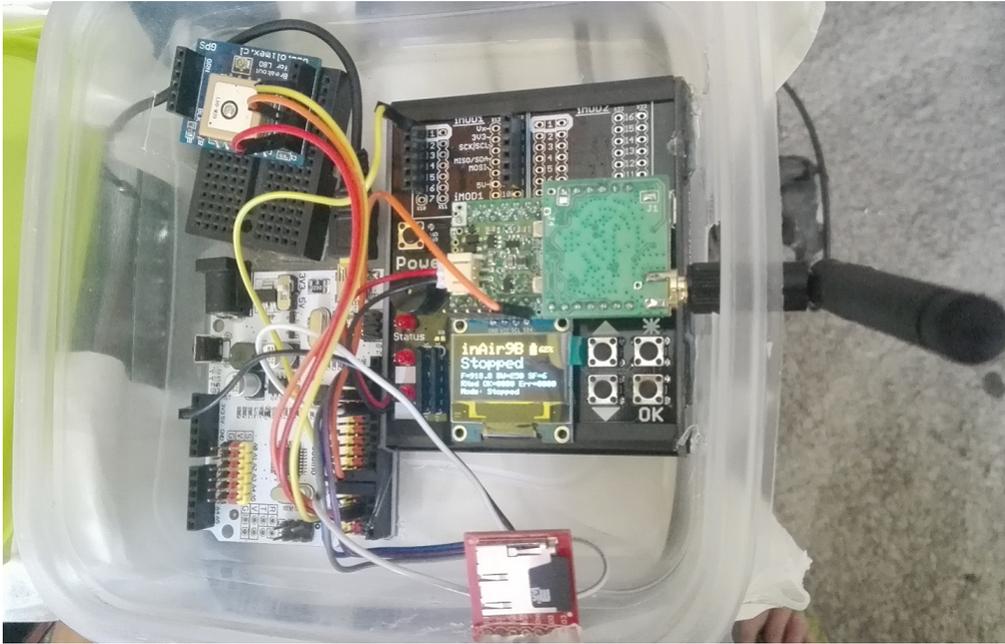


Figura 3.1: *End-device* en funcionamiento.

El *End-device* fue configurado en “Mode” “master”, dado esto, a medida que avanza arroja un paquete-ping de LoRa a la estación base por el enlace inalámbrico, espera el paquete-pong de respuesta y luego procesa el SNR y RSSI para el paquete-pong recibido. Además, el paquete-pong recibido trae la información del RSSI con el que se recibió el paquete-ping en la estación base. Finalmente, los contadores de “paquetes recibidos” y “paquetes errados” aumentan en función de la cantidad de enlaces ping-pong correctamente realizados o errados.



Figura 3.2: Estación Base en funcionamiento en borde ventana.

3.1.2. Estación Base: Nodo Receptor

La Estación Base es el nodo pareja del Dev-Kit, programado en modo “Slave 1” para ser el repetidor de los paquetes-pong. En la Fig. 3.2 se puede observar la ubicación de la estación base fijada en el borde de la ventana, este módulo fue configurado para los mismos parámetros de radio que el otro nodo. Para cada prueba se seleccionaron los mismos parámetros de BW, SF y Frecuencia que su par ubicado en el *end-device*.

3.2. Pruebas

Las pruebas fueron realizadas moviendo el *end-device* alrededor de la estación base (ubicada a unos 7[m] sobre el piso), como se ejemplifica en la Fig. 3.3. Se realizaron dos tipos de pruebas, una de forma lineal para medir el decaimiento de la potencia en función del radio y una de cobertura desplazándose por aproximadamente 500[m] a la redonda de la estación base.

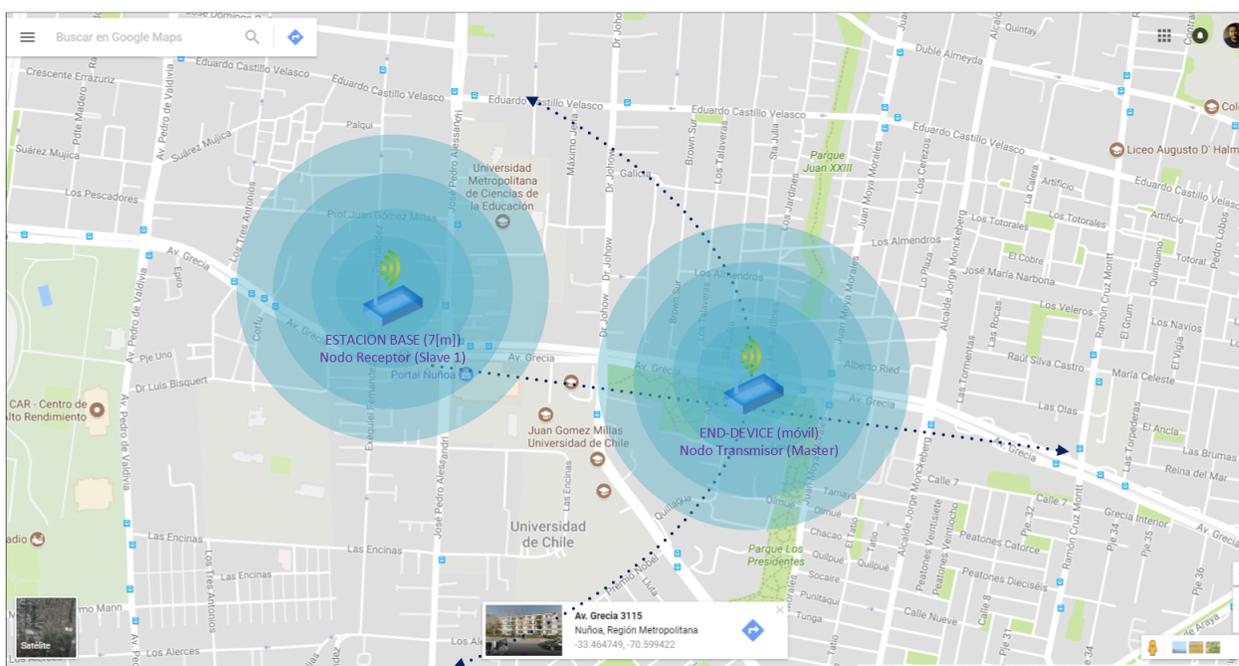


Figura 3.3: Mapa de la zona bajo estudio y el recorrido aproximado del *end-device*.

En la Fig. 3.3 se puede apreciar un mapa georreferenciado de la zona escogida y la trayectoria tentativa de los nodos durante las pruebas, se debe recordar que ambas antenas son omnidireccionales, por lo que se espera que las variaciones en la medición de RSSI dependan principalmente de las perturbaciones causadas por el ambiente sobre el canal.

3.2.1. Terreno de Pruebas

Las pruebas se llevaron a cabo en una zona urbana de Santiago de Chile, la locación escogida es en el límite entre las comunas de Ñuñoa y Macul, en la intersección de las calles Av. Jose Pedro Alessandri y Av. Grecia. La zona presenta un ambiente desafiante con mucho tendido eléctrico, un *Mall* de 5 pisos en la esquina S-E y edificaciones que bordean los 5 pisos distribuidas de manera dispersa por toda la zona. También se tiene el campus Juan Gómez Milla y el ex Pedagógico en las esquinas S-E y N-E respectivamente, los cuales presentan amplios espacios abiertos y algunas edificaciones mayores.



Figura 3.4: Foto panorámica hacia el Norte desde la ubicación de la estación base.

3.2.2. *End-device* móvil

El *end-device* se instaló en la parte trasera de la parrilla de una bicicleta para realizar el desplazamiento por el terreno de pruebas. En la Fig. 3.5 se puede apreciar el *end-device* en funcionamiento montado sobre la bicicleta y la batería externa sobre la tapa verde del case plástico.



Figura 3.5: *End-device* montado en bicicleta.

3.3. Mapas de Calor: *Heatmaps*

Los mapas de calor son una forma visual de representar los valores del RSSI o el SNR (entre otros parámetros posibles) dentro de un mapa georreferenciado. Como se puede ver en la Fig. 1.29, del capítulo 1, las distintas intensidades recibidas en cada punto del mapa son representadas por un rango de colores. En el caso de los mapas de calor de este estudio el rango de colores va desde lo más cálido a lo más frío, para sus valores más altos y más bajos respectivamente. Se comienza en el color más cálido (amarillo) pasando por el naranja y el rojo hasta llegar a lo más frío (morado).

