



DISPOSITIVO LoRa DE COMUNICACIÓN A LARGO ALCANCE Y BAJO CONSUMO ENERGÉTICO PARA APLICACIONES DEL ÁMBITO DEL DESARROLLO

JOSÉ DANIEL RODRÍGUEZ MUNCA

Asignatura

Trabajo fin de Master (técnico e investigativo)

Tutor Académico: Manuel Sierra Castañer

Tutor Profesional: Álvaro Gutiérrez Martín

GENERACIÓN DISTRIBUIDA RENOVABLE Y CONTROL INTELIGENTE INSTITUTO DE ENERGÍA SOLAR

**UNIVERSIDAD DE POLITÉCNICA DE MADRID
UNIVERSIDAD COMPLUTENSE DE MADRID
MASTER INTERUNIVERSITARIO EN ESTRATEGIAS Y
TECNOLOGÍAS PARA EL DESARROLLO
MADRID - ESPAÑA
2016**

AUTORIZACIÓN DE ENTREGA Y DEFENSA DE TRABAJO FIN DE MÁSTER

D./D Álvaro Gutiérrez Martín, Profesor Contratado Doctor en la ETSI Telecomunicación de la Universidad Politécnica de Madrid

A la vista del Trabajo Fin de Máster del Estudiante, se emite el presente informe sobre la evaluación final.

Título del Trabajo: DISPOSITIVO LoRa DE COMUNICACIÓN A LARGO ALCANCE Y BAJO CONSUMO ENERGÉTICO PARA APLICACIONES DEL ÁMBITO DEL DESARROLLO

Alumno/a: JOSÉ DANIEL RODRÍGUEZ MUNCA

Valoración de su apreciación tras el desempeño del TFM

El trabajo realizado por el alumno ha sido excelente. Por un lado se ha enfrentado a problemas técnicos de integración de dispositivos de comunicaciones de largo alcance, bajo consumo y bajo coste sobre unas plataformas de hardware libre donde el alumno ha tendido que aplicar sus conocimientos técnicos para su consecución. Por otro lado, el alumno ha realizado una revisión exhaustiva de la utilización del dispositivo en el ámbito del desarrollo (agua, saneamiento, energía, salud, etc.) planteando su utilidad, proporcionando soluciones y sentando las bases para un despliegue de los dispositivos en un futuro próximo dentro del campo del internet de las cosas (IoT) y su relación con el Big Data. El documento realizado por el alumno es un documento bien escrito, sentando las bases del sistema, accediendo a las partes más complicadas del driver implementado y proporcionando al lector una cómoda lectura del mismo. Además, el documento está planteado de una manera didáctica, para proporcionar una capacidad de réplica exquisita. Por estos motivos considero que D. José Daniel Rodríguez Munca se merece una calificación sobresaliente.

Calificación Alfabética SOBRESALIENTE

Calificación Numérica 10.0

Autorización

Autorizo la entrega y defensa del trabajo titulado **DISPOSITIVO LoRa DE COMUNICACIÓN A LARGO ALCANCE Y BAJO CONSUMO ENERGÉTICO PARA APLICACIONES DEL ÁMBITO DEL DESARROLLO** de acuerdo con la normativa vigente del Máster en Estrategias y Tecnologías para el Desarrollo Humano.

En Madrid, a 27 de enero de 2017

Firmado digitalmente por
GUTIERREZ MARTIN
ALVARO - 50110009R
Fecha: 2017.01.30 12:35:07
+01'00'

Firma : Álvaro Gutiérrez Martín

AGRADECIMIENTOS

Agradezco a la *Fundación Carolina* por brindarme la valiosa oportunidad de haber sido beneficiario del programa de becas para el master interuniversitario en Estrategias y Tecnologías para el Desarrollo durante la versión 2015-2016, permitiéndome afianzar nuevos conocimientos y mejorar mi perfil profesional. Agradezco a los docentes Álvaro Gutiérrez Martín y Manuel Sierra Castañer por compartir sus valiosos conocimientos técnicos en su calidad de tutores profesional y académico respectivamente, además del respaldo incondicional en el desarrollo de mis prácticas profesionales en la ETSIT-UPM; a los maestros de la Universidad Politécnica de Madrid y Universidad Complutense de Madrid quienes me impartieron sus conocimientos y experiencias, y a todas las personas que de una u otra manera hicieron posible la realización y culminación de esta maestría.

Considero que soy el fruto de las buenas intenciones de las personas.
Muchas gracias.

José Daniel Rodríguez Munca

TABLA DE CONTENIDO

IDENTIFICACIÓN DEL TRABAJO FIN DE MASTER	14
Título del proyecto.....	15
Facultad y programa en los que está inscrito el proyecto.....	15
Grupo y línea de investigación	15
Temática de estudio	15
Tutores	15
Estudiante investigador.....	15
INTRODUCCIÓN	17
1.1 RESUMEN.....	19
1.2 ABSTRACT	21
2. OBJETIVOS	23
2.1 Objetivo General.....	23
2.2 Objetivos específicos	23
3. MÉTODO Y FASES DEL TRABAJO.....	25
3.1 Descripción de los dispositivos utilizados.....	25
3.1.1 Transceiver Semtech LoRa	25
3.1.2 Sistemas embebidos - importancia del hardware libre.....	31
3.2 Descripción del software utilizado - Importancia del software libre.....	39
3.2.1 IDE Arduino	40
3.2.2 IDE Eclipse – Sistema operativo FreeRTOS	41
3.3 Metodología de trabajo	43
3.3.1 Enfoque de la investigación	43
3.3.2 Tipo de investigación	44
3.3.3 Método de investigación	44
3.3.4 Población y muestra	45
3.3.5 Metodología	45
4. RESULTADOS.....	63
4.1 Pruebas de alcance y consumo energético.....	63
4.1.1 Pruebas de alcance para LoRa SX1272.....	64
4.1.2 Pruebas de consumo energético y alcance para LoRa SX1278	67

4.1.3 Pruebas de dispositivos LoRa con diseño de red tipo estrella.....	80
PRINCIPIOS DEL DESARROLLO DIGITAL EN APLICACIONES PROPUESTAS PARA EL ÁMBITO DEL DESARROLLO CON DISPOSITIVOS LoRa.....	81
CONCLUSIONES	85
RECOMENDACIONES	91
BIBLIOGRAFÍA.....	95

LISTADO DE FIGURAS

<i>Figura 1. Disposición de elementos necesarios para la creación del driver con LoRa</i>	25
<i>Figura 2. Diagrama de bloques con principales características y diferencias entre SX1272 y SX1278.</i>	26
<i>Figura 3. Principales características de comunicación SPI para los dispositivos LoRa seleccionados.</i>	29
<i>Figura 4. Placa Arduino UNO.</i>	33
<i>Figura 5. Arquitectura simplificada de Arduino UNO R3.</i>	33
<i>Figura 6. Placa Arduino DUE.</i>	34
<i>Figura 7. Arquitectura simplificada de Arduino DUE.</i>	35
<i>Figura 8. Arquitectura de monitorización para una vivienda autosostenible.</i>	36
<i>Figura 9. Arquitectura a nivel de hardware de los módulos SDIN.</i>	37
<i>Figura 10. Tarjeta principal del módulo SDIN.</i>	38
<i>Figura 11. Tarjeta secundaria del módulo SDIN tipo 0: Termopares.</i>	38
<i>Figura 12. Conexión mecánica de los módulos SDIN entre la tarjeta primaria y secundaria.</i>	38
<i>Figura 13. Módulo SDIN Therm.</i>	39
<i>Figura 14. Interfaz del IDE Arduino.</i>	41
<i>Figura 15. Interfaz del IDE Eclipse.</i>	42
<i>Figura 16. Árbol de decisión dicotómica de los métodos de investigación experimental.</i>	44
<i>Figura 17. Patrón lineal de la investigación cuantitativa.</i>	45
<i>Figura 18. Esquema propuesto para pruebas del driver con los dispositivos LoRa.</i>	49
<i>Figura 19. Esquema de conexión para SX1272 de Semtech y Arduino UNO.</i>	50
<i>Figura 20. Esquema de conexión para LoRa1278 de NiceRF y Arduino UNO.</i>	51
<i>Figura 21. Esquema de conexión para LoRa1278 de NiceRF y Arduino DUE.</i>	52
<i>Figura 22. Esquema de conexión para LoRa1278 de NiceRF y módulo SDIN.</i>	53
<i>Figura 23. Espacio de pruebas Indoor para los dispositivos LoRa.</i>	54
<i>Figura 24. Espacio de pruebas Outdoor para los dispositivos LoRa.</i>	55
<i>Figura 25. Tipología estrella para la prueba del driver generado.</i>	56
<i>Figura 26. Principio de funcionamiento “ping-pong” para la realización de pruebas de alcance en dispositivos LoRa.</i>	63
<i>Figura 29. Máximo alcance en transmisión/recepción de datos para LoRa SX1272 (plano XY).</i>	65
<i>Figura 28. Esquema para determinar el alcance en transmisión/recepción de datos para LoRa SX1272 (plano XY e YZ).</i>	66
<i>Figura 29. Dispositivo esclavo conformado por sistema embebido Arduino Uno y transceiver SX1272.</i>	67
<i>Figura 30. Representación gráfica del consumo energético de los transceivers DRF1278F y LoRa1278 para modo recepción.</i>	70
<i>Figura 31. Representación gráfica del consumo energético de los transceivers DRF1278F y LoRa1278 para modo transmisión.</i>	71

<i>Figura 32. Esquema propuesto para pruebas de alcance Indoor.....</i>	72
<i>Figura 33. Máximo alcance en transmisión/recepción de datos para LoRa SX1278 (plano XY)</i>	73
<i>Figura 34. Esquema para determinar el alcance en transmisión/recepción de datos para LoRa SX1278 (plano XY e YZ)</i>	74
<i>Figura 35. Dispositivo maestro conformado por sistema embebido SDIN y transceiver SX1278 en el laboratorio B-301 de la ETSIT.</i>	75
<i>Figura 36. Dispositivo esclavo conformado por sistema embebido Arduino Uno y transceiver SX1278 en sótano del edificio A de la ETSIT.....</i>	75
<i>Figura 37. Esquema propuesto para pruebas de alcance “Outdoor”.</i>	76
<i>Figura 38. Dispositivo maestro conformado por sistema embebido Arduino Uno, transceiver SX1278 y batería de 5V en la terraza del edificio C de la ETSIT.....</i>	76
<i>Figura 39. Dispositivo esclavo conformado por sistema embebido Arduino Uno, transceiver SX1278 en el parque Dehesa de la Villa.....</i>	77
<i>Figura 40. Máximo alcance en espacio “outdoor” utilizando transceivers SX1278.</i>	78
<i>Figura 41. Perfil de elevación entre la terraza del edificio C y el punto de máximo alcance en el parque Dehesa de la Villa.....</i>	79
<i>Figura 42. Esquema para determinar el alcance en transmisión/recepción de datos para LoRa SX1278 (plano YZ).....</i>	79
<i>Figura 43. Vínculo entre las tecnologías y el desarrollo humano.....</i>	85
<i>Figura 44. IoT: la red de redes.....</i>	86
<i>Figura 45. Comparación del ancho de banda requerido (frecuencia) con respecto al rango de cobertura para los principales estándares inalámbricos.</i>	87
<i>Figura 46. Comparación de los niveles de voltaje entre Arduino Uno y los transceiver LoRa.</i>	91
<i>Figura 47. Esquema de conexión para visualizar el nivel de alimentación con respecto a los modos transmisión y recepción en los dispositivos LoRa.</i>	92
<i>Figura 48. Diagrama de tiempos de la señal de alimentación con respecto a los modos transmisión y recepción en los dispositivos LoRa utilizando Arduino Uno.</i>	92
<i>Figura 49. Diagrama de tiempos de la señal de alimentación con respecto a los modos transmisión y recepción en los dispositivos LoRa utilizando Arduino DUE.</i>	93

LISTADO DE TABLAS

<i>Tabla 1. Relación de frecuencias de trabajo para el SX1272 y SX1278.</i>	27
<i>Tabla 2. Banda de frecuencia ISM compatibles con los transceivers de radio propuestos.</i> ...	27
<i>Tabla 3. Principales características de los dispositivos LoRa seleccionados.</i>	28
<i>Tabla 4. Principales características eléctricas de los dispositivos LoRa seleccionados.</i>	28
<i>Tabla 5. Comparación de pines de control según los fabricantes de tarjetas para los transceivers estudiados.</i>	30
<i>Tabla 6. Descripción del uso de los pines RXEN y TXEN para la tarjeta LoRa1278 de NiceRF.</i>	30
<i>Tabla 7. Fotografías de los transceivers estudiados.</i>	31
<i>Tabla 8. Nomenclatura utilizada para la conexión entre la tarjeta SX1272RF1 de Semtech y la tarjeta Arduino UNO.</i>	50
<i>Tabla 9. Nomenclatura utilizada para la conexión entre la tarjeta LoRa1278 de NiceRF y la tarjeta Arduino UNO.</i>	51
<i>Tabla 10. Nomenclatura utilizada para la conexión entre la tarjeta LoRa1278 de NiceRF y la tarjeta Arduino DUE.</i>	52
<i>Tabla 11. Nomenclatura utilizada para la conexión entre la tarjeta LoRa1278 de NiceRF y el módulo SDIN.</i>	53
<i>Tabla 12. Instrumento propuesto para la recopilación de datos en las pruebas con dispositivos LoRa para diferentes configuraciones.</i>	57
<i>Tabla 13. Combinación de los principales parámetros configurables para los dispositivos LoRa.</i>	58
<i>Tabla 14. Parámetros de configuración por software en los dispositivos SX1272 para obtener máximo alcance</i>	64
<i>Tabla 15. Instrumento con la recopilación de datos para las pruebas con los dispositivos LoRa para diferentes configuraciones en DRF1278F de Dorji.</i>	68
<i>Tabla 16. Instrumento con la recopilación de datos para los dispositivos LoRa a diferentes configuraciones para LoRa1278 de NiceRF.</i>	69
<i>Tabla 17. Parámetros de configuración por software en los dispositivos SX1278 para obtener máximo alcance.</i>	71
<i>Tabla 18. Cuadro comparativo de tecnologías inalámbricas LoRa, ZigBee, Bluetooth y Wi-Fi.</i>	89
<i>Tabla 19. Tarjetas embebidas STMicroelectronics de ultra bajo consumo.</i>	94

ANEXOS

Anexo A. Desarrollo del driver para el uso de los dispositivos de radio LoRa con el uso de sistemas embebidos.....99

Anexo B. Instalación y uso del driver LoRa para sistemas embebidos.....163

Anexo C. DataSheet LoRa SX1278 (digital).

Anexo D. DataSheet LoRa SX1272 (digital).

IDENTIFICACIÓN DEL TRABAJO FIN DE MASTER

Título del proyecto

DISPOSITIVO LoRa DE COMUNICACIÓN A LARGO ALCANCE Y BAJO CONSUMO ENERGÉTICO PARA APLICACIONES DEL ÁMBITO DEL DESARROLLO

Facultad y programa en los que está inscrito el proyecto

Escuela Técnica Superior de Ingenieros de Telecomunicación ETSIT.
Departamento de Tecnología Fotónica y Bioingeniería.
Universidad Politécnica de Madrid.

Programa de Maestría Interuniversitaria en Estrategias y Tecnologías para el Desarrollo.
Universidad Complutense de Madrid.
Universidad de Politécnica de Madrid.

Grupo y línea de investigación

Grupo de investigación: Generación Distribuida Renovable y Control Inteligente (GEDIRCI).
Escuela Técnica Superior de Ingenieros de Telecomunicación ETSIT.

Línea de investigación: Sistemas de control inteligente y bioinspirado.

Temática de estudio

TIC para el Desarrollo.

Tutores

Tutor Académico: Manuel Sierra Castañer
Departamento de Señales, Sistemas y Radiocomunicaciones.
Escuela Técnica Superior de Ingenieros de Telecomunicación ETSIT.

Tutor Profesional: Álvaro Gutiérrez Martín
Departamento de Tecnología Fotónica y Bioingeniería.
Escuela Técnica Superior de Ingenieros de Telecomunicación ETSIT.

Estudiante investigador

José Daniel Rodríguez Munca.
Ingeniero Electrónico. Especialista en Docencia mediada por TIC. Candidato a magíster en Ciencias de la Educación.

INTRODUCCIÓN

Comúnmente se asocia la comunicación con los medios masivos de difusión de información. Este término también involucra el diálogo e intercambio de ideas y conocimientos entre individuos, ya sea de forma verbal o utilizando medios electrónicos que facilitan estas tareas, recortan distancias y brindan precisión a la información. Esto permite la toma asertiva de decisiones en favor de una comunidad, un negocio, familia o individuo.

Según la FAO (1994), la comunicación es decisiva en la promoción del desarrollo humano aumentando en las comunidades la sensibilización, participación, capacidad, conocimientos y tecnologías. Sin embargo, estas últimas suelen estar subutilizadas. Además, “*Los programas de desarrollo sólo podrán dar todos sus frutos si los conocimientos y tecnologías se comparten efectivamente*” (FAO, 1994).

También enfatiza que el principal denominador en cuestiones relacionadas con el desarrollo es el factor humano y que se depende más de la población, del nivel de sensibilización, de la participación y las competencias. El segundo denominador es la comunicación que puede influir en cambios económicos, sociales y culturales ofreciendo mejores posibilidades y oportunidades al desarrollo.

A un nivel práctico, el término *comunicación* con el tiempo fue absorbido y generalizado con lo que hoy se le conoce como TIC o Tecnologías de la Información y las Comunicaciones; encontrando similitudes en su uso y aplicación: “...*las TIC son un instrumento cada vez más poderoso para poder participar en los mercados mundiales; promover la responsabilidad política; mejorar la provisión de servicios básicos, y realizar las oportunidades de desarrollo local. Pero sin políticas innovadoras de TIC, muchas personas de los países en desarrollo – especialmente los pobres – se quedarán atrasados*” (PNUD, 2003).

Con el auge de las Tecnologías de la Información y las Comunicaciones en el siglo XXI, aparecen nuevas tendencias como lo es el Big Data y el Internet de las cosas (IoT o IdC). Internet no solo ha permitido conectar personas, sino dispositivos, sensores y sistemas complejos. Es así como aparece el término “Internet de las cosas” o “internet de los objetos” que corresponde a la interconexión de diversos dispositivos a internet, los cuales generan datos en tiempo real.

Lo importante del “internet de las cosas” no radica en la conexión de las “cosas” a internet, sino en la información y conocimiento que estas puedan aportar para mejorar la calidad de las personas.

Con la posibilidad de obtener información valiosa de diferentes elementos a través de internet, aparecen nuevas técnicas y protocolos de comunicación de bajo consumo; es el caso de *LoRaWAN*, un protocolo de comunicación inalámbrica de bajo consumo y largo alcance para IoT (LoRa Alliance, 2016).

Sin embargo:

“La tecnología se crea en respuesta a las presiones del mercado y no de las necesidades de los pobres, que tienen escaso poder de compra... en la era de las redes, cada país necesita contar con capacidad para comprender las tecnologías mundiales y adaptarlas a las necesidades locales”, PNUD (2001).

Los dispositivos LoRa son un claro ejemplo de lo planteado por el PNUD en 2001; por tal razón se propone adaptarlos a confiables sistemas de monitorización y a su vez a posibles aplicaciones del ámbito del desarrollo.

1.1 RESUMEN

Teniendo en cuenta los avances en innovación tecnológica, se han desarrollado estándares inalámbricos abiertos de bajo consumo energético, alto radio de cobertura y bajo costo como respuesta a las tendencias y necesidades del mercado del *IoT* como lo es el estándar *LoRAWAN* (LoRa Alliance, 2016). Sin embargo, es necesario adaptar su uso para aplicaciones del ámbito del desarrollo debido a que el uso de la tecnología apropiada puede ser la clave en el aseguramiento de los servicios básicos que sustentan redes de infraestructuras como agua potable, comunicaciones, electricidad, saneamiento, entre otros; mejorando la calidad de la vida de las personas y en especial las que se encuentran más aisladas de la población. Sin embargo, esta tecnología debe ser ajustada al contexto, características y población beneficiada: “*En la era de las redes, cada país necesita contar con capacidad para comprender las tecnologías mundiales y adaptarlas a las necesidades locales*” (PNUD, 2001).

Es así que se planteó el desarrollo de un driver para los dispositivos LoRa SX127X junto con una serie de pruebas para determinar el consumo energético y distancia acorde con condiciones reales como *espacios Indoor* (obstáculos como infraestructuras y muros) y *espacio Outdoor* (obstáculos como vegetación) junto con características y particulares de los dispositivos LoRa plasmadas en este documento con ejemplos de sistemas de monitorización con sensores remotos.

Durante el desarrollo de la librería o driver para el uso y control de los dispositivos LoRa propuestos, se utilizó *Arduino UNO/DUE* (tarjeta de adquisición de datos a 8/32 bits, catalogada como hardware libre) y *SDIN* (plataforma de adquisición de datos con microcontrolador ARM de 32 bits desarrollada por el grupo de investigación GEDIRCI) mediante el uso de herramientas de software libre como el IDE *eclipse*. En el proceso fueron generadas pruebas de alcance, consumo energético y se documentó la librería respecto al uso y control de los dispositivos.

De forma transversal, se plantearon posibles aplicaciones de la tecnología explorada en aplicaciones del contexto del desarrollo (agua, saneamiento, energía, salud, entre otros), con relación al Big Data, Internet de las cosas (*IoT*) y los principios del desarrollo digital.

1.2 ABSTRACT

Taking into account the advances in technological innovation, open wireless standards of low energy consumption, high radio coverage and low cost in response to the trends and needs of the market of the IoT have been developed as it is the standard LoRAWAN. However, it is necessary to adapt its use for development applications since the use of appropriate technology can be the key in securing basic services that support infrastructure networks such as potable water, communications, electricity, sanitation, among others; improving the quality of life of people and especially those found most isolated population. However, this technology must be adjusted to the context, characteristics and population benefited: "*In the networks age, every country needs the capacity to understand and adapt global technologies for local needs*" (UNDP, 2001).

The development of a driver for LoRa SX127X devices together with a series of tests to determine the energy consumption and distance according to real conditions like Indoor spaces (obstacles like infrastructures and walls) and Outdoor space (obstacles like vegetation), together with features and particulars of the LoRa devices embodied has been developed with examples of monitoring systems with remote sensors.

During the development of the library or driver for the use and control of the proposed LoRa devices, Arduino UNO/DUE (8/32 bit data acquisition card, cataloged as open hardware) and SDIN (platform for data acquisition with 32-bit ARM microcontroller developed by the GEDIRCI research group) by means of open software tools such as the eclipse IDE. During the process, range and energy consumption tests were developed and the library was documented regarding the use and control of the devices.

In a transversal way, possible applications of the technology in applications of the development context (water, sanitation, energy, health, among others), with respect to Big Data, Internet of Things (IoT) and the digital development were studied.

2. OBJETIVOS

2.1 Objetivo General

Evaluar y documentar las características de los transceiver LoRa como dispositivo de comunicación de largo alcance y bajo consumo energético para aplicaciones del ámbito del desarrollo.

2.2 Objetivos específicos

- Plantear posibles aplicaciones de la tecnología explorada en aplicaciones del contexto del desarrollo (agua, saneamiento, energía, entre otros).
- Evaluar las características de consumo energético y alcance máximo para espacios indoor/outdoor de los dispositivos de radio LoRa SX1272 y SX1278 a partir de diferentes configuraciones de firmware.
- Generar y documentar una librería o driver estándar para los dispositivos de radio LoRa SX1272 y SX1278 con el uso de herramientas de hardware y software libre.
- Evaluar la implementación de sistemas básicos de monitorización con el uso de los dispositivos LoRa SX1272/SX1278 con sistemas embebidos Arduino y módulos SDIN.
- Comparar el estándar inalámbrico abierto LoRaWAN con respecto a tecnologías inalámbricas utilizadas para la transmisión de datos.

3. MÉTODO Y FASES DEL TRABAJO

3.1 Descripción de los dispositivos utilizados

Se propone el uso de dispositivos transceivers con la tecnología LoRa (fabricante *Semtech*) con las versiones SX1272 y SX1278. Un transceiver es un dispositivo que cuenta con circuitos electrónicos capaces de procesar información que podrá ser transmitida o recibida sobre el mismo (Weik, 1996).

Sin embargo, el transceiver requiere de la lógica de programación para ser configurado. Es así como se debe incorporar un sistema embebido. Un sistema embebido se define como un circuito electrónico o hardware diseñado para realizar un conjunto limitado de funciones específicas (Herrera, 2011); en este caso debe controlar los dispositivos LoRa. Como sistemas embebidos se propone el uso de *SDIN* (plataforma desarrollada por GEDIRCI) y *Arduino* con la versión UNO y DUE (hardware libre).

La *Figura 1* presenta la disposición de los elementos necesarios para realizar y probar el driver para los dispositivos LoRa.

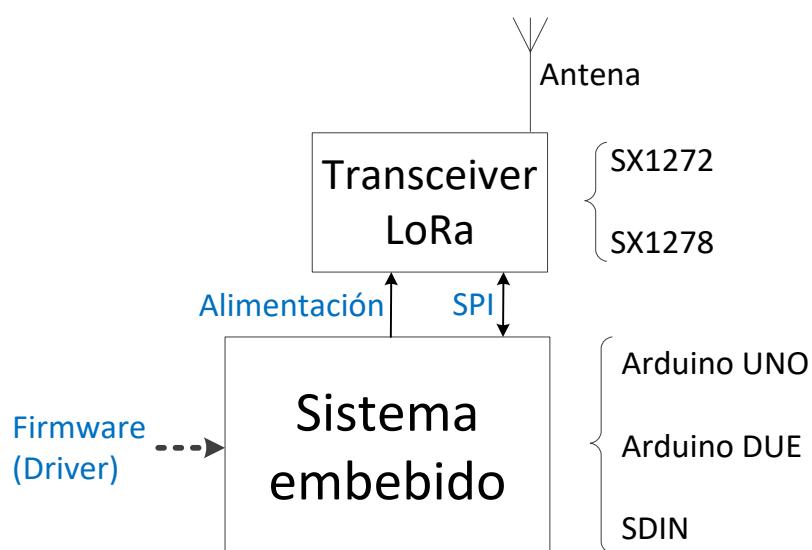


Figura 1. Disposición de elementos necesarios para la creación del driver con LoRa.

Fuente: Esquema desarrollado por el autor

3.1.1 Transceiver Semtech LoRa

Los módulos Semtech LoRa son transceivers RF (serie SX127'x') con la capacidad de comunicarse con la tipología M2M (maquina a máquina). Con estos dispositivos se pueden

crear complejas redes de comunicación de largo alcance y con la posibilidad de conectar millones de dispositivos. Estos dispositivos optimizan el consumo energético, aumentando el ciclo de vida de las baterías de los sistemas embebidos utilizados, brindando aplicaciones con redes ideales para Internet de las Cosas (IoT), automatización, meteorología, seguridad, tracking y aplicaciones M2M (Semtech, s.f.).

Los dispositivos de radio Semtech LoRa cuentan con un modem de largo alcance con alta inmunidad al ruido y mínimo consumo de corriente. La serie SX1272/78 utilizan la técnica de modulación patentada LoRa alcanzando una sensibilidad de -137dBm (SX1272) y -148 dBm (SX1278). Esta alta sensibilidad combinada con amplificador LNA (amplificador de bajo ruido) de +20 dBm optimiza el rango de alcance y brindan la robustez necesaria para aplicaciones industriales (Semtech SX1272, 2015 & Semtech SX1278, 2015).

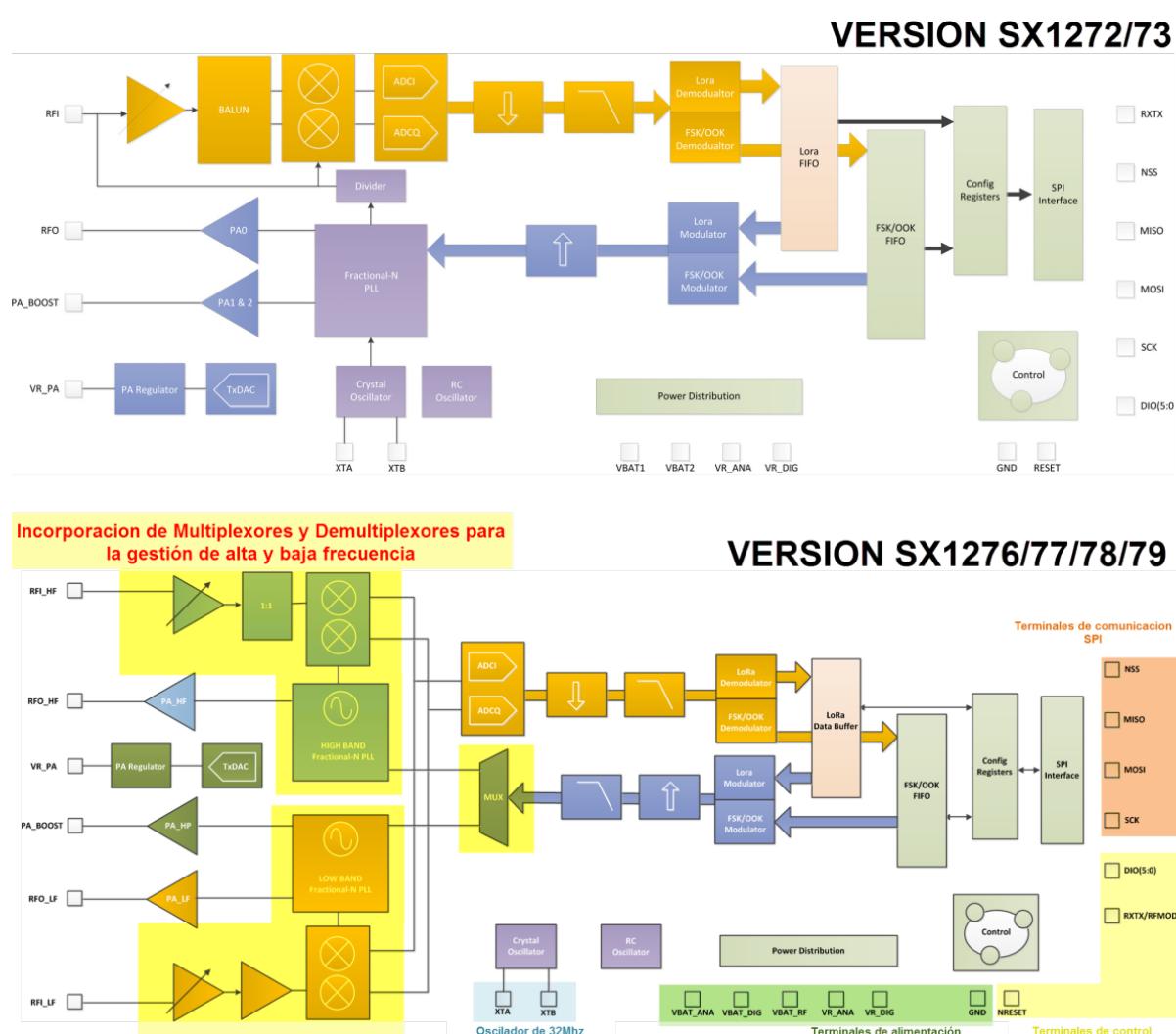


Figura 2. Diagrama de bloques con principales características y diferencias entre SX1272 y SX1278.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Editado por el autor

El consumo de corriente es nulo cuando se encuentra en modo sleep y tiene un mínimo consumo cuando se encuentra en modo recepción llegando hasta 12mA. Tiene como ventaja la incorporación de la Verificación por redundancia cíclica (CRC) de hasta 256 Bytes, característica muy útil para comprobar si los datos fueron recibidos correctamente en el transceiver (Semtech SX1278, 2015).

La versión SX1278 presenta mejoras a nivel de la arquitectura respecto al SX1272. La principal mejora es la incorporación de multiplexores y demultiplexores para el manejo de bajas y altas frecuencias (ver *Figura 2*). Para el chip radio SX1278, las bajas frecuencias (LF) corresponden a frecuencias por debajo de 525 MHz y las altas frecuencias (HF) se encuentran por encima de 779 MHz (Semtech SX1272, 2015 & Semtech SX1278, 2015).

La *Tabla 1* presenta las frecuencias máximas y mínimas de trabajo de los transceivers de radio seleccionados.

Tabla 1. Relación de frecuencias de trabajo para el SX1272 y SX1278.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015)

Dispositivo	Frecuencia mínima (MHz)	Frecuencia máxima (MHz)
SX1272	860	1020
SX1278	137	525

Existen frecuencias de libre uso conocidas como ISM (Industrial, Scientific and Medical) que son frecuencias intencionalmente libres para uso no comercial en aplicaciones de tipo industrial, científico y médico. La *Tabla 2* presenta las principales frecuencias ISM compatibles con los dispositivos LoRa propuestos.

Tabla 2. Banda de frecuencia ISM compatibles con los transceivers de radio propuestos.

Fuente: Texas Instruments (2005)

Banda ISM (MHz)	Frecuencia mínima (MHz)	Frecuencia máxima (MHz)	Ancho de Banda (MHz)	Usos por continente
433	433.050	434.790	1.84	Europa, África y parte del norte de Asia.
869	868	870	2	Europa, África, Asia y Oceanía.
900	902	928	26	Continente americano

De acuerdo a la *Tabla 2*, serán seleccionadas las frecuencias de trabajo a 433Mhz para el chip SX1278 y 869 MHz para el chip radio SX1272 teniendo en cuenta el desarrollo de pruebas en el espacio Europeo.

Algunas características importantes de los dispositivos seleccionados se presentan en la *Tabla 3*.

Tabla 3. Principales características de los dispositivos LoRa seleccionados.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015)

Radio SX	Frecuencia seleccionada MHz	Parámetros importantes del LoRa			
		Factor de alcance SF	Ancho de banda/canal (kHz)	Tasa efectiva de bits (kbps)	Sensibilidad (dBm)
1272	869.525	6 a 12	125 a 500	0.24 a 37.5	-117 a -137
1278	433.800	6 a 12	7.8 a 500	0.018 a 37.5	-111 a -148

La sensibilidad del receptor identifica el valor mínimo de potencia que necesita un receptor para poder decodificar o extraer datos para alcanzar una determinada tasa de bits o de velocidad (Ramírez, 2015). Comúnmente se expresan en dBm e indican que entre menor sea su valor, el receptor tendrá mejor sensibilidad; por lo que se concluye que el dispositivo SX1278 tiene como ventaja el tener mejor sensibilidad al recibir datos.

Otra importante diferencia a favor del dispositivo SX1278, es que cuenta con canales más reducidos de hasta 7.8kHz como valor mínimo. Sin embargo, lo anterior implica transmisión y recepción de datos con tasas efectivas de datos significativamente bajas que podrían dar lugar a pérdidas de datos o tiempos excesivos de transmisión o recepción de datos.

Tabla 4. Principales características eléctricas de los dispositivos LoRa seleccionados.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015)

Voltaje (V)			Rango de temperatura (°C)			Corriente RX (mA)	Corriente TX (mA) CONFIGURABLE		
Min	Est	Max	Min	Est	Max		Min	Est	Max
1.8	3.3	3.9	-40	25	+85	<12	10	100	240

Min: Mínimo

Max: máximo

Est: Estándar

Existen parámetros eléctricos configurables como la máxima corriente cuando el dispositivo transmite (ver la *Tabla 4*). Estos parámetros se configuran con registros internos del transceiver mediante el protocolo de comunicación síncrono SPI (Semtech SX1278, 2015).

El SPI es un protocolo de comunicación serial síncrono *full duplex*, es decir permite transmitir y recibir información al mismo tiempo. Para usar este protocolo se requiere configurar un dispositivo como maestro y los demás como esclavos (López, s.f.).

Este protocolo utiliza 4 líneas de comunicación que se describen como:

SCLK: corresponde a la señal de reloj, es generada por el maestro y sincroniza la transferencia de datos.

MOSI (Master Out Slave In): transfiere los datos del maestro hacia el esclavo.

MISO (Master In Slave Out): transfiere los datos del esclavo hacia el maestro.

NSS (Slave Select): esta línea es generada por el maestro y permite seleccionar el dispositivo esclavo para iniciar la comunicación. Con esta línea es posible hacer uso de varios dispositivos esclavos utilizando las mismas líneas de comunicación.

Los dispositivos de radio LoRa se comportan como esclavos y el microcontrolador del sistema embebido será el maestro en la comunicación SPI.

A nivel lógico de sincronización y transmisión de datos, la comunicación SPI requiere la configuración de la polaridad del reloj (CPOL) y el bit fase del reloj (CPHA).

Los dispositivos de radio LoRa SX1272/78 utilizan CPOL=0 y CPHA=0; lo que se conoce como modo 0 SPI. El bit más significativo (MSB) de un byte enviado debe ser el primero y la velocidad del SCLK no debe superar los 10MHz (Semtech SX1278, 2015).

El esquema propuesto de conexión entre los transceiver de radio LoRa y el microcontrolador del sistema embebido se presenta en la *Figura 3*.

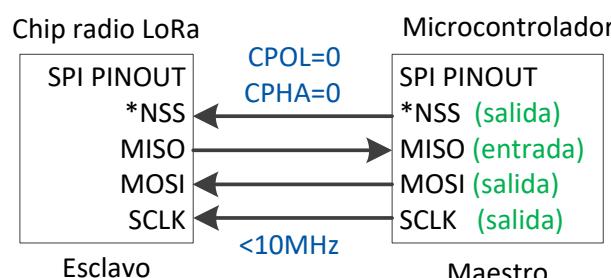


Figura 3. Principales características de comunicación SPI para los dispositivos LoRa seleccionados.

Fuente: Semtech (2013). Esquema adaptado por el autor

Algunos fabricantes realizan tarjetas de adecuación para facilitar la manipulación de los transceivers de radio LoRa. Al realizar estas tarjetas, se eliminan o incorporan pines físicos de control respecto a las características originales del chip. Los fabricantes mantienen los pines de comunicación SPI y eliminan algunos pines de control del chip radio. Las principales características de las tarjetas LoRa adquiridas para realizar las pruebas de funcionamiento se presentan en la *Tabla 5*.

Tabla 5. Comparación de pines de control según los fabricantes de tarjetas para los transceivers estudiados.

Fuente: Semtech (2013), Dorji (2015) & NiceRF (2015)

Tarjeta	Chip radio	Marca	Pines de control físicos			
			DIO0 – DIO5	RXTX	Pines SPI	RESET
SX1272RF1	SX1272	Semtech	Pines de entrada/salida, configurables por software.	Pin de salida. Se coloca en "1" cuando el dispositivo está trasmitiendo datos.		Pin de entrada. Utilizado al iniciar o arrancar el dispositivo.
DRF1278F	SX1278	Dorji	Pines de entrada/salida, configurables por software NI* DIO3–DIO5	NI*	NSS MISO MOSI SCLK	Utilizado al iniciar o arrancar el dispositivo.
LoRa1278		NiceRF				

*NI: No implementado

La versión LoRa1278 de *NiceRF* realiza adecuaciones relacionadas con el hardware incorporando 2 pines adicionales de control que permiten habilitar el uso de la antena para transmitir o recibir. La descripción de las funciones de los pines TXEN y RXEN se presentan en la *Tabla 6*.

Tabla 6. Descripción del uso de los pines RXEN y TXEN para la tarjeta LoRa1278 de NiceRF.

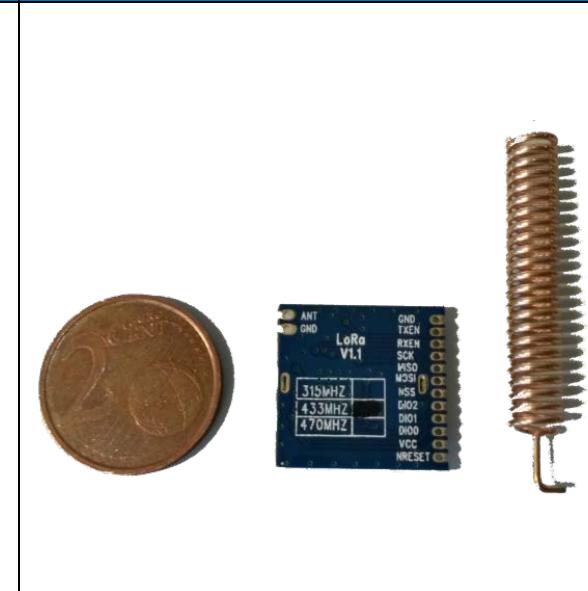
Fuente: NiceRF (2015)

Estados lógicos de pines. Pines de entrada para LoRa.		Descripción de funcionamiento para la Antena
Pin TXEN	Pin RXEN	
0	0	Modo Sleep o bajo consumo
0	1	Configurada para Recibir
1	0	Configurada para Transmitir

Físicamente, la tarjeta SX1272RF1 de Semtech es de mayor tamaño y cuenta con un conector SMA para antenas a 868Mhz e impedancia de 50Ω . Las tarjetas DRF1278F y LoRa1278 tienen el mismo tamaño (significativamente más pequeño que la versión SX1272RF1) y permiten conectar antenas pasivas tipo primavera a 433Mhz con impedancia de 50Ω . Los tamaños se pueden observar en la *Tabla 7*.

Tabla 7. Fotografías de los transceivers estudiados.

Fuente: Fotografías tomadas por el autor

Tarjeta SX1272RF1 de Semtech (868Mhz)	Tarjeta LoRa1278 de NiceRF (433Mhz)
	

3.1.2 Sistemas embebidos - importancia del hardware libre

Un sistema embebido corresponde a un conjunto de componentes electrónicos diseñado para funciones específicas (Pedre, 2012). Teniendo en cuenta el estudio de los dispositivos LoRa para aplicaciones del ámbito del desarrollo, los sistemas embebidos deben ser libremente replicados, por lo que se contempla el uso de *hardware libre*.

“El hardware libre es aquel hardware cuyo diseño se hace disponible públicamente para que cualquier persona lo pueda estudiar, modificar, distribuir, materializar y vender, tanto el original como otros objetos basados en ese diseño. Las fuentes del hardware (entendidas como los ficheros fuente) habrán de estar disponibles en un formato apropiado para poder realizar modificaciones sobre ellas. Idealmente, el hardware de fuentes abiertas utiliza componentes y materiales de alta disponibilidad, procesos estandarizados, infraestructuras abiertas, contenidos sin restricciones, y herramientas de fuentes abiertas de cara a maximizar la habilidad de los individuos para materializar y usar el hardware. El hardware de fuentes abiertas da libertad de controlar la tecnología y al mismo tiempo compartir conocimientos y estimular la comercialización por medio del intercambio abierto de diseños” (Lazalde, Torres & Vila-Viñas, 2015).

En términos prácticos, hardware libre es una tarjeta que incorpora un microcontrolador programable conectado a circuitos discretos o integrados y a su vez a pines o terminales con

los que usuario puede cómodamente conectar sensores, actuadores o dispositivos electrónicos para desarrollar aplicaciones específicas. Es así como se propone utilizar la tarjeta Arduino, que cumple con los parámetros de hardware libre, se encuentra ampliamente documentada en la web y cuenta con un IDE propio para el desarrollo del firmware o aplicación lógica de control.

Utilizar Arduino para el desarrollo de aplicaciones específicas con relación al ámbito del desarrollo tiene las siguientes ventajas (Torrente, 2013):

- *Comunidad de desarrolladores*: muchos usuarios lo utilizan, enriquecen la documentación y comparten sus ideas y conocimientos.
- *Entorno de programación multiplataforma*: es posible instalar y ejecutar en sistemas operativos Windows, Mac OS X y Linux.
- *Sencillo entorno y lenguaje de programación*: resulta fácil de aprender y de utilizar para principiantes. Además, las librerías se encuentran documentadas con ejemplos detallados y gran cantidad de proyectos publicados en diferentes formatos.
- *Placas Arduino económicas*: la placa Arduino estándar (llamada Arduino UNO) ya preensamblada y lista para funcionar cuesta alrededor de 20 euros. También se puede construir la placa adquiriendo los componentes por separado, con lo que el precio total de la placa resultante sería incluso menor.
- *Reutilizables y versátiles*: reutilizables porque se puede aprovechar la misma placa para varios proyectos (es sencillo de desconectar reconectar y reprogramarla), y versátiles porque las placas proveen varios tipos de entradas y salidas de datos, los cuales permiten capturar información de sensores y enviar señales a actuadores de múltiples formas.

Sin embargo, existen diversas versiones de la tarjetas Arduino entre las que se propone utilizar *Arduino UNO* que dispone de un procesador de 8 bits con un nivel bajo de complejidad y *Arduino DUE* de 32 bits con un nivel intermedio de complejidad. También se propone el uso de la tarjeta *SDIN* desarrollada por GEDIRCI que cuenta con 32 bits de procesador y un nivel alto de complejidad.

3.1.2.1 Arduino UNO

Arduino cuenta con un modelo estándar de tarjeta denominada Arduino UNO siendo la más utilizada de las diferentes versiones. Apareció en 2010 y ha contado con tres revisiones, por lo que el modelo actual se suele llamar UNO Rev3 o simplemente UNO R3. Cuenta con un microcontrolador de 8 bits con referencia ATmega328P de la marca Atmel que funciona con un voltaje nominal de 5V y a una frecuencia de 16 MHz. Tiene 14 entradas/salidas digitales (6 de los cuales pueden ser usadas como salidas PWM), 6 entradas analógicas, comunicación SPI, comunicación I²C y comunicación UART (Torrente, 2013).

La tarjeta Arduino Uno funciona con voltajes de 7 a 12 voltios por alimentación externa (conector Jack macho) o por USB (voltaje suministrado por el ordenador). Para programar la lógica de programación o firmware, Arduino Uno utiliza un segundo microcontrolador que le permite ser configurado por USB y realizar una depuración básica del código.

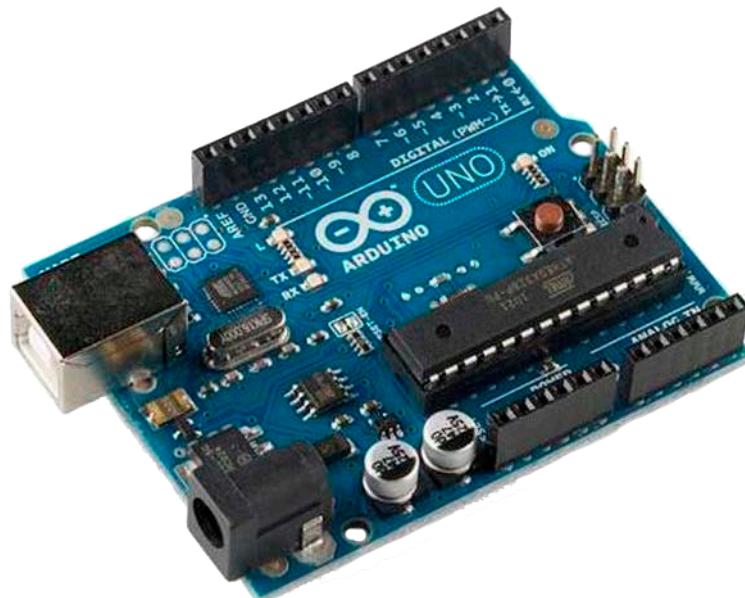


Figura 4. Placa Arduino UNO.
Fuente: Arduino Board UNO (s.f.)

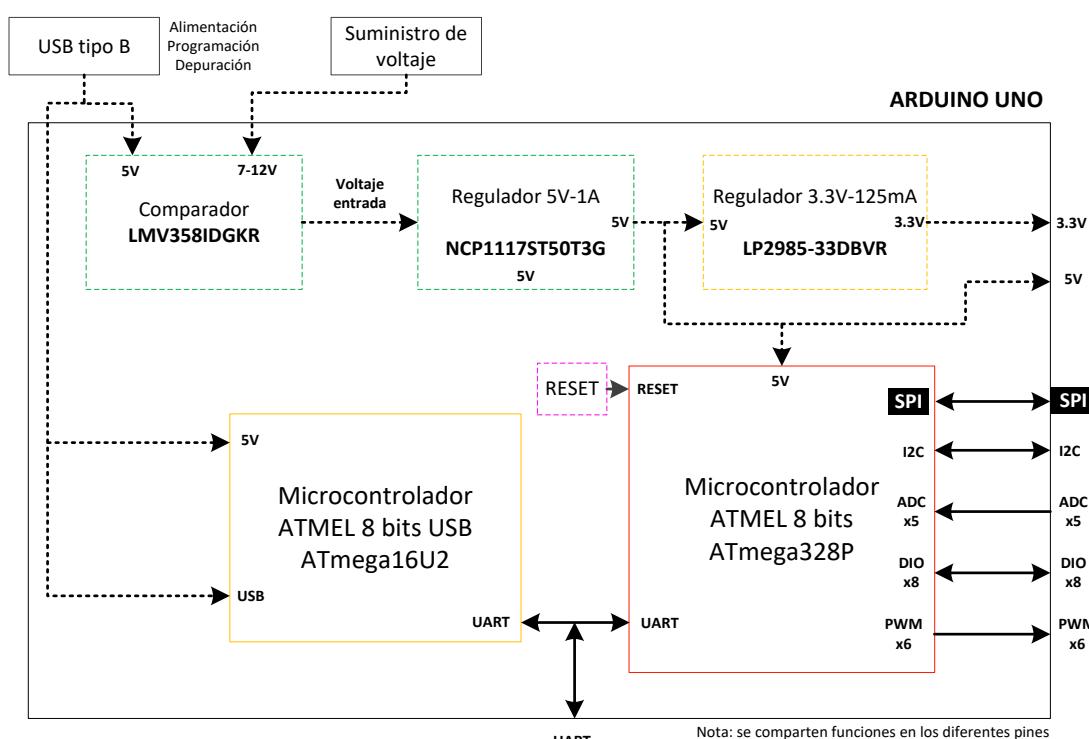


Figura 5. Arquitectura simplificada de Arduino UNO R3.
Fuente: Arduino Board UNO (s.f.). Esquema desarrollado por el autor

La *Figura 4* presenta una fotografía de la placa Arduino UNO y la *Figura 5* presenta el esquema de arquitectura aproximada utilizada por la placa en donde se puede muestra la incorporación del protocolo SPI necesario para los dispositivos LoRa.

3.1.2.2 Arduino DUE

Arduino DUE pertenece a una familia de microcontroladores de 32 bits siendo la primera tarjeta basada en ARM Cortex-M3. Dispone de un microcontrolador SAM3X8E fabricado también por Atmel que funciona con un voltaje nominal de 3.3V, a una frecuencia de 84 MHz. Dispone de 54 pines de entrada/salida digital (12 de los cuales pueden ser usados como salidas PWM), 12 entradas analógicas, 4 módulos UART, 2 conversores digitales-analógicos, 2 puertos I2C independientes, 1 puerto SPI, 1 conector USB de tipo mini-B, y un conector USB de tipo mini-A.

Arduino DUE funciona con voltajes de 7 a 12 voltios por alimentación externa (conector Jack macho) o por USB (voltaje suministrado por el ordenador). Para programar la lógica de programación o firmware, Arduino utiliza un segundo microcontrolador que le permite ser configurado por USB y realizar una depuración básica del código. Sin embargo, existe un segundo puerto USB que puede ser utilizado para propósitos relacionados con la aplicación.

La *Figura 6* presenta una fotografía de la placa Arduino DUE y la *Figura 7* presenta el esquema de la arquitectura aproximada utilizada por la placa en donde se muestra la incorporación del protocolo SPI necesario para el funcionamiento de los dispositivos LoRa.

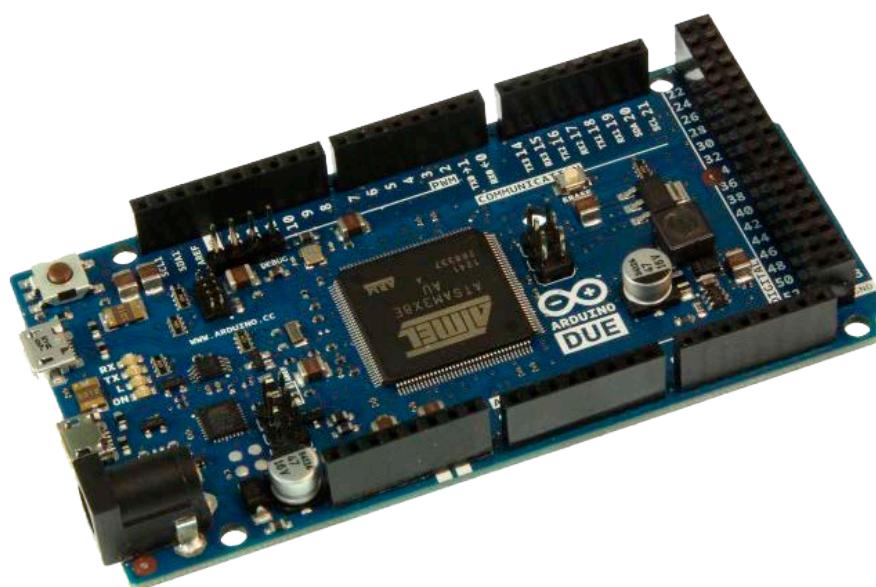


Figura 6. Placa Arduino DUE.
Fuente: Arduino Board DUE (s.f.)

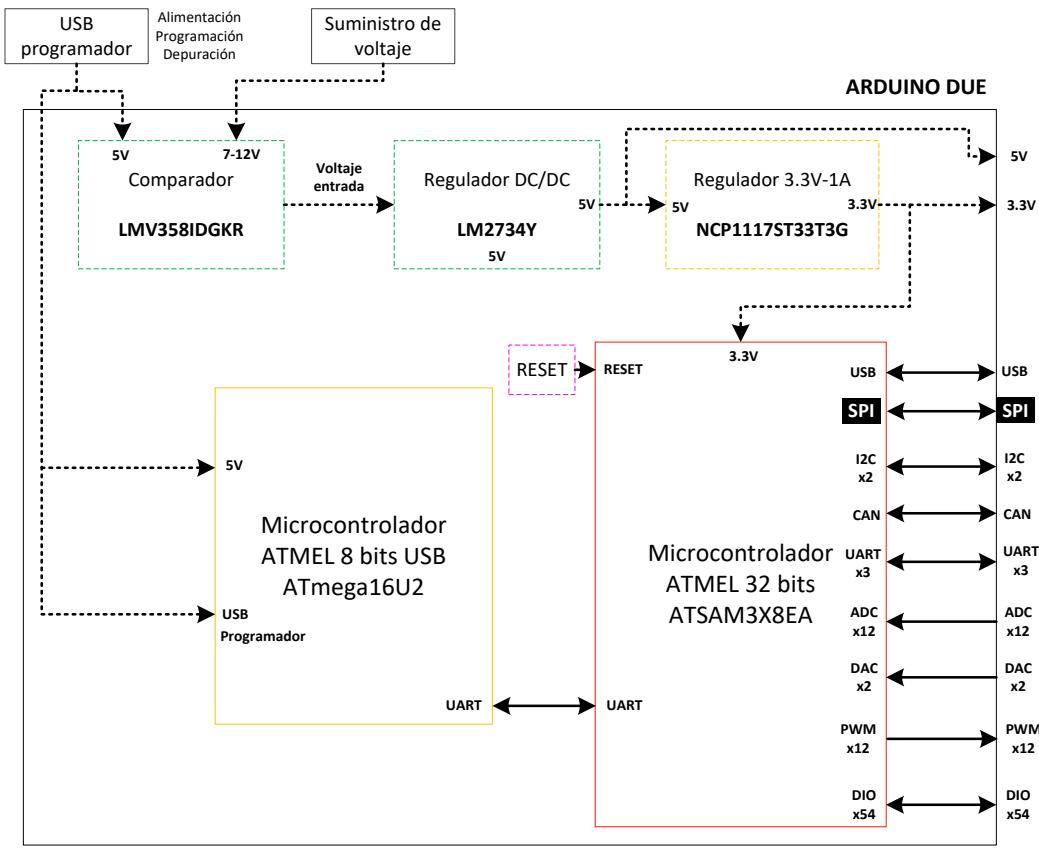


Figura 7. Arquitectura simplificada de Arduino DUE.

Fuente: Arduino Board DUE (s.f.). Esquema desarrollado por el autor

3.1.2.3 SDIN

Los módulos SDIN desarrollados por GEDIRCI, son tarjetas de adquisición de datos entre los que se encuentran, termopares, RTD, sensores analógicos (4-20mA y 0-10V) y módulos de alimentación. Estos módulos han sido ampliamente utilizados en diversos proyectos de monitorización brindando la robustez y eficiencia de tipo industrial.

Algunas de estas aplicaciones se encuentran en el desarrollo de la casa solar “*MagicBox*” presentada en la competición de viviendas autosuficientes “*Solar Decathlon*” 2005, utilizado en el Campus de Montegancedo de la Universidad Politécnica de Madrid y recientemente en las instalaciones del *Centro de Innovación para el Desarrollo de la Universidad Politécnica de Madrid itdUPM*. Sin embargo, en las ediciones del “*Solar Decathlon*” del 2014, los módulos SDIN fueron utilizados de forma masiva como dispositivo de monitorización para las viviendas autosuficientes de la competición.

El tipo de comunicación utilizada entre los módulos SDIN es RS-485 que es un protocolo de comunicación industrial serial, utiliza dos hilos y permite múltiples conexiones sobre el

mismo bus o canal. Algunas de las principales ventajas de este tipo de comunicación son las siguientes (López, s.f.):

- a) *Bajo costo*: los circuitos integrados para transmitir y recibir son económicos y solo requieren una fuente de alimentación +5V para poder generar una diferencia mínima de 1.5v entre las salidas diferenciales.
- b) *Capacidad de interconexión*: RS-485 es una interface multi-enlace con la capacidad de lograr contar con múltiples transmisores y receptores. Con una alta impedancia receptora, los enlaces con RS-485 pueden llegar a tener a lo máximo hasta 256 nodos sobre el mismo bus.
- c) *Longitud de enlace*: En un enlace RS-485 puede tener hasta 1200 metros.
- d) *Rapidez*: La razón de bits puede ser tan alta como 10 Mega bits/ segundo.

El circuito integrado utilizado para la comunicación en la placa de los módulos SDIN corresponde al MAX13487E que lo hace *half duplex*, permite contar con velocidades de transmisión de hasta 16Mbps y una conexión de hasta 128 transceivers sobre el bus (Maxim Integrated Products, 2015).

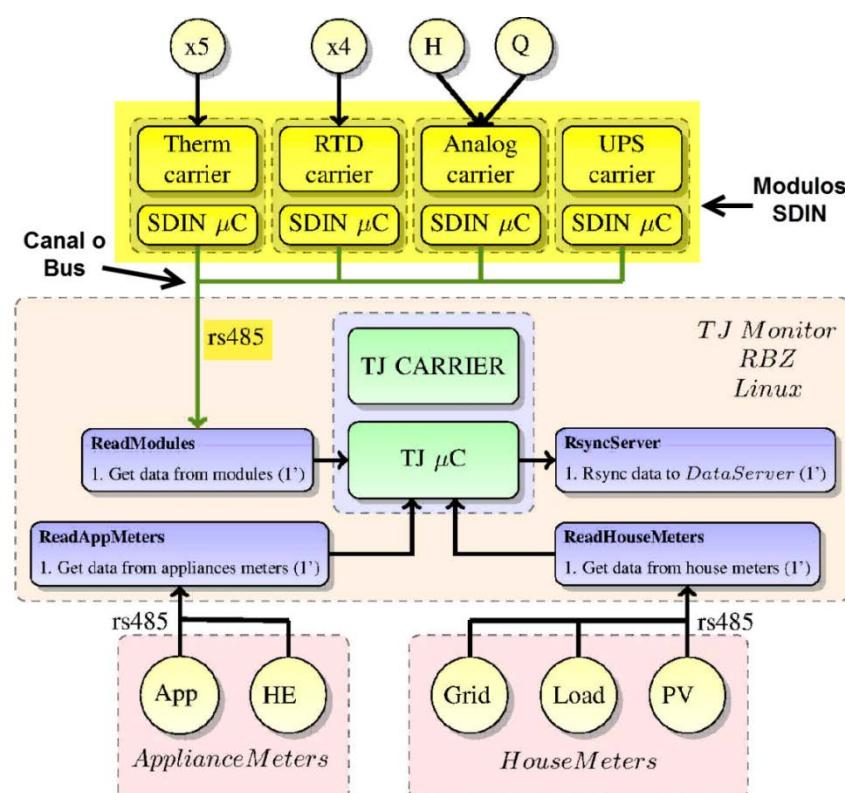


Figura 8. Arquitectura de monitorización para una vivienda autosostenible.

Fuente: Solar Decathlon Europe (2014). Editado por el autor

El esquema de la *Figura 8* presenta la arquitectura de conexión de los sistemas de monitorización para las viviendas sostenibles de la Decathlon Europe en Francia para el 2014. En este esquema se observa el TJ Monitor (equivale al dispositivo maestro del sistema), que

se encarga de recibir la información de medidores de potencia y de los módulos SDIN. Además, el TJ Monitor utiliza LINUX, por lo que el sistema está desarrollado con software libre y de código abierto.

También se presentan los módulos SDIN conectados a un mismo bus RS-485 sobre el cual se transmiten los parámetros de termopares, RTD, dispositivos analógicos, dispositivos digitales y UPS (ver *Figura 9*).

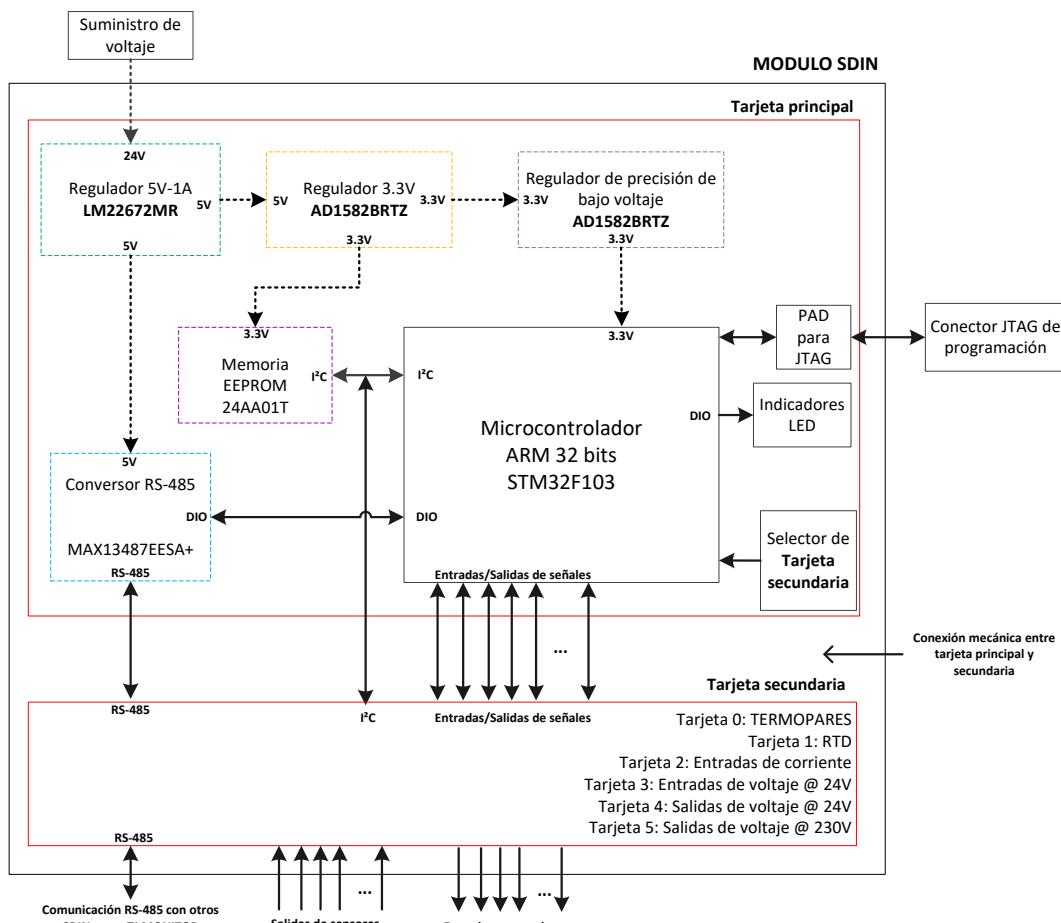


Figura 9. Arquitectura a nivel de hardware de los módulos SDIN.
Fuente: Planos compartidos por GEDIRCI. Esquema desarrollado por el autor

A nivel de hardware, los módulos SDIN están compuestos por dos tarjetas de conexión modular, una tarjeta primaria donde se encuentra el microcontrolador ARM de 32 bits (ver *Figura 10*) y la posibilidad de conexión de una tarjeta secundaria acorde a un selector que determina la función y conexión de la tarjeta secundaria.

Las tarjetas secundarias pueden ser de tipo 0: termopares, 1: RTD, 2: entradas de corriente, 3: entradas de voltaje @ 24V, 4: salidas de voltaje @ 24V y 5: salidas de voltaje @ 230V (ver *Figura 11*). La conexión de estas tarjetas se realiza de forma mecánica donde el usuario ajusta la tarjeta secundaria debajo de la tarjeta principal (ver *Figura 12*).



Figura 10. Tarjeta principal del módulo SDIN.

Fuente: Fotografía tomada por el autor

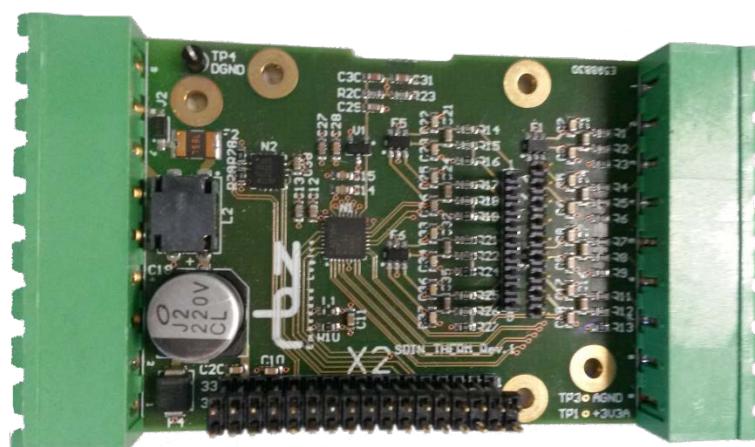


Figura 11. Tarjeta secundaria del módulo SDIN tipo 0: Termopares.

Fuente: Fotografía tomada por el autor

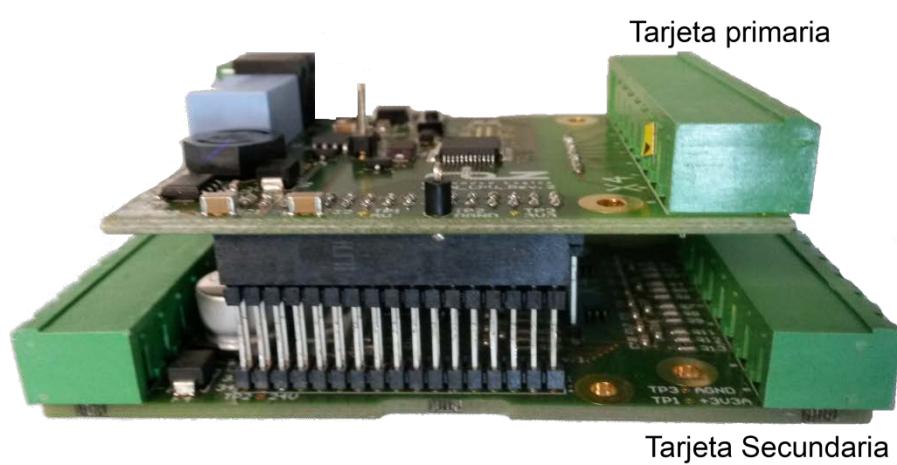


Figura 12. Conexión mecánica de los módulos SDIN entre la tarjeta primaria y secundaria.

Fuente: Fotografía tomada por el autor

En la tarjeta principal del SDIN se encuentra el microcontrolador STM32F103 que dispone del protocolo de comunicaciones SPI para dos módulos. La placa SDIN funciona a 24V. La *Figura 13* presenta una fotografía de los módulos SDIN.



Figura 13. Módulo SDIN.

Fuente: Solar Decathlon Europe (2014)

3.2 Descripción del software utilizado - Importancia del software libre

El *software libre* cuenta con un papel relevante en el ámbito de la cooperación y el desarrollo por cuanto asegura la soberanía tecnológica, impulsa la innovación nacional, optimiza los gastos fortaleciendo el desarrollo local y facilita la inclusión digital. Algunas características del software libre en relación con el ámbito del desarrollo son las siguientes (Administración pública Ecuador, s.f.):

Autonomía tecnológica: Adoptando el uso de software libre y con las posibilidades que éste ofrece de acceder al código fuente, muchos usuarios pasarán de ser consumidores a ser desarrolladores de software. Esto significa que se podrán adaptar el código o programas a necesidades específicas, y todas las modificaciones deberán realizarse siguiendo los requisitos exigidos por el modelo del software libre (un ejemplo claro se encuentra con el uso del S.O. FreeRTOS).

Estandarización e integración: El software libre es producido utilizando especificaciones y estándares tecnológicos libres y públicos, también denominados “estándares abiertos”. Esto beneficia la integración de sistemas y el intercambio de información, de forma que se garantiza la accesibilidad sin restricciones para los usuarios.

Seguridad: El hacer públicos los códigos de los programas o aplicaciones favorece la seguridad de los mismos. Utilizando software libre es posible conocer y analizar lo que está siendo ejecutado realmente por el programa, el tipo de información maneja y como es manipulada la información. Una buena seguridad debe basarse en la transparencia. El software privativo oculta estos aspectos y muchas veces no se conoce si la información se envía a servicios remotos.

Independencia de proveedores: Adquiriendo un software privativo se genera una relación de dependencia con respecto al fabricante. Una vez que se instala dicho software, se dependerá del fabricante para obtener actualizaciones. En muchos casos, el fabricante obligará a actualizar a nuevas versiones aunque no se deseé.

Democratización de la información: Las tecnologías de la información han pasado a ocupar un lugar central en la sociedad. Si bien cada vez son más los usuarios que acceden a dichas tecnologías, la “brecha tecnológica” todavía es grande y es un factor más de exclusión social. El software libre favorece la democratización de la información permitiendo la utilización de protocolos, formatos y lenguajes.

Economía: se estima que la compra de un sistema operativo más un paquete de suite de oficina, ambos con una licencia comercial, cuestan entre 300 y 600 dólares por cada computadora, y ese gasto debe renovarse cada dos o tres años debido a la dependencia hacia el fabricante. Los países en vías de desarrollo con las carencias de recursos que cuentan, pueden ahorrar una gran cantidad de recursos económicos.

El sistema operativo planteado para ser usado en el desarrollo de las pruebas de los dispositivos LoRa fue *Linux* con el uso de la versión *Ubuntu*, la cual es conocida por ser bastante intuitiva, contar con un aspecto similar a *Windows* y ser gratuita. Sobre *Linux*, se utilizaron IDE (Entorno de Desarrollo Integrado), que es un conjunto de herramientas de software para crear, depurar y compilar el código fuente.

Se utilizó el *IDE Arduino* para programar las versiones UNO y DUE de las placas Arduino. También se utilizó sobre *Linux* el *IDE eclipse* para programar el driver compatible con FreeRTOS, sistema operativo utilizado por los módulos SDIN.

3.2.1 *IDE Arduino*

Arduino cuenta con su propio IDE capaz de crear, editar, depurar y transferir el código (también llamado “*sketch*”) a la tarjeta Arduino. El IDE puede ser utilizado por cualquier versión de Arduino y es compatible con los sistemas operativos Windows, Mac OS X, y Linux. El entorno de desarrollo está escrito en JAVA, por lo que previamente se debe instalar la última versión de la JDK (Java Development Kit).

Gran cantidad de desarrolladores comparten “sketch” y librerías para las placas Arduino en *GitHub*, la cual es una plataforma para alojar proyectos utilizando un sistema de control de versiones Git, almacenados de forma gratuita y compartidos al público (*GitHub*, s.f.).

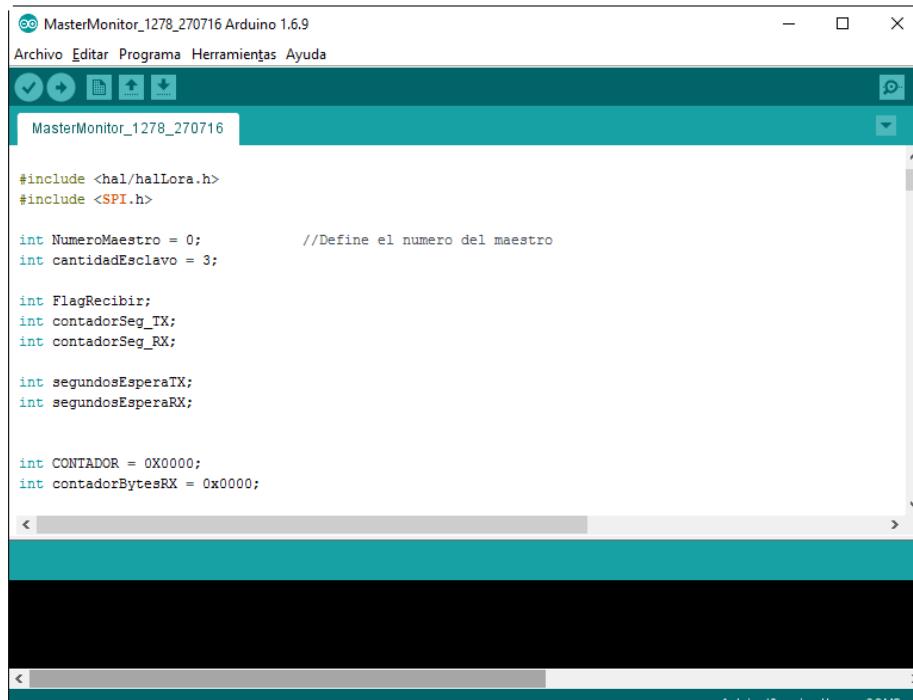


Figura 14. Interfaz del IDE Arduino.

Fuente: Captura de pantalla realizada por el autor

El IDE Arduino utiliza el lenguaje de programación Arduino que en el sentido estricto de la palabra *no* es un lenguaje, sino un conjunto de instrucciones del lenguaje C y C++ “camufladas”. Estas instrucciones están diseñadas para simplificar el desarrollo de programas para microcontroladores AVR. Es decir, al escribir un sketch en “lenguaje Arduino”, realmente se está programando en una versión simplificada del lenguaje C/C++ ocultando gran cantidad de detalles técnicos para el desarrollador, lo que facilita el uso de la plataforma Arduino en conjunto hardware-software.

Sin embargo, al reducir el nivel de complejidad en la programación, se está utilizando un subconjunto de toda la funcionalidad que puede llegar a ofrecer el lenguaje C/C++ (Torrente, 2013).

3.2.2 *IDE Eclipse – Sistema operativo FreeRTOS*

Eclipse, es una plataforma IDE (Entorno de Desarrollo Integrado) de software de código abierto para el desarrollo de aplicaciones, diseñada para ser extendida (escalable) de forma indefinida a través de *plugins* o complementos. Está compuesta por un conjunto de

herramientas de programación de código abierto multiplataforma para desarrollar aplicaciones sin la exclusividad de un lenguaje específico, sino que es un IDE genérico. Sin embargo, es popular entre las comunidades de desarrolladores para el desarrollo de aplicaciones JAVA.

Entre las principales características del IDE Eclipse se encuentran (Genbetadev, 2014):

Perspectivas, editores y vistas: en Eclipse el concepto de trabajo está basado en las perspectivas, que no es otra cosa que una pre-configuración de ventanas y editores, relacionadas entre sí, y que permiten trabajar en un determinado entorno de forma óptima.

Gestión de proyectos: El desarrollo sobre Eclipse se basa en los proyectos, que son el conjunto de recursos relacionados entre sí, como pueden ser el código fuente, librerías, documentación, ficheros configuración, árbol de directorios, archivos multimedia, entre otros. El IDE proporciona asistentes y ayudas para la creación de proyectos. Sin embargo, en la web se encuentran la documentación completa con tutoriales para principiantes.

Depurador de código: Incluye un potente depurador, de uso fácil e intuitivo, y que visualmente ayuda a optimizar el código (utilizando un botón de la interfaz). De nuevo, se tiene una perspectiva específica para la depuración de código, la perspectiva depuración, donde se muestra de forma ordenada toda la información necesaria para realizar dicha tarea.

Extensa colección de plugins (complementos): se encuentran disponibles en una gran cantidad de complementos, algunos publicados por Eclipse y otros compartidos por terceros. Algunos complementos son gratuitos, de pago, bajo distintas licencias, pero casi para cualquier necesidad existe un plugin adecuado.

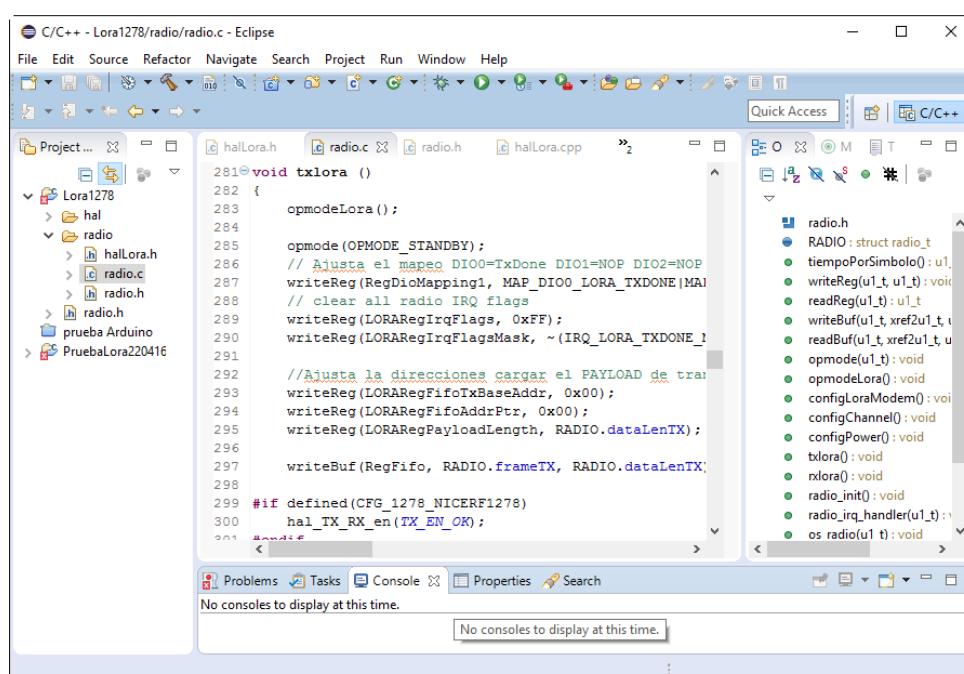


Figura 15. Interfaz del IDE Eclipse.

Fuente: Captura de pantalla realizada por el autor

Sobre Linux, se instaló el IDE Eclipse con la versión MARS 2. Se realizó la programación y depuración de las librerías de los dispositivos LoRa utilizando el lenguaje de programación C y C++ para adaptar el driver al S.O. *FreeRTOS*, utilizado por el módulo SDIN.

Los módulos SDIN utilizan como software *FreeRTOS*, que es un sistema operativo en tiempo real y de código abierto capaz de funcionar sobre un microcontrolador. Utilizar FreeRTOS como software de programación sobre los SDIN cuenta con las siguientes ventajas (Barry, 2009):

- FreeRTOS se encuentra ampliamente documentado.
- Cuenta con gran control en el tiempo de ejecución de cada tarea que se desarrolla.
- Tiene un “scheduler” (comúnmente utilizado en los sistemas operativos) diseñado de tal forma que permita ejecutar tareas en tiempos predecibles (normalmente descrito como determinístico). Esto resulta de gran utilidad en sistemas embebidos donde se busca que el sistema responda en tiempos estrictos.
- Permite establecer prioridades a las tareas a ejecutar.
- Es lo suficientemente ligero para funcionar sobre un microcontrolador.
- Es software libre y de código abierto.
- Permite aplicaciones escalables en el tiempo.
- Utiliza un estándar para las definiciones de tareas, lo que promueve la independencia de la aplicación al desarrollador del software.

3.3 Metodología de trabajo

3.3.1 Enfoque de la investigación

Es seleccionado el *enfoque cuantitativo de investigación*, por cuanto las medidas y la cuantificación de los datos constituyen el procedimiento empleado para alcanzar la objetividad en el proceso investigativo. Este enfoque promueve la búsqueda de la objetividad y la cuantificación orientadas a establecer tendencias a partir del estudio de las características de las pruebas realizadas (Monje, 2011).

El enfoque investigativo cuantitativo plantea la unidad de la ciencia, es decir, la utilización de una metodología única que es la misma utilizada por las ciencias exactas y naturales (Bonilla & Rodríguez, 1997). De acuerdo a lo anterior, los resultados se obtienen exclusivamente de la observación directa, de la comprobación y de la experimentación. Los conocimientos obtenidos se fundamentan en el análisis de hechos reales, sobre los cuales se realiza una validación neutra, objetiva y completa de los resultados.

3.3.2 Tipo de investigación

Teniendo en cuenta la aplicación de los dispositivos LoRa para el ámbito del desarrollo se propone utilizar el tipo de *investigación aplicada o tecnológica*, por cuanto se busca utilizar los conocimientos obtenidos de la investigación en aspectos prácticos y aplicados (servicios básicos), y con ello *traer beneficios a la sociedad o al sector productivo*. Además la *investigación aplicada* estructura procedimientos, innova estrategias, crea y prueba artefactos, y estima su valor pragmático (Tam, Vera & Oliveros, 2008).

3.3.3 Método de investigación

De acuerdo a las variables de la investigación (configuración de los dispositivos LoRa), se plantea el uso del método de investigación experimental. Básicamente se desarrolla mediante la manipulación de una o varias variables experimentales (configuración de los dispositivos LoRa) manipuladas por el investigador en condiciones rigurosamente controladas, con el fin de describir la evidencia causa-efecto produciendo una situación o acontecimiento (alcance máximo y consumo energético).

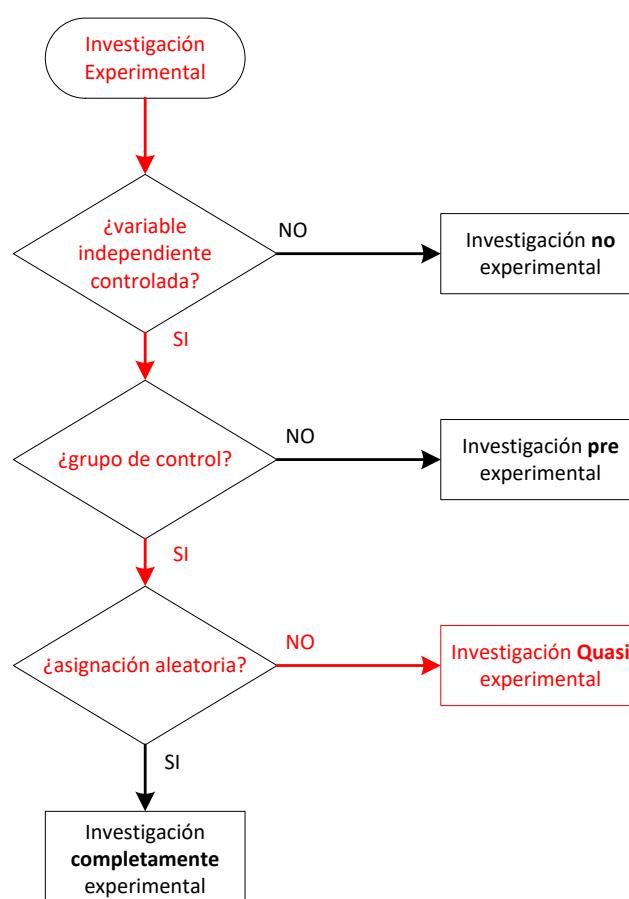


Figura 16. Árbol de decisión dicotómica de los métodos de investigación experimental.

Fuente: Tam, Vera & Oliveros (2008). Editado por el autor

La *Figura 16* presenta un flujoograma que permite determinar el subtipo de investigación experimental. Las variables independientes controladas en el proceso investigativo corresponde a los parámetros de configuración por software de los dispositivos LoRa (parámetros SF, BW y CR), al igual que el tipo de hardware utilizado (Arduino UNO, Arduino DUE y SDIN). Se establecen un subgrupo de dispositivos LoRa que corresponde a las versiones SX1272 y SX1278. Sin embargo, no se realizan pruebas aleatorias sino controladas sobre los dispositivos LoRa junto a sus respectivas configuraciones, por lo que la metodología de investigación es *Quasi-experimental*.

3.3.4 Población y muestra

La población corresponde a dispositivos transceivers de comunicación inalámbrica LoRa de Semtech junto con sistemas embebidos de hardware libre.

La muestra de la población propuesta corresponde a dispositivos transceivers de las referencias SX12782 (@868 MHz) y SX1278 (@434 MHz), al igual que tarjetas de desarrollo Arduino UNO, Arduino DUE y la tarjeta de adquisición de datos SDIN de GEDIRCI.

3.3.5 Metodología

La metodología utilizada corresponde al modelo del patrón lineal de investigación cuantitativa propuesto por Spradley (1980). La *Figura 17* presenta el esquema propuesto de etapas para el desarrollo de la investigación propuesta.

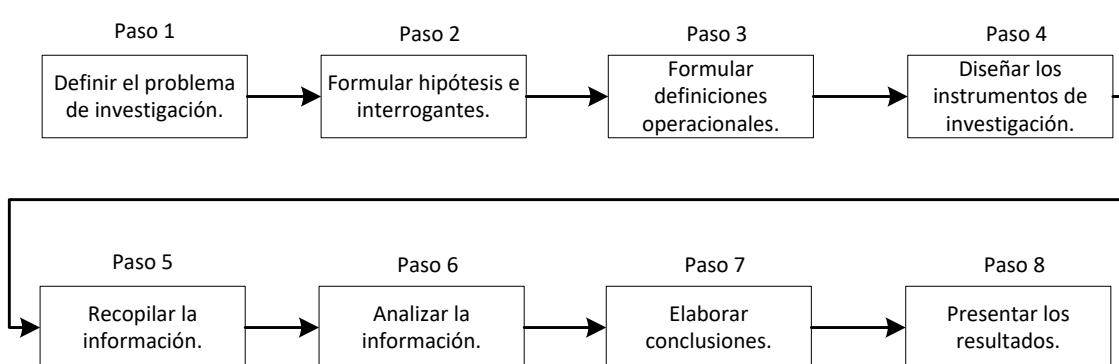


Figura 17. Patrón lineal de la investigación cuantitativa.