3.3.1. Muestras

Los datos guardados en la microSD están ordenados en un archivo .txt, como se puede observar en la Fig. 3.6. Cada fila representa un punto de medición con todos sus parámetros separados por coma, los primeros dos pertenecientes a la posición GPS y los últimos cinco pertenecientes a los datos del enlace LoRa. Luego estos datos son convertidos a un archivo .csv y tratados para ser representados por el *software* de mapeo.

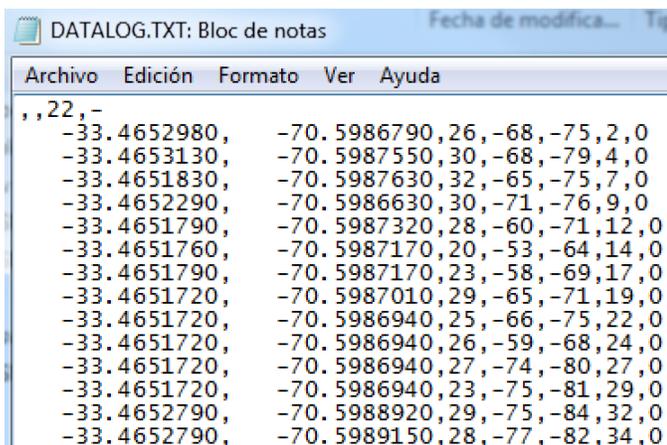


Figura 3.6: archivo .txt con las muestras.

Para transformar los archivos .txt en .csv se utilizó el programa Microsoft Excel 2010, dado que los datos están separados por comas es fácil para Excel identificar los cambios de columna, y como están ordenados por filas Excel los ordena tal cual. Como se aprecia en la Fig. 3.7 los datos del .txt ahora están distribuidos por filas y columnas, y además a cada columna se le agregó su etiqueta correspondiente en la primera fila.

A1		fx		latitude					
	A	B	C	D	E	F	G		
1	latitude	longitude	lora_SNR	rssi L	rssi R	ok	err		
2	-33.465298	-70.598679	26	-68	-75	2	0		
3	-33.465313	-70.598755	30	-68	-79	4	0		
4	-33.465183	-70.598763	32	-65	-75	7	0		
5	-33.465229	-70.598663	30	-71	-76	9	0		
6	-33.465179	-70.598732	28	-60	-71	12	0		
7	-33.465176	-70.598717	20	-53	-64	14	0		
8	-33.465179	-70.598717	23	-58	-69	17	0		
9	-33.465172	-70.598701	29	-65	-71	19	0		
10	-33.465172	-70.598694	25	-66	-75	22	0		
11	-33.465172	-70.598694	26	-59	-68	24	0		
12	-33.465172	-70.598694	27	-74	-80	27	0		
13	-33.465172	-70.598694	23	-75	-81	29	0		
14	-33.465279	-70.598892	29	-75	-84	32	0		
15	-33.465279	-70.598915	28	-77	-82	34	0		
16	-33.465305	-70.598869	24	-78	-87	37	0		
17	-33.465324	-70.598793	26	-83	-96	39	0		

Figura 3.7: archivo .csv con las muestras.

3.3.2. Ploteo de mapas

El programa utilizado para crear los mapas, se llama “myheatmap”¹ y es una aplicación *web*, requiere crear un usuario para entrar en la interfaz donde es cargado el archivo .csv. La aplicación interpreta las etiquetas del .csv para crear la representación, las primeras dos columnas deben corresponder a la latitud y longitud, cada columna de datos posterior creara un *layer* independiente en el mapa. Cada *layer* crea un mapa de calor distinto para el mismo mapa georreferenciado.

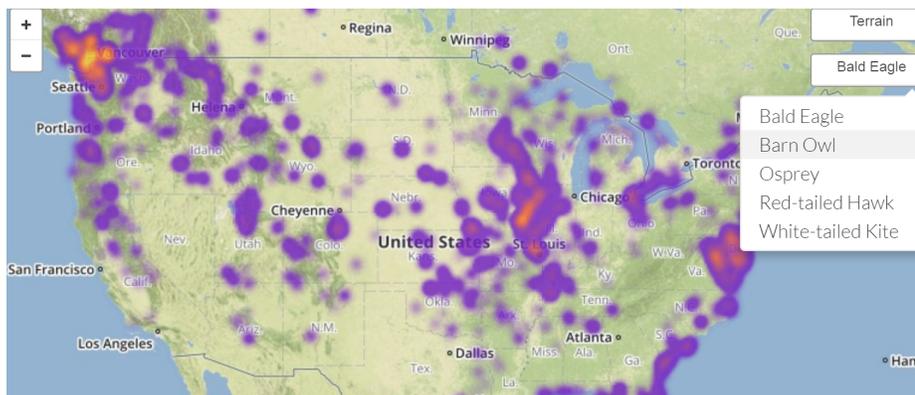


Figura 3.8: ejemplo de *heatmap*, para densidad de aves de Norteamérica.

En la Fig. 3.8 se aprecia un mapa de calor de la densidad de aves en Estados Unidos. A la derecha se ven los *layer* posibles (Bald Eagle, Barn Owl, etc) y en los casilleros superiores se ve la etiqueta correspondiente al mapa de calor mostrado y la selección del tipo de mapa, ya sea terreno (Terrain), Street (Calles marcadas) o Satellite (una imagen satelital).

¹<https://myheatmap.com/>

Capítulo 4

Análisis de resultados

En este capítulo se presenta el análisis de los datos obtenidos en las pruebas realizadas. Primero se mostrará la configuración relevante del radio enlace definido por *firmware*, luego se mostrará el análisis gráfico de las muestras obtenidas y finalmente los *heatmap* obtenidos.

4.1. Parámetros del radio enlace

En el *firmware* cargado en el STM32 se definen varios parámetros de la capa física que luego son cargados y ejecutados en el *transceiver* de LoRa SX1276. La capa de enlace y física es controlada en conjunto por ambos chips, tal como se ejemplificó en la Fig. 1.6 del capítulo 1, pero en este caso la CPU es el microcontrolador STM32 y el adaptador de red el InAir9b(SX1276). A continuación, se muestran varias definiciones importantes para la configuración de los parámetros de Radio, estas se encuentran ubicadas en distintas líneas de código de los archivos del *firmware* descargado de GitHub.

4.1.1. Configuración de Radio y Paquetes de LoRa

Primero que todo, podemos ver la selección del módulo de radio inAir9b configurado a 915[MHz] en la línea 51 de la Fig. 4.1, como se puede ver al final de la definición están los parámetros por *default* correspondientes, se puede ver que el último parámetro es la potencia de transmisión *Tx power* (17[dBm]) a la que esta configurado el módulo de radio.

```
50 //#define DEVKIT_FOR_INAIR9_868 //Freq = 868.8MHz, BW=250, SF=12, Pwr=14dBm  
51 #define DEVKIT_FOR_INAIR9B_915 //Freq = 918.8MHz, BW=250, SF=12, Pwr=17dBm  
52 //#define DEVKIT_FOR_INAIR9B_868 //Freq = 868.8MHz, BW=250, SF=12, Pwr=17dBm  
53
```

Figura 4.1: Definición del módulo de radio inAir9b y su Freq, BW, SF y Pwr por *default*.

En la definición de la línea 128 de la Fig. 4.2 se puede ver que el CRC del paquete LoRa está habilitado. Esto quiere decir que para calcular el tamaño total de información efectiva enviada se debe considerar el código de redundancia cíclico para el *payload*, ubicado al final del paquete.

```
127 #define LORA_IQ_INVERSION_ON           false    SF7..SF12
128 #define LORA_CRC_ENABLED              true      1=4/5, 2=4/6, 3=4/7, 4=4/8
```

Figura 4.2: Definición para habilitar CRC.

Otros parámetros importantes del paquete de LoRa enviado son definidos en las líneas 119, 120, 121 y 124 de la Fig. 4.3, se puede observar el SF, el valor de “n” para el *CodingRate*(CR), el tamaño del preámbulo y la habilitación del *header* respectivamente. Debido a que la definición “LORA_FIX_LENGTH_PAYLOAD_ON” está asignada como “false”, se debe considerar el *header* opcional en el paquete de LoRa.

```
119 #define LORA_SPREADING_FACTOR         12      // SF7..SF12
120 #define LORA_CODINGRATE               1      // 1=4/5, 2=4/6, 3=4/7, 4=4/8
121 #define LORA_PREAMBLE_LENGTH         8      // Same for Tx and Rx
122 #define LORA_SYMBOL_TIMEOUT           5      // Receive Timeout in Symbols
123 #define RX_TIMEOUT_VALUE              5000000 // Receive Timeout in us = 5
124 #define LORA_FIX_LENGTH_PAYLOAD_ON    false   // true = No Header(Implicit
```

Figura 4.3: SF, tamaño preámbulo, CR y habilitación *header*.