Fuente: Spradley (1980). Editada por el autor

Paso 1: Definición del problema de investigación

El seguimiento o monitoreo se define como un ejercicio a identificar de manera sistemática la calidad del desempeño de un sistema, subsistema o proceso con el fin de introducir los ajustes

o cambios pertinentes y oportunos para el logro de sus resultados y efectos en el entorno. De esta forma, el monitoreo permite analizar el estado actual de un sistema y propone acciones a tomar para lograr los objetivos, lo que se traduce en identificar estados reales o potenciales para hacer los ajustes oportunos a la ejecución (OIE - IDIE, 2010).

Respecto a infraestructuras, los sistemas de monitorización permiten determinar el estado de sus variables a partir del uso de sensores. Los sensores son elementos físicos que pertenecen a un dispositivo denominado transductor y tienen la capacidad de transformar una variable física a otro tipo de variable (generalmente eléctrica). Los sensores captan las señales necesarias para conocer el estado de un proceso y de esta manera decidir un comportamiento a futuro (Velásquez, s.f.). Es así como en múltiples campos se hace uso ampliamente de los sensores como en la automatización, en la domótica y el ámbito del desarrollo, entre otros.

En el ámbito del desarrollo, los sensores se utilizan de acuerdo al contexto, por ejemplo, para agua y saneamiento se utilizan sensores en infraestructuras para los sistemas de recolección de aguas lluvia (cisternas) con el fin de medir el nivel de agua almacenamiento, en infraestructuras de sistemas fotovoltaicos se utilizan piranómetros para medir la irradiación sobre el panel y establecer la eficiencia en la producción energética, para aplicaciones de cocinas mejoradas se utilizan sensores para medir la calidad del aire, entre cientos de aplicaciones.

Los sistemas de control o de almacenamiento de los datos para la monitorización de variables generalmente se ubican en puntos centrales para el fácil acceso, es decir, no necesariamente se ubican cerca a los sensores por lo que se deben buscar estrategias para transmitir la información de los sensores a los puntos de almacenamiento, recolección de datos o sistemas de monitorización. Tradicionalmente se utilizan cables para la transmisión de información de los sensores, solucionando un problema de comunicación siempre y cuando las distancias entre los sensores y el sistema de adquisición de datos sean cortas y si se utilizan los protocolos de comunicación adecuados para realizar esta tarea, como el RS485, CAN, RS232, entre otros. Sin embargo, algunos protocolos utilizan la técnica de aumentar los niveles de voltaje e incorporan circuitos adicionales que para aplicaciones del ámbito del desarrollo no resulta ser pertinente debido al aumento del consumo energético y aumento en los costos de implementación. Otro agravante corresponde en muchos casos a la limitante de expansión y modificaciones necesarias del hardware cuando se incorporan más sensores en los sistemas de monitorización. Por lo que el uso de comunicación cableada entre los sensores y los sistemas de adquisición de datos se podría implementar en casos muy específicos y no siempre permite ser una solución escalable.

Paso 2: Formular hipótesis e interrogantes.

Con el auge de las Tecnologías de la Información y las Comunicaciones en el siglo XXI, aparecen nuevas tendencias como lo es el Big Data y el Internet de las cosas (IoT o IdC).

Internet no solo ha permitido conectar personas, sino dispositivos, sensores y complejos sistemas. Es así como aparece el término “Internet de las cosas” o “internet de los objetos” que corresponde a la interconexión de diversos dispositivos a internet, los cuales generan datos en tiempo real.

Lo importante del “internet de las cosas” no radica en la conexión de las “cosas” a internet, sino en la información y conocimiento que estas puedan aportar para mejorar la calidad de las personas.

Con la posibilidad de obtener información valiosa de diferentes elementos a través de internet, aparecen nuevas técnicas y protocolos de comunicación a bajo consumo; es el caso de *LoRaWAN*, un protocolo de comunicación inalámbrica de bajo consumo y largo alcance para IoT (LoRa Alliance, 2016).

Sin embargo, “*La tecnología se crea en respuesta a las presiones del mercado y no de las necesidades de los pobres, que tienen escaso poder de compra... en la era de las redes, cada país necesita contar con capacidad para comprender las tecnologías mundiales y adaptarlas a las necesidades locales*”, PNUD (2001). Los dispositivos LoRa (estándar LoRaWAN) son un claro ejemplo de lo expuesto por el PNUD en 2001; por tal razón se propone adaptarlos a confiables sistemas de monitorización y a su vez a posibles aplicaciones del ámbito del desarrollo.

Por lo anteriormente presentado, es necesario plantear la preguntas: ¿Es posible adaptar los dispositivos LoRa como dispositivos de comunicación entre los sensores y los sistemas de adquisición de datos para aplicaciones del ámbito del desarrollo?, ¿qué condiciones y características deben tener los dispositivos LoRa para alcanzar su máximo rendimiento en términos consumo energético y alcance de transmisión?, ¿es posible crear un ruta guiada para el correcto uso de estos dispositivos?

Finalizado el proceso de investigación, se pretende demostrar o refutar la hipótesis: “*De acuerdo a las características de bajo consumo energético, bajo costo de adquisición, fácil configuración y alto alcance de transmisión, los dispositivos LoRa resultan ser idóneos para aplicaciones del ámbito del desarrollo como mecanismo de comunicación entre sensores y dispositivos de adquisición de datos*”

Paso 3: Formular definiciones operacionales.

Fase I: Búsqueda y documentación de los dispositivos propuestos.

En el apartado *Descripción de los dispositivos utilizados*, se presentan las principales características de los dispositivos propuestos entre los que se tiene:

- Características físicas y eléctricas de los transceiver LoRa con las versiones SX1272 y SX1278
- Definición de las bandas de frecuencia ISM (Industrial, Scientific and Medical).
- Descripción del protocolo SPI utilizado por los dispositivos LoRa.
- Definición y características de los sistemas embebidos.
- Importancia del hardware libre para el ámbito del desarrollo.
- Principales características de Arduino Uno y Arduino DUE.
- Principales características del módulo SDIN (desarrollado por GEDIRCI).
- Descripción de los IDE utilizados para la programación: Eclipse y Arduino IDE.
- Importancia del software libre para el ámbito del desarrollo.
- FreeRTOS: sistema operativo en tiempo real de código abierto.

Fase II: Desarrollo del Driver para el control de los dispositivos LoRa

El Anexo A presenta el “*Desarrollo del driver para el uso de los dispositivos de radio LoRa con sistemas embebidos*”. En el mencionado documento se presenta de forma detallada el método adecuado de configuración de los dispositivos LoRa (en el documento, ver apartado “*Configuración por software de los radio LoRa*”).

Fase III: Pruebas del driver para el control de los dispositivos LoRa

Para realizar las pruebas del driver en los dispositivos LoRa, fue utilizado el esquema presentado en la *Figura 18*. Se utilizaron transceivers compuestos por dispositivos LoRa (versiones SX1272 y SX1278) junto con sistemas embebidos como Arduino UNO, DUE y los módulos SDIN. Para determinar la pertinencia en el funcionamiento del driver desarrollado, se utilizaron instrumentos electrónicos, entre los que se tienen:

- *Osciloscopio*: útil para medir tiempos de transmisión y recepción de tramas.
- *Multímetro*: utilizado para medir la corriente consumida por el dispositivo cuando se encuentra trasmitiendo y recibiendo tramas de datos. Con la medición de la corriente,

implícitamente es posible calcular el consumo energético para las diferentes configuraciones de SF, CR y BW en los dispositivos LoRa.

- *Equipo de cómputo:* se utilizó un equipo de cómputo con sistema operativo *Linux Ubuntu 14*, sobre los cuales se instalaron las versiones IDE de Arduino y eclipse para la programación y depuración del código fuente incorporado en los sistemas embebidos (configuración de parámetros lógicos SF, CR y BW).

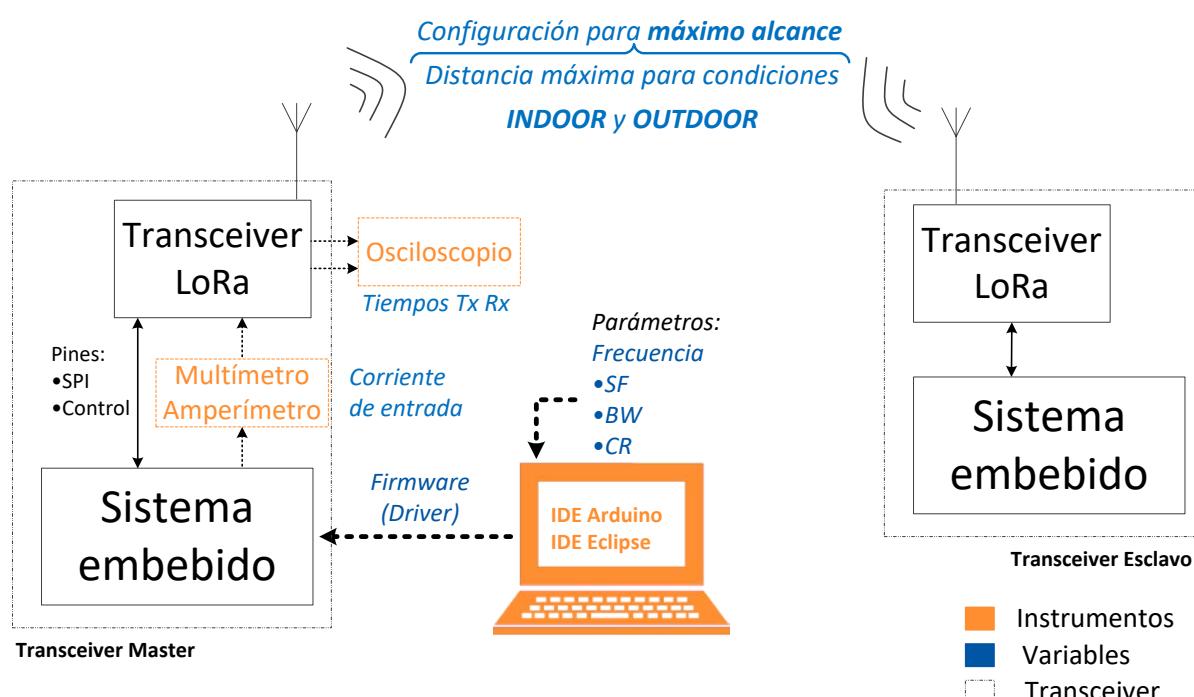


Figura 18. Esquema propuesto para pruebas del driver con los dispositivos LoRa.

Fuente: Esquema desarrollado por autor

Sin embargo, la Figura 18 resulta ser muy general, por lo que se hace necesario presentar el esquema de conexión para los diferentes dispositivos LoRa y sistemas embebidos utilizados.

La Figura 21 presenta el esquema de conexión utilizado entre los dispositivos LoRa SX1272 y el sistema embebido Arduino Uno. La Tabla 7 presenta la nomenclatura utilizada en el transceiver SX1272 y la tarjeta Arduino Uno.

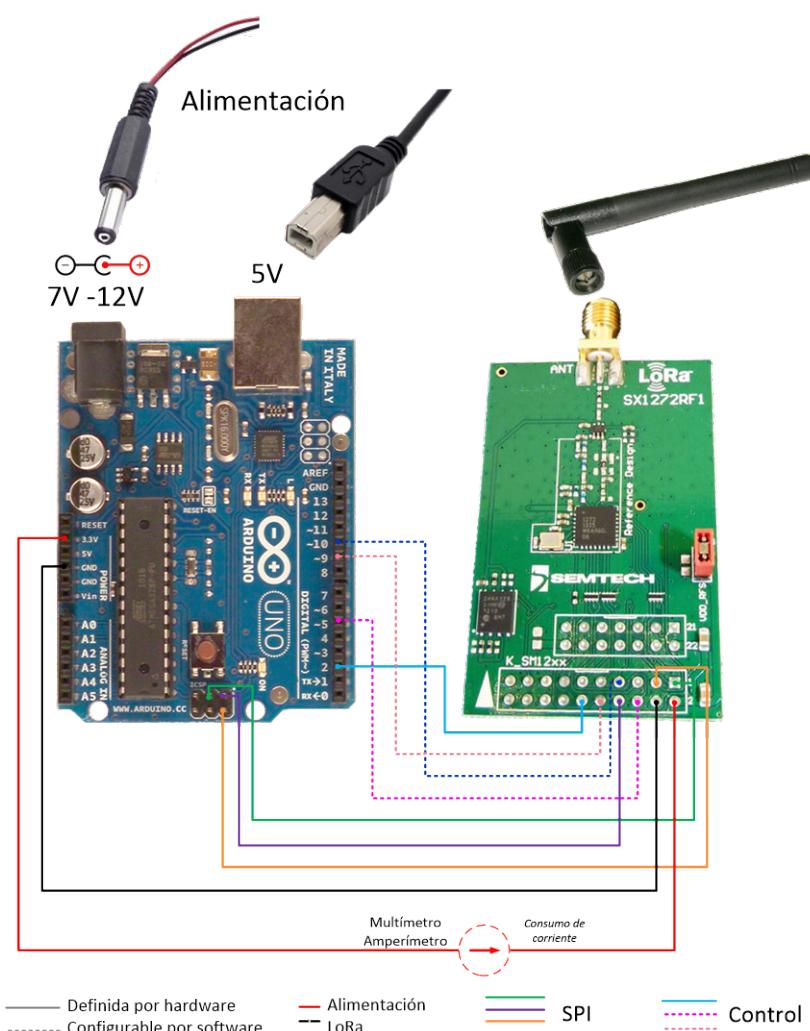


Figura 19. Esquema de conexión para SX1272 de Semtech y Arduino UNO.

Fuente: Esquema propuesto por el autor

Tabla 8. Nomenclatura utilizada para la conexión entre la tarjeta SX1272RF1 de Semtech y la tarjeta Arduino UNO.

Fuente: Semtech SX1272 (2015)

Tipo	Nomenclatura estándar	Notación en tarjetas	
		SX1272	Arduino UNO
Alimentación	VCC	2	3.3V
	GND	1	GND
SPI	NSS	7	10
	MOSI	3	11
	MISO	8	12
	SCK	1	13
CONTROL	RESET	10	9
	DIO0	12	2
	DIO3	6	5

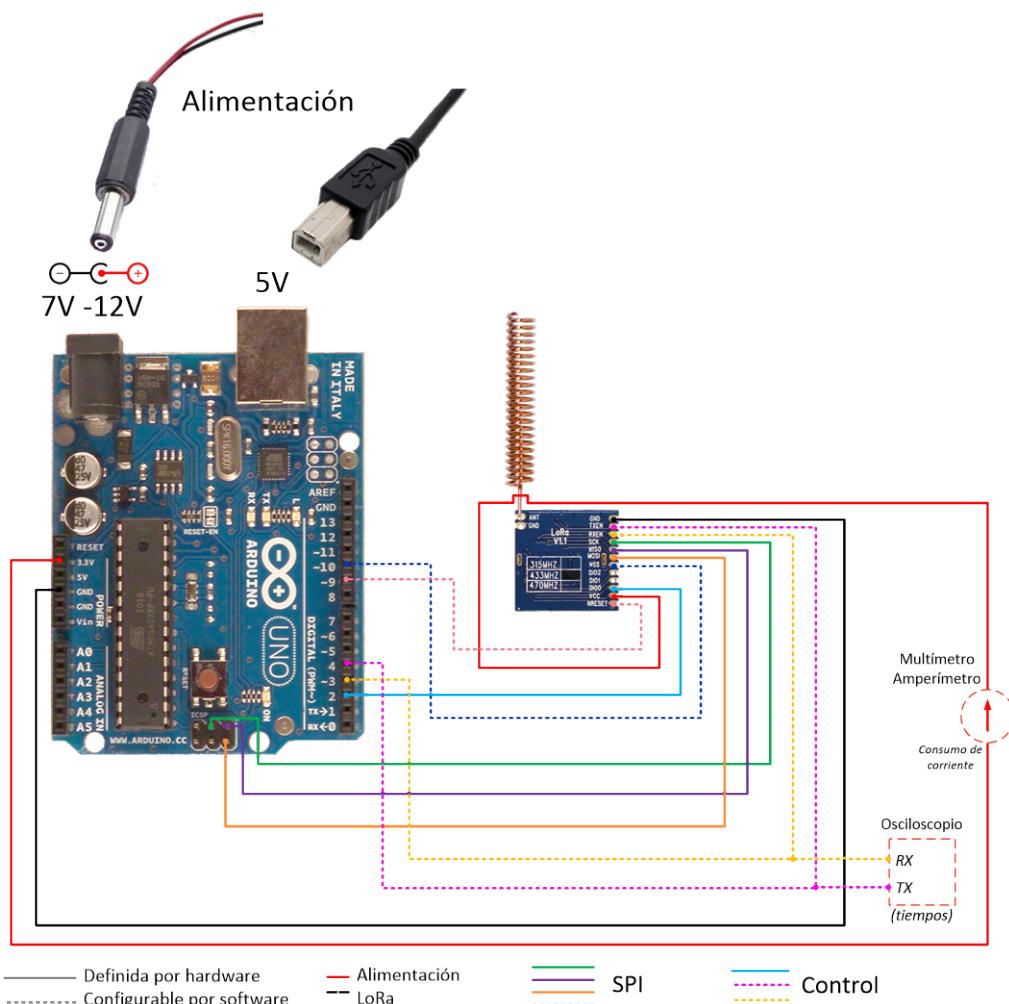


Figura 20. Esquema de conexión para LoRaJ278 de NiceRF y Arduino UNO.

Fuente: Esquema propuesto por el autor.

Tabla 9. Nomenclatura utilizada para la conexión entre la tarjeta LoRa1278 de NiceRF y la tarjeta Arduino UNO

Fuente: NiceRE (2015)

Tipo	Nomenclatura est醤dar	Notaci髇 en tarjetas	
		LoRa1278	Arduino UNO
Alimentaci髇	VCC	VCC	3.3V
	GND	GND	GND
SPI	NSS	NSS	10
	MOSI	MOSI	11
	MISO	MISO	12
	SCK	SCK	13
CONTROL	RESET	NRESET	9
	DIO0	DIO0	2
	TXEN	TXEN	3
	RXEN	RXEN	4

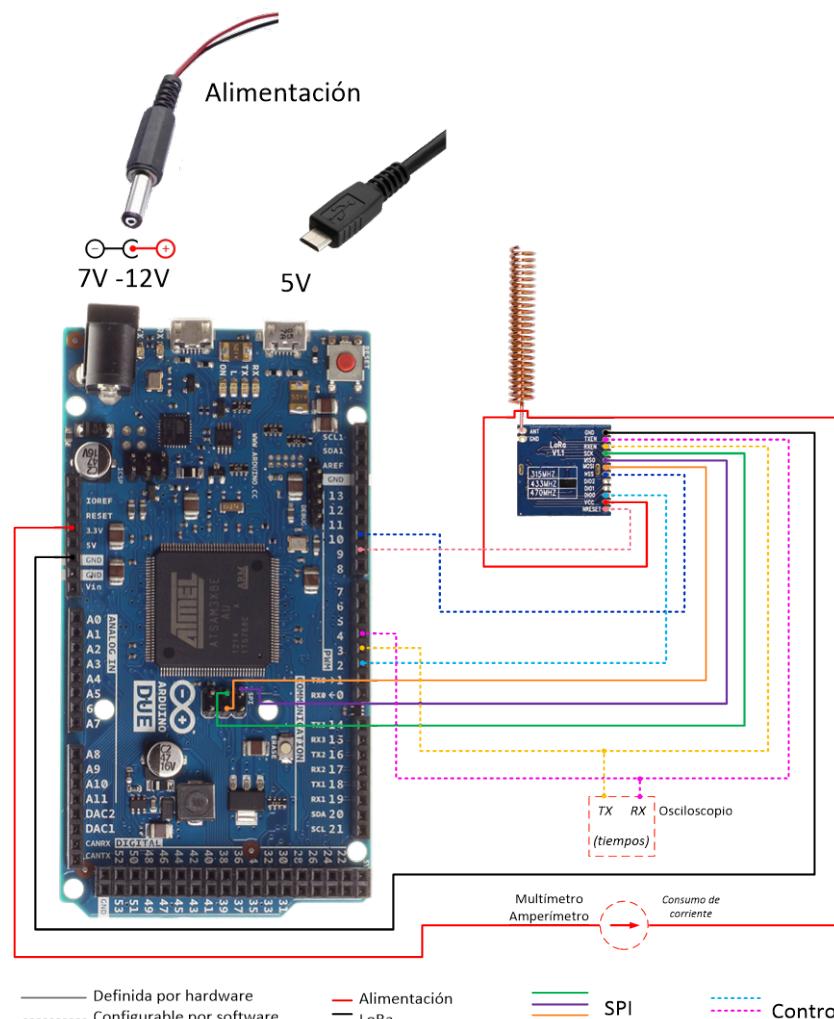


Figura 21. Esquema de conexión para LoRa1278 de NiceRF y Arduino DUE.

Fuente: Esquema propuesto por el autor

Tabla 10. Nomenclatura utilizada para la conexión entre la tarjeta LoRa1278 de NiceRF y la tarjeta Arduino DUE.

Fuente: NiceRF (2015)

Tipo	Nomenclatura estándar	Notación en tarjetas	
		LoRa1278	Arduino DUE
Alimentación	VCC	VCC	3.3V
	GND	GND	GND
SPI	NSS	NSS	10
	MOSI	MOSI	MOSI
	MISO	MISO	MISO
	SCK	SCK	SCK
CONTROL	RESET	NRESET	9
	DIO0	DIO0	2
	TXEN	TXEN	3
	RXEN	RXEN	4

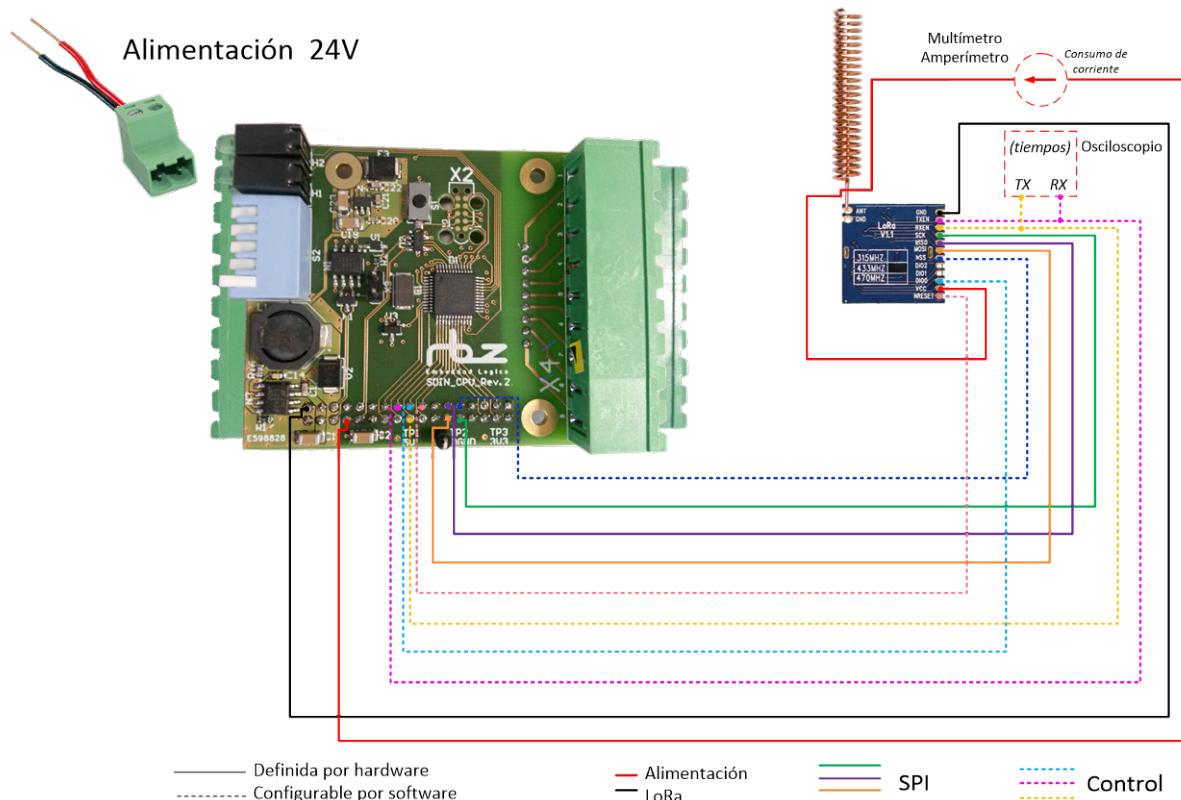


Figura 22. Esquema de conexión para LoRa1278 de NiceRF y módulo SDIN.

Fuente: Esquema propuesto por el autor

Tabla 11. Nomenclatura utilizada para la conexión entre la tarjeta LoRa1278 de NiceRF y el módulo SDIN.

Fuente: NiceRF (2015)

Tipo	Nomenclatura estándar	Notación en tarjetas	
		LoRa1278	SDIN
Alimentación	VCC	VCC	3.3V
	GND	GND	GND
SPI	NSS	NSS	PB12
	MOSI	MOSI	PB15
	MISO	MISO	PB14
	SCK	SCK	PB13
CONTROL	RESET	NRESET	PA7
	DIO0	DIO0	PA5
	TXEN	TXEN	PA3
	RXEN	RXEN	PA4

En las *Figuras 19 (Tabla 8)* a la *Figura 22 (Tabla 11)* se presentan los esquemas de conexión en donde se puede apreciar los instrumentos utilizados para determinar las variables relacionadas con el consumo de corriente, utilizando un amperímetro conectado en serie entre el terminal de salida de voltaje de 3.3V del sistema embebido y el pin de alimentación del dispositivo LoRa; y los tiempos de transmisión y recepción de tramas utilizando un osciloscopio. Con la corriente de consumo de corriente y los tiempos de transmisión y recepción de datos es posible determinar el consumo energético en los modos transmisión y recepción.

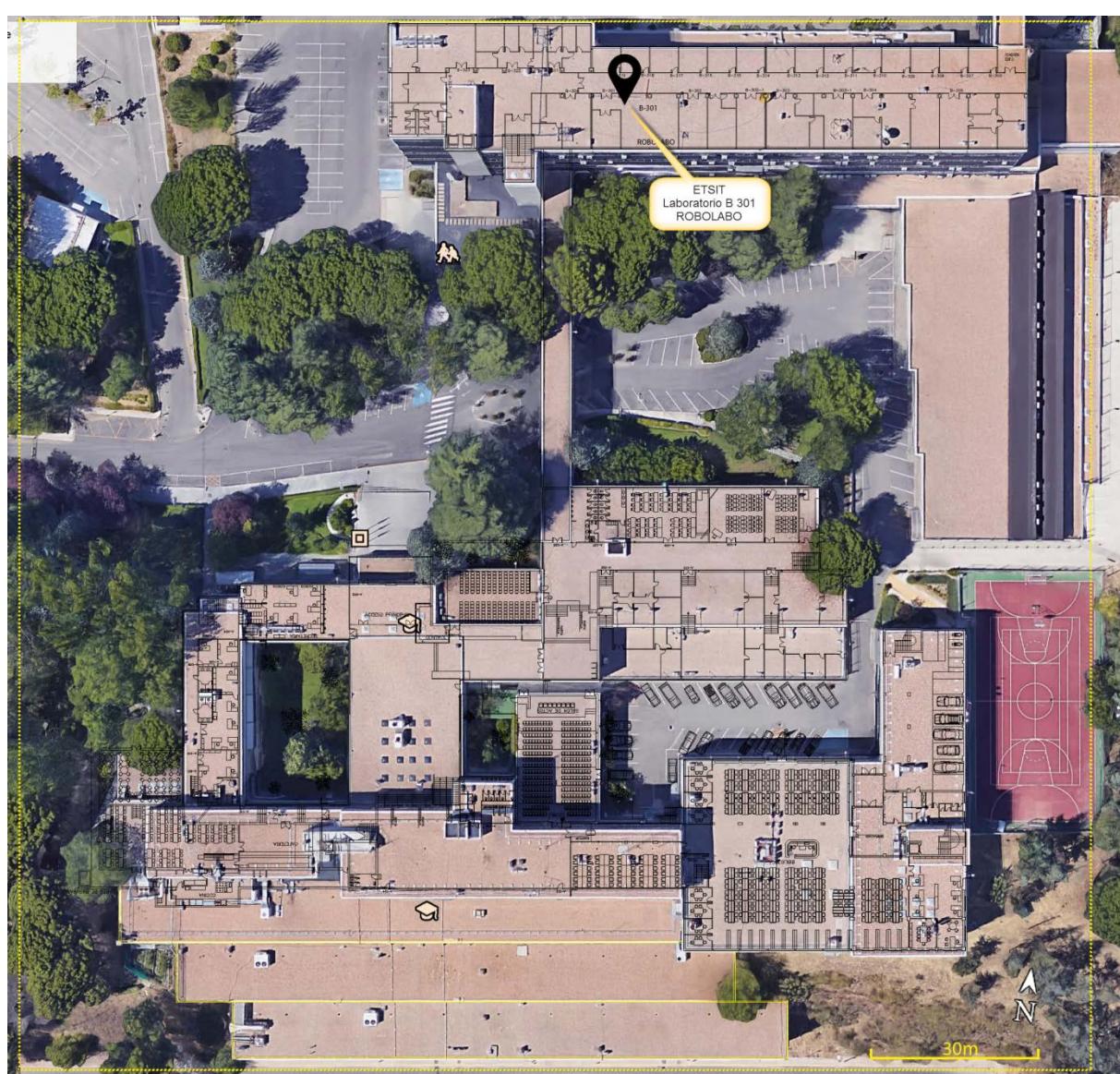


Figura 23. Espacio de pruebas Indoor para los dispositivos LoRa.
Fuente: Google earth 7.1.5.1557 y planos compartidos por la ETSIT

Las pruebas para los diferentes dispositivos se desarrollaron en las instalaciones de la Escuela Técnica Superior de Ingenieros de Telecomunicación ETSIT de la Universidad Politécnica de

Madrid. Tanto los dispositivos como los instrumentos para el desarrollo del driver y para la ejecución de las pruebas fueron suministrados por GEDIRCI, explícitamente por el *laboratorio B-301 de Robolabo* del Departamento de Tecnología Fotónica y Bioingeniería.

Se desarrollaron pruebas de máximo alcance para espacio Indoor y Outdoor. Las pruebas de máximo alcance buscan encontrar la mejor configuración de los dispositivos para obtener la máxima distancia de transmisión sin comprometer la veracidad de los datos, es decir que los datos transmitidos y recibidos sean correctos y además que no se comprometan los consumos energéticos. Las pruebas de configuración a máximo alcance para espacios Indoor (con obstáculos como infraestructuras) se desarrollaron en las instalaciones de la ETSIT – UPM (ver *Figura 23*).

Las pruebas de configuración a máximo alcance para espacios Outdoor (con línea de vista u obstáculos como vegetación en espacios amplios) se desarrollaron en el *parque Dehesa de la Villa*, espacio muy próximo a las instalaciones de la ETSIT – UPM como se observa en la *Figura 24*.



Figura 24. Espacio de pruebas Outdoor para los dispositivos LoRa.

Fuente: Google earth 7.1.5.1557

Fase IV: Pruebas del driver para una red tipo estrella

Finalmente se aumenta la complejidad del diseño creando una red estrella con un nodo central (circuito maestro) y 3 dispositivos esclavos. Como circuitos esclavos utilizan diversas versiones de sistemas embebidos. La *Figura 25* presenta el esquema propuesto donde se aprecia la comunicación simultánea de 3 dispositivos hacia un nodo central. Esta prueba demuestra la eficacia del driver desarrollado para aplicaciones complejas con diversos sensores ubicados en sitios remotos. Cabe resaltar que los dispositivos LoRa deben ser de la misma referencia, estar configurados a la misma frecuencia y con los parámetros SF, BW y CR idénticos.

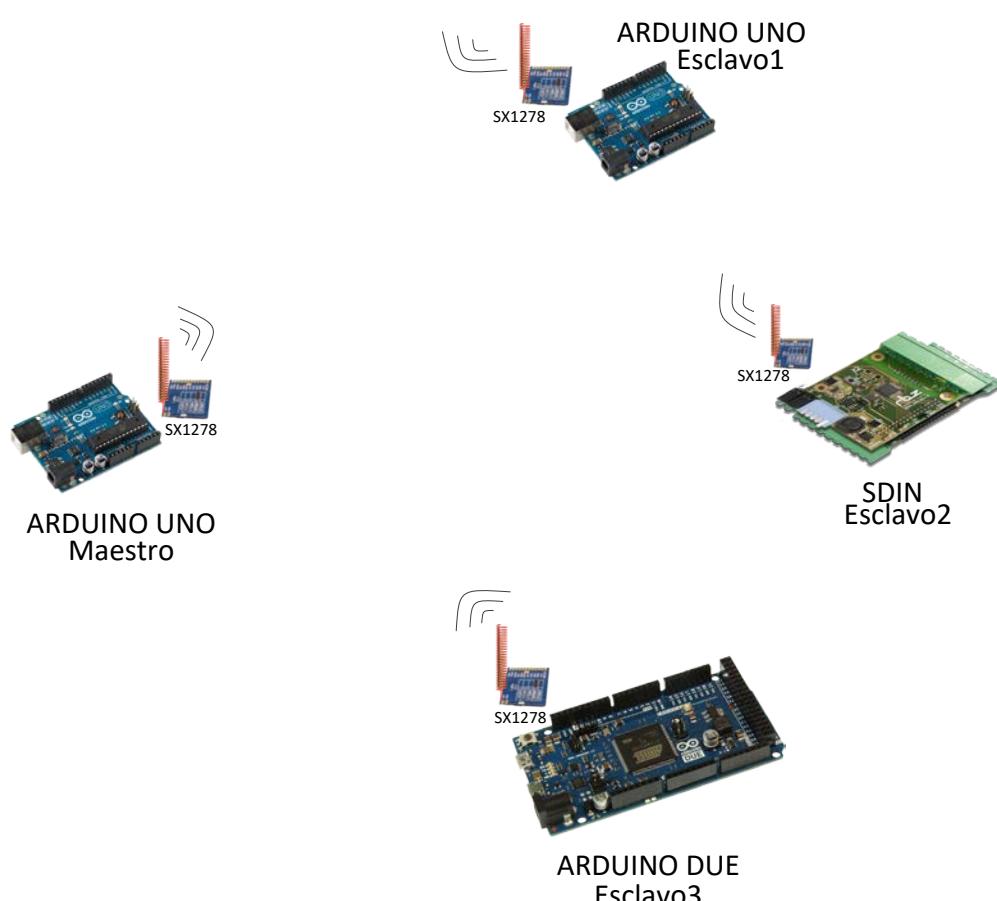


Figura 25. Tipología estrella para la prueba del driver generado.

Fuente: Esquema propuesto por el autor

Paso 4: Diseñar los instrumentos de investigación

Una vez establecidas las "*definiciones operacionales*", se proponen los instrumentos para recolectar información valida y confiable con el fin de corroborar o refutar la hipótesis planteada. Según Rojas (1989), la información recolectada es clasificada en información

primaria y secundaria: para la investigación, la información primaria corresponde a la información obtenida directamente de los instrumentos planteados por el investigador (ver *Tabla 12*); y la información secundaria se refiere a información extraída de fuentes documentales, que para el caso particular corresponde a hojas de especificaciones técnicas resumidas en el apartado *Transceiver Semtech LoRa*.

La *Tabla 12* presenta el instrumento propuesto para determinar los consumos energéticos de acuerdo a diferentes configuraciones por software en los transceivers LoRa. Este instrumento se utilizará para el dispositivo LoRa (SX1272 o SX1278) que presente el mayor alcance y estabilidad en la transmisión y recepción de los datos.

Tabla 12. Instrumento propuesto para la recopilación de datos en las pruebas con dispositivos LoRa para diferentes configuraciones.

Fuente: Instrumento propuesto por el autor

Parámetros configurables por software			Consumo de corriente mA			Tiempos segundos		Consumo energético mW·s	
SF	BW	CR	TX	RX	Sleep	TX	RX	TX	RX

Parámetros

- Configurables por software
- Medidos con multímetro
- Determinados por osciloscopio
- Calculados a partir de los datos

Paso 5: Recopilar la información.

La técnica utilizada para recopilar información corresponde a observaciones directas que corresponde al método más preciso para obtener información para todas las variables debido a obtención de datos cuantitativos por medio de la medición del fenómeno que se esté observando (Hernández, s.f.).

Acorde con la metodología de investigación, las variables utilizadas son de tipo cuantitativo, las cuales según las características o propiedades pueden presentarse en diversos grados o intensidad y tienen un carácter numérico o cuantitativo.

De acuerdo a las relaciones de causalidad, las variables se definen como independientes y dependientes. Las variables independientes son los parámetros configurables por software SF,

BW y CR que según las combinaciones entre estos permiten determinar el alcance o distancia máxima de recepción de datos, la velocidad de los datos, sobrecarga por corrección o detección de errores en la información trasmisita y el consumo energético al transmitir y recibir datos. Estos valores deben ser los mismos en todos los dispositivos radio LoRa de la red inalámbrica. Los parámetros independientes de configuración se definen como (Semtech SX1272, 2015 & Semtech SX1278, 2015):

BW: Ancho de banda (entre más bajo sea este valor, el tiempo de transmisión será mayor)

SF rate: Es el factor de alcance expresado como logaritmo de base 2. Cuanto mayor sea este valor, mejor será el rendimiento en la transmisión de datos.

CR: Tasa de codificación de errores: Entre mayor sea este valor, mayor será la fiabilidad de los datos, pero con una sobrecarga en el tiempo de transmisión.

La *Tabla 13* presenta la combinación de los principales variables independientes para la investigación. Cabe aclarar que estos valores deben ser los mismos en todos los dispositivos radio LoRa de la red inalámbrica.

Tabla 13. Combinación de los principales parámetros configurables para los dispositivos LoRa.

Fuente: Semtech SX1278 (2015) & Semtech SX1272 (2015)

Ancho de banda BW (kHz)		Factor de alcance SF		Tasa de codificación de errores. CR	
Hardware	Software RADIO.BW	Hardware	Software RADIO.sf	Hardware	Software RADIO.cr
500	BW500	6 → 64 chips / symbol (modo utilizado para FSK)	SF_6	4/5	CR_4_5
250	BW250	7 → 128 chips / symbol	SF_7	4/6	CR_4_6
125	BW125	8 → 256 chips / symbol	SF_8	4/7	CR_4_7
62.5*	BW62_5*	9 → 512 chips / symbol	SF_9	4/8	CR_4_8
41.7*	BW41_7*	10 → 1024 chips / symbol	SF_10		
31.25*	BW31_25*	11 → 2048 chips / symbol	SF_11		
20.8*	BW20_8*	12 → 4096 chips / symbol	SF_12		
15.6*	BW15_6*				
10.4*	BW10_4*				
7.8*	BW7_8*				

*: Aplica solamente para los transceiver SX1278

Los parámetros son configurados en el driver (Ver Anexo A: Desarrollo del driver para el uso de los dispositivos de radio LoRa con sistemas embebidos – apartado “*Registros de configuración SF, BW y CR del radio LoRa*”) mediante la definición de las variables *RADIO.sf*, *RADIO.bw* y *RADIO.cr*.

A partir de la combinación de los parámetros SF, BW y CR se realizan las mediciones de consumos de corriente cuando el dispositivo se encuentra transmisión y recepción datos.

En las *Figuras 19 (Tabla 8)* a la *Figura 22 (Tabla 11)* se presentan los esquemas de conexión propuestos para el desarrollo de las pruebas, donde se muestran los instrumentos utilizados para determinar las variables dependientes relacionadas con el consumo de corriente. El consumo de corriente se determina utilizando un amperímetro conectado en serie entre el terminal de salida de voltaje de 3.3V del sistema embebido y el pin de alimentación del dispositivo LoRa; y los tiempos de transmisión y recepción de tramas son determinados utilizando un osciloscopio. Con la información de consumo de corriente y los tiempos de transmisión y recepción de datos, es posible determinar el consumo energético en los modos transmisión y recepción.

El modelo matemático para determinar la potencia necesaria para transmitir y recibir datos está dada por (1):

$$P_E = V_{CC} \cdot I_{CC} \quad (1)$$

Dónde:

P_E : Potencia eléctrica necesaria para transmitir o recibir datos. Se presenta en vatios.

V_{CC} : Voltaje nominal de alimentación. Según las especificaciones técnicas de los dispositivos LoRa corresponde a 3.3V de voltaje constante. Esta variable será constante o invariante.

I_{CC} : Corriente consumida cuando el dispositivo transmite o recibe datos. El valor máximo de corriente consumida puede ser configurado para 100mA o 240mA. Para el caso de las pruebas, se plantea un máximo de consumo de hasta 100mA. Esta variable será adquirida con el uso de un amperímetro según las *Figura 19* a la *Figura 22*.

Sin embargo, la potencia también se define como la razón de cambio del trabajo o energía por unidad de tiempo (Chapman, 2012), como se observa en la ecuación (2):

$$P_E = \frac{W}{t} \quad (2)$$

Dónde:

W : corresponde a la energía.

P_E : Potencia eléctrica.

t : Tiempo en segundos.

De acuerdo a (2) se define la energía como:

$$W = P_E \cdot t \quad (3)$$

Por lo que se concluye que la energía está dada como la potencia por unidad de tiempo. Para determinar la energía necesaria para transmitir y recibir datos en los dispositivos LoRa, se utilizan los modelos (1) y (3):

$$W = (V_{CC} \cdot I_{CC}) \cdot t \quad (4)$$

$$W_{Tx} = (3.3 \cdot I_{Tx}) \cdot t_{Tx} \quad (5)$$

$$W_{Rx} = 3.3 \cdot I_{Rx} \cdot t_{Rx} \quad (6)$$

$$W_{Rx} = 3.3 \cdot I_{Rx} \cdot t_{Rx} \quad (7)$$

Según las ecuaciones (6) y (7), a partir de la corriente eléctrica necesaria para transmitir y recibir datos (utilizando un amperímetro) y los tiempos necesarios para transmitir y recibir información, se establecen los valores de energía necesaria para transmitir y para recibir datos del instrumento de la *Tabla 12*.

Paso 6: Analizar la información.

El instrumento de la *Tabla 12* diligenciado será presentado y analizado en el apartado *RESULTADOS*, donde se determinará las características de consumo energético según la configuración de los parámetros SF, BW y CR. Además se presentan los resultados de las pruebas de máximo alcance para espacios *Indoor* como *Outdoor* (ver Figura 25 y Figura 26).

Paso 7: Elaborar conclusiones.

A partir del apartado *RESULTADOS*, se desarrollan las *CONCLUSIONES* junto con comentarios y recomendaciones a partir de la observación y uso de los dispositivos LoRa con los sistemas embebidos Arduino y el módulo SDIN.

Paso 8: Presentar los resultados.

Los resultados se presentan en el apartado *RESULTADOS*. Estos resultados se basan en los anexos creados: Anexo A: “Desarrollo del driver para el uso de los dispositivos de radio LoRa con sistemas embebidos” y Anexo B: “Instalación y uso del driver LoRa para sistemas embebidos”. El driver y algunos ejemplos son presentados en el espacio GitHub del grupo de investigación GEDIRCI con el fin de compartir la información de forma gratuita y colaborar con diferentes desarrolladores (ver <https://github.com/Robolab0>). Finalmente, se realiza un video resumen de la investigación desarrollada.

4. RESULTADOS

4.1 Pruebas de alcance y consumo energético

De acuerdo con la *Metodología*, en el apartado “*Fase III: Pruebas del driver para el control de los dispositivos LoRa*” se plantea el uso de dos sistemas embebidos, uno utilizado como maestro y el otro utilizado como esclavo.

El dispositivo maestro se encarga de realizar la petición de datos a un dispositivo esclavo ubicado en un sitio remoto, el cual obtiene datos de sensores. El esclavo remite la información solicitada y continuamente se encuentra esperando peticiones del dispositivo maestro. El fluajograma y código fuente empleado para la prueba se encuentra en el “*ANEXO A Desarrollo del driver para el uso de los dispositivos de radio LoRa con sistemas embebidos*”.

En la *Figura 28* se presenta un diagrama esquemático de la aplicación propuesta: maestro-esclavo (sensor) con la utilización de dispositivo LoRa para la realización de las pruebas de alcance (prueba “*ping-pong*”).



Figura 26. Principio de funcionamiento “ping-pong” para la realización de pruebas de alcance en dispositivos LoRa.

Fuente: Desarrollado por el autor

Básicamente esta prueba consiste en el envío continuo de 3 Bytes codificados desde el dispositivo maestro para realizar la petición de datos. Una vez el esclavo recibe estos 3 Bytes, captura los datos de un sensor, los codifica en 5 Bytes y los trámite hacia el maestro. En caso que el maestro detecte información errónea (utilizando el CDC o verificación por redundancia cíclica), vuelve a solicitar la información al dispositivo esclavo.

Para realizar las pruebas de máximo alcance, se realizaron desplazamientos entre los edificios A y B de la Escuela Técnica Superior de Ingenieros en Telecomunicación - ETSIT. El dispositivo maestro permaneció en el edificio B, laboratorio B-301 Robolabo; mientras que con el dispositivo esclavo se realizaron los desplazamientos entre el edificio B y el edificio A.

Una vez se logró establecer el enlace de comunicación entre ambos dispositivos, se realizó la comprobación en tiempo real de los datos trasmisidos/recibidos y la detección de errores utilizando el “*Serial Monitor*” en el IDE Arduino (para sistemas embebidos Arduino) y la herramienta de depuración o “*debugger*” en el IDE Eclipse (para el módulo SDIN). Cabe aclarar que ambos dispositivos (maestro y esclavo) contaron con configuraciones idénticas para mantener la comunicación.

Durante el proceso se cambió la configuración de los principales parámetros para los dispositivos LoRa. Estos parámetros corresponden a SF, BW y CR cuyas opciones se presentan en la *Tabla 13*.

4.1.1 Pruebas de alcance para LoRa SX1272

La configuración más estable y de mayor alcance para los dispositivos SX1272 LoRa detectada a partir de pruebas de comunicación, se presenta en la *Tabla 13*.

Tabla 14. Parámetros de configuración por software en los dispositivos SX1272 para obtener máximo alcance

Fuente: Información recolectada por el autor

Frecuencia Freq (Mhz)		Ancho de banda BW (kHz)		Factor de alcance SF		Tasa de codificación de errores. CR	
Hardware	Software RADIO.freq	Hardware	Software RADIO.BW	Hardware	Software RADIO.sf	Hardware	Software RADIO.cr
869.525	869525000	125	BW125	12 → 4096 chips / symbol	SF_12	4/5	CR_4_5

De acuerdo a los parámetros de la *Tabla 14*, se encuentra lo siguiente:

- Entre menor sea el BW, mayor tiempo tomara la comunicación y esto se debe a que la frecuencia es inversamente proporcional al tiempo. Sin embargo entre menor sea la frecuencia, mayor será el alcance de transmisión esperado. El valor de BW125 corresponde al mínimo valor posible para el SX1272.
- El valor de SF o factor de alcance determina el rendimiento en la transmisión de los datos, es decir que entre mayor sea este valor, el dispositivo tendrá menor probabilidad de recibir datos errados y mayor radio de cobertura.

- El CR asegura la fiabilidad de los datos. Entre mayor sea este valor, mayor será la fiabilidad de los datos, pero con una sobrecarga en el tiempo de transmisión.

La configuración presentada en la *Tabla 14*, asegura el máximo alcance y una fiabilidad de la información trasmisita/recibida razonable. Sin embargo estas características favorables para la comunicación, implican un tiempo de transmisión mayor y por ende un mayor consumo energético.

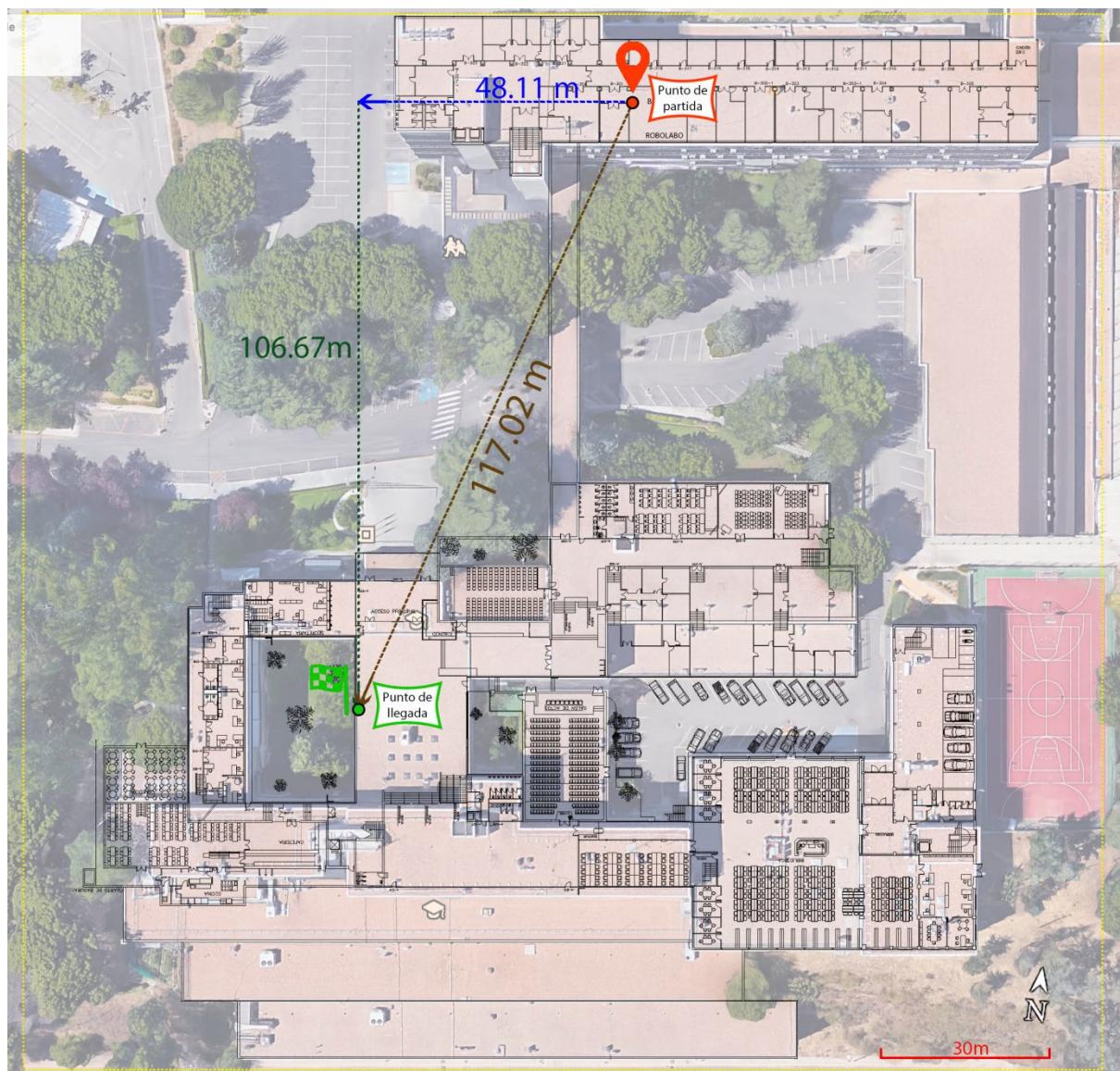


Figura 27. Máximo alcance en transmisión/recepción de datos para LoRa SX1272 (plano XY).

Fuente: Google earth 7.1.5.1557 y planos compartidos por la ETSIT. Escala 1:30

En términos de distancia o máximo alcance para la transmisión y recepción de datos en espacios *Indoor*, se realizaron los cálculos a partir de planos de las instalaciones de la Escuela técnica Superior de Ingenieros en Telecomunicación. El punto de partida fue el laboratorio B-

65

301 de Robolabo y el punto de máximo alcance o final corresponde al intermedio del salón del acceso principal del edificio A de la ETSIT.

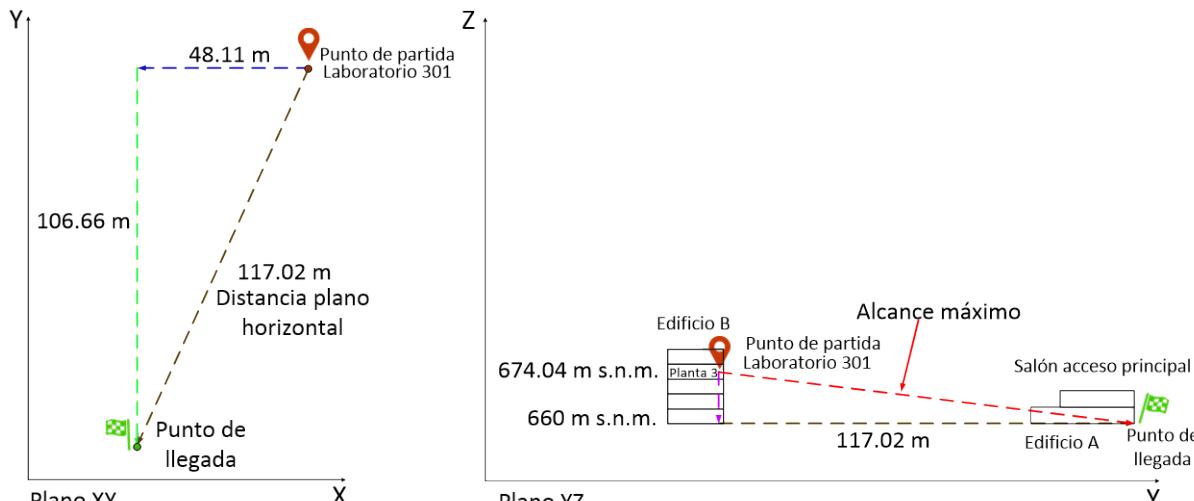


Figura 28. Esquema para determinar el alcance en transmisión/recepción de datos para LoRa SX1272 (plano XY e YZ).

Fuente: Información obtenida por el autor y planos compartidos por la ETSIT

La Figura 28 presenta los planos XY e YZ de referencia para realizar el cálculo del máximo alcance en las instalaciones de la ETSIT. Para determinar la distancia desde el punto de partida hasta el punto de llegada o de máximo alcance, se realizan cálculos trigonométricos. En la Figura 28 (izquierda), se presenta la máxima distancia alcanzada con referencia del suelo o plano horizontal, esta distancia se calculó utilizando el teorema de Pitágoras con referencia a las distancias de los ejes X (48.11m) y Y (106.66m). Para determinar el máximo alcance con referencia a los transceivers, se utiliza la Figura 28 (derecha). La diferencia de altura entre los dispositivos se calcula utilizando el siguiente modelo:

$$h_{entre\ dispositivos} = h_{dispositivo\ maestro} - h_{dispositivo\ esclavo} \quad (8)$$

$$h_{entre\ dispositivos} = 674.04 - 660 \quad (9)$$

$$h_{entre\ dispositivos} = 14.04 \text{ m} \quad (10)$$

Finalmente, utilizando el teorema de Pitágoras se calcula la distancia de alcance máximo lograda por la configuración presentada en la Tabla 13 por el dispositivo LoRa SX1272:

$$distancia_{máx\ alcance} = \sqrt{h_{entre\ dispositivos}^2 + distancia_{referencia\ suelo}^2} \quad (11)$$

$$distancia_{máx\ alcance} = \sqrt{14.04^2 + 117.02^2} \quad (12)$$

$$distancia_{máx\ alcance\ SX1272} = 117.86\ m \quad (13)$$

El máximo alcance logrado para los dispositivos SX1272 con la configuración que permite obtener el mejor rendimiento en términos de alcance (*Tabla 14*) es de 118 metros en aplicaciones *Indoor*.

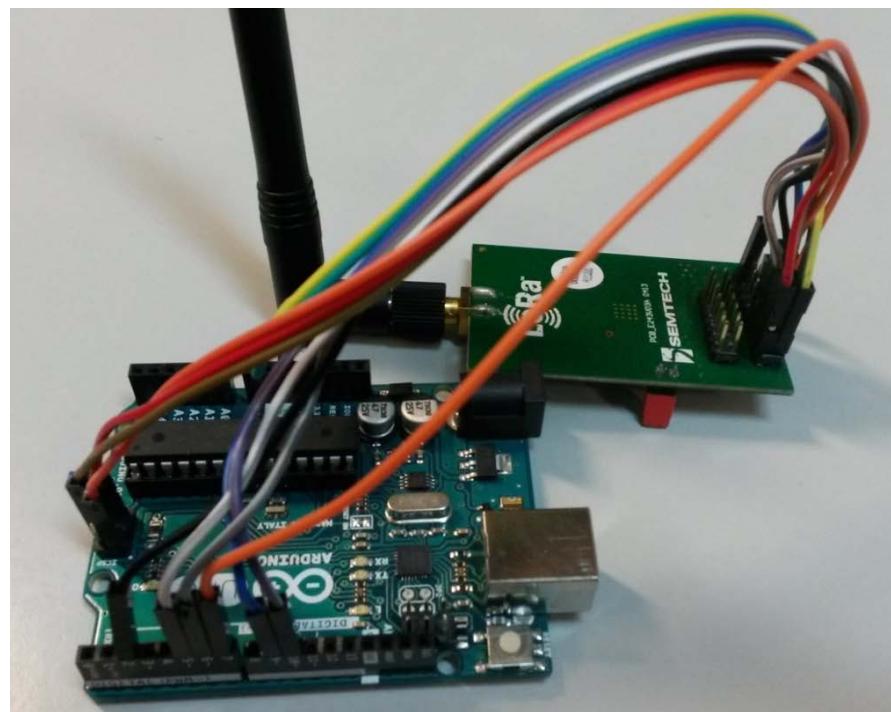


Figura 29. Dispositivo esclavo conformado por sistema embebido Arduino Uno y transceiver SX1272.

Fuente: Fotografía tomada por el autor

4.1.2 Pruebas de consumo energético y alcance para LoRa SX1278

En el caso del LoRa SX1278, se utilizaron dos fabricantes de tarjetas que realizan adecuaciones sobre el chip para facilitar la manipulación. Las referencias de las tarjetas utilizadas son las *DRF1278F* de *Dorji* y *LoRa1278* de *NiceRF* (ver *Tabla 5*). Para cada tarjeta se realizó la comprobación de consumos energéticos a diferentes configuraciones con la frecuencia de 433.800Mhz.

De acuerdo con la *Metodología*, en el apartado “*Paso 4: Diseñar los instrumentos de investigación*” se presentaron los instrumentos planteados para determinar la relación entre los parámetros configurables por software de los dispositivos LoRa y el consumo energético.

4.1.2.1 Pruebas de consumo energético para DRF1278F de Dorji

La *Tabla 15* presenta la recopilación de las pruebas de consumo energético a diferentes configuraciones para los dispositivos *DRF1278F de Dorji*. El cálculo del consumo energético se presenta en el apartado “*Paso 5: Recopilar la información*”, con el modelo (7).

Tabla 15. Instrumento con la recopilación de datos para las pruebas con los dispositivos LoRa para diferentes configuraciones en DRF1278F de Dorji.

Fuente: Información recolectada por el autor

# C on f	Parámetros configurables por software			Consumo de corriente mA					Tiempos segundos		Consumo energético mili-Vatios·s		Comentarios
	SF	BW	CR	I_{TXmin}	I_{TX}	I_{TXmax}	I_{RX}	I_{Sleep}	t_{TX}	t_{RX}	W_{TX}	W_{RX}	
1	9	500	4_5	50	52,5	55	11,5	1,7	0,026	0,035	4,50	1,33	Estable
2	12	500	4_5	25	57,5	90	11,5	1,7	0,21	0,22	39,85	8,35	Estable
3	9	125	4_5	38	54	70	11,7	1,7	0,105	0,13	18,71	5,02	Estable
4	12	125	4_5	80	95	110	12	1,7	0,8	0,9	250,80	35,64	Estable
5	9	61,5	4_5	30	62,5	95	11,7	1,7	0,205	0,23	42,28	8,88	Estable
6	12	61,5	4_5	84	98,5	113	11,2	1,7	1,6	1,8	520,08	66,53	Estable
7	9	41,7	4_5	60	82	104	11	1,7	0,3	0,48	81,18	17,42	Estable
8	12	41,7	4_5	79	96	113	11,1	1,7	2,5	2,6	792,00	95,24	Estable
9	9	31,5	4_5	93	101	109	11,9	1,7	0,4	0,505	133,32	19,83	Estable
10	12	31,5	4_5	113	113	11,9	1,7	3,5	4	1305,1	157,0	Estable	
11	9	20,8	4_5	99	106	113	11,6	1,7	0,75	0,8	262,35	30,62	Estable
12	7	20,8	4_5	28	60	92	11	1,7	0,5	0,6	99,00	21,78	Estable
13	9	7,8	4_5	113	113	113	11,7	1,7	X	X	X	X	Inestable

#Conf: Número de configuración

Parámetros

- Configurables
- Medidos con multímetro (multitester)
- Determinados por osciloscopio
- Calculados a partir de los datos

De acuerdo a los resultados presentados en la *Tabla 15*, la configuración SF=12, BW=32.5 y CR=4/5 (ver configuración # **10**), cuenta con un significativo consumo energético 1305 mW·s para transmitir 3 Bytes y de 157 mW·s para recibir 5 Bytes. Si bien es la configuración que logra establecer el máximo alcance debido al que el parámetro BW es el más bajo, en la práctica se evidenciaron problemas por transmitir /recibir gran cantidad de datos errados. Por las anteriores razones, se establece el máximo rendimiento en términos de alcance con la configuración # **8**: SF=12, BW=41.7 y CR=4/5.

4.1.2.2 Pruebas de consumo energético para LoRa1278 de NiceRF

La *Tabla 16* presenta la recopilación de las pruebas de consumo energético a diferentes configuraciones para los dispositivos *LoRa1278 de NiceRF*. El cálculo del consumo energético se presenta en el apartado “*Paso 5: Recopilar la información*”, con el modelo (7).

Tabla 16. Instrumento con la recopilación de datos para los dispositivos LoRa a diferentes configuraciones para LoRa1278 de NiceRF.

Fuente: Información recolectada por el autor

# C on f	Parámetros configurables por software			Consumo de corriente mA					Tiempos segundos		Consumo energético mili-Vatios·s		Coment arios
	SF	BW	CR	I_{TXmin}	I_{TX}	I_{TXmax}	I_{RX}	I_{Sleep}	t_{TX}	t_{RX}	W_{TX}	W_{RX}	
1	9	500	4_5	45	46	47	11,5	1,7	0,025	0,03	3,80	1,14	Estable
2	12	500	4_5	25	52,5	80	11,5	1,7	0,21	0,23	36,38	8,73	Estable
3	9	125	4_5	34	47	60	11,7	1,7	0,105	0,13	16,29	5,02	Estable
4	12	125	4_5	37	58,5	80	11,7	1,7	0,8	0,9	154,44	34,75	Estable
5	9	61,5	4_5	20	40	60	11,7	1,7	0,205	0,26	27,06	10,04	Estable
6	12	61,5	4_5	80	85	90	11,2	1,7	1,7	1,8	476,85	66,53	Estable
7	9	41,7	4_5	40	60	80	11	1,7	0,3	0,4	59,40	14,52	Estable
8	12	41,7	4_5	80	85	90	11	1,7	2,4	2,6	673,20	94,38	Estable
9	9	31,5	4_5	50	65	80	11,5	1,7	0,4	0,52	85,80	19,73	Estable
10	12	32,5	4_5	80	80	80	11	1,7	3,5	4	924,00	145,2	Estable
11	9	20,8	4_5	60	75	90	11	1,7	X	X	X	X	Inestable
12	7	20,8	4_5	30	45	60	11	1,7	X	X	X	X	Inestable

#Conf: Número de configuración

Parámetros

Configurables

Medidos con multímetro (multitester)

Determinados por osciloscopio

Calculados a partir de los datos

De acuerdo a los resultados presentados en la *Tabla 16*, la configuración SF=12, BW=32.5 y CR=4/5 (ver configuración # **10**), cuenta con un significativo consumo energético 924 mW·s para transmitir 3 Bytes y de 145.2 mW·s para recibir 5 Bytes. Si bien es la configuración que logra establecer el máximo alcance debido al que el parámetro BW es el más bajo, en la práctica se evidenciaron problemas por transmitir /recibir gran cantidad de datos errados. Por las anteriores razones, se establece el máximo rendimiento en términos de alcance con la configuración # **8**: SF=12, BW=41.7 y CR=4/5

4.1.2.3 Comparación de consumo energético entre DRF1278F de Dorji y LoRa1278 de NiceRF

De acuerdo a la *Tabla 15* y *Tabla 16*, se establece un bajo consumo en corriente de 1.7 mA cuando los transceivers se encuentran en modo SLEEP para las dos versiones de tarjetas SX1278. Se puede apreciar que el consumo de corriente cuando el transceiver se encuentra modo transmisión I_{TX} es superior para la versión *DRF1278F de Dorji* e incluso supera los umbrales de los 100mA. Además, el consumo de corriente se encuentra entre los 11 a 12 mA cuando el transceiver se encuentra en modo recepción para ambas versiones. También se puede observar que no existen diferencias significativas en los tiempos de transmisión y recepción entre las versiones *DRF1278F* y *LoRa1278*.

La Figura 30 representa la comparación de configuraciones y consumos energéticos entre las versiones *DRF1278F* y *LoRa1278* cuando los transceivers se encuentran en modo recepción. Se puede observar que no existen diferencias significativas en los consumos energéticos entre las dos versiones para el modo recepción.

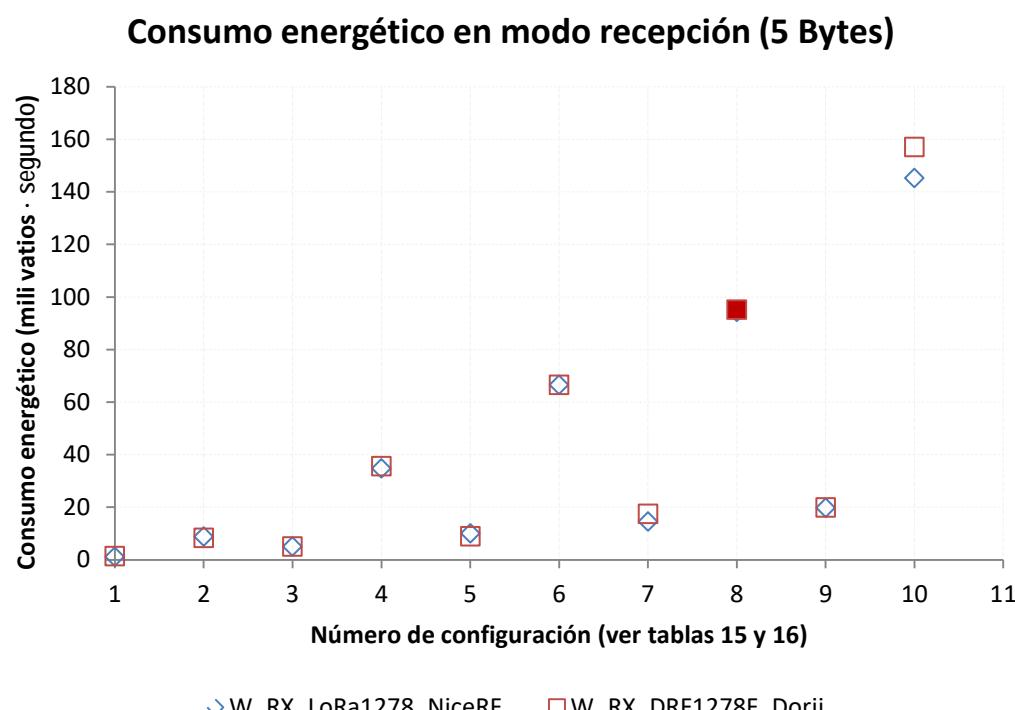


Figura 30. Representación gráfica del consumo energético de los transceivers *DRF1278F* y *LoRa1278* para modo recepción.

Fuente: Información recolectada por el autor

La Figura 31 representa la comparación de configuraciones y consumos energéticos entre las versiones *DRF1278F* y *LoRa1278* cuando los transceivers se encuentran en modo transmisión. Se puede apreciar que el dispositivo *DRF1278F* cuenta con un mayor consumo de corriente con respecto al dispositivo *LoRa1278*. Esta característica se hace más evidente en la configuración # 8 y # 10 (ver Tabla 15 y Tabla 16) que corresponde a los parámetros donde se obtiene el máximo alcance.

Por lo anterior, se establece un mejor rendimiento en términos del consumo energético por parte de la versión *LoRa1278* de NiceRF y se continúa con esta versión para realizar las pruebas de máximo alcance en condiciones *Indoor* y *Outdoor*.

Consumo energético en modo transmisión (3 Bytes)

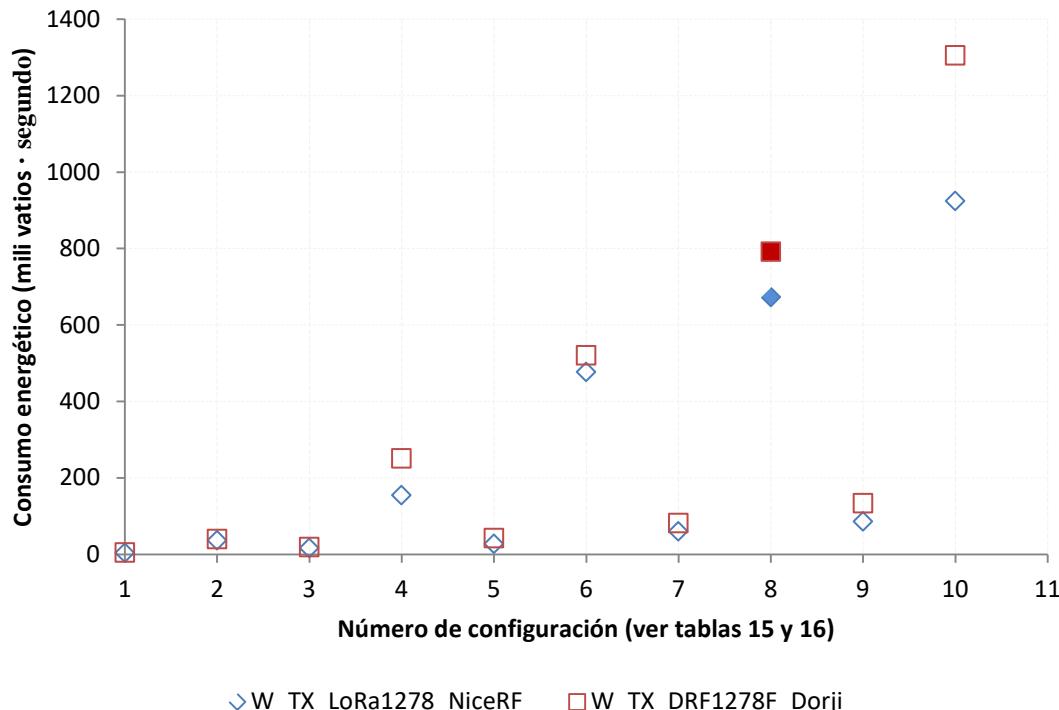


Figura 31. Representación gráfica del consumo energético de los transceivers DFR1278F y LoRa1278 para modo transmisión.

Fuente: Información recolectada por el autor

4.1.2.4 Pruebas de alcance para LoRa SX128 (LoRa1278 de NiceRF)

La configuración más estable y de mayor alcance para los dispositivos LoRa1278 de NiceRF detectada a partir de “Pruebas de consumo energético para LoRa1278 de NiceRF”, se presenta en la Tabla 17.

Tabla 17. Parámetros de configuración por software en los dispositivos SX1278 para obtener máximo alcance.

Fuente: Información recolectada por el autor

Frecuencia Freq (Mhz)		Ancho de banda BW (kHz)		Factor de alcance SF		Tasa de codificación de errores. CR	
Hardware	Software RADIO.freq	Hardware	Software RADIO.BW	Hardware	Software RADIO.sf	Hardware	Software RADIO.cr
433.800	433800000	41.7	BW41_7	12 → 4096 chips / symbol	SF_12	4/5	CR_4_5

De acuerdo a los parámetros de la *Tabla 17*, se encuentra lo siguiente:

- Entre menor sea el BW, mayor tiempo tomara la comunicación y esto se debe a que la frecuencia es inversamente proporcional al tiempo. Sin embargo entre menor sea la frecuencia, mayor será el alcance de transmisión esperado.
- El valor de SF o factor de alcance determina el rendimiento en la transmisión de los datos, es decir que entre mayor sea este valor, el dispositivo tendrá menor probabilidad de recibir datos errados y mayor radio de cobertura.
- El CR asegura la fiabilidad de los datos. Entre mayor sea este valor, mayor será la fiabilidad de los datos, pero con una sobrecarga en el tiempo de transmisión.

La configuración presentada en la *Tabla 17*, asegura el máximo alcance y una fiabilidad de la información trasmisita/recibida razonable. Sin embargo estas características favorables para la comunicación, implican un tiempo de transmisión mayor y por ende un mayor consumo energético.

4.1.2.4.1 Pruebas de alcance “Indoor” para LoRa SX128 (LoRa1278 de NiceRF)

En términos de distancia o máximo alcance para transmisión y recepción de datos *Indoor*, se realizaron los cálculos a partir de planos de las instalaciones de la Escuela Técnica Superior de Ingenieros en Telecomunicación. El punto de partida fue el laboratorio B-301 Robolabo y el punto de máximo alcance o final corresponde al sótano del almacén de mantenimiento del edificio A de la ETSIT. Esta representación se observa en la *Figura 32*.

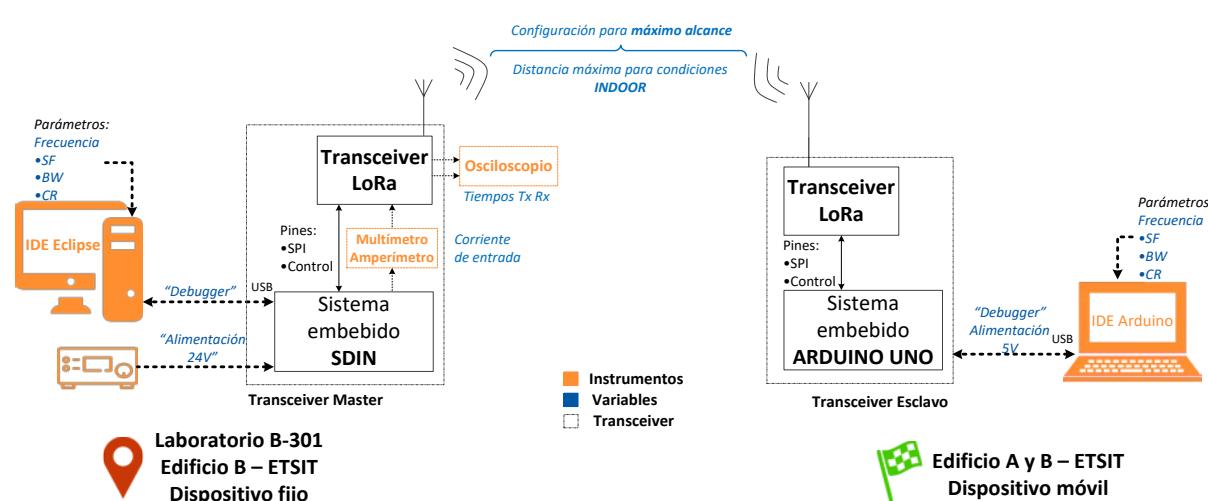


Figura 32. Esquema propuesto para pruebas de alcance Indoor.

Fuente: Esquema desarrollado por el autor



Figura 33. Máximo alcance en transmisión/recepción de datos para LoRa SX1278 (plano XY).

Fuente: Google earth 7.1.5.1557 y planos compartidos por la ETSIT. Escala 1:30

La Figura 34 presenta los planos XY e YZ de referencia para realizar el cálculo del máximo alcance en las instalaciones de la ETSIT. Para determinar la distancia desde el punto de partida hasta el punto de llegada o de máximo alcance, se realizan cálculos a partir de cálculos trigonométricos. En la Figura 34 (izquierda), se presenta la máxima distancia alcanzada con referencia del suelo, esta distancia se calculó utilizando el teorema de Pitágoras con referencia a las distancias de los ejes X (27.96m) y Y (167.73m).

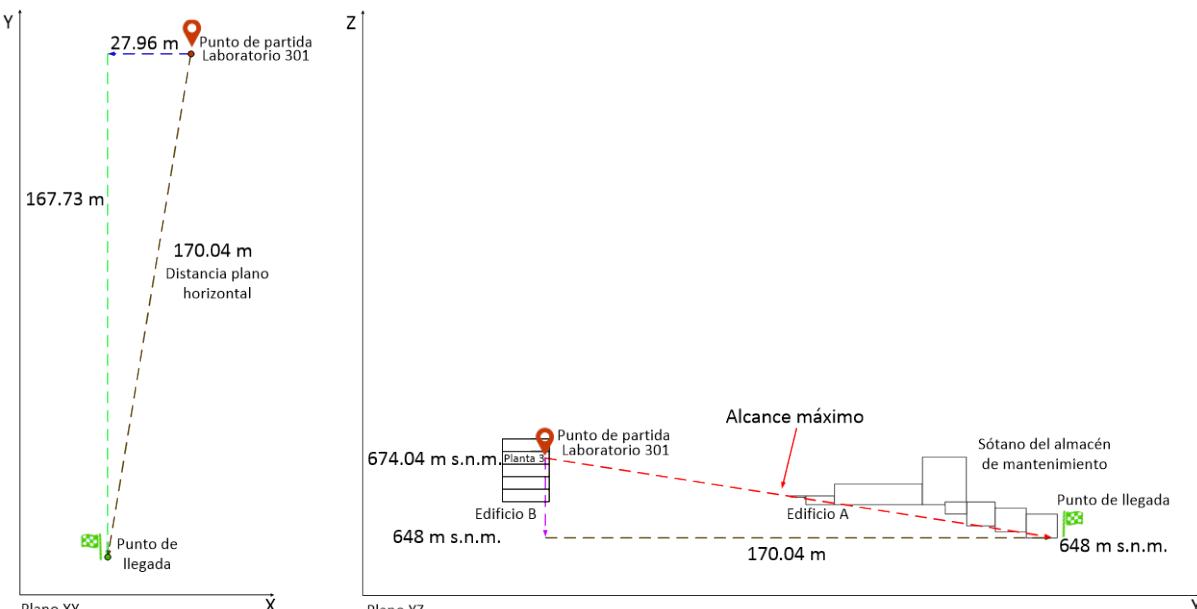


Figura 34. Esquema para determinar el alcance en transmisión/recepción de datos para LoRa SX1278 (plano XY e YZ).

Fuente: Información obtenida por el autor y planos compartidos por la ETSIT

Para determinar el máximo alcance con referencia a los transceivers, se utiliza la *Figura 34* (derecha). La diferencia de altura entre los dispositivos se calcula utilizando el modelo (8):

$$h_{entre\ dispositivos} = h_{dispositivo\ maestro} - h_{dispositivo\ esclavo} \quad (8)$$

$$h_{entre\ dispositivos} = 674.04 - 648 \quad (9)$$

$$h_{entre\ dispositivos} = 26.04\text{ m} \quad (10)$$

Finalmente, utilizando el teorema de Pitágoras se calcula la distancia de alcance máximo lograda por la configuración presentada en la *Tabla 17* por el dispositivo LoRa SX1278:

$$distancia_{máx\ alcance} = \sqrt{h_{entre\ dispositivos}^2 + distancia_{referencia\ suelo}^2} \quad (11)$$

$$distancia_{máx\ alcance} = \sqrt{26.04^2 + 170.04^2} \quad (12)$$

$$distancia_{máx\ alcance\ SX128} = 172.03\text{ m} \quad (13)$$

El máximo alcance logrado para los dispositivos SX1278 con la configuración que permite obtener el mejor rendimiento en términos de alcance (*Tabla 17*) es de 172 metros en aplicaciones *Indoor*.

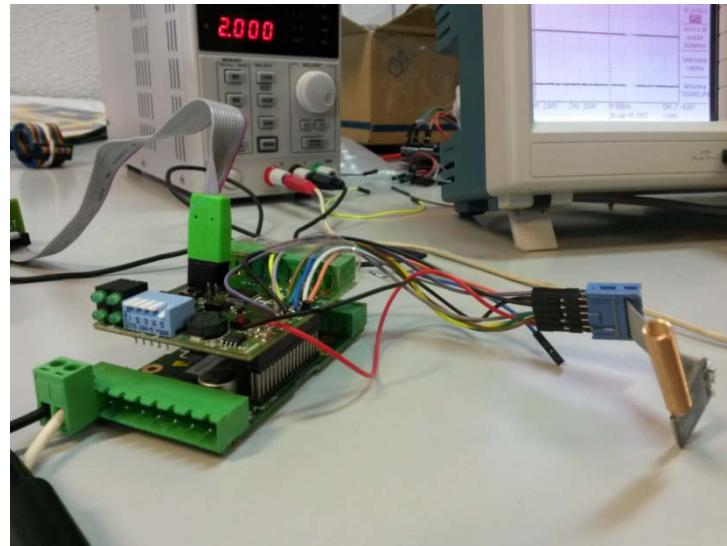


Figura 35. Dispositivo maestro conformado por sistema embebido SDIN y transceiver SX1278 en el laboratorio B-301 de la ETSIT.

Fuente: Fotografía tomada por el autor

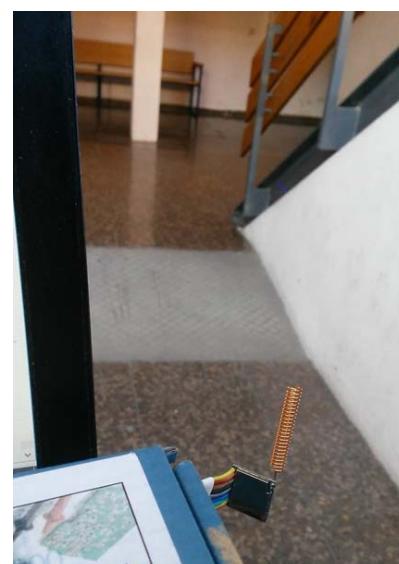


Figura 36. Dispositivo esclavo conformado por sistema embebido Arduino Uno y transceiver SX1278 en sótano del edificio A de la ETSIT.

Fuente: Fotografía tomada por el autor

4.1.2.4.2 Pruebas de alcance “Outdoor” para LoRa SX128 (LoRa1278 de NiceRF)

Las pruebas de configuración a máximo alcance para espacios *Outdoor* (con línea de vista u obstáculos como vegetación en espacios amplios) se desarrollaron en el *parque Dehesa de la Villa*, espacio muy próximo a las instalaciones de la ETSIT – UPM como se observa en la Figura 24.

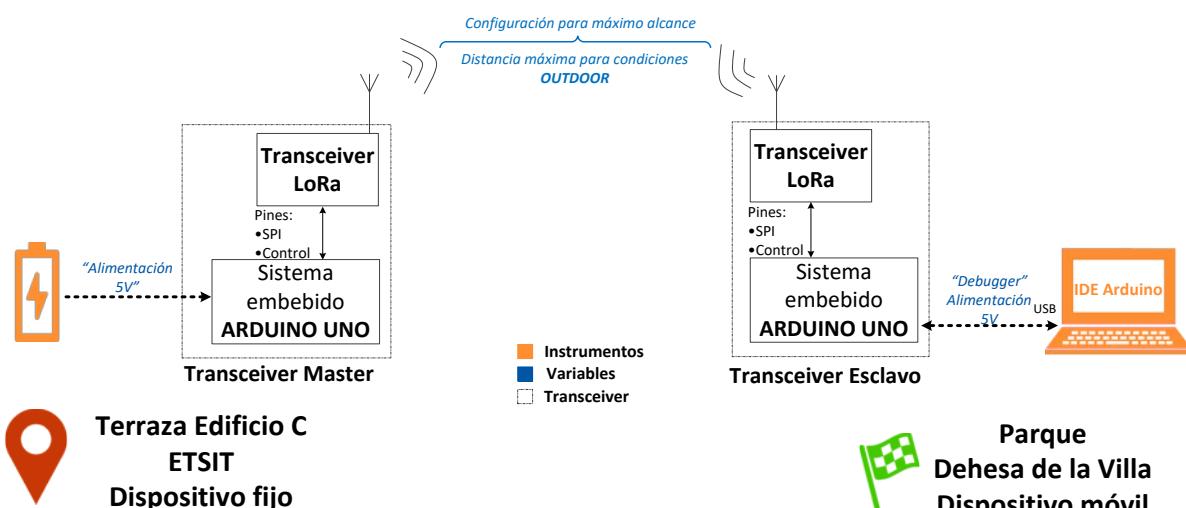


Figura 37. Esquema propuesto para pruebas de alcance “Outdoor”.

Fuente: Esquema desarrollado por el autor

La Figura 37 presenta el esquema de pruebas de máximo alcance para espacios Outdoor. El dispositivo maestro utilizado para el desarrollo de la prueba, estuvo compuesto por un sistema embebido Arduino UNO conectado al transceiver LoRa1278 de NiceRF y alimentado por una batería de 5V. Este dispositivo fue ubicado en la terraza del edificio C de la ETSIT como se observa en la Figura 38 (dispuesto como dispositivo de ubicación fija).



Figura 38. Dispositivo maestro conformado por sistema embebido Arduino Uno, transceiver SX1278 y batería de 5V en la terraza del edificio C de la ETSIT.

Fuente: Fotografía tomada por el autor

Como dispositivo esclavo se utilizó la tarjeta Arduino UNO como sistema embebido, conectado al transceiver LoRa1278 de NiceRF y alimentado por un equipo de cómputo portátil utilizando la salida de voltaje del puerto USB (5 voltios). La tarjeta Arduino Uno se encontraba continuamente conectada al equipo de cómputo para el suministro de voltaje y para comprobar la efectiva comunicación de los datos con el uso de la herramienta “*Serial Monitor*” del IDE Arduino. Este dispositivo se desplazó en el *parque Dehesa de la Villa* debido a que es el espacio abierto más cercano a las instalaciones de la ETSIT y para establecer condiciones similares a las que se encuentran en aplicaciones del ámbito del desarrollo (ver *Figura 39*).



Figura 39. Dispositivo esclavo conformado por sistema embebido Arduino Uno, transceiver SX1278 en el parque Dehesa de la Villa.

Fuente: Fotografía tomada por el autor

La *Figura 40* presenta el máximo alcance (en amarillo) para espacios *Outdoor* determinado con el uso de los transceivers SX1278. El esquema presenta el recorrido realizado (en azul) para encontrar el punto de máximo alcance de transmisión/recepción de datos y determinar la línea de vista generada entre los transceivers ubicados en la terraza del edificio C de la ETSIT y el parque Dehesa de la Villa.



Figura 40. Máximo alcance en espacio “outdoor” utilizando transceivers SX1278.

Fuente: Google earth 7.1.5.1557

Utilizando *google earth* con la herramienta “perfil de elevación”, se estableció la posición y la diferencia de elevación para calcular la distancia en línea recta o línea de vista. El resultado se presenta en la *Figura 41*.