Finalmente, en las Fig.s 4.4 y 4.5 se aprecia el tamaño de la *data* insertada en el campo *Payload* de los paquetes. En la primera imagen se observa que en las líneas 369 y 370 se inserta la *data* en el *payload* del paquete “Ping” enviado por el *end-device* a la estación base. En la segunda figura, en las líneas 1264, 1265 y 1266, se puede ver la definición de la *data* insertada en el mensaje “Pong”, enviado como respuesta desde la estación base al *end-device*.

```
366 //Send PING message to slave. Message has following format:
367 // - Byte0:      Slave address
368 // - Byte1-4:    "pInG"
369 radioData[0].txBuf.put(appConfig.remotelAdr);
370 radioData[0].txBuf.putArray((unsigned char*)pingMsg, 4);
371     }
372 }
```

Figura 4.4: Formato del mensaje Ping.

Todos los parámetros definidos en el código son enviados como información desde el STM32 al inAir9b mediante el canal de comunicación SPI definido en el iMod2.

```

1262     if(pRadioData->rxLen > 2) {
1263         //If Master, check for reply PONG message. Message has following format:
1264         // - Byte0:      Slave address
1265         // - Byte1-4:    "pOnG"
1266         // - Bytes5-6:  RSSI value_slave received our "PING" message at

```

Figura 4.5: Formato del mensaje Pong.

4.1.2. Resumen de Parámetros

Dadas las definiciones anteriores, en la Tbl.4.1 se resumen los principales parámetros para el módulo inAir9b, estos parámetros están definidos por *default* en el *firmware* pero son configurables mediante la interfaz del DevKit (salvo el Tx pwr). Por otro lado, en la Tbl. 4.2 se resumen los parámetros principales utilizados para calcular el tamaño de los paquetes enviados por el radio enlace.

Freq Transmisión	918.8[MHz]	BW	250[kHz]
SF	12	Tx pwr	17[dBm]

Tabla 4.1: Parámetros de Radio por *default*.

Preámbulo	8 [bytes]	Optional Header	habilitado
CR	4/5	Optional CRC	habilitado
CRC Payload	1 [byte]	Pong Payload	7 [bytes]
Ping Payload	5 [bytes]	–	–

Tabla 4.2: Parámetros de los paquetes LoRa por *default*.

4.1.3. Calculo de parámetros del enlace LoRa

Con la información anterior se pueden calcular algunos parámetros relevantes del enlace, correspondiente a la información presentada en la sección 1.2.3. El *Data-rate* o R_b es calculado utilizando la Ecu. 1.6, luego, utilizando el BW como el parámetro R_c y la ecuación 1.5 se calculó la G_p . Este cálculo se realizó respectivamente para dos BW distintos, 250[kHz] (*default*) y 500[kHz] (prueba de cobertura). Como se puede ver por los resultados, para el doble del BW se obtiene una pérdida de 3[dBm] similar a las pérdidas esperadas según la Tbl. 1.2, del capítulo 1.

BW_1	250[kHz]	BW_2	500[kHz]
R_{b1}	585.9[bps]	R_{b2}	1171.9[bps]
G_{p1}	26.3[dBm]	G_{p2}	23.3[dBm]

Tabla 4.3: Parámetros calculados del enlace LoRa.

Con estos parámetros calculados, se decidió hacer dos pruebas, la prueba uno utilizando los parámetros por *default* para analizar el comportamiento del enlace al variar solo la distancia en una dirección y la prueba dos, donde se aumenta el BW para disminuir la ganancia y tener un mapa de cobertura en todas direcciones pero mas compacto. Cabe destacar que se decidió por los parámetros por *default* ya que al utilizar ciertas combinaciones de BW y SF la comunicación fallaba.

4.2. Prueba 1: Prueba de decaimiento

La primera prueba fue realizada con el objetivo de medir el alcance para las características por *default* del Dev-Kit. Debido a que la potencia recibida en la estación base disminuye a medida que aumenta la distancia al *end-device*, a razón de $\frac{1}{r^2}$, la prueba se realizó tratando de mantener las coordenadas (polares) Θ y Φ con la menor variación posible mientras que el *end-device* se alejaba de la estación base por la coordenada R . En palabras simples el *end-device* avanzó desde la estación base hacia el Este (ref. hacia la cordillera) siguiendo por la calle Grecia hasta llegar a la rotonda Grecia.

Los parámetros ambientales y la configuración de los parámetros de radio se pueden ver en la Tbl. 4.4. La prueba completa recolectó 500 datos, pero se utilizaron entre 110 y 216 para hacer el análisis gráfico.

Velocidad max.	5 km/h	Tx Power	17[dBm]
T° a la sombra	24°C	BW	250kHz
HR %	35 %	SF	12
Freq	918.8[MHz]	–	–

Tabla 4.4: Parámetros ambientales y configuración de radio prueba 1.

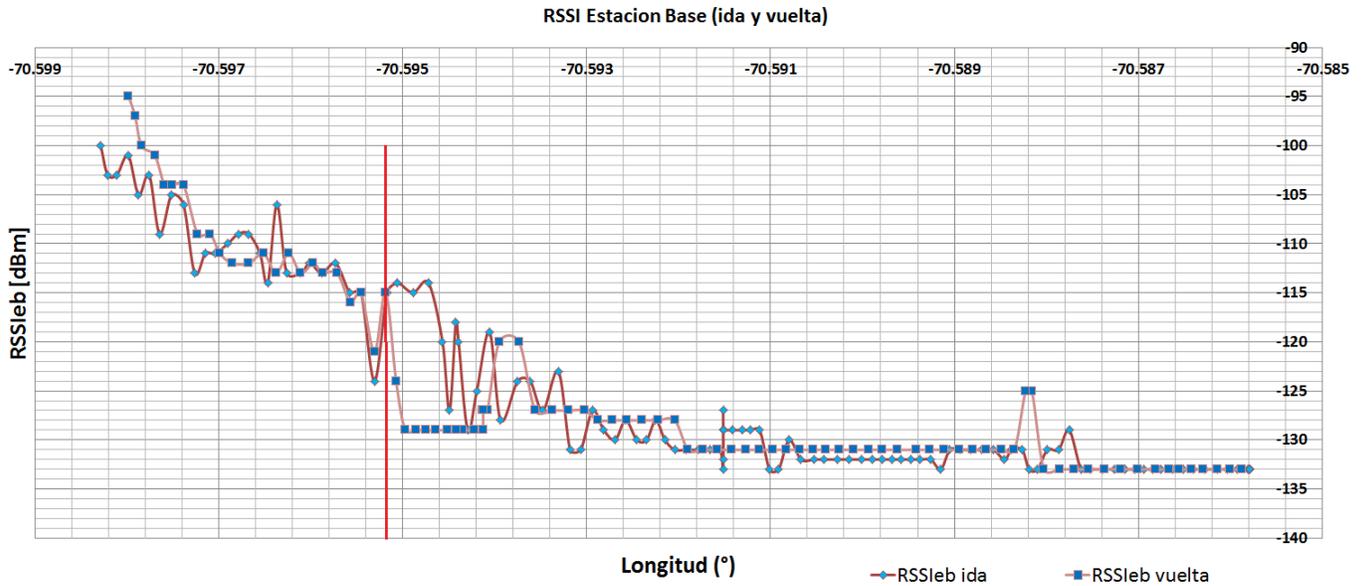


Figura 4.6: Decaimiento del RSSI de la estación Base vs distancia.

4.2.1. Análisis gráficos

En la Fig. 4.6 se puede observar como decae el RSSI medido en la estación base a medida que aumenta la distancia recorrida por el *end-device*, tal como se espera tiene un decaimiento similar a $\frac{1}{r^2}$. El gráfico muestra dos series de datos, la primera, “RSSIeb ida” generada por las muestras recolectadas mientras el *end-device* se alejaba de la estación base, y la segunda, “RSSIeb vuelta” generada por las muestras recolectadas mientras el *end-device* se acercaba a la estación base. La primera serie tiene 120 muestras y la segunda 96 muestras, y algunas longitudes se repiten en puntos con latitudes cercanas o en el punto de detención (semáforo).