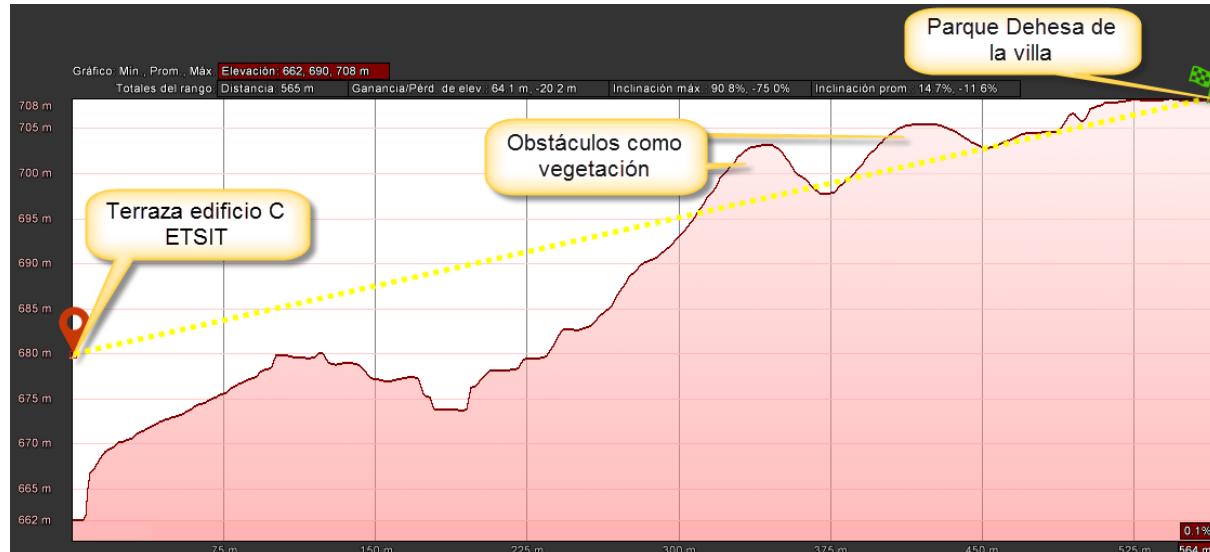


Figura 41. Perfil de elevación entre la terraza del edificio C y el punto de máximo alcance en el parque Dehesa de la Villa.

Fuente: Google earth 7.1.5.1557

Utilizando los datos suministrados por el “perfil de elevación” de la Figura 41, se realizan los cálculos para establecer la distancia de máximo alcance. La distancia sobre el plano horizontal es de 564 metros y la diferencia de altura entre los transceivers es de 28 metros.

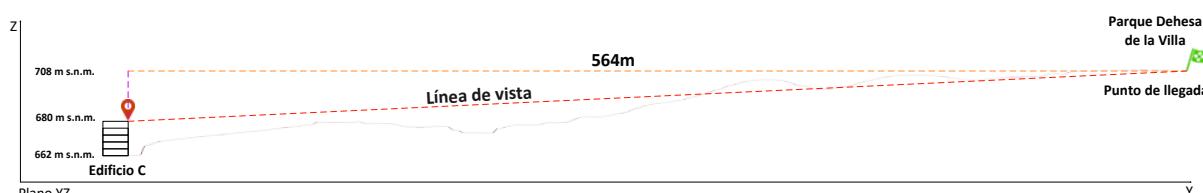


Figura 42. Esquema para determinar el alcance en transmisión/recepción de datos para LoRa SX1278 (plano YZ).

Fuente: Información del autor y perfil de elevación de google earth 7.1.5.1557

Para determinar el máximo alcance con referencia a los transceivers se utiliza la Figura 44. La diferencia de altura entre los dispositivos se calcula utilizando el modelo (14):

$$h_{entre\ dispositivos} = h_{dispositivo\ esclavo} - h_{dispositivo\ maestro} \quad (14)$$

$$h_{entre\ dispositivos} = 708 - 680 \quad (15)$$

$$h_{entre\ dispositivos} = 28\text{ m} \quad (16)$$

Finalmente, utilizando el teorema de Pitágoras se calcula la distancia de alcance máximo lograda por la configuración presentada en la *Tabla 17* por el dispositivo LoRa SX1278:

$$\text{distancia}_{\text{máx alcance}} = \sqrt{h_{\text{entre dispositivos}}^2 + \text{distancia}_{\text{referencia suelo}}^2} \quad (17)$$

$$\text{distancia}_{\text{máx alcance}} = \sqrt{28^2 + 564^2} \quad (18)$$

$$\text{distancia}_{\text{máx alcance SX1278}} = 565 \text{ m} \quad (19)$$

El máximo alcance logrado para los dispositivos SX1278 con la configuración que permite obtener el mejor rendimiento en términos de alcance (*Tabla 17*) es de 565 metros en aplicaciones *Outdoor*.

4.1.3 Pruebas de dispositivos LoRa con diseño de red tipo estrella

Finalmente se aumentó la complejidad del diseño creando una red estrella con un nodo central (circuito maestro) y 3 dispositivos esclavos. Como circuitos esclavos, se utilizan diversas versiones de sistemas embebidos (Arduino Uno, Arduino DUE y SDIN). Cabe resaltar que los dispositivos LoRa deben ser de la misma referencia, estar configurados a la misma frecuencia y con los parámetros SF, BW y CR idénticos. Estos parámetros se presentan en la *Tabla 16*.

El modem LoRa es *half-duplex*, es decir que los LoRa no puede transmitir o recibir datos en el mismo instante; esto significa que los transceiver LoRa deben ser configurados para transmitir o recibir datos. Cuando se incorpora más de dos dispositivos en un protocolo de comunicación *half dúplex*, se deben establecer estrategias que mitiguen los efectos adversos por colisión indeseada de datos cuando se recibe información.

Si los dispositivos no cuentan con estrategias para evitar la transmisión de datos sobre el mismo canal de comunicación por parte de los dispositivos esclavos en un mismo instante, los datos llegarán errados al dispositivo maestro y la comunicación de datos será incoherente.

La solución para evitar la colisión de datos, corresponde a establecer tiempos de muestreo y tiempos aleatorios de “des-sincronización” cuando se trasmiten datos; es decir evitar la colisión de datos con los tiempos adicionales de transmisión.

Una vez se establece el tiempo en el que no se presentan colisiones, se almacena dicho valor para ser utilizado consecutivamente en las próximas trasmisiones de datos y con lo anterior el sistema *half dúplex* se comporta sincronizado. En el documento Anexo A: “Desarrollo del driver para el uso de los dispositivos de radio LoRa con sistemas embebidos” – apartado “*Ejemplo de aplicación: recepción de datos de varios dispositivos remotos (red punto – multipunto)*”, se describe de manera detallada el proceso de implementación de la prueba desarrollada para crear la red punto-multipunto con dispositivos LoRa.

PRINCIPIOS DEL DESARROLLO DIGITAL EN APLICACIONES PROPUESTAS PARA EL ÁMBITO DEL DESARROLLO CON DISPOSITIVOS LoRa

Teniendo en cuenta que las TIC son un medio y no el fin, una vez desarrollado el driver de uso de los dispositivos LoRa junto con las parámetros de máximo alcance y consumo energético, se plantean algunas aplicaciones para el ámbito del desarrollo con el uso de los dispositivos LoRa:

Agua y saneamiento: Una de las principales funciones del agua es cubrir las necesidades básicas de las personas para su ingesta: agua potable apta para el consumo humano, saneamiento básico, higiene personal, alimentación, producción de alimentos a escala familiar y limpieza del entorno (Pérez & Jiménez, 2011). Basados en la necesidad de dotar a las poblaciones de agua potable, existen diversos programas que han contado con gran acogida debido a los impactos producidos en los beneficiarios. Entre estos programas se encuentran las cisternas de placa ampliamente difundidas en Brasil. Estos programas buscan a través de la captación de agua de los tejados, almacenar el agua en cisternas y sobrellevar las épocas más secas del año. Con el uso de sensores de nivel de agua, se podría realizar la monitorización del estado actual de las cisternas en términos de capacidad de almacenamiento en tiempo real y buscar una redistribución de este recurso hídrico.

Energía: El acceso a la energía está directamente relacionado con el bienestar de las personas. La energía es necesaria para cocinar, para la iluminación, para la conservación de los alimentos, climatización y servicios básicos que cualquier hogar debería tener cubiertos para asegurar unas condiciones mínimas de confort (García & Mundó, 2014).

La energía solar con el uso de paneles solares se ha convertido en una solución rentable para las zonas aisladas (alejadas de las redes de distribución eléctrica) como un mecanismo eficaz para suplir el déficit energético. Se propone el uso de redes LoRa con el uso de piranómetros para monitorizar la producción energética respecto a la eficiencia de las instalaciones fotovoltaicas.

Producción alimenticia- agricultura: La agricultura se podría considerar como una de las estrategias más importantes para desarrollar los pilares de los Objetivos de Desarrollo Sostenible. Entre estos pilares se tienen la erradicación de la pobreza extrema, combatir la desigualdad y la injusticia, y dar soluciones al cambio climático. Según la FAO, el crecimiento agrícola es en promedio dos veces más eficaz para reducir la pobreza y cinco veces más eficaz que el impulsado por otros sectores (FAO, 2012). Ahora bien, al promover estrategias y metodologías óptimas como la aplicación de la agricultura de conservación, la rotación de los cultivos, la modificación genética de las semillas, la monitorización y seguimiento continuo a los cultivos; este impacto podría aumentar significativamente. Se propone utilizar las redes LoRa para monitorizar diversas variables de los cultivos como PH, humedad, temperatura y de esta manera optimizar el suministro de agua o hacer seguimiento a plagas u otros factores de interés.

La agricultura es fundamental para los sectores menos favorecidos, por cuanto “más del 75% de los pobres viven en zonas rurales y dependen en gran medida de la agricultura” (FAO - ODS, 2015).

Los anteriores ejemplos de aplicación con el uso de redes LoRa deben ajustarse a los “*Principios para el Desarrollo Digital*” que son un conjunto de criterios utilizados en la implementación de programas basados en tecnología. Tienen el propósito de informar, pero no dictar, el diseño de programas de desarrollo (Digital principles, s. f.):

1. Diseñe con el usuario

Desarrolle soluciones adecuadas al contexto conforme a las necesidades del usuario. Esto hace referencia a establecer la mejor configuración de los transceiver LoRa acorde a las necesidades en términos de alcance de transmisión necesario y consumo energético deseado. Las *Tabla 15* y *Tabla 16* presentan esta relación para diferentes configuraciones.

2. Comprenda el ecosistema

Alinéese con políticas tecnológicas, legales y normativas vigentes. Debido a que el estándar LoRa es abierto, es posible utilizar frecuencias abiertas para la comunicación entre los dispositivos. Se recomienda utilizar bandas de frecuencias ISM que cumplan con el libre uso de acuerdo a la normativa de cada país. La relación de frecuencias de libre uso se presenta en la *Tabla 2*.

3. Diseñe para multiplicar la eficiencia

Diseñe con el fin de multiplicar la eficiencia desde el principio, evalúe y mitigue las dependencias que podrían limitar la capacidad de escalar. Una de las ventajas del estándar LoRa es ser inalámbrico. Esto implica la independencia de la infraestructura cableada y la escalabilidad del sistema en caso de ser necesaria. En caso de ser necesario el aumento en la cantidad de sensores y dispositivos transceivers, se realizarían cambios a nivel de configuración de software (el hardware podría mantenerse sin modificaciones), permitiendo la escalabilidad del sistema.

4. Siente las bases para la sustentabilidad

Planifique el desarrollo de sustentabilidad desde el principio, incluida la planificación de salud financiera a largo plazo, por ejemplo, al evaluar el costo total de propiedad. Una de las ventajas de los chip radio LoRa es su precio asequible (el par de transeivers tiene un costo de 17€ en julio de 2016). El protocolo LoRaWAN está respaldado por la LoRa Alliance (creado en 2015), lo que repercute en el respaldo y seguridad en el soporte de esta tecnología.

5. Cuente con un servicio basado en datos

Utilice información en tiempo real para monitorear e informar decisiones de gestión en todos los niveles. De acuerdo a la configuración y tiempo de muestreo personalizado por el usuario

en los LoRa, los datos trasmisidos pueden ser almacenados y estudiados para verificar tendencias que influyan en la mejora de toma de acciones de la sociedad beneficiaria.

6. Utilice datos abiertos, estándares abiertos, código abierto, innovación abierta
Desarrolle software de código abierto por defecto con el código disponible para repositorios públicos y comunidades respaldadas por desarrolladores. El código fuente e instrucciones propuestas en el Anexo A: “Desarrollo del driver para el uso de los dispositivos de radio LoRa con sistemas embebidos”, se encuentran documentadas y fue basadas en código abierto, probado con las tarjetas Arduino (hardware abierto, ver apartado “*Sistemas embebidos - importancia del hardware libre*”) y SDIN (sistema embebido desarrollado por GEDIRCI). Además los IDE utilizados para la implementación del driver, se desarrollaron utilizando software libre (ver apartado “*Descripción del software utilizado- Importancia del software libre*”). Por lo tanto:

- El estándar LoRaWAN es un *estándar abierto*.
- Los IDE utilizados para el desarrollo del driver son *software libre*.
- El hardware utilizado para las pruebas y propuesto para el desarrollo de aplicaciones es *hardware libre*.
- El driver desarrollado se encuentra documentado con el fin de compartir la información de forma gratuita y colaborar con diferentes desarrolladores (ver <https://github.com/Robolabo>).

7. Reutilice y mejore.

Utilice, modifique y extienda herramientas, plataformas y marcos existentes cuando sea posible. El código e instrucciones propuestas en el Anexo A: “*Desarrollo del driver para el uso de los dispositivos de radio LoRa con sistemas embebidos*” se encuentra documentadas y fue basado en código abierto, reutilizado y optimizado para brindar mayor independencia al usuario en términos de configuración y funcionamiento. Además fue desarrollado el anexo B: “*Instalación y uso del driver LoRa para sistemas embebidos*” que presenta de forma detallada la estructura del driver, junto con el proceso de instalación y uso para que desarrolladores puedan utilizar el driver en diversas aplicaciones.

8. Aborde la privacidad y la seguridad

Tenga en cuenta el contexto y las necesidades de privacidad de la información de identificación personal en el diseño de soluciones y mitigue en consecuencia. Debido a que es un estándar, el protocolo se encuentra protegido según se presenta en las especificaciones técnicas presentadas en la Tabla 18. Sin embargo, el desarrollador podrá generar métodos de encriptación de datos de acuerdo con la configuración de las tramas de datos trasmisidos.

9. Sea Cooperativo

Documente trabajos, resultados, procesos y mejores prácticas, y compártalos ampliamente. El desarrollo del driver para uso con microcontroladores de los LoRa fue documentado

utilizando las especificaciones técnicas del fabricante con el uso de diagramas de flujo y comentarios explícitos del por qué y para qué se realizan las instrucciones.

Se documentó el proceso de configuración, uso y aplicación de estos transceivers según el documento Anexo A: “*Desarrollo del driver para el uso de los dispositivos de radio LoRa con sistemas embebidos*”. Este proceso busca que cualquier usuario pueda configurar dispositivos de radio sin la necesidad de contar con conocimientos avanzados en desarrollo de software o en la interpretación de hojas de especificaciones técnicas; y busca ser un tutorial de configuración con la posibilidad de ajustarse según la necesidad (alcance, consumo energético, frecuencia utilizada, entre otros).

CONCLUSIONES

Las TIC o tecnologías de la información, corresponden al conjunto de tecnologías que permiten la adquisición, almacenamiento y manipulación de la información. Se encuentran presentes en diferentes áreas de la sociedad siendo un elemento transversal como su uso y aplicación en el gobierno, en la participación de la sociedad, salud, educación, medio ambiente, entre otras áreas. Por ser considerado un elemento transversal en el desarrollo, se considera como apoyo, respaldo o una herramienta.

En el año 2001, el PNUD dedicó su informe anual haciendo referencias importantes respecto a los adelantos tecnológicos como herramienta al servicio del desarrollo humano. El informe presenta relaciones y vínculos entre la innovación tecnológica y la brecha digital con el desarrollo humano.

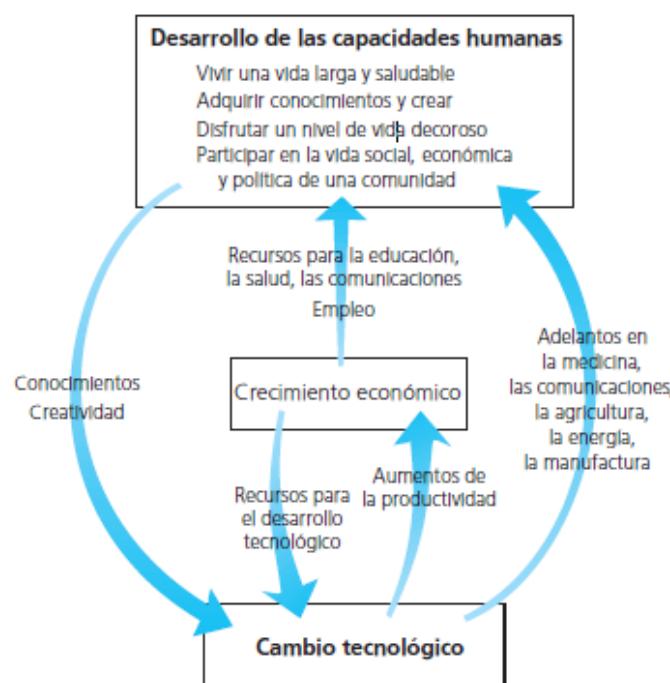


Figura 43. Vínculo entre las tecnologías y el desarrollo humano.

Fuente: PNUD (2001)

De acuerdo a la *Figura 43*, las tecnologías influyen en el desarrollo humano debido a que las innovaciones aumentan la capacidad de las personas gracias a sus aportaciones en sectores prioritarios como la salud, servicios básicos y la educación; y se constituye en un medio para generar crecimiento económico gracias a la incidencia de los cambios tecnológicos.

El auge de las Tecnologías de la Información y las Comunicaciones en el siglo XXI ha permitido almacenar grandes volúmenes de información debió a la dinámica de creación

intencionada o no de información por parte de usuarios de diferentes tecnologías, llámense teléfonos móviles, rastros de posición GPS, equipos de cómputo, cámaras de seguridad, diferentes sensores, entre otras. Esta información proviene en su mayoría de datos generados en internet. Con la aparición de internet se revolucionó la vida de los seres humanos por ofrecer la posibilidad de compartir conocimientos, experiencias y facilitar la comunicación entre personas y organizaciones de diversas formas.

Sin embargo, internet no solo ha permitido conectar personas, sino dispositivos, sensores y complejos sistemas. Es así como aparece el término “Internet de las cosas” o “internet de los objetos” o “IoT” que corresponde a la interconexión de diversos dispositivos a internet que generan datos en tiempo real.

Lo importante del “internet de las cosas” no radica en la conexión de las “cosas” a internet, sino en la información y conocimiento que estas puedan aportar para mejorar la calidad de las personas.

El IoT se compone por una colección dispersa de redes diferentes y con distintos fines, por lo que se considera como la red de redes.

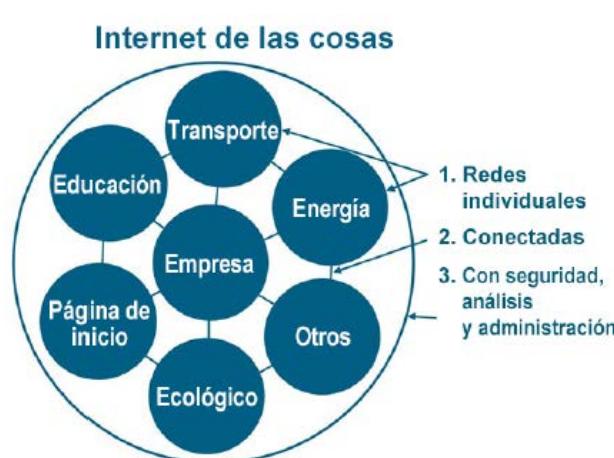


Figura 44. IoT: la red de redes.

Fuente: Cisco IBSG (2011)

Según Cisco IBSG (2011), se encuentran en marcha proyectos IoT que prometen cerrar la brecha de desigualdad, mejorar la distribución de los recursos en el mundo para quienes más los necesitan y ayudar a comprender el planeta. Sin embargo, existen barreras que amenazan con retrasar el desarrollo de IoT, como la transición del protocolo IPv4 a IPv6, el establecimiento de un conjunto de normas en común, el desarrollo de fuentes propias de energías para sensores diminutos y la optimización en el consumo energético en las redes de comunicación entre los objetos.

Respecto a la optimización del consumo energético de las redes, este es un importante campo de estudio del IoT debido a que se requieren considerables consumos energéticos para estar conectados a internet. Es así como aparecen proyectos relacionados con “Chirp Networks” o

redes alternativas que consumen significativamente menos energía que Wi-Fi o Bluetooth. La clave en la reducción energética consiste en lograr desarrollar protocolos de comunicación que varíen la modulación de la frecuencia para transmitir información, reduciendo la potencia y por ende mejorando el consumo energético. Un ejemplo claro de la solución en reducción de consumo energético con la variación del ancho de banda y el aumento en el radio de cobertura son las redes *LPWAN Low power wide area network o red de área extendida de bajo consumo* (Link Labs, 2016). La comparación grafica del ancho banda requerido para la comunicación con respecto al radio de cobertura se presenta en la *Figura 45*.

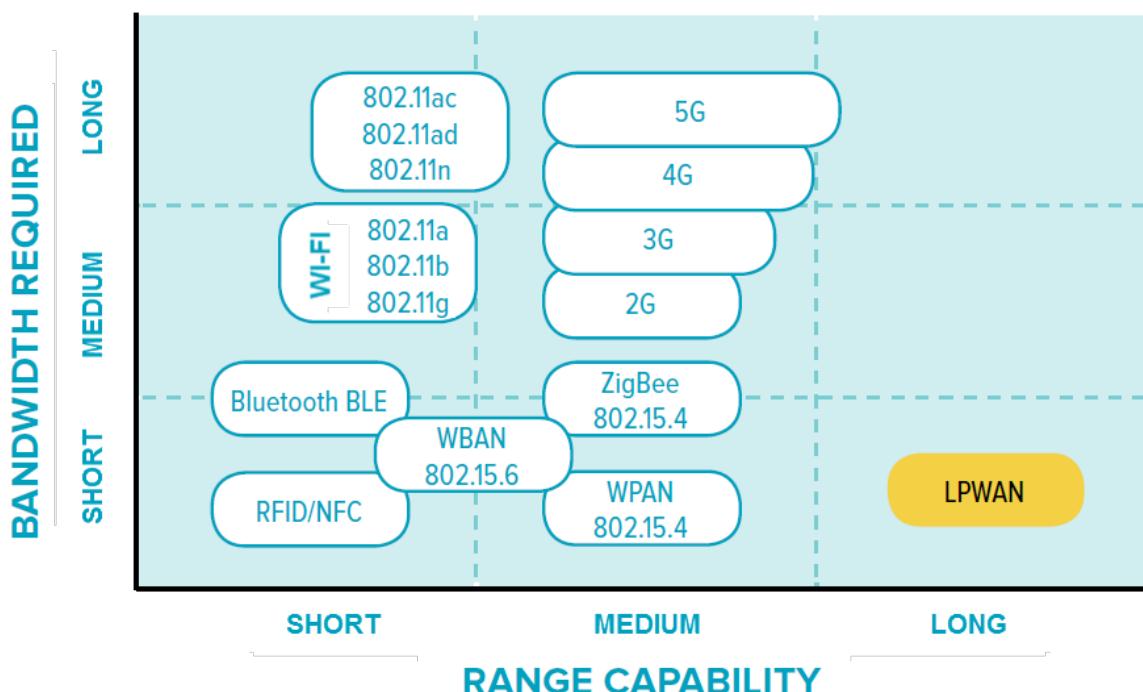


Figura 45. Comparación del ancho de banda requerido (frecuencia) con respecto al rango de cobertura para los principales estándares inalámbricos.

Fuente: Link Labs (2016)

Si bien este protocolo (LPWAN) aparece como respuesta a las tendencias y necesidades del mercado del IoT, es necesario adaptar su uso para aplicaciones del ámbito del desarrollo debido a que el uso de la tecnología apropiada puede ser la clave en el aseguramiento de los servicios básicos que sustentan redes de infraestructuras como agua potable, comunicaciones, electricidad, saneamiento, entre otros; mejorando la calidad de la vida de las personas y en especial las que se encuentran más aisladas de la población. Sin embargo, esta tecnología debe ser ajustada al contexto, características y población beneficiada: “*En la era de las redes, cada país necesita contar con capacidad para comprender las tecnologías mundiales y adaptarlas a las necesidades locales*”, PNUD (2001).

Es así que se planteó el desarrollo de un driver para los dispositivos LoRa SX127X junto con una serie de pruebas para determinar el consumo energético y distancia acorde con

condiciones reales como *espacios Indoor* (obstáculos como infraestructuras y muros) y *espacio Outdoor* (obstáculos como vegetación) junto con características y particulares de los dispositivos LoRa plasmadas en este documento.

Los dispositivos LoRa, hacen parte de la red LoRaWAN, la cual es una red de área extensa (LPWAN: *Low power wide area network*) que proporciona bajo consumo energético, bajo costo y largo alcance de cobertura.

Este estándar facilita la interoperabilidad entre las cosas o dispositivos inteligentes sin necesidad de instalaciones locales complejas y brinda la libertad al usuario, desarrollador o empresas en el despliegue para IoT.

Actualmente existe la “LoRa Aliance”, la cual es una asociación abierta, sin fines de lucro e iniciada por la industria *Semtech* con la misión de estandarizar las redes de baja potencia de área extendida (LPWAN) desplegadas en todo el mundo para permitir la conexión con IoT, dispositivos máquina a máquina (M2M), Smart City y aplicaciones industriales. Los miembros de la Alianza colaborarán para impulsar el éxito global del protocolo de LoRa (LoRaWAN), mediante el intercambio de conocimiento y experiencia para garantizar la interoperabilidad entre los operadores en un estándar global abierto (LoRa Alliance, 2016).

Una de las principales razones por las cuales fue seleccionado el estándar LoRaWAN, corresponde a que es un estándar abierto e ideal para el uso con sensores (Link Labs, 2016), lo que implica que es adecuado para aplicaciones del ámbito del desarrollo.

Sin embargo, fue necesario explorar las especificaciones técnicas de otros estándares inalámbricos debido a que existen en los mercados estándares muy bien documentados y conocidos.

La *Tabla 18* presenta el cuadro comparativo de las tecnológicas inalámbricas LoRa, ZigBee, Bluetooth y Wi-Fi. Esta información se ha complementado con las pruebas de alcance desarrolladas en el apartado “*Pruebas de consumo energético y alcance para LoRa SX1278*”, junto con el *Anexo C: DataSheet LoRa SX1278*.

Se puede observar la superioridad del alcance nominal de los dispositivos LoRa con respecto a las demás tecnologías (hasta 5 veces mayor alcance que Wi-Fi). En cuanto a la máxima tasa de datos, la baja tasa de transmisión de datos de hasta 37.5 Kbps en los dispositivos LoRa los hace poco eficientes para aplicaciones en la que sea necesaria la transmisión de video o audio; es decir, el estándar LoRa es adecuado para la transmisión de datos de sensores (sensórica) que no necesiten control estricto en tiempo real; es decir, aplicaciones en la que el tiempo de respuesta no sea crítico. Respecto a la frecuencia de trabajo, el estándar LoRa utiliza la más baja frecuencia y esto tiene sentido debido a que la frecuencia es inversamente proporcional al tiempo; lo que implica que si la frecuencia se reduce, el tiempo de transmisión de datos aumenta (transmisión lenta). Sin embargo, al trabajar a tan bajas frecuencias, es posible utilizar las frecuencias ISM (Industrial, Scientific and Medical), las cuales son

intencionalmente libres para uso no comercial en aplicaciones de tipo industrial, científico y médico (ver *Tabla 2*: Banda de frecuencia ISM compatibles con los transceivers LoRa).

Tabla 18. Cuadro comparativo de tecnologías inalámbricas LoRa, ZigBee, Bluetooth y Wi-Fi.

Fuente: Jin-Shyan, Yu-Wei & Chung-Chou (2007). Información basada en las pruebas desarrolladas por el autor

Característica	LoRa	ZigBee	Bluetooth	Wi-Fi
Especificación IEEE	802.15.4g	802.15.4	802.15.1	802.11a/b/c
Frecuencia de trabajo	137 – 1020 MHz	868/915 MHz; 2.4 GHz	2.4 GHz	2.4 GHz; 5 GHz
Ancho de banda de canales BW	7.8 – 500 kHz	0.3/0.6 MHz; 2 MHz	1 MHz	22 MHz
Máxima tasa de datos	0.018 – 37.5 Kb/s	250 Kb/s	1 Mb/s	54 Mb/s
Alcance nominal	172 m (Indoor)* 565 m (Outdoor)*	10 -100 m	10 m	10 -100 m
Máxima potencia nominal de TX	Hasta 20 dBm	(-25) - 0 dBm	0 - 10 dBm	15 - 20 dBm
Protección de datos (Error de redundancia cíclica)	16-bit CRC	16-bit CRC	16-bit CRC	32-bit CRC
Tiempo de conexión	Menos de un segundo*	Menos de 30 ms	Entre 3 - 5 segundos	Máximo 10 segundos
Tipo de modulación	FSK/OOK/LoRa™	BPSK (+ ASK), O-QPSK	GFSK	BPSK, QPSK COFDM, CCK, M-QAM
Consumo energético	Muy bajo*	Bajo	Intermedio	Alto
Tipo de comunicación	Half dúplex	Half dúplex	Full dúplex	Full dúplex
Tipo de red	LPWAN Red de área ampliada de bajo consumo	WPAN Red de área personal	WPAN Red de área personal	WLAN Red de área local
Topologías de red	Punto a punto, punto a multipunto, estrella, malla	Punto a punto, punto a multipunto, estrella, malla, arbol	Scatternet (propio de Bluetooth)	Punto a hub ESS malla
Aplicaciones	Automatización, Domótica, sistemas de alarmas inalámbricas, monitorización y control industrial, sistemas de irrigación, sensórica*	Dispositivos portátiles, Domótica (control y monitorización), automatización, juguetes, sistemas de irrigación, sensórica	Dispositivos portátiles (periféricos como auriculares, teclados, mouse, entre otros) y algunos dispositivos industriales de comunicación	Se utiliza para acceder a Internet y para interconexión de computadoras

*Información obtenida a partir de las pruebas de alcance y consumo energético del presente documento.

La comparación evidencia que el estándar LoRa es el de menor consumo energético con respecto a los estándares ZigBee, Bluetooth y Wi-Fi. Sin embargo, el estándar LoRa presenta como principal desventaja el utilizar comunicación *half dúplex*; es decir, que los dispositivos pueden transmitir o recibir datos en un instante. Cuando se crean complejas redes de comunicación, el utilizar comunicación *half dúplex* resulta ser una limitante si no se implementan estrategias adecuadas para efectuar la correcta comunicación. En el documento Anexo A: “*Desarrollo del driver para el uso de los dispositivos de radio LoRa con sistemas embebidos*” – apartado “*Ejemplo de aplicación: recepción de datos de varios dispositivos remotos (red punto - multipunto)*”, se describe de manera detallada estrategias para utilizar el tipo de comunicación *half dúplex* en complejas redes punto-multipunto para sensores con dispositivos LoRa.

Con respecto a las redes cableadas para aplicaciones del ámbito del desarrollo, el uso de medios cableados aumenta los costos de implementación debido al uso de cables (si se requieren largas distancias) y por ende se aumenta el costo de mantenimiento en la red. En algunos casos se hace necesario utilizar cables blindados para evitar la interferencia del ruido electromagnético (mejorar la calidad del cableado). Además, para asegurar la correcta transmisión de los datos se deben aumentar los niveles de voltaje, por lo que en algunos casos no estarían acordes como tecnologías para el desarrollo.

Los transceiver LoRa debido a la adaptación según uso que deseé darle el usuario, el bajo consumo energético y el alto alcance de cobertura, resulta ser un estándar idóneo para diferentes áreas enmarcadas en el desarrollo.

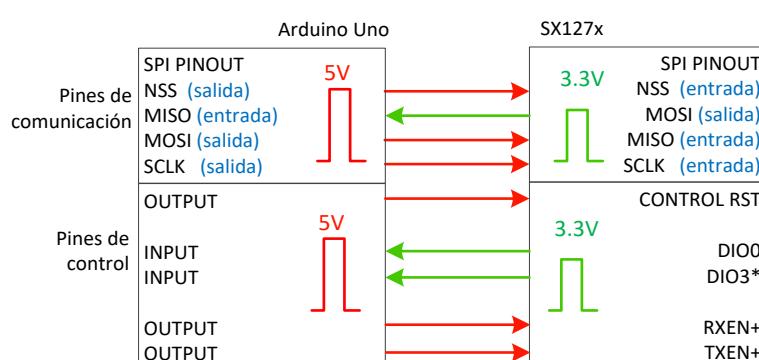
Sin embargo, *el uso de estos dispositivos no debe ser tomado como un fin sino como un medio y deben adaptarse a las necesidades de los países en desarrollo*.

De acuerdo con los argumentos y pruebas desarrolladas en el presente documento, se confirma la hipótesis planteada en el apartado *Paso 2: Formular hipótesis e interrogantes* con lo que se confirma la hipótesis: “*De acuerdo a las características de bajo consumo energético, bajo costo de adquisición, fácil configuración y alto alcance de transmisión, los dispositivos LoRa resultan ser idóneos para aplicaciones del ámbito del desarrollo como mecanismo de comunicación entre sensores y dispositivos de adquisición de datos*”.

RECOMENDACIONES

Estos dispositivos LoRa SX1272 y SX1278 fueron probados con tarjetas Arduino, las cuales tienen como especial característica el ser hardware libre, estar documentados en la web, contar con un costo de adquisición razonable, tener aceptación por parte de la comunidad de desarrolladores y posibilidad de implementar sistemas básicos de monitorización y control con el uso de sensores. En el apartado “*Sistemas embebidos - importancia del hardware libre*” se presenta la importancia del uso de hardware libre para el ámbito del desarrollo. “*El hardware de fuentes abiertas da libertad de controlar la tecnología y al mismo tiempo compartir conocimientos y estimular la comercialización por medio del intercambio abierto de diseños*” (Lazalde, Torres & Vila-Viñas, 2015).

Si bien Arduino es un referente importante al indagar sobre hardware abierto, durante las pruebas desarrolladas con los dispositivos LoRa se evidenciaron problemas particulares. El principal problema corresponde a los niveles incompatibles de voltaje del sistema embebido Arduino Uno y los transceivers. Arduino utiliza 5 voltios para pines de control y comunicación SPI (Torrente, 2013), mientras los transceiver LoRa utilizan 3.3 voltios para pines de control y comunicación (Semtech SX1272, 2015 & Semtech SX1278, 2015). Si bien las pruebas no se vieron perturbadas por la incompatibilidad con niveles de voltaje, si continuamente permanecen conectados estos dispositivos, el transceiver LoRa podría verse afectado y hasta destruirse por recibir más voltaje que el nominal necesario. Este planteamiento se representa en la *Figura 46*.



*No implementado en LoRa1278

+Implementado en LoRa1278

Figura 46. Comparación de los niveles de voltaje entre Arduino Uno y los transceiver LoRa.
Fuente: Imagen planteada por el autor a partir de Torrente (2013) & Semtech SX1278 (2015)

Sin embargo, este no fue el único problema evidenciado con el uso de la tarjeta Arduino Uno como sistema embebido. Al realizar las mediciones de los niveles de voltaje de alimentación en el transceiver LoRa con respecto a los modos transmisión y recepción, se evidencio un aumento de voltaje sobre el pin de alimentación del transceiver. El esquema de conexión utilizado se presenta en la *Figura 47*.

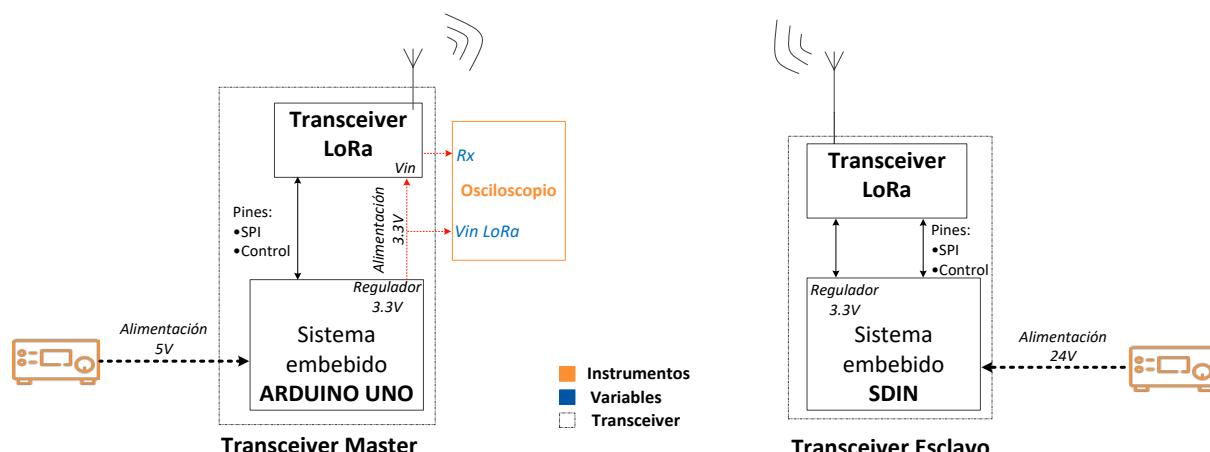


Figura 47. Esquema de conexión para visualizar el nivel de alimentación con respecto a los modos transmisión y recepción en los dispositivos LoRa.

Fuente: Esquema planteado por el autor

Cuando el dispositivo se encuentra en modo recepción de datos, existe un efecto de “sobre”-alimentación que pasa de los 3.3 voltios nominales de alimentación, hasta los 4.1 voltios. Esto ocurre para cualquiera de las configuraciones planteadas en la Tabla 15 y Tabla 16.

Según las especificaciones del fabricante de tarjetas Arduino Uno, el regulador utilizado como alimentación de 3.3 voltios corresponde a la referencia LP2985-33DBVR, el cual es un regulador de bajo ruido con voltaje de salida nominal de 3.3V y máxima corriente de suministro hasta de 150mA (Texas Instruments, 2004). Este regulador debería mantener el voltaje estable y no permitir el aumento o reducción en el voltaje. Sin embargo, como se observa en la Figura 48, existe un efecto nocivo de “sobre”-alimentación en los dispositivos LoRa que con el uso continuo con Arduino Uno, podría llegar a destruir el transceiver.

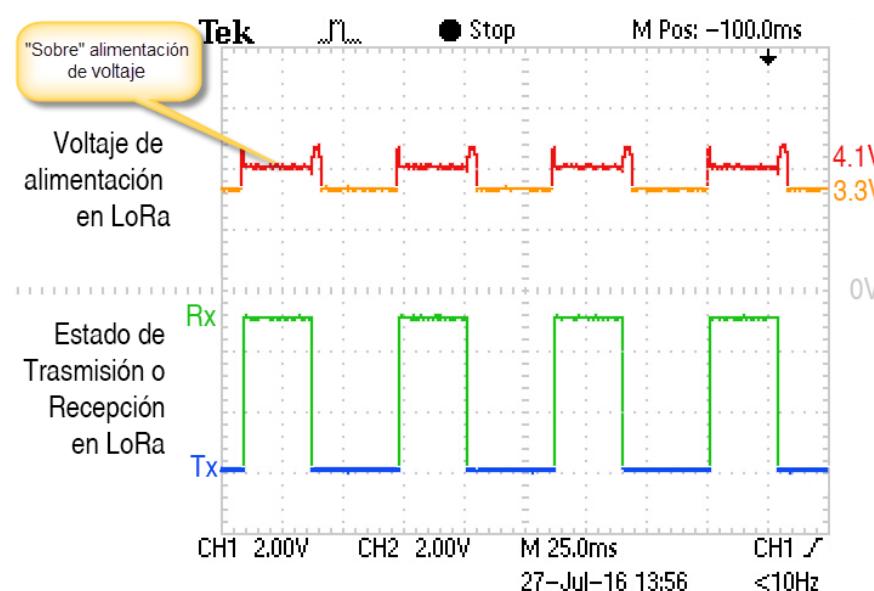


Figura 48. Diagrama de tiempos de la señal de alimentación con respecto a los modos transmisión y recepción en los dispositivos LoRa utilizando Arduino Uno.

Fuente: Captura de pantalla de osciloscopio Tektronix

Sin embargo, un efecto nocivo para los dispositivos LoRa también ocurrió cuando se utilizó Arduino DUE. Cuando el dispositivo se encuentra en modo transmisión, el suministro de voltaje se reduce a 2.0 voltios. Según las especificaciones del fabricante de tarjetas Arduino DUE, el regulador utilizado como alimentación de 3.3 voltios corresponde a la referencia *NCP1117ST33T3G*, el cual cuenta con un voltaje de salida nominal de 3.3V y máxima corriente de suministro de 1A (ON semiconductor, 2014). Este regulador debería mantener el voltaje estable y no permitir el aumento o reducción en el mismo. Sin embargo, como se observa en la *Figura 49*, existe un efecto nocivo de “reducción” en la *alimentación* de voltaje de los dispositivos LoRa, lo que podría llegar a dañar el transceiver.

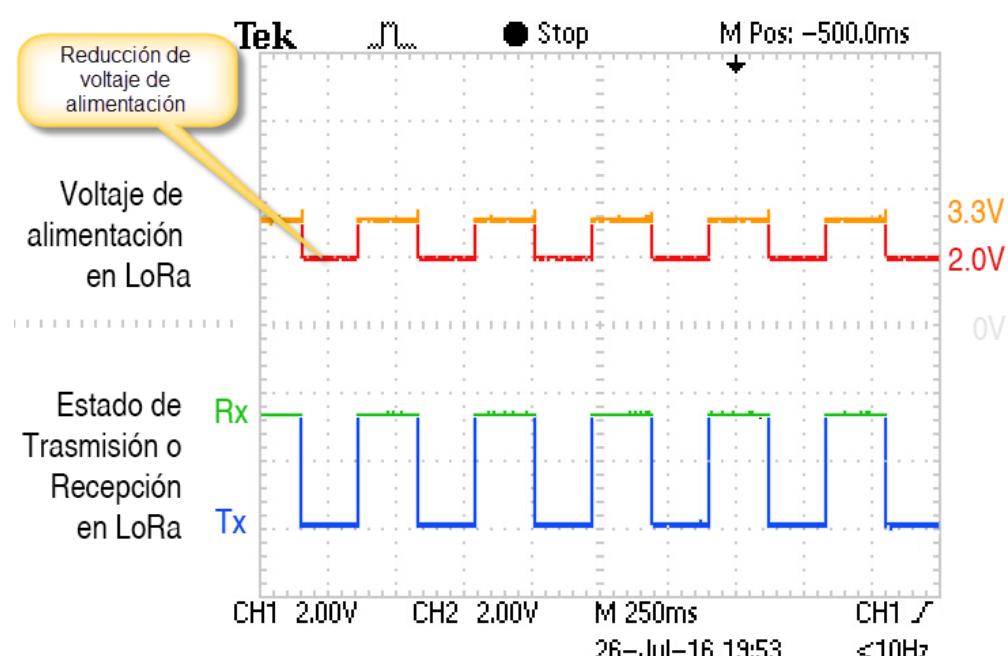


Figura 49. Diagrama de tiempos de la señal de alimentación con respecto a los modos transmisión y recepción en los dispositivos LoRa utilizando Arduino DUE.

Fuente: Captura de pantalla de osciloscopio Tektronix

Por los motivos expuestos anteriormente, se recomienda el uso de Arduino con dispositivos LoRa siempre y cuando se utilice una fuente de alimentación externa de 3.3V como alimentación de los dispositivos LoRa, evitando utilizar los reguladores de las tarjetas Arduino. Para realizar sistemas complejos y funcionales de monitorización con los transceiver LoRa en aplicaciones del ámbito del desarrollo, se recomienda el uso el módulo SDIN (utilizados durante las pruebas) o tarjetas de embebidas STMicroelectronics, específicamente los módulos de la familia STM32 con microcontroladores CORTEX M0, M3 o M4 de ultra bajo consumo, series L0, L1 y L4 respectivamente. La *Tabla 19* presenta algunos sistemas embebidos propuestos para la familia de STM32 a ultra bajo consumo energético.

Tabla 19. Tarjetas embebidas STMicroelectronics de ultra bajo consumo.

Fuente: ST (2016)

Tipo de Microcontrolador	STM32L0	STM32L1	STM32L4
Tarjeta de desarrollo ST	STM32L073RZ STM32L011K4 STM32L053R8 STM32L031K6 STM32L053C8 STM32L073VZ	STM32L152RE STM32L100RC STM32L152RC STM32L152VB STM32L152ZD	STM32L476RG STM32L432KC STM32L476VG STM32L476ZG

BIBLIOGRAFÍA

Administración publica Ecuador (s.f.). Importancia del Software Libre para un país. Secretaría Nacional de la Administración Pública de Ecuador. Recuperado de <http://www.administracionpublica.gob.ec/importancia-del-software-libre-para-un-pais/>. Ecuador. Consultado el 29 de julio de 2016.

Arduino Board Due (2016). STM32 MCU Eval Tools. Recuperado de http://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools.html?querycriteria=productId=SS1532, Suiza. Consultado el 29 de julio de 2016.

Arduino Board DUE (s.f.). Arduino DUE. Recuperado de <https://www.arduino.cc/en/Main/ArduinoBoardDue>, Italia. Consultado el 29 de julio de 2016.

Arduino Board UNO (s.f.). Arduino UNO & Genuino UNO. Recuperado de <https://www.arduino.cc/en/Main/ArduinoBoardUno>, Italia. Consultado el 29 de julio de 2016.

Barry, R. (2009). Using the FreeRTOS Real Time Kernel - A Practical Guide. FREERTOS. Estados Unidos

Bonilla, E. & Rodríguez, P. (1997). Más allá del dilema de los métodos. La investigación en ciencias sociales. 3^a Ed, Ediciones Uniandes, Colombia.

Chapman, S. (2012). Máquinas eléctricas. 5a edición. McGraw-Hill. México.

Cisco IBSG (2011). Internet de las cosas: cómo la próxima evolución de Internet lo cambia todo. Internet Business Solutions Group. Estados Unidos.

Digital principles (s. f.). Principios para el desarrollo digital. Recuperado de <http://digitalprinciples.org/>. Estados Unidos. Consultado el 30 de junio de 2016.

Dorji (2015). DRF1278F - 20dBm LoRa Long Range RF Front-end Module. Version 1.11. China.

FAO - ODS (2015). Cinco agentes de cambio para un mundo sostenible. Recuperado de <http://www.fao.org/post-2015-mdg/news/detail-news/es/c/281558/>, Italia. Consultado el 06 de diciembre de 2015.

FAO (1994). La comunicación clave para el desarrollo humano. Repositorio de documentos de la FAO. Recuperado de <http://www.fao.org/docrep/t1815s/t1815s01.htm>. Italia.

FAO (2012). EL ESTADO MUNDIAL DE LA AGRICULTURA Y LA ALIMENTACIÓN: invertir en la agricultura para construir un futuro mejor. Roma. Organización de las Naciones Unidas para la alimentación y la agricultura. Italia

Genbetadev (2014). Eclipse IDE. Recuperado de <http://www.genbetadev.com/herramientas/eclipse-ide>, Español. Consultado el 29 de julio de 2016.

GitHub (s.f.). Conociendo GitHub: ¿Qué es GitHub?. Recuperado de <http://conociendogithub.readthedocs.io/en/latest/data/introduccion/>, Estados Unidos. Consultado el 30 de septiembre de 2016.

Hernández, J. (s.f.). Recopilación de la información. Conceptos de estadística y su clasificación. Departamento de ciencias básicas. Instituto Tecnológico de Apizaco. México.

Herrera J. (2011). Programación en tiempo real y bases de datos: Un enfoque práctico. Universidad de Cataluña. España

Jin-Shyan, L., Yu-Wei, S. & Chung-Chou, S. (2007). A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi, The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECO5). Taiwán.

Lazalde, A., Torres, J. & Vila-Viñas, D. (2015). Hardware: ecosistemas de innovación y producción basados en hardware. Buen Conocer - FLOK Society, Ecuador.

Link Labs (2016). A COMPREHENSIVE LOOK AT Low Power, Wide Area Networks for ‘Internet of Things’ Engineers and Decision Makers, Estados Unidos

López, E. (s.f.). Protocolo RS-485. Ingeniería en microcontroladores. México.

López, E. (s.f.). Protocolo SPI (Serial Peripherical Interface). Ingeniería en microcontroladores. México

LoRa Alliance (2016). Wide Area Network for IoT. Recuperado de <https://www.lora-alliance.org>. Estados Unidos. Consultado el 30 de junio de 2016.

Maxim Integrated Products (2015). MAX13487E/MAX13488E Half-Duplex RS-485-/RS-422-Compatible Transceiver with AutoDirection Control. Estados Unidos.

Monje, C. (2011). Metodología de la investigación Cuantitativa y Cualitativa. Guía didáctica. Universidad Surcolombiana. Colombia.

García, M. & Mundó, J. (2014). La energía como derecho. Cómo afrontar la pobreza energética. Dossier Catalunya Social número 38. España..

NiceRF (2015). LoRa1278 100 mW 4 km larga distancia y alta sensibilidad (-139 dBm) 433 MHz módulo de transceptor inalámbrico. Recuperado de <http://es.aliexpress.com/item/2pcs-lot-LoRa1278-100mW-4km-Long-Distance-and-High-Sensitivity-139-dBm-433MHz-Wireless-Transceiver-Module/32461365864.html>, China. Consultado el 22 de junio de 2016.

OIE - IDIE (2010). Monitoreo e indicadores. Texto de apoyo al proceso deconstrucción de un Sistema Regional de Indicadores sobre Atención y Educación Inicial. Junta de Andalucía. Instituto para el Desarrollo y la Innovación Educativa. Nicaragua.

ON semiconductor (2014). NCP1117, NCV1117 1.0A Low-Dropout Positive Fixed and Adjustable Voltage Regulators. Datasheet. Estados Unidos.

Pedre, S. (2012). Sistemas embebidos. Laboratorio de Robótica y Sistemas Embebidos. Departamento de Computación - FCEN – UBA. Argentina.

Pérez-Foguet, A., Jiménez, A. (2011). El agua como elemento clave para el desarrollo. CanalEduca. España

PNUD (2001). Informe de desarrollo humano 2001: poner el adelanto tecnológico al servicio del desarrollo humano. Programa de las Naciones unidas para el Desarrollo. Estados Unidos.

PNUD (2003). Informe sobre desarrollo humano 2003. Los Objetivos de Desarrollo del Milenio: un pacto entre las naciones para eliminar la pobreza. Programa de las Naciones unidas para el Desarrollo. Estados Unidos.

Ramírez, R. (2015). Sistemas de radiocomunicaciones. Paraninfo, ciclos formativos. Sistemas de Telecomunicación e Informáticos. España.

Rojas, R. (1989). Guía para realizar investigaciones sociales, Plaza y Valdés. México.

Semtech (2013). SX1272 Development kit. User guide. Semtech Corporation. Estados Unidos

Semtech (s.f.). LoRa® Product Family. Recuperado de <http://www.semtech.com/wireless-rf/lora.html>, Estados Unidos. Consultado el 29 de agosto de 2016.

Semtech SX1272 (2015). SX1272/73 - 860 MHz to 1020 MHz Low Power Long Range Transceiver. Datasheet. Semtech Corporation. Estados Unidos.

Semtech SX1278 (2015). SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver. Datasheet. Semtech Corporation. Estados Unidos.

Solar Decathlon Europe (2014). Módulos SDIN – Manual de desarrollo. Equipo de monitorización. Francia

Spradley, J. (1980). Observación Participante. Waveland Press Inc. Estados Unidos.

ST (2016). STM32 MCU Eval Tools. Recuperado de http://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools.html?querycriteria=productId=SS1532, Suiza. Consultado el 29 de julio de 2016.

Tam, J., Vera, G. & Oliveros, R. (2008). Tipos, métodos y estrategias de investigación. Pensamiento y Acción 5, pp. 145-154. Perú.

Texas Instruments (2004). LP2985 150-mA Low-noise Low-dropout Regulator with Shutdown. Datasheet. Estados Unidos.

Texas Instruments (2005). SM-Band and Short Range Device Regulatory Compliance Overview. Estados Unidos.

Torrente, O. (2013). ARDUINO: Curso práctico de formación. RC libros. España.

Velásquez, J. (s.f.). Los sensores en la producción. Universidad Ricardo Palma. Perú.

Weik, M. (1996). Communications Standar Dictionary. 3a edición. Chapman & Hall. Estados Unidos.



POLITÉCNICA



ANEXO A

Desarrollo del driver para el uso de los dispositivos de radio LoRa con sistemas embebidos.

JOSÉ DANIEL RODRÍGUEZ MUNCA

Tutor Académico: Manuel Sierra Castañer

Tutor Profesional: Álvaro Gutiérrez Martín

**UNIVERSIDAD DE POLITÉCNICA DE MADRID
UNIVERSIDAD COMPLUTENSE DE MADRID**

**MASTER INTERUNIVERSITARIO EN ESTRATEGIAS Y TECNOLOGÍAS PARA
EL DESARROLLO
MADRID - ESPAÑA
2016**

TABLA DE CONTENIDO

CONFIGURACIÓN A NIVEL DE HARDWARE	107
Lectura y escritura de registros SPI.....	113
Modos de operación de los LoRa	114
CONFIGURACIÓN POR SOFTWARE DE LOS TRANSCEIVERS LORA	116
Comprobación de comunicación entre LoRa y MC	117
Registros de configuración SF, BW y CR de los transceivers LoRa	118
Registros de configuración de potencia del radio.....	129
Registros de configuración de frecuencia del radio.....	135
Configuración por software del radio lora para la transmisión datos.....	136
Configuración por software del radio lora para la recepción de datos	141
Ejemplo de aplicación: red punto a punto (aplicación ping-pong).....	147
Ejemplo de aplicación: recepción de datos de varios dispositivos remotos (red punto - multipunto)	150
BIBLIOGRAFÍA.....	160

LISTADO DE FIGURAS

<i>Figura 1. Principales características de comunicación SPI para los dispositivos LoRa seleccionados.</i>	107
<i>Figura 2. Configuración de pines de control para la tarjeta SX1272RF1 de Semtech.</i>	109
<i>Figura 3. Configuración de pines de control para la tarjeta DRF1278F de Dorji.</i>	110
<i>Figura 4. Configuración de pines de control para la tarjeta LoRa1278 de NiceRF.</i>	110
<i>Figura 5. Esquema de configuración de pines de control y SPI para el microcontrolador.</i> ..	111
<i>Figura 6. Diagrama de tiempos para iniciar los transceivers SX1272 y SX1278.</i>	112
<i>Figura 7. Flujograma propuesto para iniciar chip SX1272 y SX1278.</i>	112
<i>Figura 8. Flujograma propuesto para la lectura de registros en los radio SX1272 y SX1278.</i>	113
<i>Figura 9. Flujograma propuesto para la escritura de registros en los radio SX1272 y SX1278.</i>	113
<i>Figura 10. Configuración de los modos de operación en registro RegOpMode (0x01).</i>	115
<i>Figura 11. Configuración de modo LoRa en registro RegOpMode (0x01).</i>	116
<i>Figura 12. Esquema propuesto para inicio en la configuración del LoRa.</i>	117
<i>Figura 13. Comprobación de comunicación SPI entre el microcontrolador y el LoRa seleccionado.</i>	118
<i>Figura 14. Esquema para determinar el TS según BW y SF.</i>	123
<i>Figura 15. Esquema lógico de configuración de los registros RegModemConfig1_0x1D y RegModemConfig2_0x1E para SX1272 (parte I).</i>	124
<i>Figura 16. Esquema lógico de configuración de los registros RegModemConfig1_0x1D y RegModemConfig2_0x1E para SX1272 (parte II).</i>	125
<i>Figura 17. Esquema lógico de configuración de los registros RegModemConfig1_0x1D y RegModemConfig2_0x1E para SX1272 (parte III).</i>	126
<i>Figura 18. Esquema lógico de configuración de los registros RegModemConfig1_0x1D, RegModemConfig2_0x1E y RegModemConfig3_0x26 para SX1278 (parte I).</i>	127
<i>Figura 19. Esquema lógico de configuración de los registros RegModemConfig1_0x1D, RegModemConfig2_0x1E y RegModemConfig3_0x26 para SX1278 (parte II).</i>	128
<i>Figura 20. Esquema lógico de configuración de los registros RegModemConfig1_0x1D, RegModemConfig2_0x1E y RegModemConfig3_0x26 para SX1278 (parte III).</i>	129
<i>Figura 21. Esquema lógico de configuración de los registros RegLna_0x0C, RegPaConfig_0x09, RegOcp_0x0B y RegPaDac para los dispositivos LoRa (parte I).</i>	133
<i>Figura 22. Esquema lógico de configuración de los registros RegLna_0x0C, RegPaConfig_0x09, RegOcp_0x0B y RegPaDac para los dispositivos LoRa (parte II).</i>	134
<i>Figura 23. Esquema lógico de configuración de frecuencia para los radio LoRa.</i>	136
<i>Figura 24. Esquema lógico de almacenamiento de datos sobre la FIFO del LoRa.</i>	139
<i>Figura 25. Esquema lógico de configuración para transmisión de datos en el LoRa.</i>	140
<i>Figura 26. Esquema lógico propuesto para gestión por interrupción cuando se efectúa una transmisión de datos en el LoRa.</i>	141
<i>Figura 27. Esquema lógico de configuración en el LoRa para recepción de datos.</i>	142

<i>Figura 28. Esquema lógico de atención a la interrupción por recepción o transmisión de datos en el LoRa (parte I).....</i>	143
<i>Figura 29. Esquema lógico de atención a la interrupción por recepción o transmisión de datos en el LoRa (parte II).....</i>	144
<i>Figura 30. Esquema lógico de almacenamiento de datos sobre la FIFO del LoRa.....</i>	146
<i>Figura 31. Esquema lógico para dispositivo maestro utilizando radio LoRa (parte I).....</i>	147
<i>Figura 32. Esquema lógico para dispositivo maestro utilizando radio LoRa (parte II).</i>	148
<i>Figura 33. Esquema lógico para dispositivo esclavo utilizando radio LoRa (parte I).....</i>	148
<i>Figura 34. Esquema lógico para dispositivo esclavo utilizando radio LoRa (parte II).....</i>	149
<i>Figura 35. Principio de funcionamiento “ping-pong” para la realización de pruebas de alcance en dispositivos LoRa.</i>	150
<i>Figura 36. Tipología estrella para comunicación con diversos dispositivos.</i>	151
<i>Figura 37. Líneas de tiempos para dispositivos esclavos y dispositivo maestro sin estrategias adecuadas para la transmisión de información.</i>	152
<i>Figura 38. Líneas de tiempos para dispositivos esclavos y dispositivo maestro con tiempos aleatorios para la transmisión adecuada de datos.</i>	153
<i>Figura 39. Esquema lógico para dispositivo maestro en aplicación punto-multipunto (parte I).</i>	154
<i>Figura 40. Esquema lógico para dispositivo maestro en aplicación punto-multipunto (parte II).</i>	155
<i>Figura 41. Esquema lógico para dispositivos esclavo en aplicación punto-multipunto (parte I).</i>	156
<i>Figura 42. Esquema lógico para dispositivos esclavo en aplicación punto-multipunto (parte II).</i>	157
<i>Figura 43. Esquema lógico para dispositivos esclavo en aplicación punto-multipunto (parte III).</i>	158

LISTADO DE TABLAS

<i>Tabla 1. Comparación de pines de control según los fabricantes de tarjetas de los LoRa seleccionados.</i>	108
<i>Tabla 2. Descripción del uso de los pines RXEN y TXEN para la tarjeta LoRa1278 de NiceRF.</i>	108
<i>Tabla 3. Fotografías de los transceivers estudiados.</i>	109
<i>Tabla 4. Configuración de pines de control para los LoRa seleccionados.</i>	110
<i>Tabla 5. Descripción del registro RegOpMode (0x01).</i>	114
<i>Tabla 6. Descripción del registro modo de operación del LoRa.</i>	115
<i>Tabla 7. Descripción del registro RegModemConfig1_0x1D del LoRa SX1272.</i>	119
<i>Tabla 8. Descripción del registro RegModemConfig2_0x1E del LoRa SX1272.</i>	120
<i>Tabla 9. Descripción del registro RegModemConfig1_0x1D del LoRa SX1278.</i>	120
<i>Tabla 10. Descripción del registro RegModemConfig2_0x1E del LoRa SX1278.</i>	121
<i>Tabla 11. Descripción del registro RegModemConfig3_0x26 del LoRa SX1278.</i>	122
<i>Tabla 12. Descripción del registro RegLna_0x0C del LoRa.</i>	130
<i>Tabla 13. Descripción del registro RegPaConfig_0x09 del LoRa.</i>	131
<i>Tabla 14. Descripción del registro RegPaConfig_0x09 del LoRa.</i>	131
<i>Tabla 15. Descripción del registro RegOcp_0x0B del LoRa.</i>	132
<i>Tabla 16. Descripción del registro RegPaDac* del LoRa.</i>	132
<i>Tabla 17. Banda de frecuencia ISM compatibles con los transceivers de radio LoRa.</i>	135
<i>Tabla 18. Descripción del registro de configuración de frecuencia en los radio LoRa.</i>	135
<i>Tabla 19. Descripción del registro RegIrqFlagsMask_0x11 para habilitación de interrupciones.</i>	137
<i>Tabla 20. Descripción del registro RegIrqFlags_0x12 para la detección de interrupciones por software.</i>	137
<i>Tabla 21. Configuración de pines de control con el registro RegDioMapping1_0x40 para el uso con interrupciones en los pines DIO0-3.</i>	138
<i>Tabla 22. Configuración de pines de control con el registro RegDioMapping2_0x41 para el uso con interrupciones en los pines DIO4-5.</i>	138

CONFIGURACIÓN A NIVEL DE HARDWARE

Los parámetros de configuración de los transceivers LoRa se establecen modificando registros internos del chip mediante el protocolo de comunicación síncrono SPI.

A nivel lógico de sincronización y transmisión de datos, la comunicación SPI requiere la configuración de la polaridad del reloj (CPOL) y el bit fase del reloj (CPHA).

Los dispositivos de radio LoRa SX1272/78 utilizan CPOL=0 y CPHA=0; lo que se conoce como modo 0 SPI. El bit más significativo (MSB) de un byte enviado debe ser el primero y la velocidad del SCLK no debe superar los 10MHz (Semtech-SX1278, 2015).

El esquema propuesto de conexión entre los transceiver LoRa y el microcontrolador (sistema embebido) se presenta en la *Figura 1*. Los dispositivos de radio LoRa se comportan como esclavos y el microcontrolador será el maestro en la comunicación SPI.

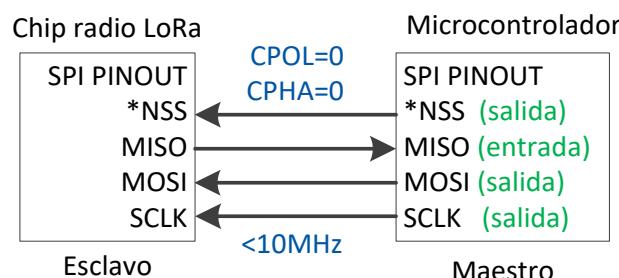


Figura 1. Principales características de comunicación SPI para los dispositivos LoRa seleccionados.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

Algunos fabricantes realizan tarjetas de adecuación para facilitar el manejo de los dispositivos de radio LoRa. Al realizar estas tarjetas, se eliminan o incorporan pines físicos de control respecto a las características originales del chip. La siguiente tabla presenta las tarjetas propuestas para realizar el driver de los radio SX1272/78. Los fabricantes mantienen los pines de comunicación SPI y eliminan algunos pines de control del chip. Las principales características de las tarjetas LoRa adquiridas para realizar las pruebas de funcionamiento y desarrollo del driver, se presentan en la *Tabla 5*.

Tabla 1. Comparación de pines de control según los fabricantes de tarjetas de los LoRa seleccionados.

Fuente: Semtech (2013), Dorji (2015) & NiceRF (2015)

Tarjeta	Chip radio	Marca	Pines de control físicos			
			DIO0 – DIO5	RXTX	Pines SPI	RESET
SX1272RF1	SX1272	Semtech	Pines de entrada/ salida, configurables por software	Pin de salida. Se coloca en “1” cuando el dispositivo está trasmitiendo datos	NSS MISO MOSI SCLK	Pin de entrada. Utilizado al iniciar o arrancar el dispositivo.
DRF1278F		Dorji				
LoRa1278	SX1278	NiceRF	Pines de entrada/ salida, configurables por software NI* DIO3–DIO5	NI*		

*NI: No implementado

La versión LoRa1278 de NiceRF realiza adecuaciones con la incorporación de 2 pines adicionales de control que permiten habilitar el uso de la antena para trasmisir o recibir. La descripción de las funciones de los pines TXEN y RXEN se presentan en la *Tabla 2*.

Tabla 2. Descripción del uso de los pines RXEN y TXEN para la tarjeta LoRa1278 de NiceRF.

Fuente: NiceRF (2015)

Estados lógicos de pines		Descripción de funcionamiento para la Antena
Pin TXEN	Pin RXEN	
0	0	Modo Sleep o bajo consumo
0	1	Configurada para Recibir
1	0	Configurada para Transmitir

Físicamente, la tarjeta SX1272RF1 de *Semtech* es de mayor tamaño y cuenta con un conector SMA para antenas a 868Mhz e impedancia de 50Ω. Las tarjetas DRF1278F y LoRa1278 tienen el mismo tamaño (significativamente más pequeño que la versión SX1272RF1) y permiten conectar antenas pasivas tipo primavera a 433Mhz con impedancia de 50Ω. Los tamaños de pueden observar en las *Tabla 3*.

Tabla 3. Fotografías de los transceivers estudiados.

Fuente: Fotografías tomadas por el autor



Las *Figuras 2, 3 y 4* presentan la conexión básica de los pines de control para las tres tarjetas LoRa propuestas y la configuración del microcontrolador en relación a pines de entrada o salida. La configuración de los pines es una de las tareas prioritarias al iniciar cualquier aplicación con sistemas embebidos.

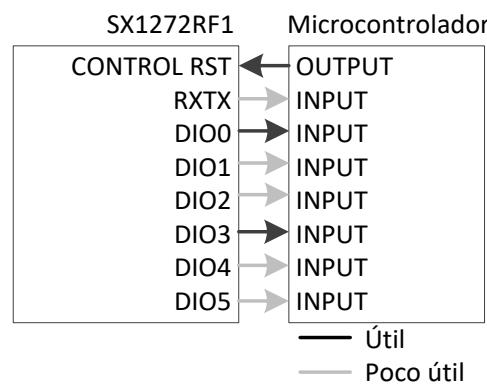


Figura 2. Configuración de pines de control para la tarjeta SX1272RF1 de Semtech.

Fuente: Semtech SX1272 (2015). Desarrollada por el autor

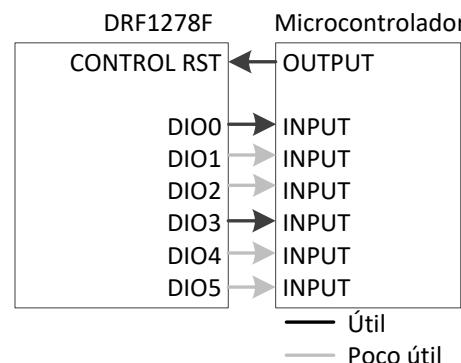


Figura 3. Configuración de pines de control para la tarjeta DRF1278F de Dorji.

Fuente: Dorji (2015). Desarrollada por el autor

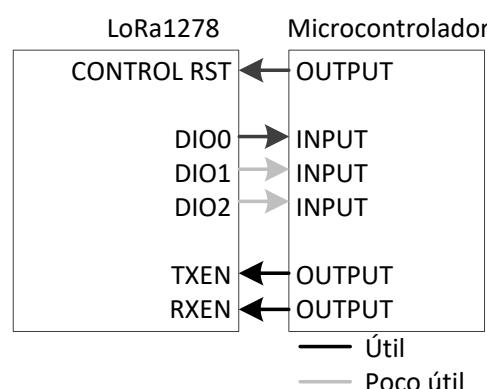


Figura 4. Configuración de pines de control para la tarjeta LoRa1278 de NiceRF.

Fuente: NiceRF (2015). Desarrollada por el autor

Los pines DIO de control más importantes de acuerdo la información suministrada por el fabricante, son los DIO0 y DIO3. La *Tabla 4* presenta las funciones de los pines de control de acuerdo a la configuración.

Tabla 4. Configuración de pines de control para los LoRa seleccionados.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015)

Conf. de mapeo	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
00	ModeReady	CadDetected	CadDone	FhssChangeChannel	RxTimeout	RxDone
01	ClkOut	PllLock	ValidHeader	FhssChangeChannel	FhssChangeChannel	TxDone
10	ClkOut	PllLock	PayloadCrcError	FhssChangeChannel	CadDetected	CadDone
11	Sin uso					

De acuerdo a la configuración, el pin DIO0 permite conocer si el chip radio LoRa ha transmitido correctamente los datos (TXDONE) o si el chip ha recibido datos y está a la espera de la gestión por parte del microcontrolador (RXDONE).

El DIO3 permite conocer si los datos llegaron correctamente de acuerdo con la Verificación por redundancia cíclica (CRC).

Teniendo en cuenta lo anterior, se plantea en la *Figura 5* el esquema de configuración de pines de control y pines SPI para el microcontrolador:

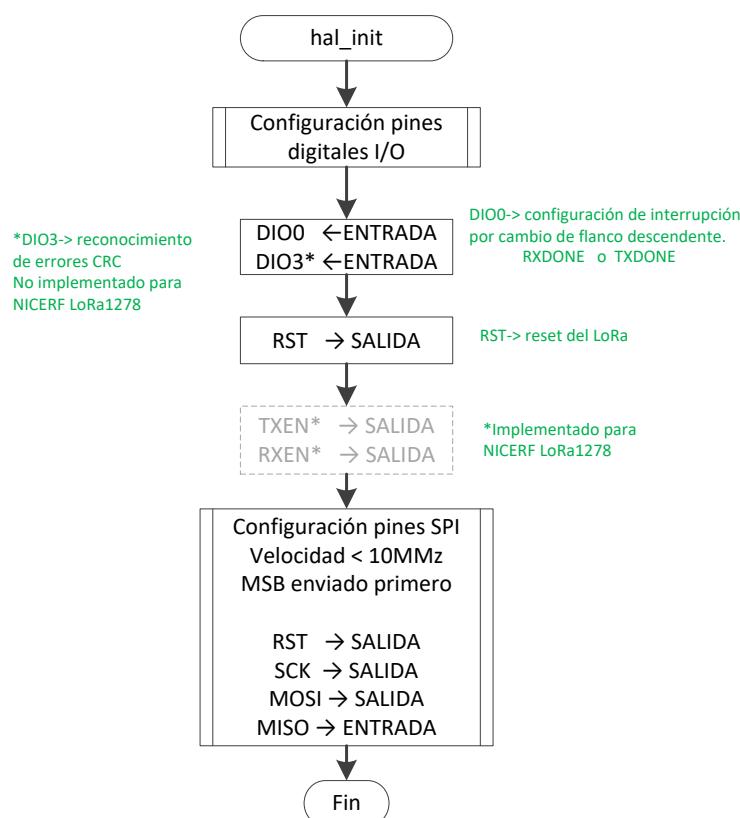


Figura 5. Esquema de configuración de pines de control y SPI para el microcontrolador.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

Una vez se tengan configurados los pines SPI y de control en el microcontrolador, el fabricante presenta el protocolo de inicio o arranque del chip LoRa para el SX1272 y para el SX1278 (ver *Figura 6*). La principal diferencia el estado del pulso para el reset del chip, para el SX1272 se realiza con un estado lógico “0”, mientras que para el SX1278 se efectúa con un estado lógico “1” durante 100 microsegundos. A nivel lógico, la *Figura 7* presenta el protocolo de inicio a nivel de hardware del chip LoRa.

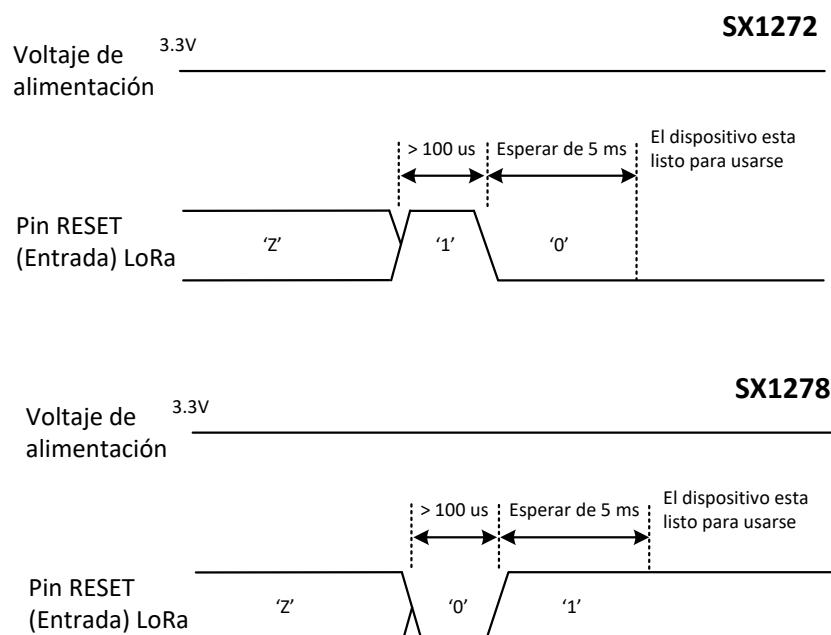


Figura 6. Diagrama de tiempos para iniciar los transceivers SX1272 y SX1278.
Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

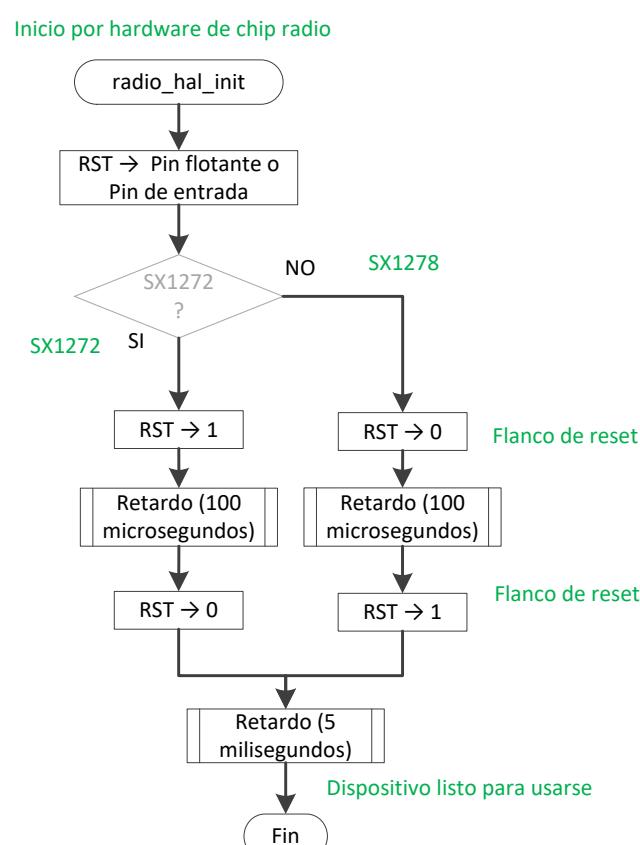


Figura 7. Flujograma propuesto para iniciar chip SX1272 y SX1278.
Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

Lectura y escritura de registros SPI

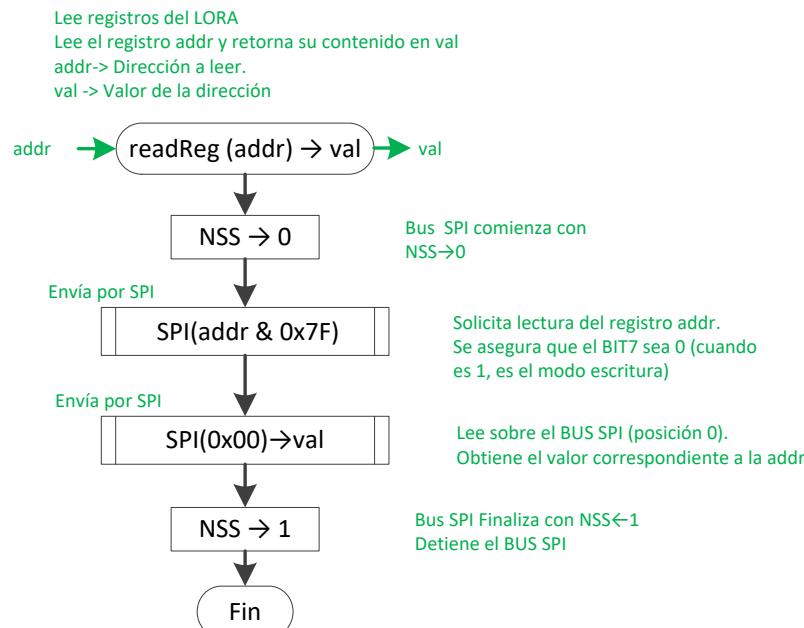


Figura 8. Flujograma propuesto para la lectura de registros en los radio SX1272 y SX1278.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

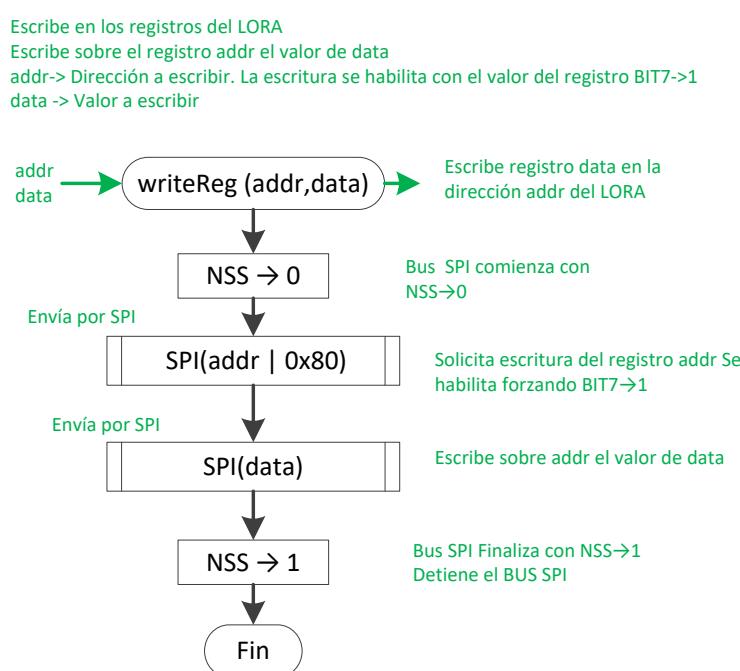


Figura 9. Flujograma propuesto para la escritura de registros en los radio SX1272 y SX1278.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

La lectura y escritura de registros en los radio LoRa se realiza utilizando el bus SPI (siempre desde los modos SLEEP o STANDBY). Para realizar la lectura, se escribe sobre el bus SPI la dirección del registro, forzando que el bit 7 sea “0” y luego se hace escritura con el valor “cero”, obteniendo el valor del registro.

Para realizar la escritura, se escribe sobre el bus SPI la dirección del registro forzando que el bit 7 (MSB) sea “1” y luego se hace la escritura del dato. Lo anterior se presenta en el flujo de las *Figuras 8 y 9*.

Modos de operación de los LoRa

El registro *RegOpMode_0x01* permite configurar el modo de operación de transceiver LoRa. Este registro se podría considerar como el más usado en cualquier aplicación. La descripción del registro *RegOpMode* se presenta en la *Tabla 5*.

Tabla 5. Descripción del registro RegOpMode (0x01).
Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015)

Bit	Nombre de la variable	Uso	Defecto	Descripción
7	LongRangeMode	r w	0x00	0 → FSK/OOK Modo 1 → LoRa™ Modo
6	AccessSharedReg	r w	0x00	0 → Acceso registros LoRa 1 → Acceso registros FSK
5-4	reserved	r	0x00	Reservado
3	LowFrequencyModeOn	r w	0x01	0 → Modo High Frequency Mode (Acceso registros HF) 1 → Modo Low Frequency Mode (Acceso registros LF)
2-0	Mode	r w	0x01	Modos del dispositivo 000 → SLEEP 001 → STDBY 010 → Frequency synthesis TX (FSTX) 011 → Transmit (TX) 100 → Frequency synthesis RX (FSRX) 101 → Receive continuous (RXCONTINUOUS) 110 → receive single (RXSINGLE) 111 → Channel activity detection (CAD)

Nota r: lectura. w: escritura. La variable **LowFrequencyModeOn** aplica para el SX1278 (SX1272 no está implementado)

Los modos de operación del LoRa más utilizados se presentan en la *Tabla 6* y se habilitan siguiendo el diagrama de la *Figura 10*. Los modos de operación utilizados para la configuración de los dispositivos LoRa son SLEEP y STANDBY.

Tabla 6. Descripción del registro modo de operación del LoRa.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015)

Modo	Descripción	Valor de 'mode' en RegOpMode
SLEEP	Es el modo de bajo consumo. Solo los registros SPI del LoRa tienen acceso en este modo. La FIFO no tiene acceso	0x00
STANDBY	El LoRa se coloca en este modo al transmitir o tener pendiente la gestión en la recepción de datos. Los registro SPI y de la FIFO del LoRa tienen acceso en este modo.	0x01
TX	Es el modo utilizado cuando se tienen configurada la interrupción por TX, datos en la FIFO y el LoRa está listo para trasmisir. Una vez transmite, el LoRa cambia automáticamente al modo STANDBY.	0x03
RXCONTINUOUS	Es el modo utilizado cuando se tienen configurada la interrupción por RX, el LoRa tiene datos en la FIFO y el chip está a espera de la gestión de los datos. Una vez el LoRa recibe los datos y valida los datos (CRC), el LoRa cambia automáticamente al modo STANDBY. Recibe varios bytes en una trama.	0x05

OPMODE_LORA	0x80
OPMODE_SLEEP	0x00
OPMODE_STANDBY	0x01
OPMODE_TX	0x03
OPMODE_RX	0x05

Escribe el modo 'mode' en el registro RegOpMode{01}

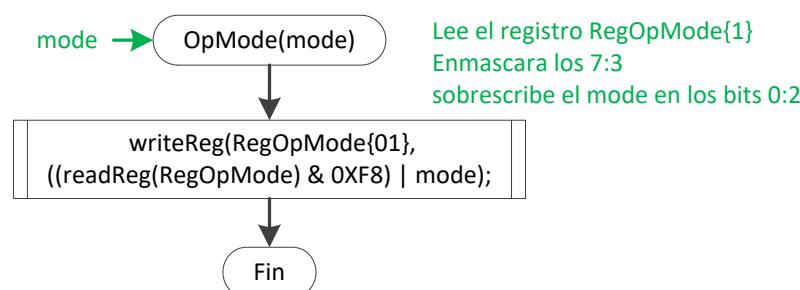


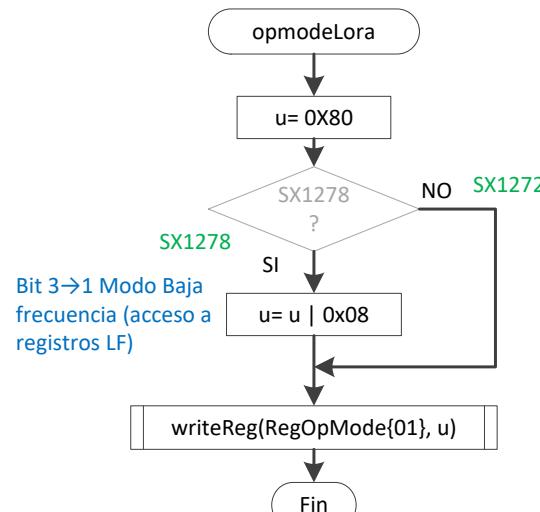
Figura 10. Configuración de los modos de operación en registro RegOpMode (0x01).

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

Para realizar cualquier configuración sobre los registros SPI del LoRa es necesario que el modo de operación pase de SLEEP a STANDBY.

Para usar el “modo LoRa”, se debe escribir inicialmente sobre el registro *RegOpMode_0X01* un “1” en el bit 7 que corresponde a la variable *LongRangeMode* (ver Tabla 6). Basta con habilitar dicho bit una sola vez al iniciar cualquier aplicación (ver Figura 11).

Escribe sobre el
RegOpMode{1} el Modo Lora Bit7 →1
y el modo SLEEP Bit2:0 → 000



Habilita el modo LoRa y SLEEP en el chip radio

Figura 11. Configuración de modo LoRa en registro RegOpMode (0x01).

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

CONFIGURACIÓN POR SOFTWARE DE LOS TRANSCEIVERS LORA

La *Figura 12* presenta el esquema lógico propuesto para el inicio en la configuración de los dispositivos LoRa. El conjunto de instrucciones puede ser ejecutado al alimentar el sistema embebido (microcontrolador) y los LoRa. El esquema lógico de instrucciones basta con ser ejecutado una sola vez por el microcontrolador. Las funciones *radio_Checkpoint_Version()*, *configLoraModem()*, *configChannel()* y *configPower()* se presentan en los siguientes apartados.