En la figura se puede ver una marca roja que señala el límite en que el RSSI comienza a decaer bajo su 70%. Luego de este punto, la tasa de paquetes recibidos comienza a decaer hasta llegar a la cola del gráfico donde difícilmente se lograba entablar la comunicación. La distancia entre la estación base y el punto de comunicación aceptable, es decir que se mantiene aproximadamente sobre el 70% del RSSI, es de 270[m]. Al superar la longitud (-70.592 S/E), aproximadamente a unos 590[m] de la estación base, el RSSI llega a su piso (cola del gráfico). Luego de este punto se retoma la comunicación a una distancia mayor, pero este punto que se observa solo en la serie “RSSIeb ida” será visto más adelante.

El SNR y el RSSI medido en el *end-device* mientras este se alejaba de la estación base se puede observar en la Fig. 4.7, donde se aprecia que ambos parámetros cambian a la par, comportamiento esperado pues ambos están relacionados. Se puede apreciar que antes de los 590[m] (long.-70.592 S/E) donde las curvas llegan a su piso, se obtienen valores del RSSI superiores a -130[dBm], con valores de SNR entre los -30[dBm] y los -50[dBm]. Este comportamiento al llegar a la cola del gráfico, define donde termina la zona de cobertura primaria, pues, aunque se llega al piso de sensibilidad se tiene un SNR suficiente para enlazar

de vez en cuando.

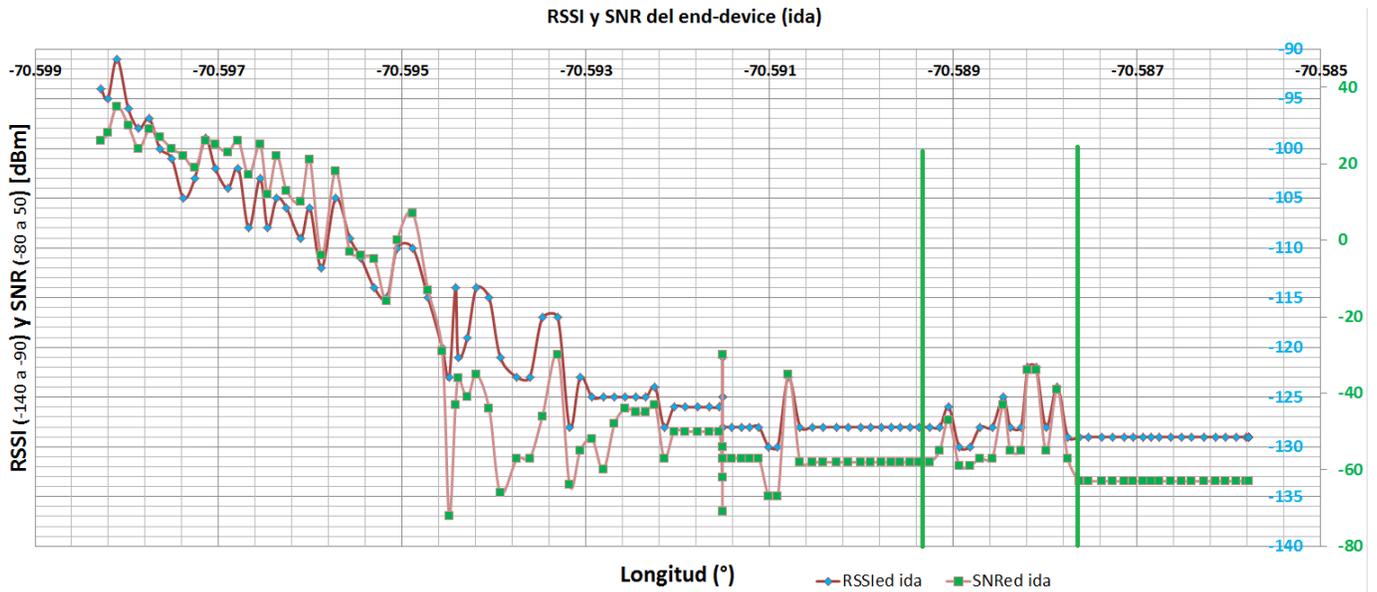


Figura 4.7: Decaimiento del RSSI y SNR *end-device* ida (120 puntos).

Luego de la zona de cobertura primaria viene una zona donde no se puede enlazar la comunicación, hasta llegar a la zona delimitada entre líneas verdes. La zona entre las líneas verdes es una ventana de recepción, causada seguramente por la disminución del *shadowing* presente en el ambiente. Esta ventana está ubicada desde los 910[m] hasta los 1050[m] aproximadamente. Luego de este punto el SNR en el *end-device* llega a -63[dBm] y no se vuelven a encontrar ventanas de transmisión más adelante.

Finalmente se comparó el SNR del *end-device* con la cantidad acumulada de paquetes recibidos ok (ECok), la cantidad de paquetes recibidos determina la cantidad de enlaces “Ping-Pong” completados de manera correcta. En la Fig. 4.8, se aprecian tres zonas:

1. La primera, entre líneas azules muestra que, aunque se dejó de recibir paquetes mientras el *end-device* se desplazaba (“ECok” plano antes y después del punto de detención), la conexión se retomó de manera sostenida por un par de muestras cuando el *end-device* se detuvo (por eso se ven puntos apilados uno sobre otro).
2. La segunda, con su número en rojo, muestra una zona donde se perdió la comunicación de manera sostenida.
3. La tercera zona, entre las líneas verdes, muestra la ventana extra de recepción donde se logró retomar la comunicación aproximadamente 1[km] de la estación base.

Además de lo señalado anteriormente, en la Fig. 4.8, también se puede apreciar que la zona de cobertura primaria (590[m]) llega hasta la primera línea azul. Esta zona es equivalente a la zona donde el parámetro “ECok” se incrementa de manera sostenida a una tasa constante, más allá de esta zona (zonas 1, 2 y 3 en Fig. 4.8) se encuentran intermitentemente problemas de comunicación, hasta llegar al piso del SNR donde ya no hay comunicación.

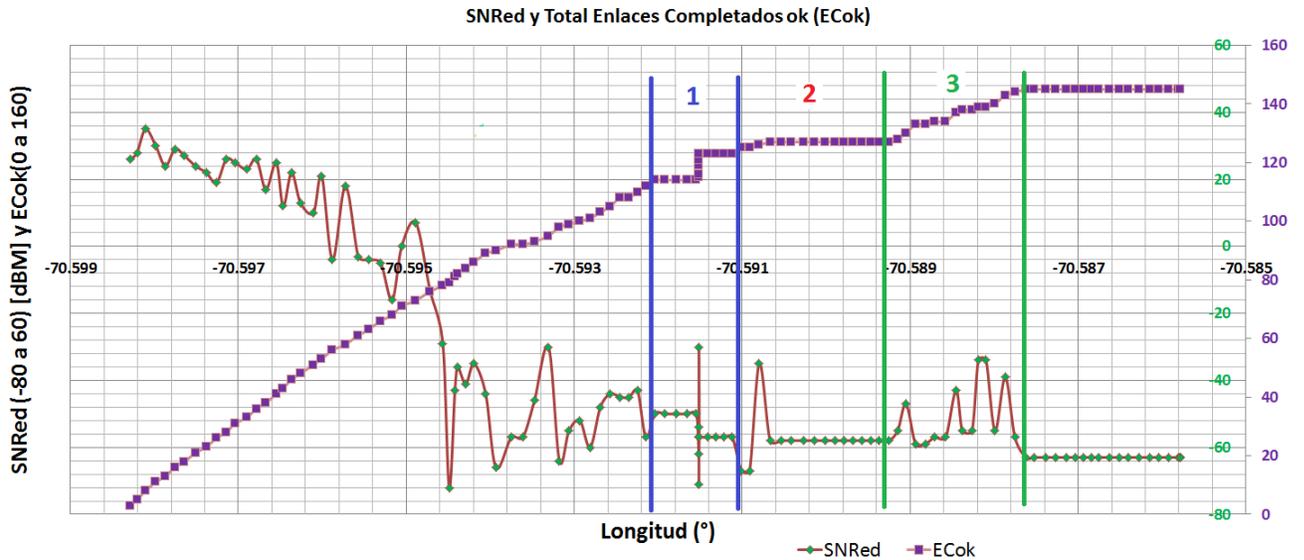


Figura 4.8: SNRed y ECok en función de la longitud.