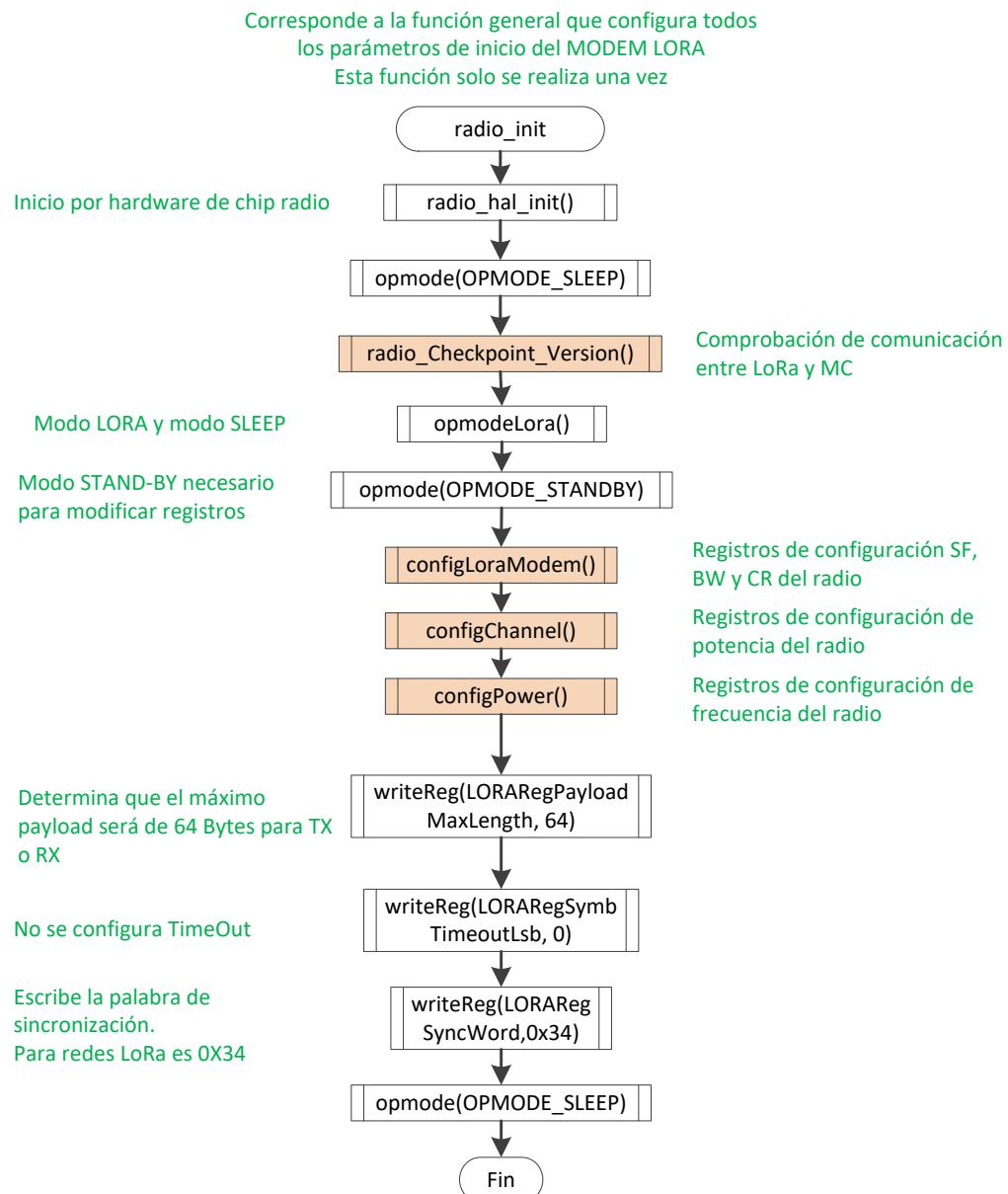


Figura 12. Esquema propuesto para inicio en la configuración del LoRa.
Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

Comprobación de comunicación entre LoRa y MC

Se recomienda realizar la comprobación de la versión del transceiver LoRa. Esto permitirá reconocer si la comunicación entre el microcontrolador y el chip radio LoRa es correcta y si se está configurando el dispositivo correctamente. La *Figura 13* presenta el esquema lógico propuesto.

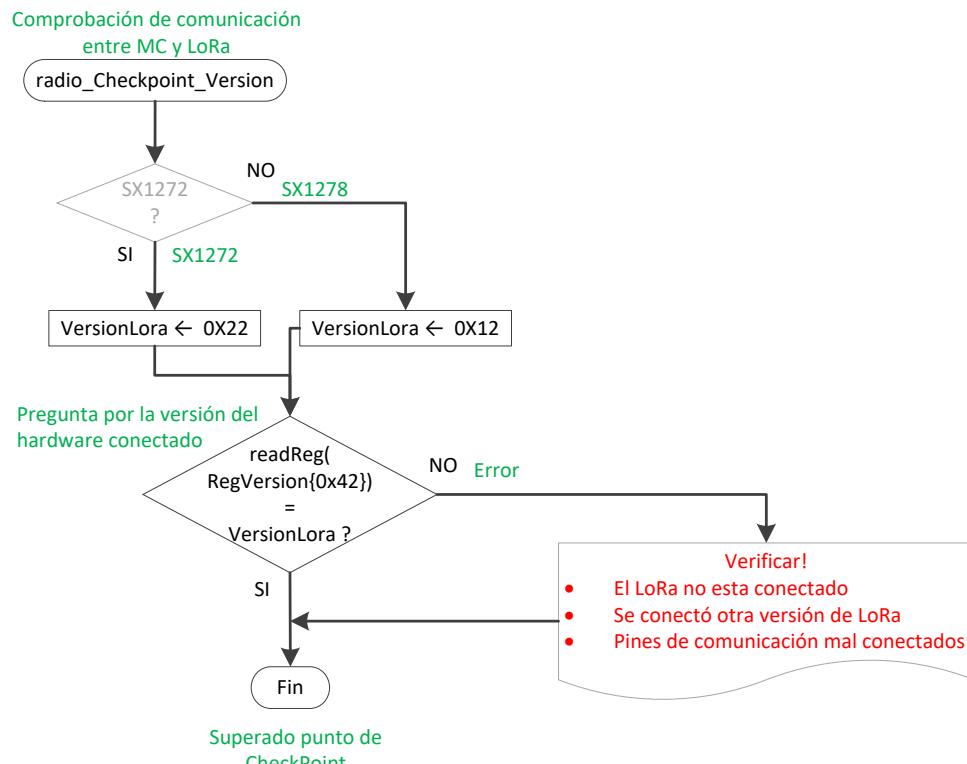


Figura 13. Comprobación de comunicación SPI entre el microcontrolador y el LoRa seleccionado.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

Registros de configuración SF, BW y CR de los transceivers LoRa

Los dispositivos LoRa cuentan con los parámetros *SF*, *BW* y *CR* que según las combinaciones entre estos, permite determinar el alcance o distancia máxima de recepción de datos, la velocidad de los datos y sobrecarga por corrección o detección de errores en los datos. Estos valores deben ser los mismos en todos los dispositivos radio LoRa de la red inalámbrica.

Los valores de *SF*, *BW* y *CR* se establecen en los registros *RegModemConfig1_0x1D* y *RegModemConfig2_0x1E* para los radio SX1272 y SX1278. Se incorpora un registro adicional denominado *RegModemConfig3_0x26* para el SX1278.

Existen otros parámetros de interés en estos registros de configuración, como las variables *ImplicitHeaderModeOn* (modo del encabezado), *RxPayloadCrcOn* (habilitación de errores por verificación CRC), *AgcAutoOn* (ajuste en el amplificador adicional LNA) y *LowDataRateOptimize* (bit de optimización de datos a baja velocidad).

Las Tablas 7 y 8 presentan la descripción de los registros *RegModemConfig1_0x1D* y *RegModemConfig2_0x1E* respectivamente para el SX1272. Las Tablas 9, 10 y 11 presentan la descripción de los registros *RegModemConfig1_0x1D*, *RegModemConfig2_0x1E* y *RegModemConfig3_0x26* para el SX1278.

Tabla 7. Descripción del registro RegModemConfig1_0x1D del LoRa SX1272.

Fuente: Semtech SX1272 (2015)

Bit	Nombre de la variable	Uso	Defecto	Descripción
7-6	Bw	rw	0x00	Ancho de banda (entre más bajo sea este valor, el tiempo de duración de la transmisión será mayor)
				00 → 125 kHz
				01 → 250 kHz
				10 → 500 kHz
				11 → reservado
5-3	CodingRate	rw	'001'	Tasa de codificación de errores: Entre mayor sea este valor, mayor será la fiabilidad de los datos, pero con una sobrecarga en el tiempo de transmisión.
				001 → 4/5
				010 → 4/6
				011 → 4/7
				100 → 4/8
				Otros valores → reservado
2	ImplicitHeaderModeOn	rw	0x00	Tipo de Encabezado utilizado por el LoRa. Se recomienda el <u>encabezado explícito</u> por contar con información adicional como la cantidad de bytes trasmítidos/recibidos y si contienen errores de CRC.
				0 → Modo del Header explícito
				1 → Modo del Header implícito
1	RxPayloadCrcOn	rw	0x0	Habilita la generación de CRC Verificación por redundancia cíclica en la recepción de los datos
				0 → CRC no habilitado
				1 → CRC habilitado
0	LowDataRateOptimize	rw	0x0	Habilita la optimización en la recepción de los datos cuando esta configurado el LoRa en baja transmisión de velocidad de los datos
				0 → no habilitado
				1 → habilitado. Solo debe ser habilitado cuando está configurado SF11 o SF12 a un BW=125kHz

Nota r: lectura. w: escritura.

Tabla 8. Descripción del registro RegModemConfig2_0x1E del LoRa SX1272.

Fuente: Semtech SX1272 (2015)

Bit	Nombre de la variable	Uso	Defecto	Descripción
7-4	SpreadingFactor	r w	0x07	SF rate Es el factor de alcance expresado como logaritmo de base 2. Entre mayor sea este valor, se tendrá un mejor rendimiento en la transmisión de datos.
				6 → 64 chips / symbol (modo utilizado para FSK)
				7 → 128 chips / symbol
				8 → 256 chips / symbol
				9 → 512 chips / symbol
				10 → 1024 chips / symbol
				11 → 2048 chips / symbol
				12 → 4096 chips / symbol
				Otros valores → reservado
				0 → modo normal de transmisión simple de paquetes. 1 → modo continuo de transmisión. Múltiples paquetes enviados a través de la FIFO.
2	AgcAutoOn	r w	0x00	Habilita la generación de CRC Verificación por redundancia cíclica en la recepción de los datos.
				0 → Ajuste de ganancia en el LNA según el registro LNA LnaGain 1 → Ajuste de ganancia en el LNA por loop interno en el AGC
1-0	SymbTimeout(9:8)	r w	0x00	Timeout para la espera de datos. Se recomienda 0.

Nota r: lectura. w: escritura.

Tabla 9. Descripción del registro RegModemConfig1_0x1D del LoRa SX1278.

Fuente: Semtech SX1278 (2015)

Bit	Nombre de la variable	Uso	Defecto	Descripción
7-4	Bw	r w	0x07	Ancho de banda (entre más bajo sea este valor, el tiempo de duración de la transmisión será mayor)
				0000 → 7.8 kHz
				0001 → 10.4 kHz
				0010 → 15.6 kHz
				0011 → 20.8kHz
				0100 → 31.25 kHz
				0101 → 41.7 kHz
				0110 → 62.5 kHz
				0111 → 125 kHz
				1000 → 250 kHz
				1001 → 500 kHz
				Otros valores → reservado

Bit	Nombre de la variable	Uso	Defecto	Descripción
3-1	CodingRate	r w	'001'	Tasa de codificación de errores: Entre mayor sea este valor, mayor será la fiabilidad de los datos, pero con una sobrecarga en el tiempo de transmisión.
				001 → 4/5
				010 → 4/6
				011 → 4/7
				100 → 4/8
				Otros valores → reservado
0	ImplicitHeaderModeOn	r w	0x00	Tipo de Encabezado utilizado por el LoRa. Se recomienda el <u>encabezado explícito</u> por contar con información adicional como la cantidad de bytes trasmítidos/recibidos y si contienen errores de CRC.
				0 → Modo del Header explícito
				1 → Modo del Header implícito

Nota r: lectura. w: escritura.

Tabla 10. Descripción del registro RegModemConfig2_0x1E del LoRa SX1278.

Fuente: Semtech SX1278 (2015)

Bit	Nombre de la variable	Uso	Defecto	Descripción
7-4	SpreadingFactor	r w	0x07	SF rate Es el factor de alcance expresado como logaritmo de base 2. Entre mayor sea este valor, se tendrá un mejor rendimiento en la transmisión de datos.
				6 → 64 chips / symbol (modo utilizado para FSK)
				7 → 128 chips / symbol
				8 → 256 chips / symbol
				9 → 512 chips / symbol
				10 → 1024 chips / symbol
				11 → 2048 chips / symbol
				12 → 4096 chips / symbol
				Otros valores → reservado
3	TxContinuousMode	r w	0x0	0 → modo normal de transmisión simple de paquetes.
				1 → modo continuo de transmisión. Múltiples paquetes enviados a través de la FIFO.
2	RxPayloadCrcOn	r w	0x00	Habilita la generación de CRC Verificación por redundancia cíclica en la recepción de los datos.
				0 → CRC no habilitado
				1 → CRC habilitado
1-0	SymbTimeout(9:8)	r w	0x00	Timeout para la espera de datos. Se recomienda 0.

Nota r: lectura. w: escritura.

Tabla 11. Descripción del registro *RegModemConfig3_0x26* del LoRa SX1278.

Fuente: Semtech SX1278 (2015)

Bit	Nombre de la variable	Uso	Defecto	Descripción
7-4	Sin uso	r	0x00	reservado
3	LowDataRateOptimize	r w	0x0	Habilita la optimización en la recepción de los datos cuando está configurado el LoRa en baja transmisión de velocidad de los datos
				0 → no habilitado
				1 → habilitado. Solo debe ser habilitado cuando excede los 16ms en el tiempo por símbolo T_S .
2	AgcAutoOn	r w	0x00	Habilita la generación de CRC Verificación por redundancia cíclica en la recepción de los datos.
				0 → Ajuste de ganancia en el LNA según el registro LNA LnaGain
				1 → Ajuste de ganancia en el LNA por loop interno en el AGC
1-0	Sin uso	r	0x00	reservado

Nota r: lectura. w: escritura.

El registro *RegModemConfig3_0x26* aplica únicamente para el radio SX1278. Una de las variables de este registro es el bit de *LowDataRateOptimize*. El criterio para habilitar este bit corresponde a que se excedan los 16ms en el tiempo por símbolo (T_S). La ecuación para el cálculo de este parámetro la comparte el fabricante y se presenta en (1).

$$T_S = \frac{2^{SF}}{BW} \quad (1)$$

El T_S se define como el periodo o tiempo en el aire de cada símbolo.

La *Figura 14* presenta el esquema lógico propuesto para determinar el TS según los valores de SF y BW.

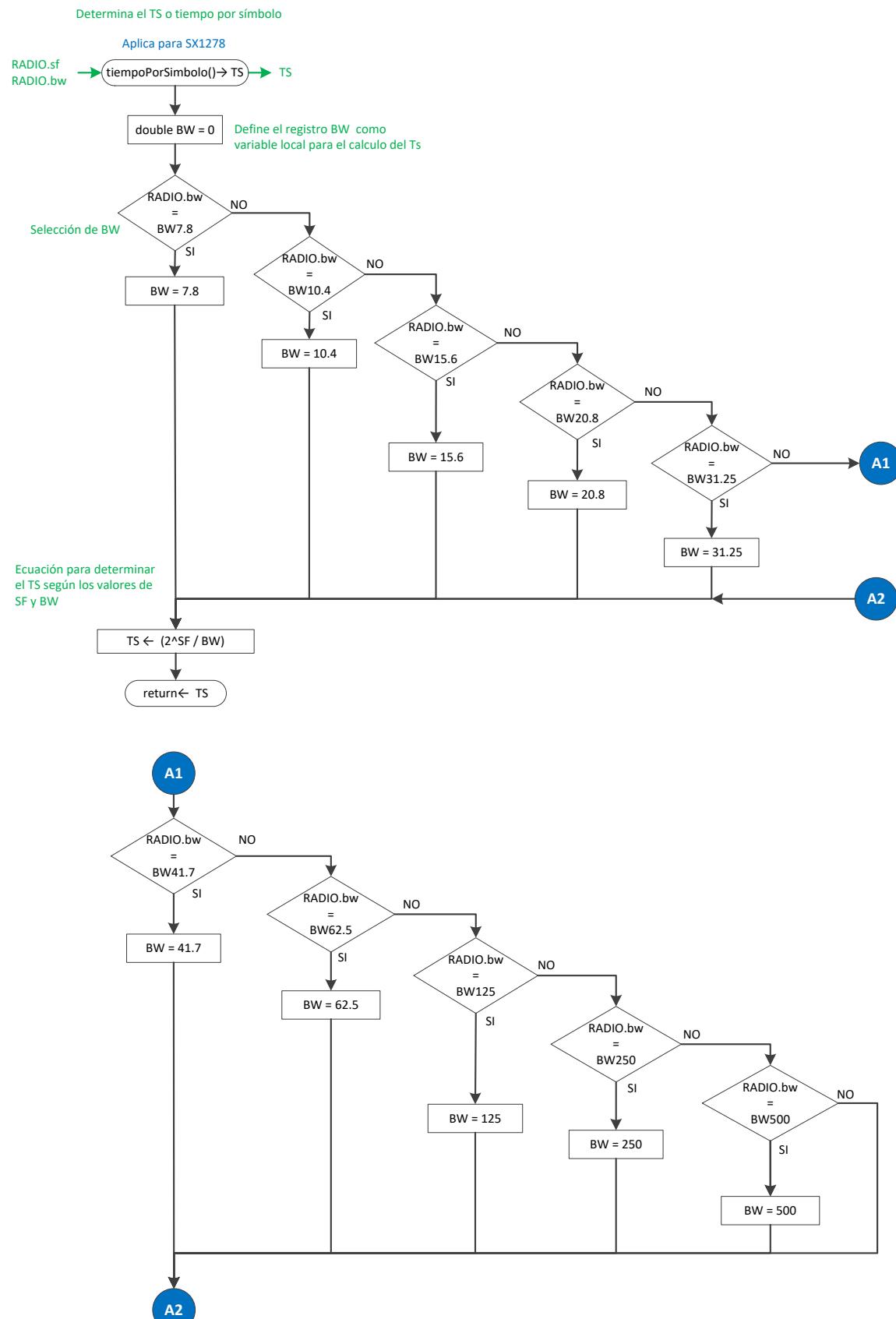


Figura 14. Esquema para determinar el TS según BW y SF.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

Para la configuración de los registros *RegModemConfig1_0x1D*, *RegModemConfig2_0x1E* y *RegModemConfig3_0x26* (aplica para el SX1278), se requiere de los valores **RADIO.sf**, **RADIO.bw** y **RADIO.cr** definidos por el usuario.

Los flujoograma presentado en las *Figuras 15, 16 y 17*, representan el esquema propuesto de configuración para los registros *RegModemConfig1_0x1D* y *RegModemConfig2_0x1E* para el chip radio SX1272. Los flujoogramas presentados en las 18, 19 y 20, representan el esquema propuesto de configuración de registros *RegModemConfig1_0x1D*, *RegModemConfig2_0x1E* y *RegModemConfig3_0x26* para el chip radio SX1278.

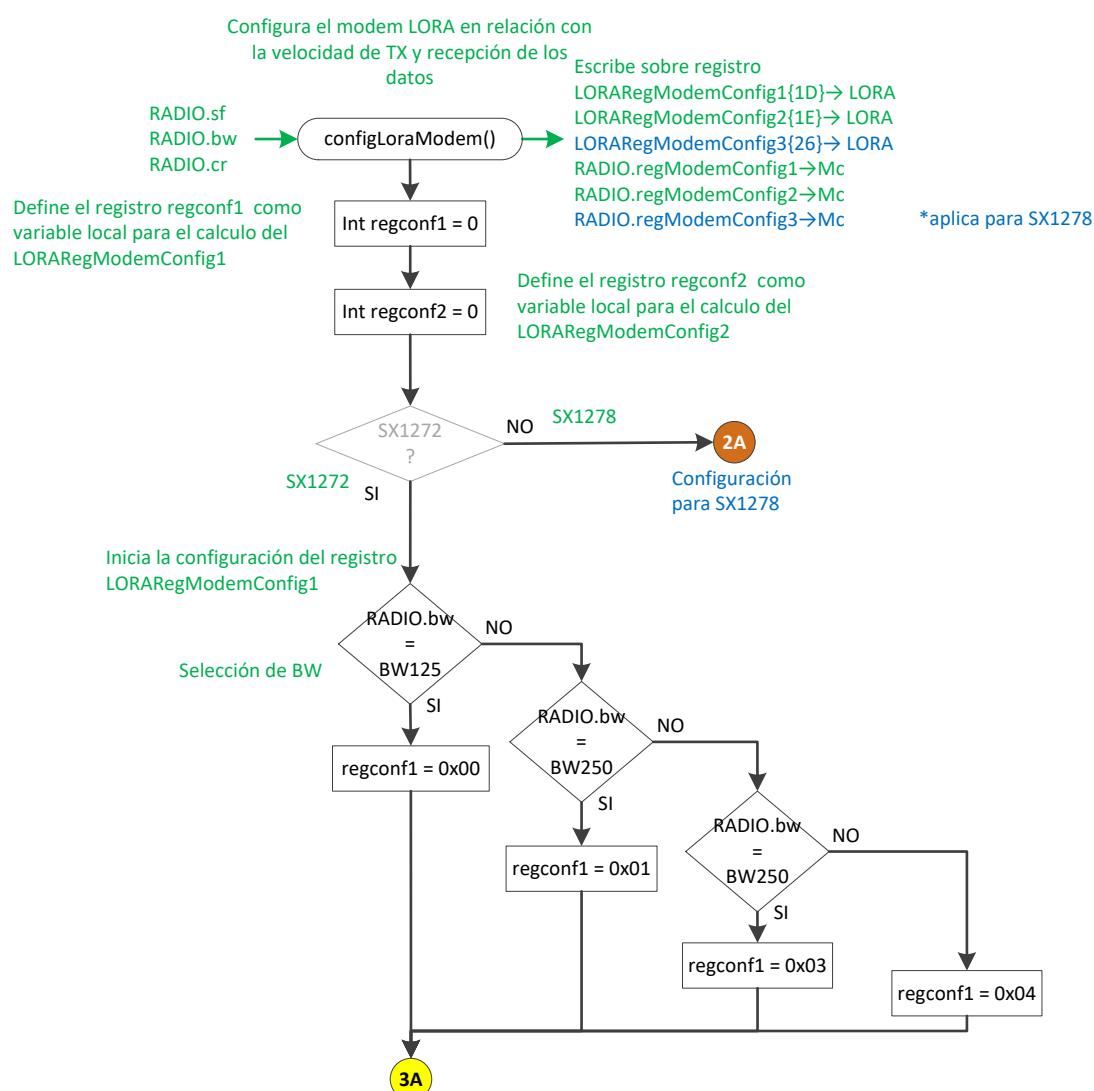


Figura 15. Esquema lógico de configuración de los registros *RegModemConfig1_0x1D* y *RegModemConfig2_0x1E* para SX1272 (parte I).

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

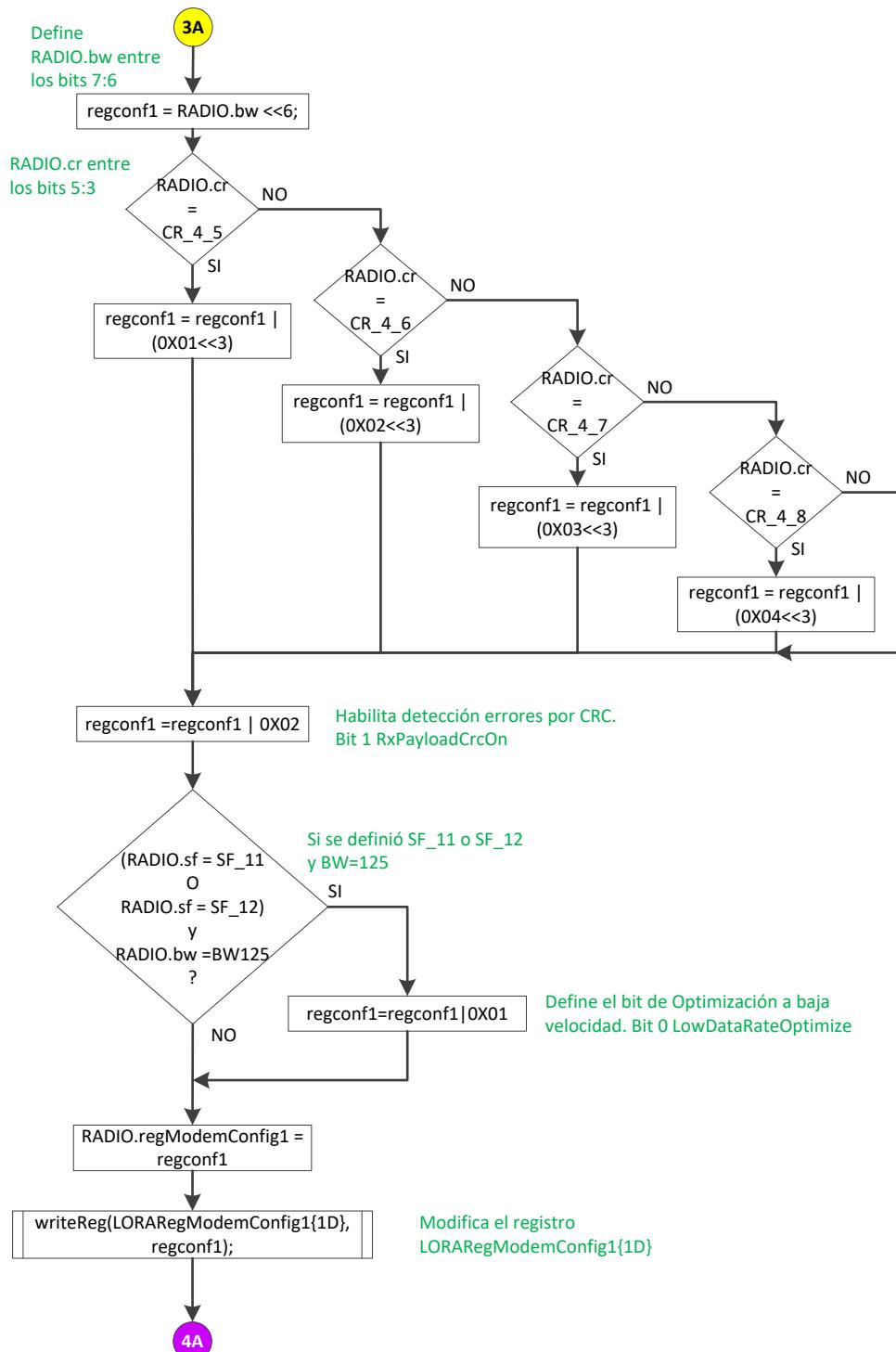


Figura 16. Esquema lógico de configuración de los registros RegModemConfig1_0x1D y RegModemConfig2_0x1E para SX1272 (parte II).

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

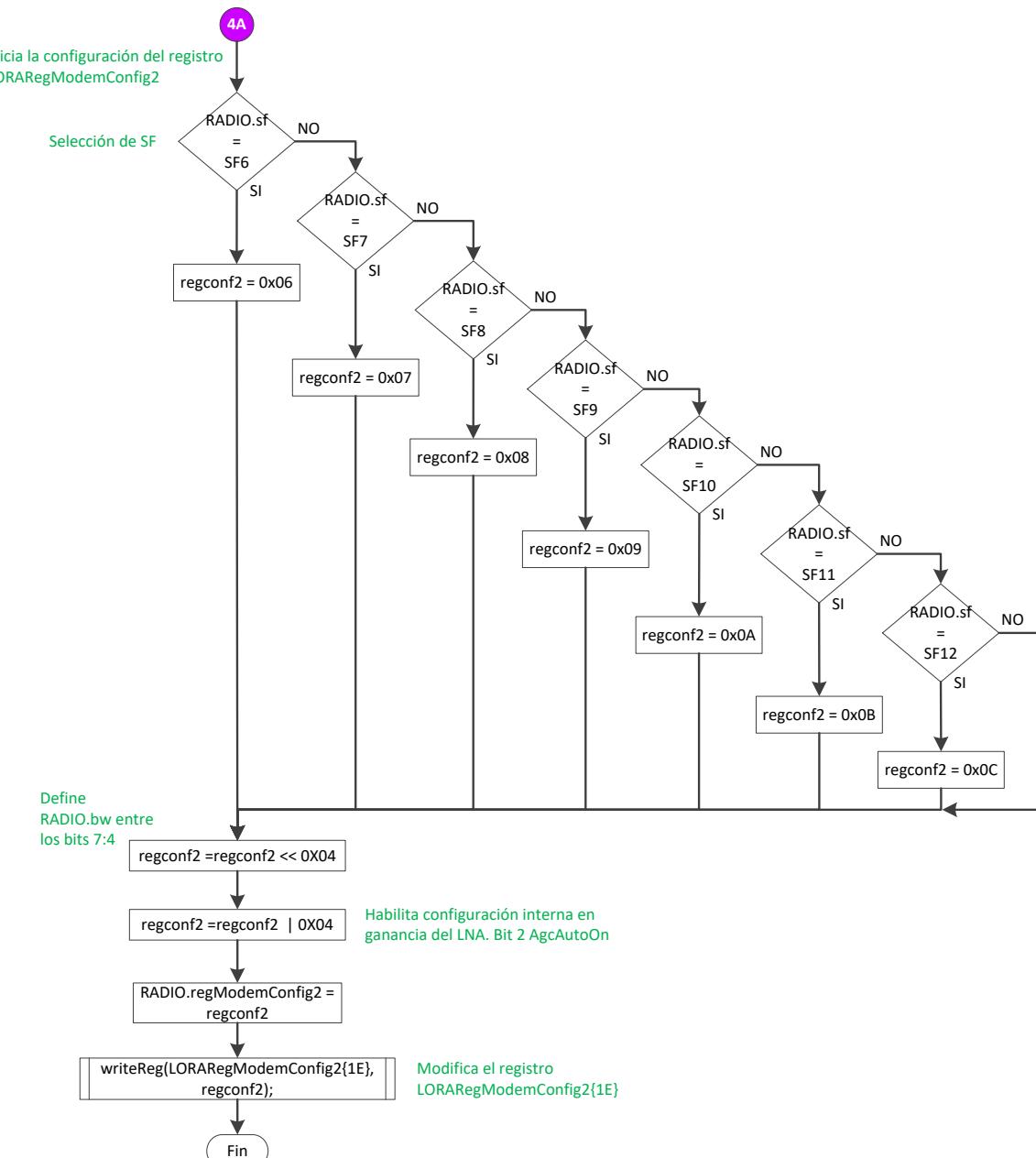


Figura 17. Esquema lógico de configuración de los registros RegModemConfig1_0x1D y RegModemConfig2_0x1E para SX1272 (parte III).

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

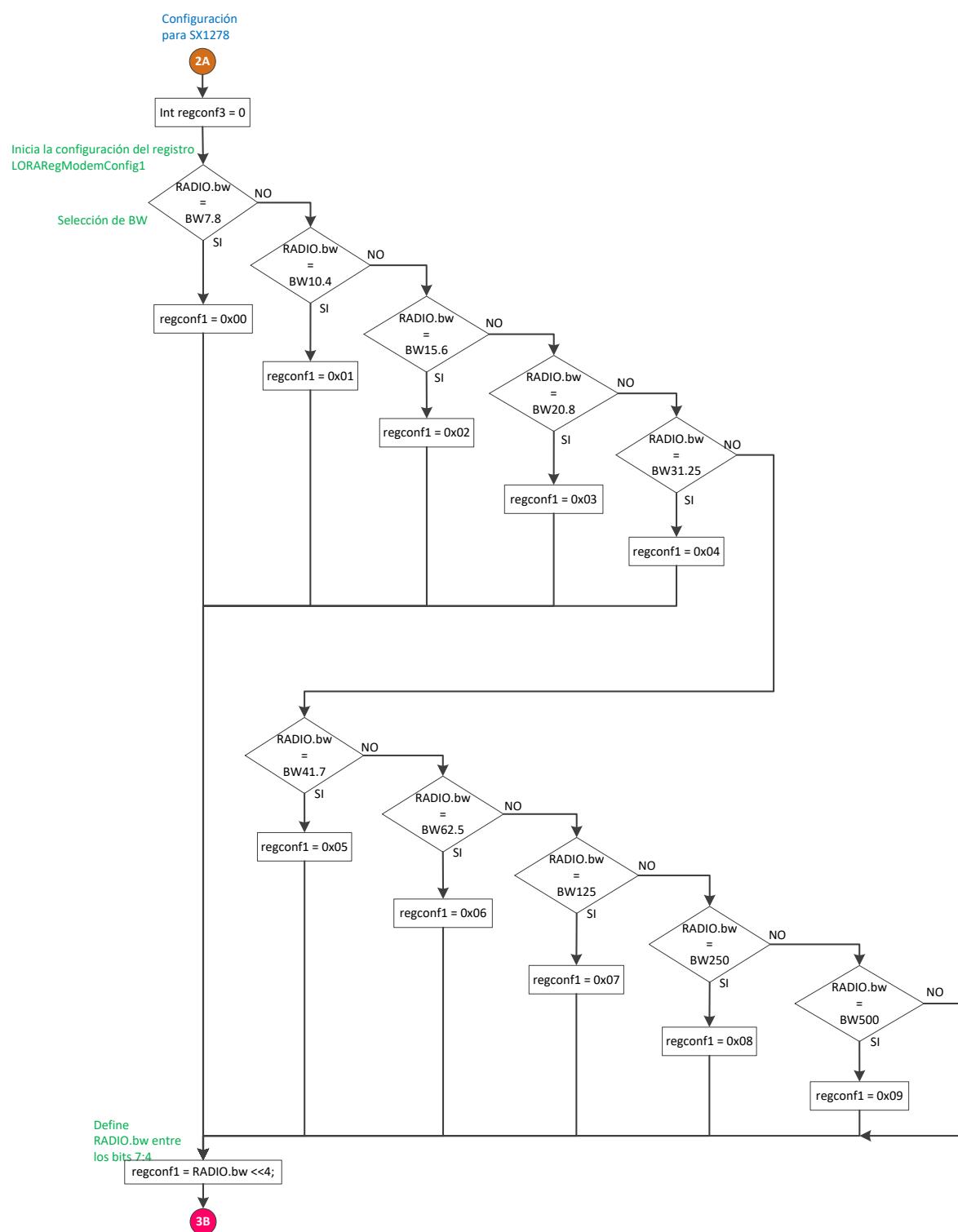


Figura 18. Esquema lógico de configuración de los registros RegModemConfig1_0x1D, RegModemConfig2_0x1E y RegModemConfig3_0x26 para SX1278 (parte I).

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

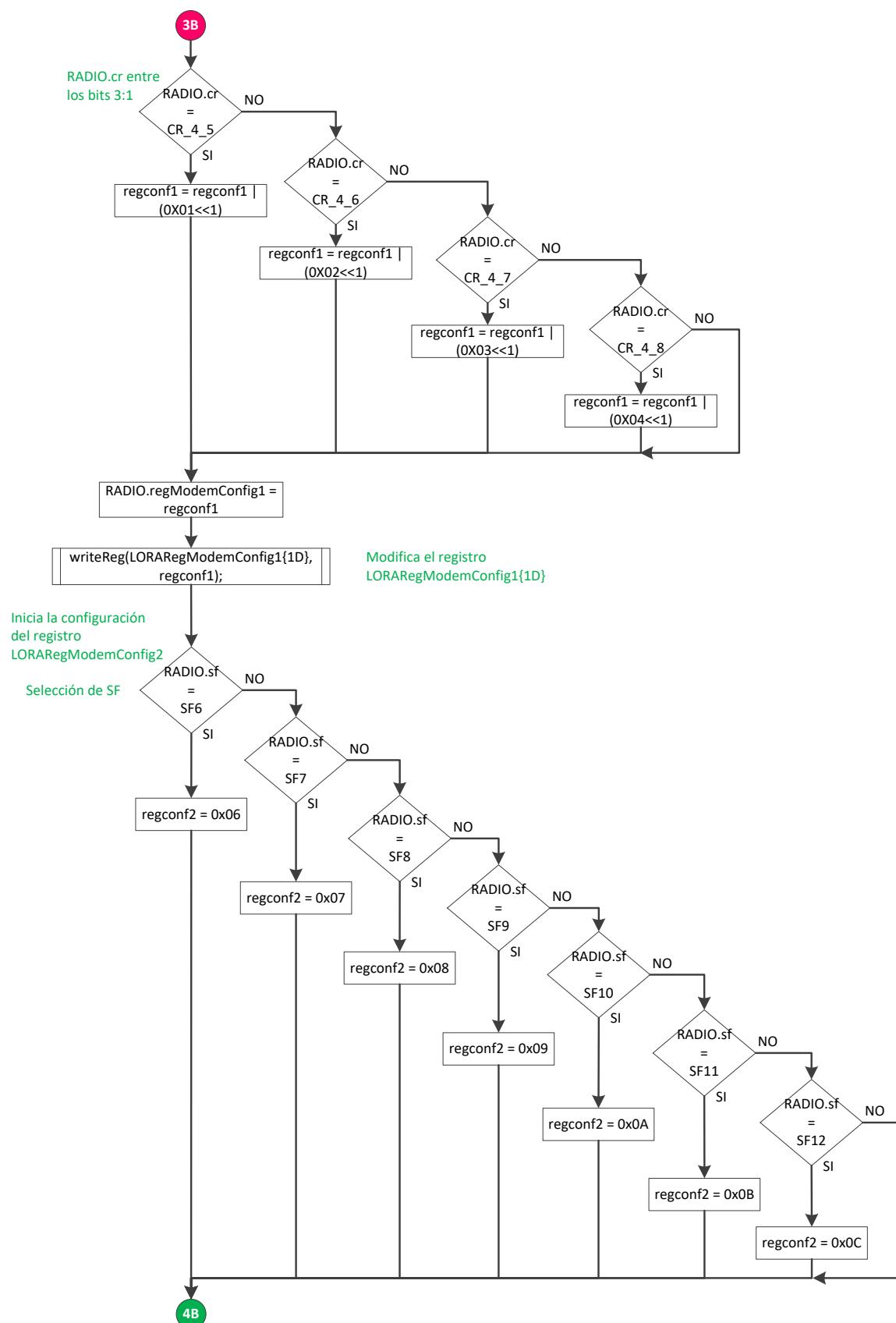


Figura 19. Esquema lógico de configuración de los registros RegModemConfig1_0x1D, RegModemConfig2_0x1E y RegModemConfig3_0x26 para SX1278 (parte II).

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

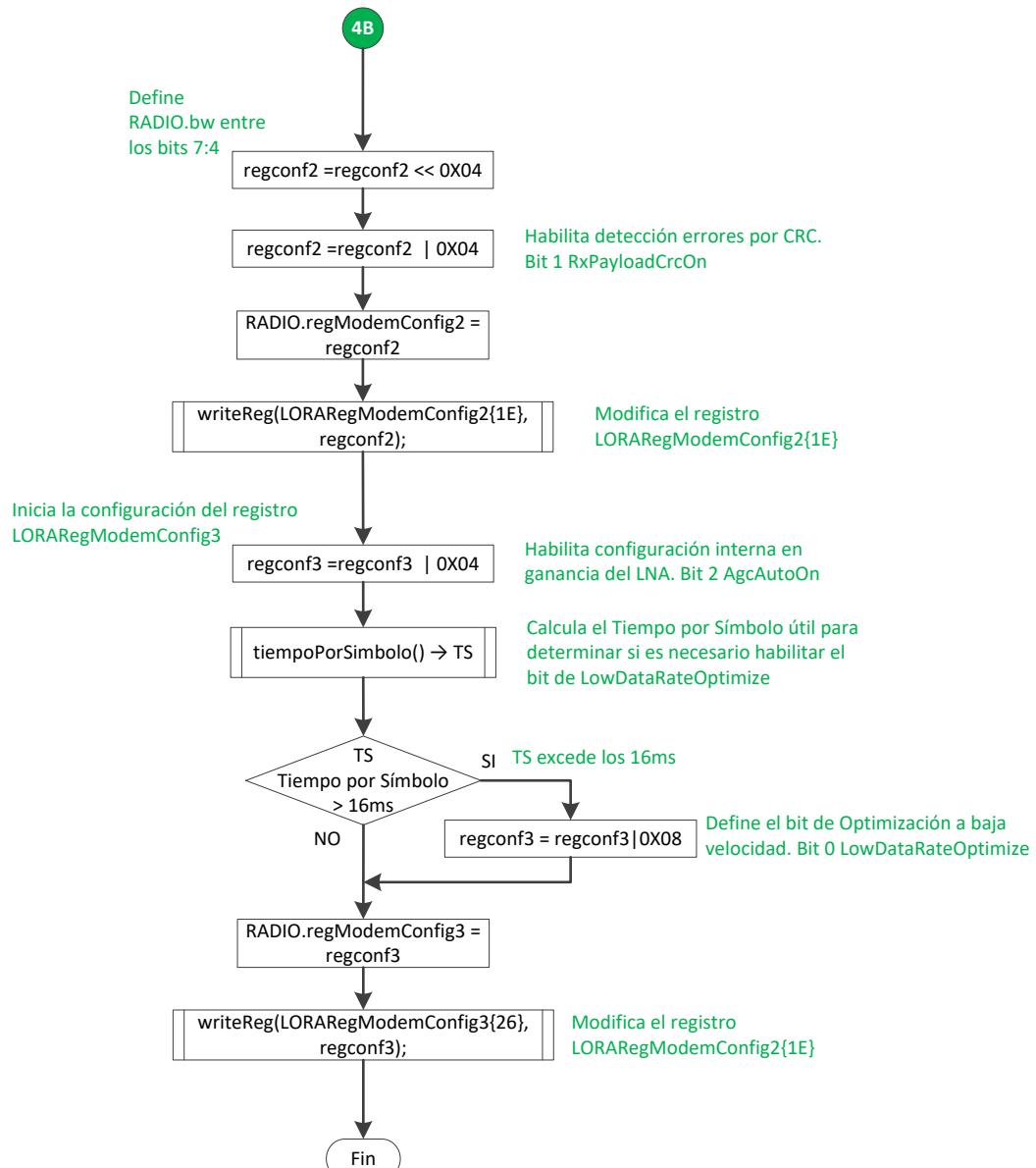


Figura 20. Esquema lógico de configuración de los registros RegModemConfig1_0x1D, RegModemConfig2_0x1E y RegModemConfig3_0x26 para SX1278 (parte III).

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

Registros de configuración de potencia del radio

Los chip de radio SX1272 y SX1278 permiten configurar la potencia máxima utilizada para trasmisir datos, máxima corriente permitida y ganancia del LNA (HF y LF para el SX1278).

El registro *RegLna_0x0C* permite configurar las características del LNA para HF y LF (LF aplica solo para el SX1278). La *Tabla 12* presenta las variables del registro *RegLna_0x0C* como la configuración de ganancia y los ajustes en corriente para alta frecuencia (HF) y baja frecuencia (LS).

Tabla 12. Descripción del registro RegLna_0x0C del LoRa.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015)

Bit	Nombre de la variable	Uso	Defecto	Descripción
7-5	LnaGain	r w	0x01	Ajustes en la ganancia del LNA
				000 → no usado
				001 → G1 = máxima ganancia
				010 → G2
				011 → G3
				100 → G4
				101 → G5
				110 → G6 = mínima ganancia
				111 → no usado
4-3	LnaBoostLf*	r w	0x00	Ajuste en corriente a baja frecuencia del LNA (RFI_LF)
				00 → Corriente por defecto
				Otro → Reservado
2	Reservado	r	0x00	Reservado
1-0	LnaBoostHf	r w	0x00	Ajuste en corriente a alta frecuencia del LNA (RFI_HF)
				00 → Corriente por defecto
				11 → Boost on, 150% de corriente en el LNA

Nota r: lectura. w: escritura. *: Aplica para SX1278

El registro *RegPaConfig_0x09* permite configurar las características del amplificador de potencia (PA). La *Tabla 12* presenta las principales variables del registro *RegPaConfig_0x09* para el chip radio SX1278 en las que se recomienda habilitar el *PA_BOOST* para contar con una potencia máxima de salida de hasta +20dBm, un valor de 0x0F para *MaxPower* ($P_{max} \approx +20\text{dBm}$) y 0x0F para *OutputPower*.