Para concluir se agregó la línea de tendencia que mejor se ajusta al decaimiento del RSSI obtenido en la estación base, entregando una similitud cercana al 90 % ($R^2 = 0,92$). Con esto se puede calcular el valor esperado del RSSI (y) en función de la longitud (x), con cierto grado de certeza. Para este gráfico se normalizaron los datos dejando solo valores mayores a cero y quitando los puntos repetidos en el eje y , se utilizaron 110 datos. La línea de tendencia exponencial es particularmente buena para variables que decaen (exponente negativo) o incrementan (exponente positivo) a una tasa decreciente hasta finalmente estabilizarse cerca de una cota. Es preciso mencionar que, esta es la tendencia para los valores medidos en la práctica, en el caso de la teoría, donde se pueden calcular más puntos del decaimiento al inicio y al final del gráfico, se puede esperar una tendencia mas cercana a la potencial ($y = a * \frac{1}{x^b}$).

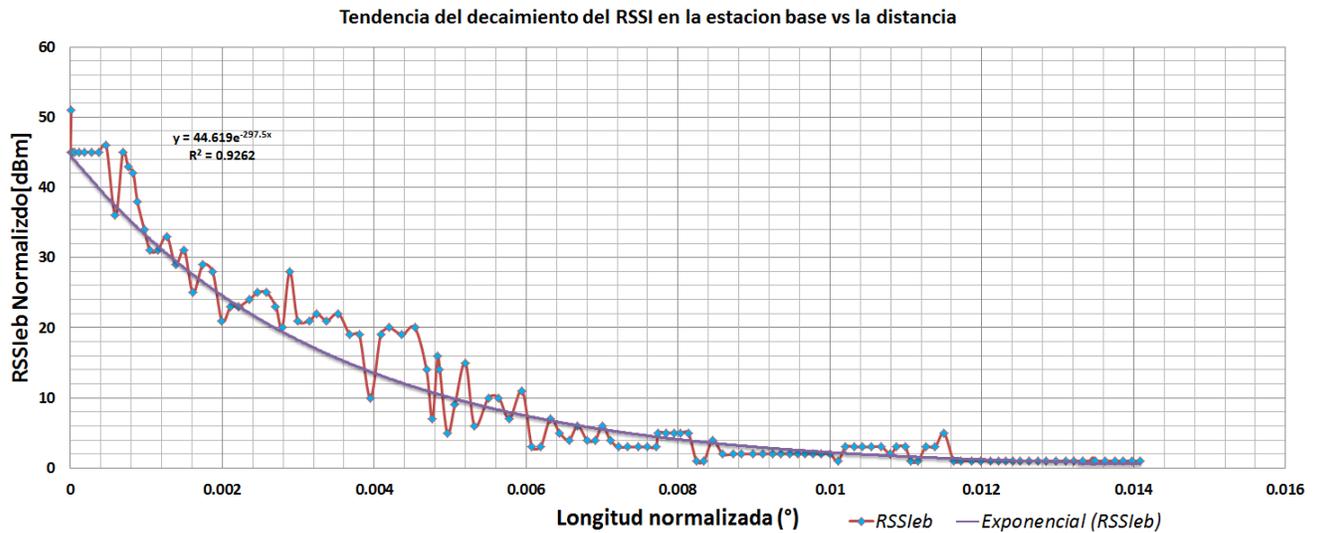


Figura 4.9: RSSIeb y su tendencia vs Longitud.

4.2.2. Heatmap Prueba 1

En la Tbl. 4.5 se resumen los principales umbrales encontrados al analizar los datos de la prueba 1. Por una parte se tiene la zona de cobertura primaria (Z12), donde el piso de sensibilidad es aproximadamente $-130[\text{dBm}]$ y llega hasta los $590[\text{m}]$, y por otra, se tiene una ventana de recepción (Z3) a aproximadamente $1[\text{km}]$ de la estación base donde se logra superar otra vez el piso de sensibilidad. También se tiene la zona de mejor recepción (Z11), donde se obtiene una potencia superior al 70% del RSSI, y por último una zona de *shadowing* (Z2) donde se corta el enlace.

Zona	inicio [m]	fin [m]	RSSI esperado
Z11	<270	270	>70 %RSSI
Z12	270	590	> $-130[\text{dBm}]$
Z2	590	910	< $-130[\text{dBm}]$
Z3	910	1040	> $-130[\text{dBm}]$
Z4	1040	>1040	< $-130[\text{dBm}]$

Tabla 4.5: Resumen Zonas prueba 1.

En la Fig. 4.10 se puede apreciar el “heatmap” de la Prueba 1. Este Layer (PRok2) representa las zonas donde se obtuvo comunicación, es decir las zonas más cálidas representan donde se logró establecer uno o más enlaces “ping-pong” (1, 2 o hasta 3). En el mapa se pueden apreciar las marcas correspondientes a la Z11, Z12, Z2, Z3 y Z4, la última zona (Z4) se extiende hasta llegar a la rotonda Grecia, no obstante, como no se encontraron más ventanas de transmisión se omitieron esas muestras pues no agregan información relevante al análisis.

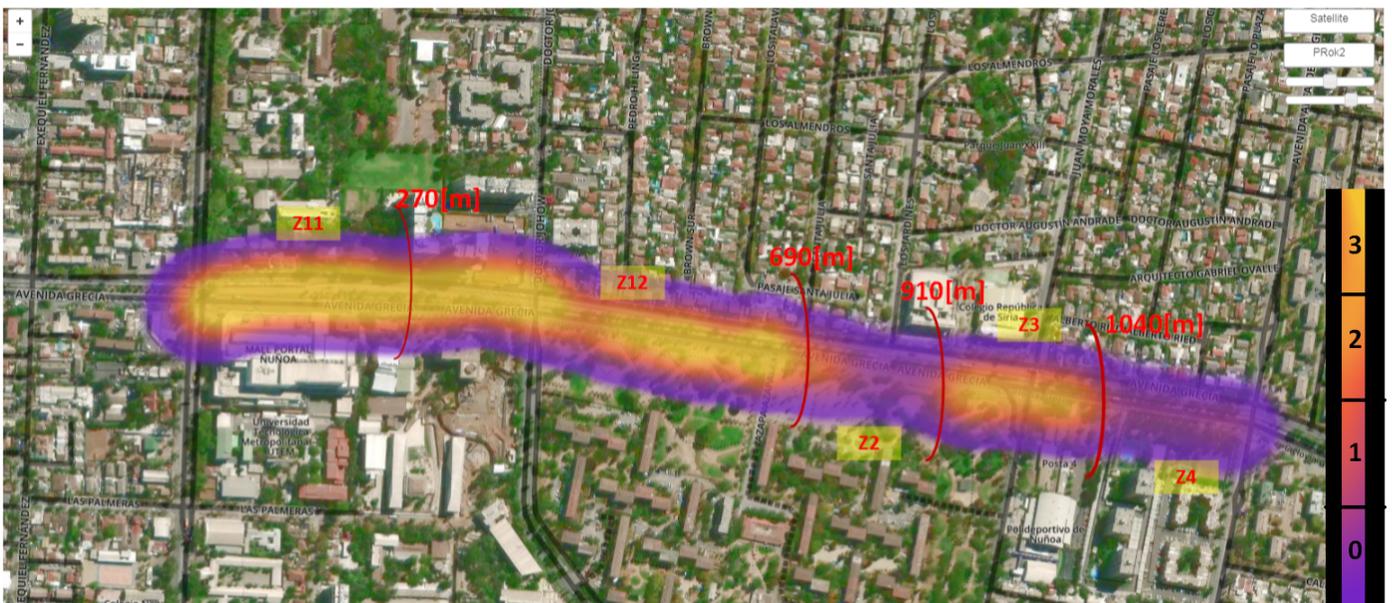


Figura 4.10: Heatmap de comunicación LoRa Prueba 1.

4.3. Prueba 2: Prueba de cobertura

Para concluir este análisis, se muestra el gráfico de cobertura realizado como un barrido del terreno, con una radio aprox. de 600[m]. Considerando que se tiene una pérdida de 3[dBm] en el *linkbudget* al duplicar el BW para esta prueba, se asume que la zona de comunicación primaria de este enlace LoRa será menor al observado en la prueba 1, por este motivo se determina que un radio en el rango de 500[m] a 700[m] es suficiente.

En la Tbl. 4.6 se resumen los parámetros ambientales y de configuración de radio, se puede ver que las condiciones de la prueba 2 son muy similares a las de la prueba 1 salvo por el aumento en el BW.

Velocidad max.	5 km/h	Tx Power	17[dBm]
T° a la sombra	28°C	BW	500[kHz]
HR %	34 %	SF	12

Tabla 4.6: Parámetros ambientales y configuración de radio prueba 2.

4.3.1. Heatmap Prueba 2

En el *heatmap* presentado en la Fig. 4.11 se puede visualizar el *Layer* (PRok2), este *layer* representa las zonas donde se obtuvo comunicación, es decir las zonas más cálidas (amarillo-naranja) representan donde se logró establecer uno o más enlaces “ping-pong” (1, 2 o hasta 3) mientras que las zonas más frías (púrpuras) representan las muestras tomadas donde no había comunicación.

En este *heatmap* se pueden ver 5 puntos verdes, estos puntos representan las muestras con cobertura LoRa obtenidas a mayor distancia de la estación base. Hacia el Oeste se obtuvo el punto “PO” con una cobertura de 500[m], hacia el Norte se obtuvieron los puntos “PN1” y “PN2” que rondan los 450[m], hacia el Este se obtuvo el punto “PE” con 430[m], y a hacia el Sur se obtuvo el punto “PS” que arrojó la mayor cobertura llegando a los 670[m]. Se debe recordar que, hacia el Norte, entre los puntos “PN1” y “PN2” existen 3 edificios de más de 10 pisos, como se puede observar en la Fig. 3.4, estos edificios generan una zona de *shadowing* extra que explica parte de la pérdida de cobertura tras los puntos antes señalados.

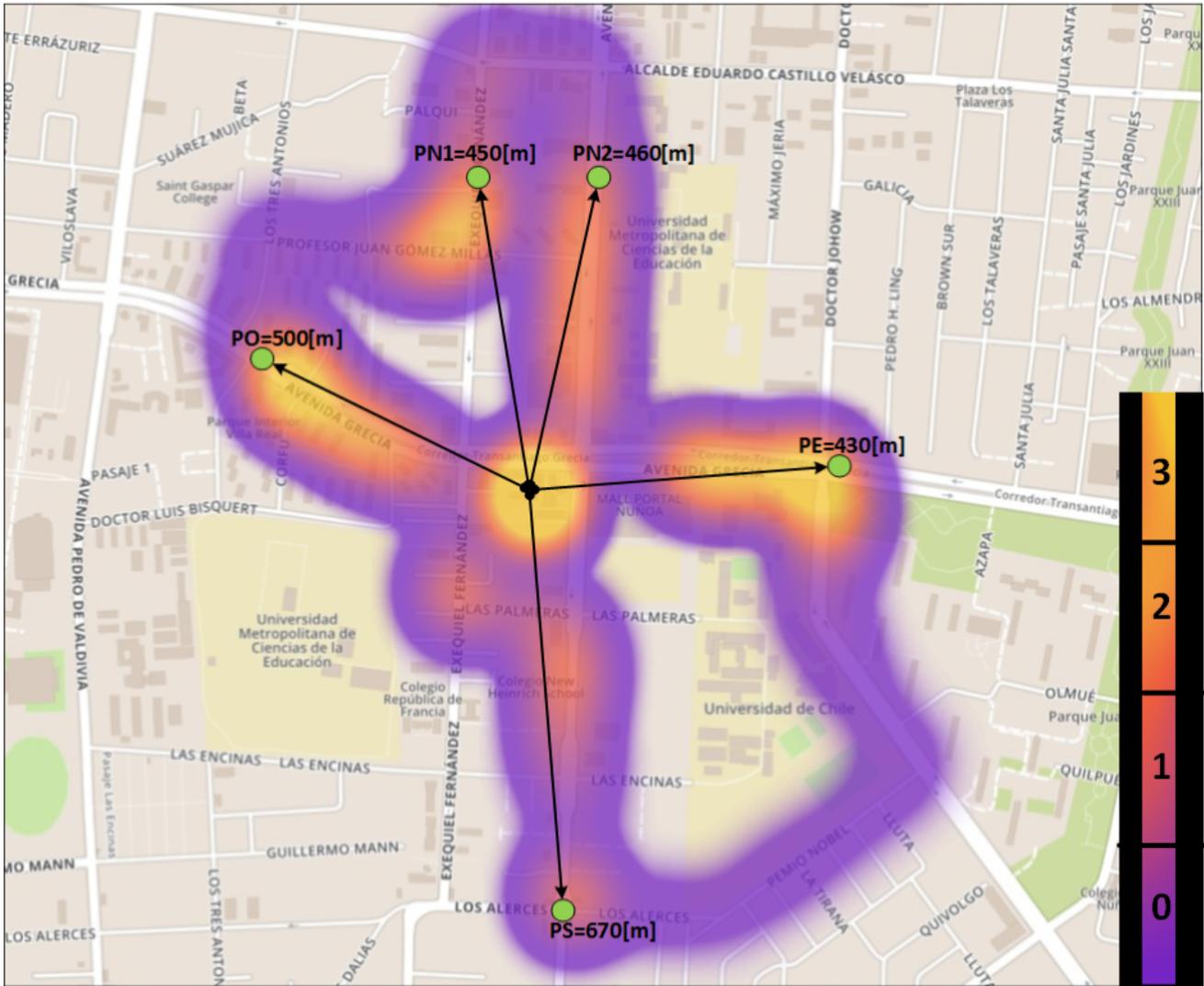


Figura 4.11: *Heatmap* de comunicación LoRa Prueba 2.

Tal como se espera de este ambiente urbano el mapa de cobertura presenta una asimetría debido a las perturbaciones características del terreno (obstáculos e interferencias). Se puede apreciar que debido a los grandes bloqueos (mall, campus y edificios sobre 10 pisos) hacia el Noreste el área de cobertura es menor, en cambio, hacia el lado Suroeste se obtuvo un alcance mayor pues las edificaciones son mas parejas y bordean los 5 pisos.

A pesar de las pérdidas de intensidad generadas por el ambiente, la comunicación inalámbrica sigue teniendo una cobertura considerable y omnidireccional. Este resultado es esperable para otros ambientes urbanos que presenten características similares a las descritas en el punto 3.2.1. Además, considerando los resultados de la prueba 1, es veraz decir que el enlace de comunicación LoRa punto a punto puede llegar a tener una cobertura omnidireccional mayor a 1 kilómetro en zonas urbanas de Santiago. Al menos en zonas que presenten tantos o menos obstáculos como los descritos en este trabajo, inclusive se puede ampliar la cobertura aun mas si se mejora la ganancia y/o la altura de la estación base (mejora de la zona de Fresnel).

Conclusión

Concluida esta memoria de título se puede decir con respecto a los Objetivos planteados, que:

1. Se presentó una explicación detallada para comprender los aspectos técnicos detrás de las redes inalámbricas M2M y LPWAN. Además se entrega información relevante para entender la teoría aplicada en la capa física y de enlace, particularmente para los enlaces inalámbricos de difusión compartida.
2. Luego se entrega una descripción detallada de los aspecto teórico/técnicos detrás de la modulación LoRa. Así mismo se explica en detalle las principales causas por las que se generan las pérdidas que sufren los enlaces inalámbricos en ambientes urbanos.
3. Se resumieron y expusieron resultados obtenidos en estudios similares realizados en Francia y Finlandia.
4. Se diseñó e implementó un sistema M2M de medición de datos, capaz de almacenar la posición GPS y los parámetros del radio-enlace (SNR y RSSI) para un sistema de comunicación inalámbrico con modulación LoRa.
5. Se desplegaron los Datos obtenidos en forma de *heatmap* sobre planos satelitales de la zona estudiada.

En referencia a los resultados se puede señalar que, fue posible observar empíricamente el comportamiento de la modulación LoRa en los alrededores de la intersección de Av. Macul con Av. Grecia. Este escenario se definió como un ambiente urbano con características desafiantes para el correcto funcionamiento de un enlace inalámbrico, por la presencia de *shadowing*, *fading* e interferencia. Los resultados obtenidos para las pruebas 1 y 2, corresponden a un canal de comunicación de 250[kHz] y 500[kHz] respectivamente, con ambos canales centrados en la frecuencia de portadora 918.8[MHz] (perteneciente a la banda ISM de 915[MHz]) y un *Spread Factor* de 12. Tal información permitió analizar la veracidad de la hipótesis planteada, con respecto a esto se puede decir que:

1. Se corroboró que la modulación LoRa es capaz de transmitir paquetes de información a más de 500[m] para dos configuraciones del parámetro de radio BW. Considerando que con el SF=12 se tiene el mejor piso de sensibilidad para el *end-device*.
2. Se observó que dentro de la zona de cobertura primaria la calidad del enlace se mantiene, llegando hasta los 690[m] para el BW de 250[kHz].
3. En un caso particular se pudo alcanzar distancias de hasta 1[km], aunque, se debe analizar empíricamente el patrón de *shadowing* del ambiente para encontrar tales ventanas.

Los resultados corroboran la capacidad de LoRa para generar una LPWAN dentro de un ambiente urbano de Santiago (*Wide Area*), según las características descritas en [4]. Se debe considerar que se puede incrementar la distancia de transmisión aún más con algunos *upgrades* como se define en la tabla 4.7.

Beneficio	upgrade	upgrade	upgrade
Aumenta Rango de Cobertura	Disminuir el BW	Aumentar G de las antenas	Aumentar la altura y PTx de la Estación Base
Disminuye PER	Aumentar el CRC	Ajustar a frecuencia con menor interferencia	

Tabla 4.7: Beneficios y Upgrades Necesarios

Dado los alcances de este trabajo solo se analizó la cobertura y no la característica de bajo consumo (*Low Power*). Además podemos señalar que, considerando la información estudiada para elaborar este documento, LoRa satisface las características típicas de las LPWAN:

- **Alta densidad de *end-points***: LoRaWAN soporta miles de nodos gracias al tamaño de su *Address Name*.
- **Costo de Hardware reducido**: Chip Semtech SX1276 cuesta 5[USD] la unidad por sobre las 500 unidades ¹.
- **Bajo *data rate***: Del orden de los [kbps], con respecto a las tecnologías 4G,5G o WiFi que llegan a los [Mbps].

Trabajo Futuro

A continuación se plantean algunos estudios que se podrían realizar utilizando el sistema de comunicación desarrollado en esta memoria.

- Utilizando otros datos como hora o tiempo de procesamiento, *packet size*, *on-air time*, entre otros se puede analizar parámetros de la capacidad de Red como *duty cycle* o *troughput*.
- Se puede agregar un sensor de corriente para medir el consumo real del dispositivo. Esto permitiría estudiar el perfil de consumo y compararlo con las características esperadas para un sistema *low-power*.
- Se puede obtener el dato de velocidad del nodo desde los paquetes GPRMC, con esto se pueden realizar pruebas de tasa de error en función de la velocidad.

¹<https://www.digikey.com/product-detail/en/semtech-corporation/SX1276IMLTRT/SX1276IMLTRTCT-ND/4259551>

Anexos

Anexo A

Lista de Siglas

IoT: Internet of Things

LPWAN: Low Power Wide Area Networks

ISO: International Organization for Standardization

OSI: Open System Interconnection

CCITT: Comité Consultivo Internacional Telegráfico y Telefónico

UIT: Unión Internacional de Telecomunicaciones

IEEE: Institute of Electrical and Electronics Engineers

WAN: Wireless Area Network

PAN: Personal Area Network

LAN: Local Area Network

MAN: Metropolitan Area Network

WSN: Wireless Sensor Network

MAC: Medium Access Control

M2M: Machine to Machine Communication

SS: Spread Spectrum

DSSS: Direct Sequence Spread Spectrum

CDMA: Code Division Multiple Access

PER: Packet Error Rate

LoRa: Long Range
CSS: Chirp Spread Sprectrum
PCB: Printed Circuit Board
FIFO: First Input First Output
GPS: Global Positioning System
TTL: Transistor-Transistor Logic
SPI: Serial Peripheral Interface
RSSI: Received Signal Strength Indicator
NMEA: National Marine Electronics Association
IDE: Integrated Development Environment
ISI: Inter Symbol Interference
QoS: Quality of Service
ADR: Adaptative Data Rate
SNR: Signal to Noise Ratio
LOS: Line of Sight
MFSK: Multiple Frecuency Shift Keying
FCC: Federal Communications Commission
ISM: Industrial, Scientific and Medical

Anexo B

Código Principal del Datalogger LoRa

```

#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <SPI.h>
#include <SD.h>

SoftwareSerial SerialGPS(2, 3);
SoftwareSerial SerialLora(5, 6);
TinyGPSPlus gps;
unsigned long time_base;
unsigned long time_now;
char buffer[50];
char sd_msg[100];
char lat[15];
char lng[15];
const int chipSelect = 4;

bool isTimeout(unsigned long t_start, unsigned long t_current, unsigned long timeout, bool print_timeout);

void setup()
{
  Serial.begin(9600);
  if (!SD.begin(chipSelect)) { //Por default son los pines 13,12,11 mas el ChipSelect
    return;
  }
}

void loop()
{
  SerialGPS.begin(9600);
  time_base = millis();
  int cerrar=0;
  while (!isTimeout(time_base, millis(), 5000ul, true)) {
    while (SerialGPS.available() > 0) {
      char c = SerialGPS.read();
      gps.encode(c);
    }
    if (gps.location.isUpdated()) {
      strncpy(lat, "", 15);
      strncpy(lng, "", 15);
      if (gps.location.isValid()) {
        dtostrf(gps.location.lng(), 14, 7, lng);
        dtostrf(gps.location.lat(), 14, 7, lat);
      }
      cerrar=1;
      //break;
    }
  }
  SerialGPS.end();
  SerialLora.begin(19200);
  time_base = millis();
  time_now = millis();
  char* start_msg;
  memset(buffer, 0, sizeof(buffer));
  int pos = 0;
  while (!isTimeout(time_base, time_now, 2000ul, false)) {
    time_now = millis();
    while (SerialLora.available() > 0) {
      char c = SerialLora.read();
      buffer[pos++ % (sizeof(buffer) - 1)] = c;
      char* end_msg = strchr(buffer, '(');
      if (end_msg) {
        start_msg = strchr(buffer, '(');
        if (start_msg) {
          start_msg++; //saltar el primer parentesis
          *end_msg = 0; //terminar mensaje
          goto end_lora;
        }
      }
      pos=0;
      memset(buffer, 0, sizeof(buffer));
    }
    time_now = millis();
  }
}

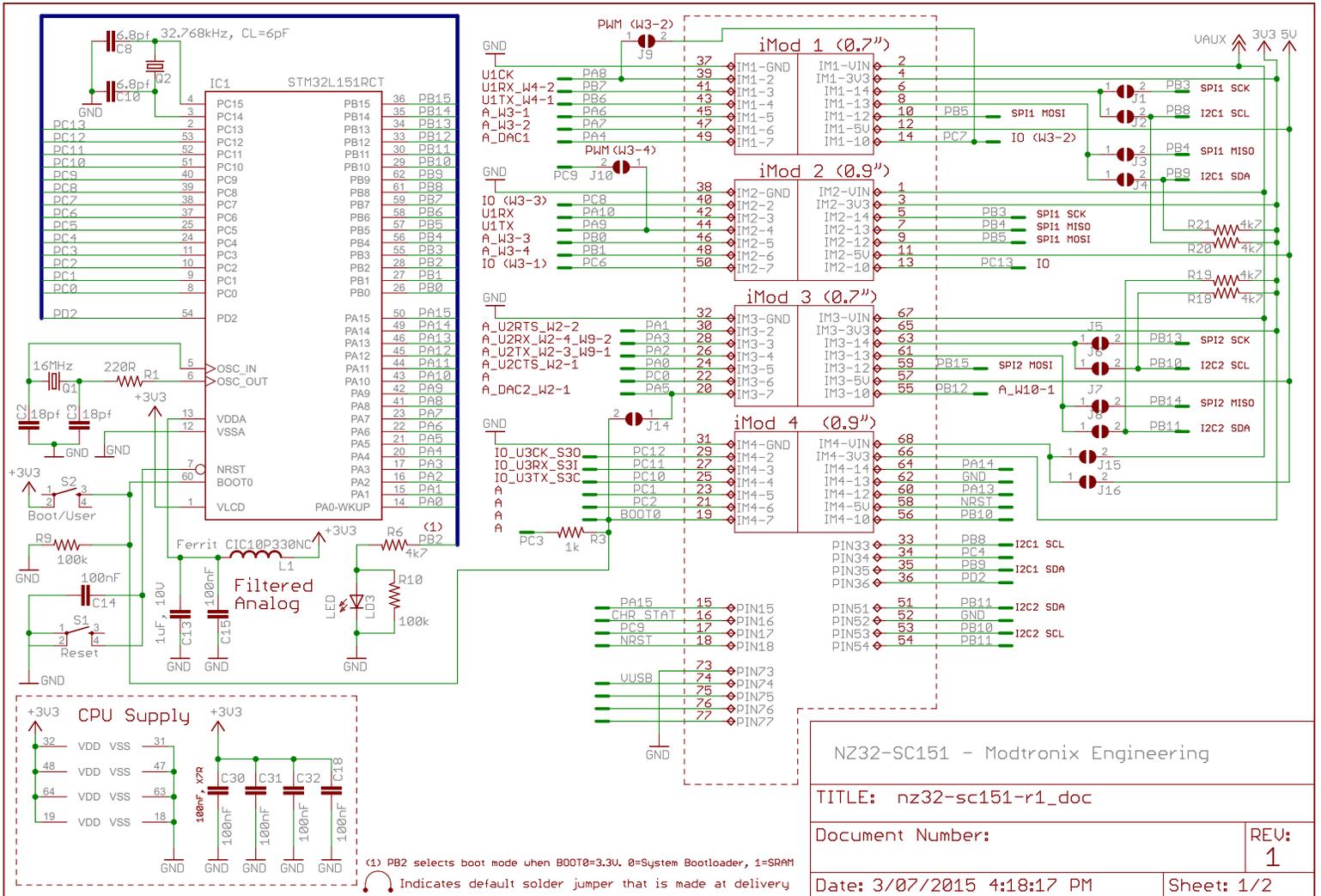
end_lora:
SerialLora.end();
sprintf(sd_msg, "%s,%s,%s", lat, lng, start_msg);
File dataFile = SD.open("datalog.txt", FILE_WRITE);
if (dataFile) {
  dataFile.println(sd_msg);
  dataFile.close();
}
else {
};
}

bool isTimeout(unsigned long t_start, unsigned long t_current, unsigned long timeout, bool print_timeout) {
  if (timeout > 600000) {
    timeout = 10000;
  }
  if (t_current - t_start >= timeout ||
      t_current - t_start < 0) {
    if (print_timeout)
      return true;
  }
}

```

Anexo C

Esquema de conexión PCB NZ32-SC151



NZ32-SC151 - Modtronix Engineering

TITLE: nz32-sc151-r1_doc

Document Number:

REV: 1

Date: 3/07/2015 4:18:17 PM

Sheet: 1/2

(1) PB2 selects boot mode when BOOT0=3.3V. 0=System Bootloader, 1=SRAM

Indicates default solder jumper that is made at delivery

Anexo D

Esquema de conexión PCB InAir9B

Bibliografía

- [1] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 6th ed. New Jersey: Pearson Education, 2013.
- [2] Semtech Corporation, “Main web,” 2017, [accessed 20-sept-2017]. [Online]. Available: <http://www.semtech.com/>
- [3] IBM Research, “Low Power Networking Technology from IBM and Semtech to Help Enable Telcos to Launch New Services for the Internet of Things,” 2015, [accessed 20-sept-2017]. [Online]. Available: <https://www-03.ibm.com/press/us/en/pressrelease/46287.wss>
- [4] A. Markkanen, “LPWA market tracker by Machina Research,” pp. 1–22, February 2016. [Online]. Available: <https://machinaresearch.com/report/lpwa-market-tracker-september-2016/>
- [5] Ingenu Inc., “Ingenu main web,” 2017, [accessed 20-sept-2017]. [Online]. Available: <https://www.ingenu.com/>
- [6] Sigfox, “Sigfox main web,” 2017, [accessed 20-sept-2017]. [Online]. Available: <https://www.sigfox.com/en>
- [7] W. Tomasi, *Sistemas de comunicaciones electrónicas*, 4th ed. PEARSON Educación, 2003.
- [8] W. Stallings, *Wireless communications and networks*, 2nd ed. New Jersey: PEARSON Prentice Hall.
- [9] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication 1*. Cambridge University Press, 2004.
- [10] Semtech, “LoRa™ Modulation Basics, Application Note,” pp. 1–26, May 2015.
- [11] J. Morrish, “With 3 billion connections, LPWA will dominate wide area wireless connectivity for M2M by 2023,” pp. 1–11, 2015.
- [12] Waviot, “Waviot NB-FI Products and Tech Description,” pp. 1–29, 2016.
- [13] Technical Marketing Workgroup of LoRa alliance, “What is it?, A technical overview of

LoRa and LoRaWAN,” Nov 2015.

- [14] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, “A Study of LoRa: Long Range & Low Power Networks for the Internet of Things,” pp. 1–18, 2016.
- [15] Nanotron Technologies, “Chirp Spread Spectrum,” pp. 1–3, March 2017.
- [16] Semtech Corporation, “Datasheet SX1276/77/78/79,” no. Rev 5, 2016, [accessed 20-sept-2017]. [Online]. Available: <http://modtronix.com/prod/components/wireless/sx1276.pdf>
- [17] B. Sklar, *Digital Communications: Fundamentals and Applications*, 2nd ed. New Jersey: Prentice Hall PTR, 2001.
- [18] Ministerio de Transportes y Telecomunicaciones; Subsecretaria de Telecomunicaciones, “Resolución 755 EXENTA,” 2017.
- [19] J. Petajajarvi, K. Mikhaylov, A. Roivainen, and T. Hanninen, “On the coverage of LPWANs: Range Evaluation and Channel Attenuation Model for LoRa Technology,” *14th international Conference on ITS Telecommunications*, 2015.
- [20] Modtronix Engineering, “Development kit Semtech LoRa SX1276 / SX1278,” 2015, [accessed 20-sept-2017]. [Online]. Available: <http://modtronix.com/devkit-lora.html>
- [21] Semtech Corporation, “Product SX1276,” 2015, [accessed 20-sept-2017]. [Online]. Available: <http://www.semtech.com/apps/product.php?pn=SX1276>
- [22] Modtronix Engineering, “NZ32-SC151,” 2015, [accessed 20-sept-2017]. [Online]. Available: <http://modtronix.com/nz32-sc151.html>
- [23] STMicroelectronics, “STM32L151RC,” 2015, [accessed 20-sept-2017]. [Online]. Available: <http://www.st.com/en/microcontrollers/stm32l151rc.html>
- [24] avrProgrammers, “ATmega328p Power Consumption,” 2016, [accessed 20-sept-2017]. [Online]. Available: <https://www.avrprogrammers.com/howto/atmega328-power>
- [25] STMicroelectronics, “DataSheet STM32L151RC,” 2015, [accessed 20-sept-2017]. [Online]. Available: http://modtronix.com/prod/components/stm32/L1_STM32L15xxCC_RC_VC_UC.pdf
- [26] Modtronix Engineering, “InAir9b page,” 2015, [accessed 20-sept-2017]. [Online]. Available: <http://modtronix.com/inair9b.html>
- [27] Semtech Corporation, “SX1276RF1JAS,” 2017, [accessed 20-sept-2017]. [Online]. Available: http://www.semtech.com/apps/filedown/down.php?file=SX1276RF1JAS_e268v01a.zip
- [28] —, “SX1276MB1LAS,” 2015, [accessed 20-sept-2017]. [Online]. Available: http://www.semtech.com/apps/filedown/down.php?file=SX1276MB1LAS_

e311v02a_prod_folder.zip

- [29] Quectel, “L80 Hardware Design,” 2013, [accessed 20-sept-2017]. [Online]. Available: http://www.quectel.com/UploadImage/Downlad/L80_Hardware_Design_V1.1.pdf
- [30] Society of Robots, “MICROCONTROLLER UART TUTORIAL,” 2014, [accessed 20-sept-2017]. [Online]. Available: http://www.societyofrobots.com/microcontroller_uart.shtml
- [31] SparkFun Electronics, “SparkFun microSD Transflash Breakout,” 2014, [accessed 20-sept-2017]. [Online]. Available: <https://www.sparkfun.com/products/544>
- [32] NT Creativas, “Freaduino Page,” [accessed 20-sept-2017]. [Online]. Available: <http://roboticay3d.com/componentes/77-freaduino-uno-rev-18.html>
- [33] Mikal Hart, “TinyGPS++,” 2014, [accessed 20-sept-2017]. [Online]. Available: <http://arduiniana.org/libraries/tinygpsplus/>
- [34] Modtronix Engineering, “Firmware for Modtronix Devkit-LoRa Development Kit,” 2015, [accessed 20-sept-2017]. [Online]. Available: https://github.com/modtronix-com/devkit_sx1276
- [35] Open STM32, “Installing System Workbench for STM32 with installer,” 2015, [accessed 20-sept-2017]. [Online]. Available: <http://www.openstm32.org/Installing%2BSystem%2BWorkbench%2Bfor%2BSTM32%2Bwith%2Binstaller>