Tabla 13. Descripción del registro *RegPaConfig_0x09* del LoRa.

Fuente: Semtech SX1278 (2015)

Bit	Nombre de la variable	Uso	Defecto	Descripción
7	PaSelect	r w	0x00	Selección de la arquitectura del Amplificador de Potencia PA (pin de salida del PA)
				0 → RFO pin. Salida de potencia (Output power) limitada hasta +14 dBm.
				1 → PA_BOOST pin. Salida de potencia (Output power) limitada hasta +20 dBm.
6-4	MaxPower	r w	0x04	Selección de potencia máxima de salida de potencia: $P_{max}=10.8+0.6*MaxPower$ [dBm] ($0 \leq MaxPower \leq 15$)
3-0	OutputPower	r w	0x0F	$Pout=P_{max}-(15-OutputPower)$ SI PaSelect ← 0 (RFO pin)
				$Pout=17-(15-OutputPower)$ SI PaSelect ← 1 (PA_BOOST pin)

Nota r: lectura. w: escritura.

La Tabla 14 presenta las principales variables del registro *RegPaConfig_0x09* para el transceiver SX1272 en las que se recomienda habilitar el *PA_BOOST* para contar con una potencia máxima de salida de hasta +20dBm y un valor de 0x0F para *OutputPower*.

Tabla 14. Descripción del registro *RegPaConfig_0x09* del LoRa.

Fuente: Semtech SX1272 (2015)

Bit	Nombre de la variable	Uso	Defecto	Descripción
7	PaSelect	r w	0x00	Selección de la arquitectura del Amplificador de Potencia PA (pin de salida del PA)
				0 → RFO pin. Salida de potencia (Output power) limitada hasta +13 dBm.
				1 → PA_BOOST pin. Salida de potencia (Output power) limitada hasta +20 dBm.
6-4	Sin uso	r	0x04	Sin uso
3-0	OutputPower	r w	0x0F	$Pout = -1 + OutputPower(3:0)$ SI PaSelect ← 0 (RFO pin)
				$Pout = 2 + OutputPower(3:0)$ SI PaSelect ← 1 (PA_BOOST pin)

Nota r: lectura. w: escritura.

Los radio LoRa cuentan con un circuito de protección para evitar el *sobre-exceso* de corriente al transmitir (OCP) en el Amplificador de Potencia (PA). Esta característica de los LoRa permite aumentar el ciclo de vida útil de las baterías, utilizar bajos consumos energéticos y evitar picos excesivos al trasmisir datos.

El registro *RegOcp_0x0B* permite regular los consumos de corrientes pico al trasmisir hasta de 240mA cuando el transceiver está en modo transmisión. Para valores significativos de corrientes (mayores a 125mA), se recomienda verificar la corriente máxima de suministro soportada por la fuente o batería de alimentación. Algunos sistemas embebidos como Arduino cuentan con reguladores con corriente máxima de suministro hasta de 120mA, por lo que al realizar una inadecuada configuración en el LoRa, podría destruirse el regulador y dañar la placa del sistema embebido.

La *Tabla 15* presenta los valores para el registro *RegOcp_0x0B* de los transceivers SX1272 y SX1278. El registro *RegPaDac** establece la máxima potencia en el *PA_BOOST* de hasta +20dBm. La descripción es este registro representa en la *Tabla 16*.

Tabla 15. Descripción del registro RegOcp_0x0B del LoRa.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015)

Bit	Nombre de la variable	Uso	Defecto	Descripción
7-6	Sin uso	r	0x00	Sin uso
5	OcpOn	r w	0x01	Habilita el circuito de protección por exceso de corriente al trasmisir (OCP) el amplificador de potencia (PA):
				0 → OCP no lo habilita
				1 → OCP lo habilita
4-0	OcpTrim	r w	0x0B	Corriente máxima soportada por el OCP:
				$I_{max} = 45+5 \cdot OcpTrim [mA]$ SI $OcpTrim \leq 15$ (I_{max} hasta 120mA)
				$I_{max} = -30+10 \cdot OcpTrim [mA]$ SI $15 < OcpTrim \leq 27$ (130 < I_{max} < 240 mA)
				$I_{max} = 240mA$ SI $OcpTrim > 27$
				Por defecto $I_{max} = 100mA$

Nota r: lectura. w: escritura.

Tabla 16. Descripción del registro RegPaDac del LoRa.*

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015)

Bit	Nombre de la variable	Uso	Defecto	Descripción
7-3	Reservado	r w	0x10	Reservado
2-0	PaDac	r w	0x04	Habilita la opción de +20dBm SI <i>PA_BOOST</i> pin-> 1
				0x04 → Valor por defecto
				0x07 → +20dBm en <i>PA_BOOST</i> cuando <i>OutputPower=1111</i>

Nota r: lectura. w: escritura. *La dirección para el SX1272 es 0x5A y para SX1278 es 0x4D

Se plantea en la *Figura 21* el esquema lógico de configuración de los registros relacionados con la potencia de los transceivers LoRa utilizando las variables `RADIO.txpow` y `RADIO.imax` configurables por el usuario.

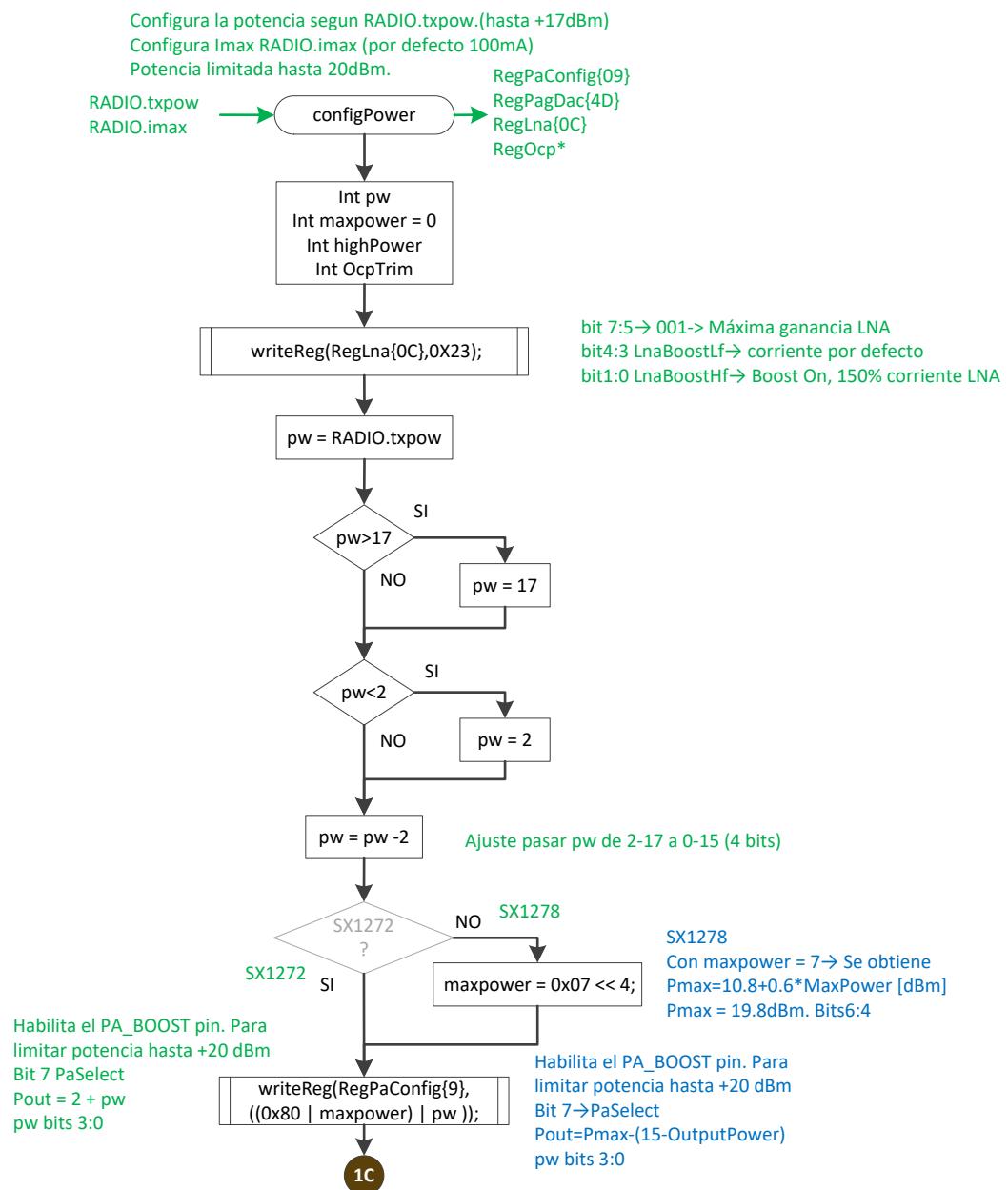


Figura 21. Esquema lógico de configuración de los registros `RegLna_0x0C`, `RegPaConfig_0x09`, `RegOcp_0x0B` y `RegPdAc` para los dispositivos LoRa (parte I).
Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

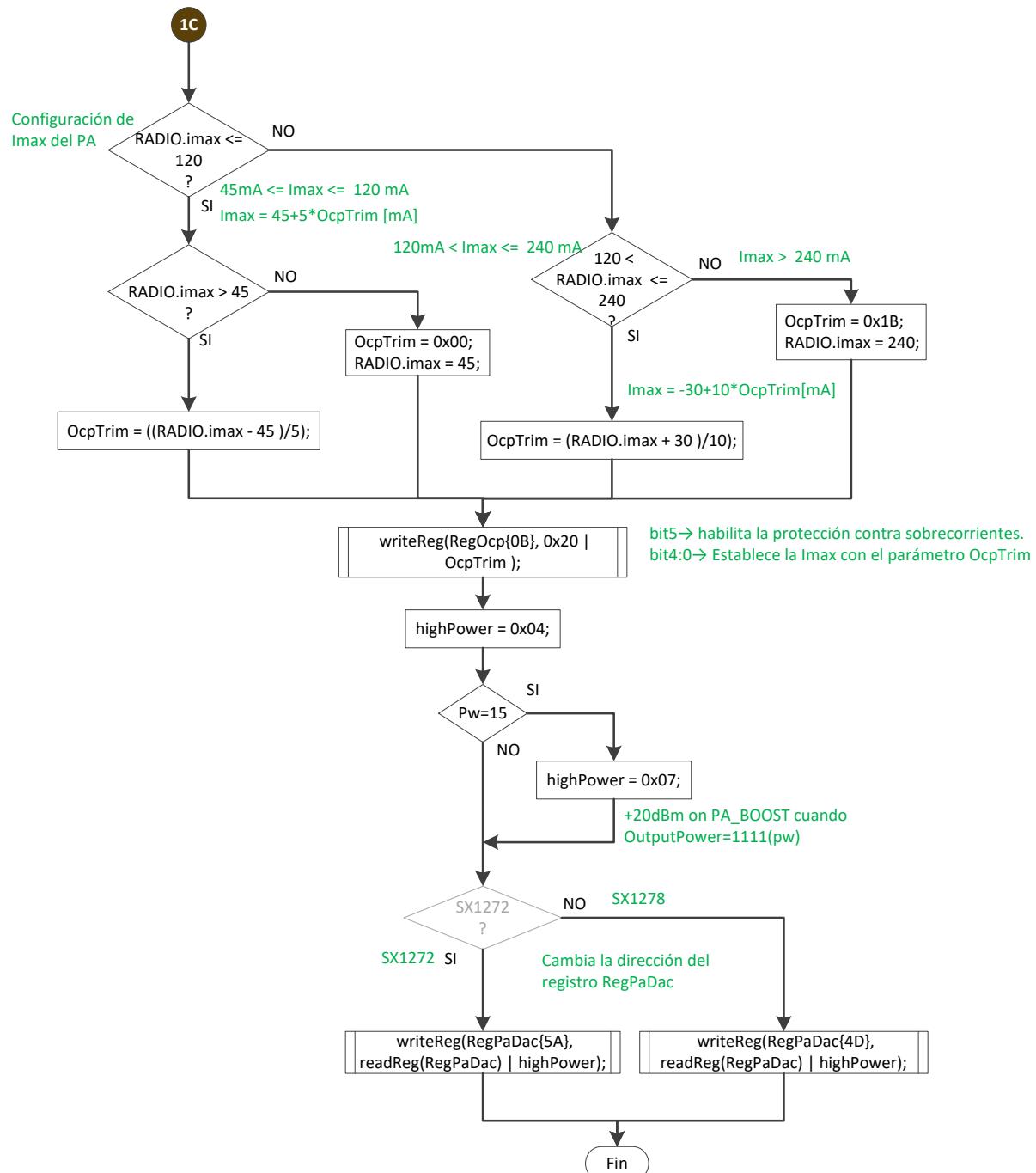


Figura 22. Esquema lógico de configuración de los registros RegLna_0x0C, RegPaConfig_0x09, RegOcp_0x0B y RegPaDac para los dispositivos LoRa (parte II).
Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

Registros de configuración de frecuencia del radio

Teniendo en cuenta que existen bandas de frecuencias de uso privativo (telefonía, internet, radio, entre otras), existen frecuencias de libre uso conocidas como ISM (Industrial, Scientific and Medical) que son frecuencias intencionalmente libres para uso no comercial en aplicaciones de tipo industrial, científico y médico. Se recomienda al usuario verificar la compatibilidad de la frecuencia de trabajo con la *Tabla 17* (bandas de uso ISM).

Tabla 17. Banda de frecuencia ISM compatibles con los transceivers de radio LoRa.

Fuente: Texas Instruments (2005)

Banda ISM (MHz)	Frecuencia mínima (MHz)	Frecuencia máxima (MHz)	Ancho de Banda (MHz)	Usos por continente
433	433.050	434.790	1.84	Europa, África y parte del norte de Asia.
869	868	870	2	Europa, África, Asia y Oceanía.
900	902	928	26	Continente americano

Para fijar la frecuencia de trabajo en el los transceivers LoRa, el fabricante presenta la ecuación (2):

$$f_{RF} = \frac{2^{19} \cdot Frecuencia}{F_{OSC}} \quad (2)$$

Dónde:

f_{RF} : Frecuencia utilizada en los registros SPI *RegFrLsb*, *RegFrMid* y *RegFrMsb*

F_{OSC} =Frecuencia del oscilador del LoRa (32 MHz)

Frecuencia : Frecuencia de trabajo fijada por el usuario.

La *Tabla 18* presenta la descripción de los registros de configuración de frecuencia.

Tabla 18. Descripción del registro de configuración de frecuencia en los radio LoRa.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015)

Nombre del registro	Dirección del registro	Uso	Defecto		Descripción
			SX1272	SX1278	
RegFrLsb	0x08	r w	0x00	0x00	Bits 7:0 del valor calculado de f_{RF}
RegFrMid	0x07	r w	0xC0	0x80	Bits 15:8 del valor calculado de f_{RF}
RegFrMsb	0x06	r w	0xE4	0x6C	Bits 15:23 del valor calculado de f_{RF}

Nota r: lectura. w: escritura.

La *Figura 23* presenta el esquema lógico propuesto para fijar la frecuencia de trabajo en el chip LoRa. La variable `RADIO.Freq` será definida por el usuario en Hz.

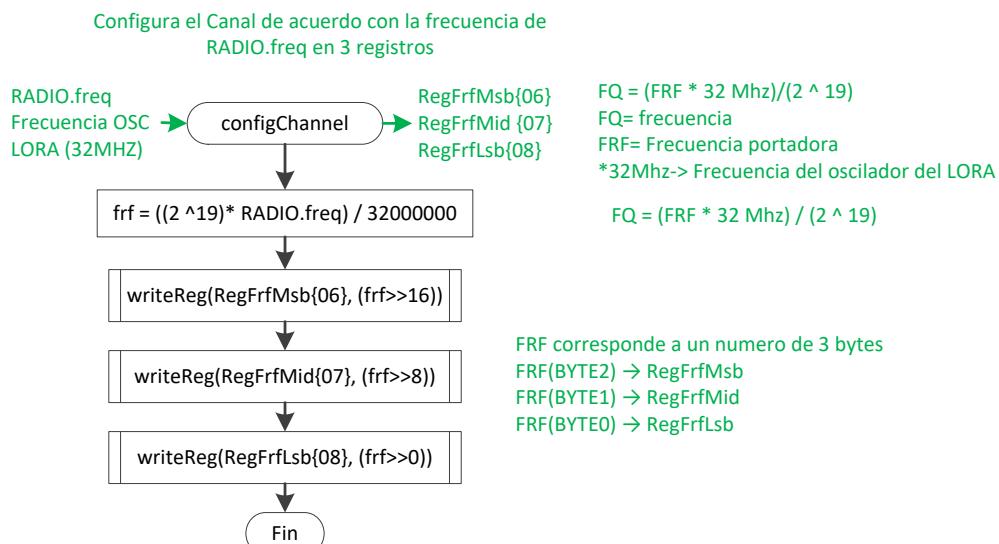


Figura 23. Esquema lógico de configuración de frecuencia para los radio LoRa.
Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

Configuración por software del radio lora para la transmisión datos

Para trasmisir los datos, es necesario utilizar los registros relacionados con la FIFO (memoria RAM de almacenamiento de los datos que serán transmitidos o que gestiona la recepción de datos). Los transceivers LoRa cuentan con una capacidad de 256 bytes en la FIFO.

Se recomienda el uso de interrupciones las cuales resultan ser útiles para la transmisión de datos por cuanto una vez se escriben los datos en la FIFO, de acuerdo a la configuración (*BW*, *SF* y *CR*), los transceivers LoRa tardan cierto tiempo en realizar esta tarea y la interrupción *TXDONE* permite preparar al LoRa ya sea para una nueva transmisión, colocarse en modo *SLEEP* o prepararse para recibir datos.

Cada una de las interrupciones utilizadas por los dispositivos LoRa se relaciona con un bit o pin físico de control (los bits de control son los DIO y se presentan en las *Figuras 2, 3 y 4*).

La *Tabla 19* presenta la descripción del registro `RegIrqFlagsMask_0x11` que permite habilitar el uso de las interrupciones. La *Tabla 20* presenta el registro de `RegIrqFlags_0x12` que

permite determinar si se ha presentado una interrupción por software. Sin embargo, los pines DIO se encuentran relacionados con estos flags y pueden ser utilizados para detectar interrupciones por cambio de flanco ascendente (Low-High), lo cual resulta ser muy útil en cuanto a la optimización de tareas con el uso de microcontroladores. Las *Tablas 21* y *22* presentan la descripción de los registros *RegDioMapping1_0x40* y *RegDioMapping2_0x41* para la configuración de pines DIO con el uso de interrupciones por hardware.

Tabla 19. Descripción del registro RegIrqFlagsMask_0x11 para habilitación de interrupciones.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015)

Bit	Nombre de la variable	Uso	Defecto	Descripción
7	RxTimeoutMask	r w	0x00	Mascara de interrupción para Timeout. Con "0" se habilita la interrupción.
6	RxDoneMask	r w	0x00	Mascara de interrupción para RXDONE o recepción completa de paquetes. Con "0" se habilita la interrupción.
5	PayloadCrcErrorMask	r w	0x00	Mascara de interrupción para detección de error por CRC en recepción de datos. Con "0" se habilita la interrupción.
4	ValidHeaderMask	r w	0x00	Mascara de interrupción para detección Header o encabezado en recepción de datos. Con "0" se habilita la interrupción
3	TxDoneMask	r w	0x00	Mascara de interrupción para TXDONE o transmisión completa de datos en la FIFO. Con "0" se habilita la interrupción.
2	CadDoneMask	r w	0x00	Mascara de interrupción para CADDONE o detección de preámbulo. Con "0" se habilita la interrupción.
1	FhssChangeChannelMask	r w	0x00	Mascara de interrupción para FhssChangeChannel o cambio de canal. Con "0" se habilita la interrupción.
0	CadDetectedMask	r w	0x00	Mascara de interrupción para detección del CAD. Con "0" se habilita la interrupción.

Nota r: lectura. w: escritura.

Tabla 20. Descripción del registro RegIrqFlags_0x12 para la detección de interrupciones por software.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015)

Bit	Nombre de la variable	Uso	Defecto	Descripción
7	RxTimeout	r c	0x00	Flag de interrupción por Timeout. Escribiendo "1" se limpia a IRQ.
6	RxDone	r c	0x00	Flag de interrupción por RXDONE o recepción completa de paquetes. Escribiendo "1" se limpia a IRQ.
5	PayloadCrcError	r c	0x00	Flag de interrupción para detección de error por CRC en recepción de datos. Escribiendo "1" se limpia a IRQ.
4	ValidHeader	r c	0x00	Flag de interrupción para detección Header o encabezado en recepción de datos. Escribiendo "1" se limpia a IRQ.
3	TxDone	r c	0x00	Flag de interrupción para TXDONE o transmisión completa de datos en la FIFO. Escribiendo "1" se limpia a IRQ.
2	CadDone	r c	0x00	Flag de interrupción para CADDONE o detección de preámbulo. Escribiendo "1" se limpia a IRQ.
1	FhssChangeChannel	r c	0x00	Flag de interrupción para FhssChangeChannel o cambio de canal. Escribiendo "1" se limpia a IRQ.
0	CadDetected	r c	0x00	Flag de interrupción para detección del CAD. Escribiendo "1" se limpia a IRQ.

Nota r: lectura. c: limpiar.

Tabla 21. Configuración de pines de control con el registro *RegDioMapping1_0x40* para el uso con interrupciones en los pines DIO0-3.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015)

Bit	Nombre de la variable	Uso	Defecto	Descripción
7-6	Dio0Mapping	rw	0x00	00 → RxDone. Se activa el pin cuando se ha recibido datos en la FIFO.
				01 → TxDone. Se activa el pin cuando se ha transmitido datos desde la FIFO.
				10 → CadDone
				11 → Sin uso del pin
5-4	Dio1Mapping	rw	0x00	00 → RxTimeout
				01 → FhssChangeChannel
				10 → CadDetected
				11 → Sin uso del pin
3-2	Dio2Mapping	rw	0x00	00 → FhssChangeChannel
				01 → FhssChangeChannel
				10 → FhssChangeChannel
				11 → Sin uso del pin
1-0	Dio3Mapping	rw	0x00	00 → CadDone
				01 → ValidHeader
				10 → PayloadCrcError. Se activa el pin cuando se detecta un error por verificación por redundancia cíclica (CRC)
				11 → Sin uso del pin

Nota r: lectura. w: escritura.

Tabla 22. Configuración de pines de control con el registro *RegDioMapping2_0x41* para el uso con interrupciones en los pines DIO4-5.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015)

Bit	Nombre de la variable	Uso	Defecto	Descripción
7-6	Dio4Mapping	rw	0x00	00 → CadDetected
				01 → PllLock
				10 → PllLock
				11 → Sin uso del pin
5-4	Dio5Mapping	rw	0x00	00 → ModeReady
				01 → ClkOut
				10 → ClkOut
				11 → Sin uso del pin

Para escribir datos sobre la FIFO de los transceivers LoRa, se utiliza el registro *RegFifo_0x00* para indicar la posición de escritura de los datos. Para tener acceso a la FIFO, el chip radio debe estar en modo STAND-BY. La Figura 20 presenta el esquema lógico propuesto para almacenar datos sobre la FIFO antes de preparar el chip para transmitirlos. Para utilizar la propuesta de configuración se requiere que el usuario defina sobre **RADIO.frameTX** los valores que desea transmitir y en **RADIO.dataLenTX** la cantidad o longitud de datos a transmitir.

Utilizada para escribir sobre la FIFO del LORA
RegFiFO → Dirección inicial FIFO
RADIO.frameTX → Vector referencia para leer y luego escribir sobre la FIFO
RADIO.dataLenTX → Longitud de datos a escribir

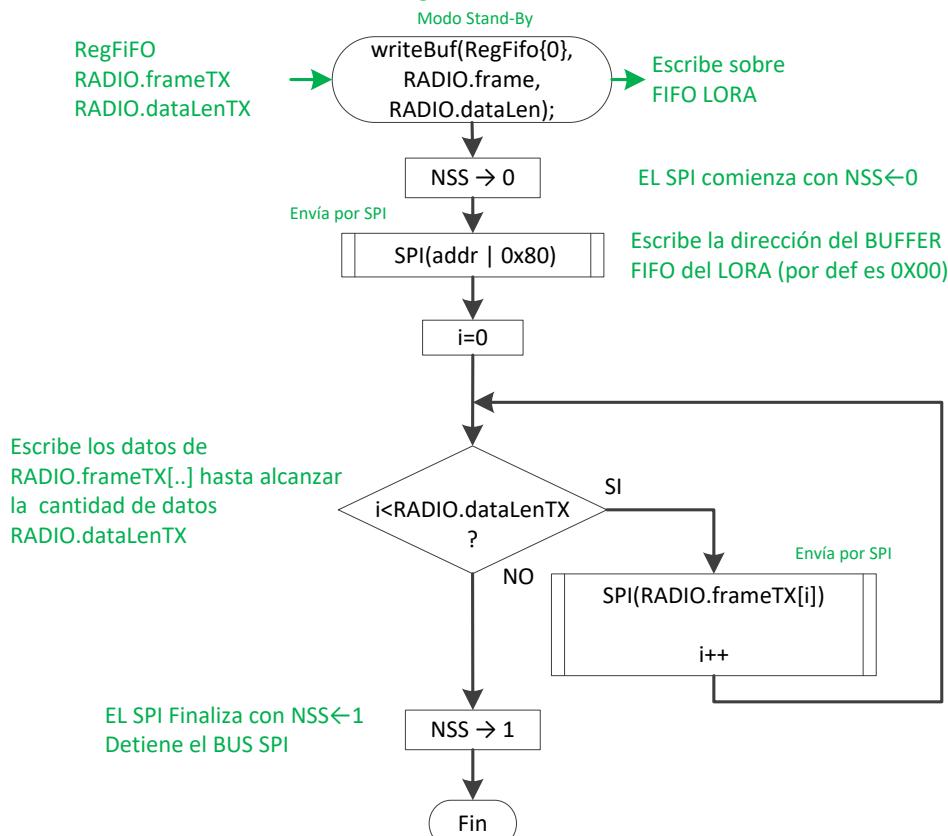


Figura 24. Esquema lógico de almacenamiento de datos sobre la FIFO del LoRa.
Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

La *Figura 24* presenta el esquema lógico propuesto para trasmisir datos previamente almacenados en la FIFO. Para utilizar el esquema, se requiere que el usuario defina sobre **RADIO.frameTX** los valores que desea transmitir y en **RADIO.dataLenTX** la cantidad o longitud de datos. Además, antes de finalizar el flujo grama se recomienda que el usuario configure una interrupción por cambio de flanco ascendente en el pin conectado al terminal del control DIO0.

Una vez se establece el modo *OPMODE_TX* en los dispositivos LoRa, comienza formalmente la transmisión de los datos escritos sobre la FIFO. Cuando finaliza la transmisión, el flag *TxDone* del registro *RegIrqFlags_0x12* se habilitara al igual que el pin *DIO0*. Si no se habilita ninguno de los dos (flag *TxDone* o pin *DIO0*), el usuario deberá reenviar los datos. El esquema lógico para determinar si se deben retransmitir los datos se presenta en la *Figura 25*.

Configura el modo a modo TX
 Para configurar el lora siempre debe cambiarse de modo SLEEP a Modo STANDBY{}
 Carga los datos de RADIO.frame al FIFO del LORA. La cantidad de datos es de
 RADIO.dataLen

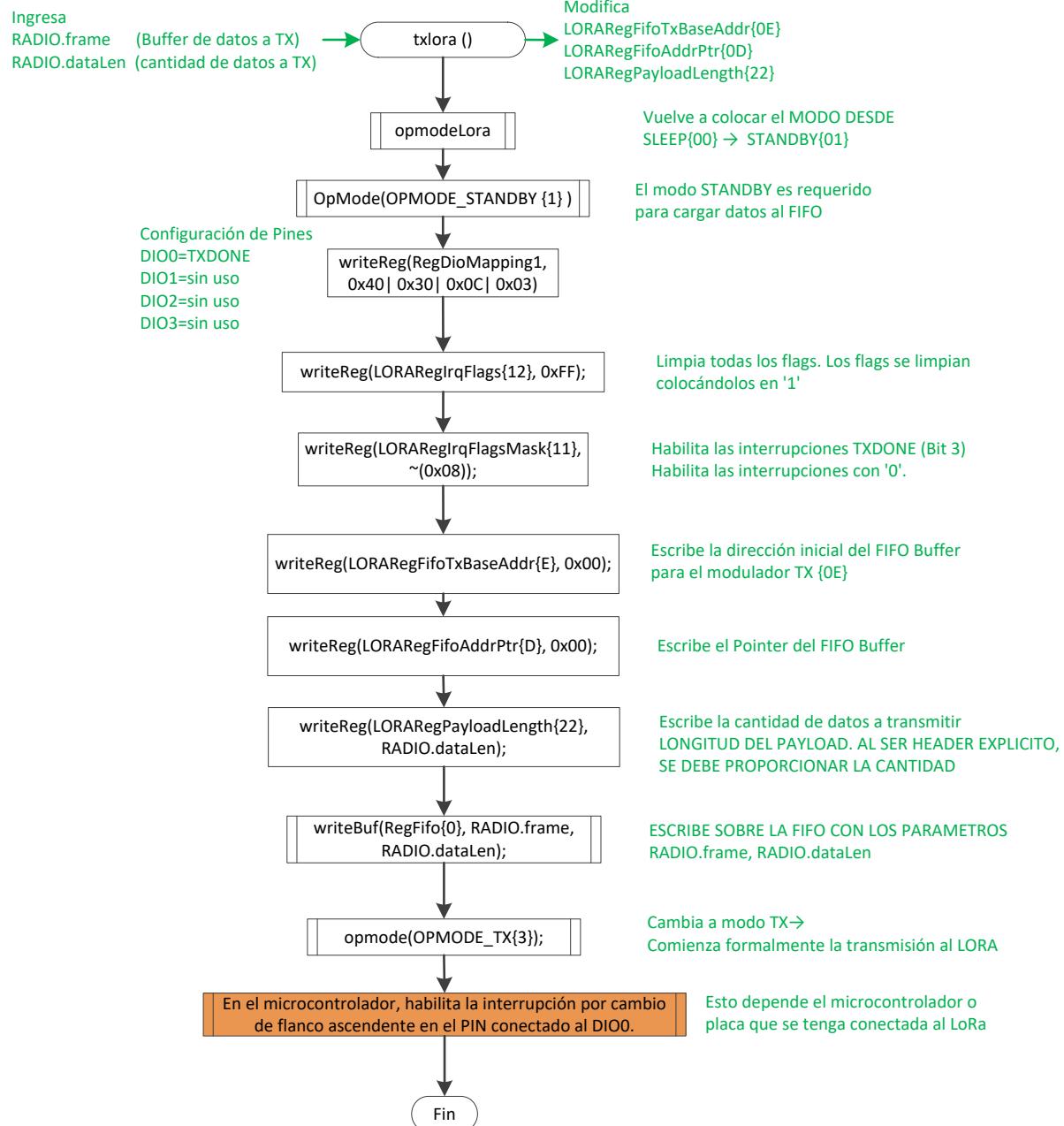


Figura 25. Esquema lógico de configuración para transmisión de datos en el LoRa.
 Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

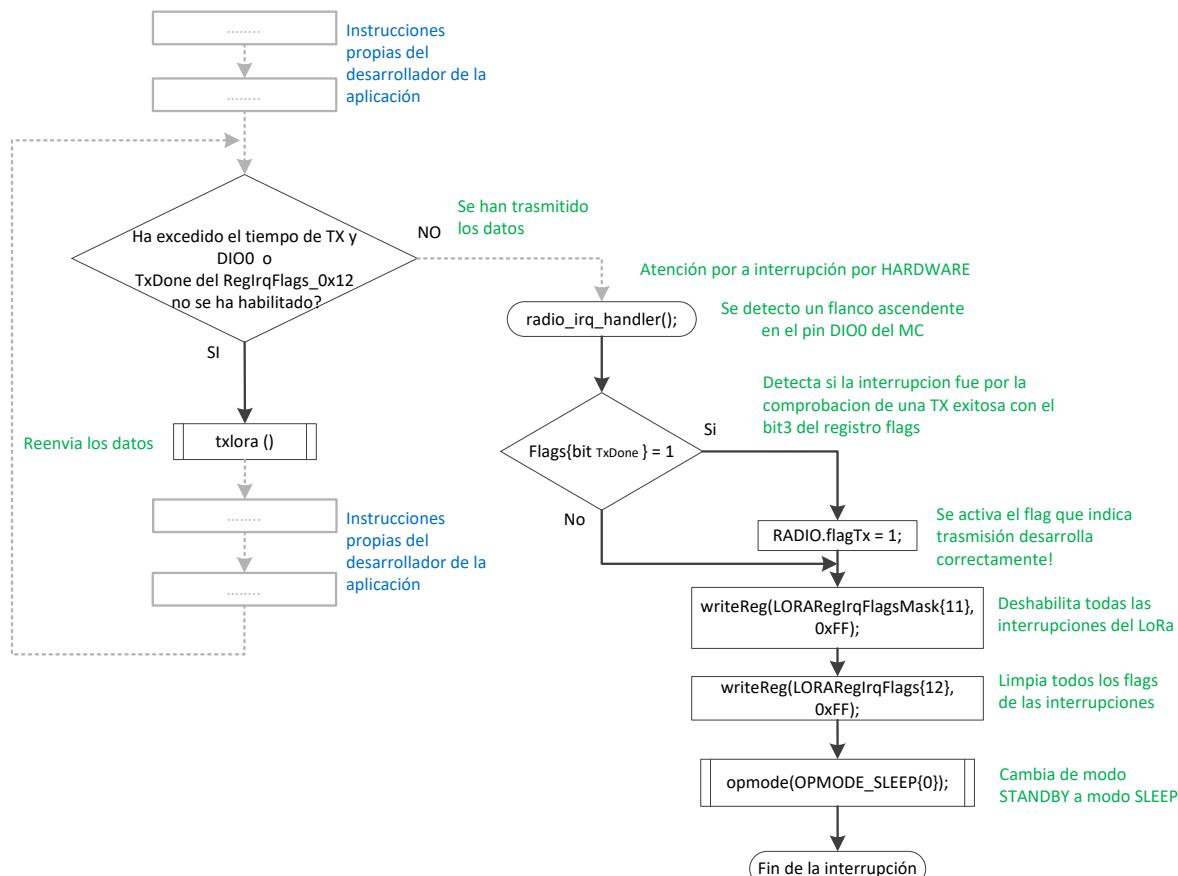


Figura 26. Esquema lógico propuesto para gestión por interrupción cuando se efectúa una transmisión de datos en el LoRa.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

Configuración por software del radio lora para la recepción de datos

El modem LoRa es half-duplex, es decir que no es posible trasmisir o recibir datos en el mismo instante; esto significa que se debe configurar el modem para trasmisir (`txlora`) o recibir datos (`rxlora`). El LoRa tiene la capacidad de recibir hasta 256 bytes de manera simultánea (capacidad de la FIFO).

Para recibir datos, se debe configurar el modem junto con las interrupciones de configuración para dicha tarea. La interrupción para detectar la recepción y gestión necesaria de datos sobre la FIFO se habilita utilizando el bit 6 o *RxDoneMask* del registro *RegIrqFlagsMask_0x11* (ver Tabla 18). Esta interrupción está relacionada con el pin de control DIO0 (ver Tabla 20).

Una ventaja adicional del chip LoRa es la incorporación de la verificación por redundancia cíclica (CRC), es decir que en caso de detectar un error al recibir tramas de datos sobre la FIFO, el LoRa tiene la posibilidad de emitir o presentar un error. La detección de errores en la recepción de datos CRC se habilita utilizando el bit 5- *PayloadCrcErrorMask* del registro *RegIrqFlagsMask_0x11* (ver *Tabla 18*). Esta interrupción está relacionada con el pin de control DIO3 (ver *Tabla 20*) y solo será activada cuando se reciban datos errados; por lo que se activará el bit 5 – *PayloadCrcError* del registro *RegIrqFlags_0x12* “o” el pin DIO3 del microcontrolador (el pin DIO3 se encuentra físicamente para las placas SX1272RF1 y DRF1278F, para la versión LoRa1278 de NiceRF se utiliza únicamente el bit 5 del registro *PayloadCrcError*).

La *Figura 27* presenta el esquema propuesto de configuración de registros en el LoRa para la recepción de datos utilizando la interrupción *RXDONE* y la detección de errores en la recepción de datos *PayloadCrcError*.

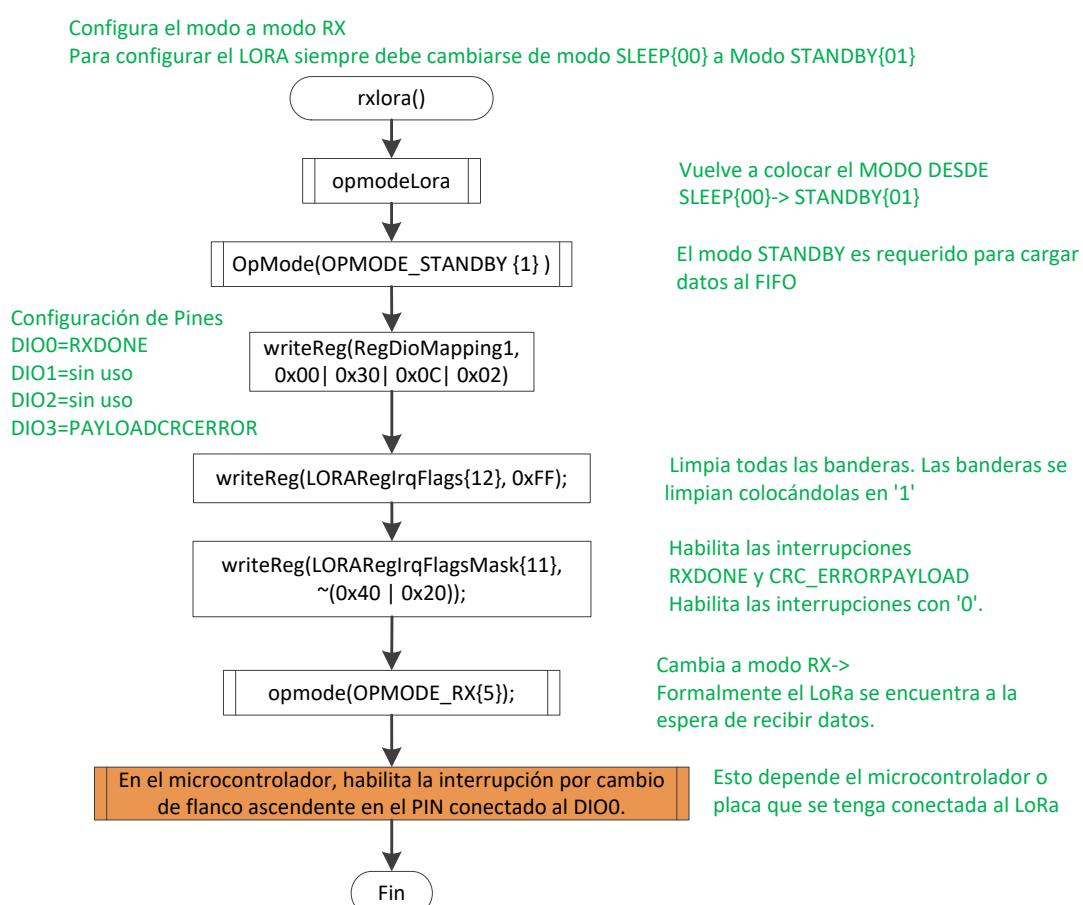


Figura 27. Esquema lógico de configuración en el LoRa para recepción de datos.
Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

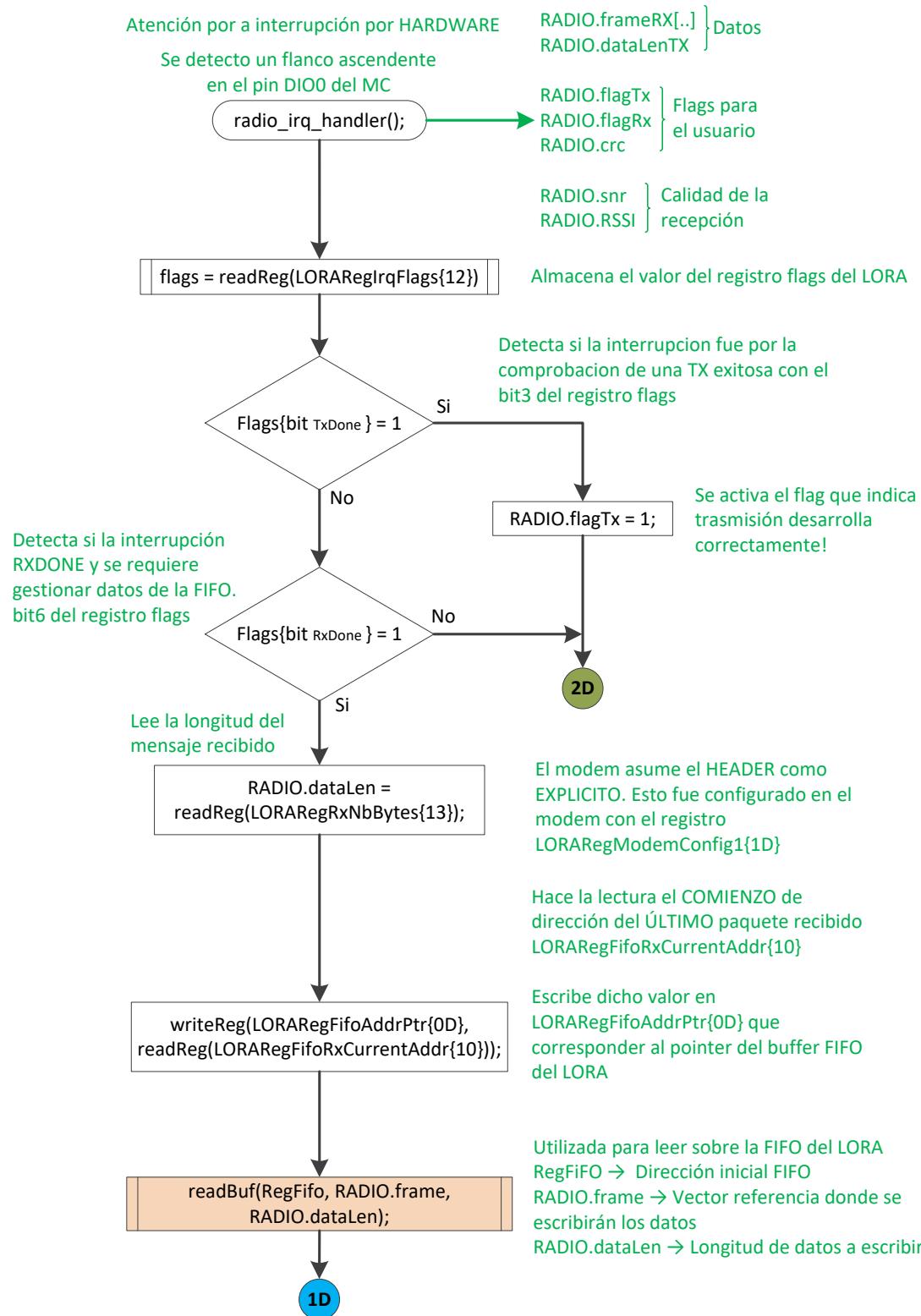


Figura 28. Esquema lógico de atención a la interrupción por recepción o transmisión de datos en el LoRa (parte I).

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

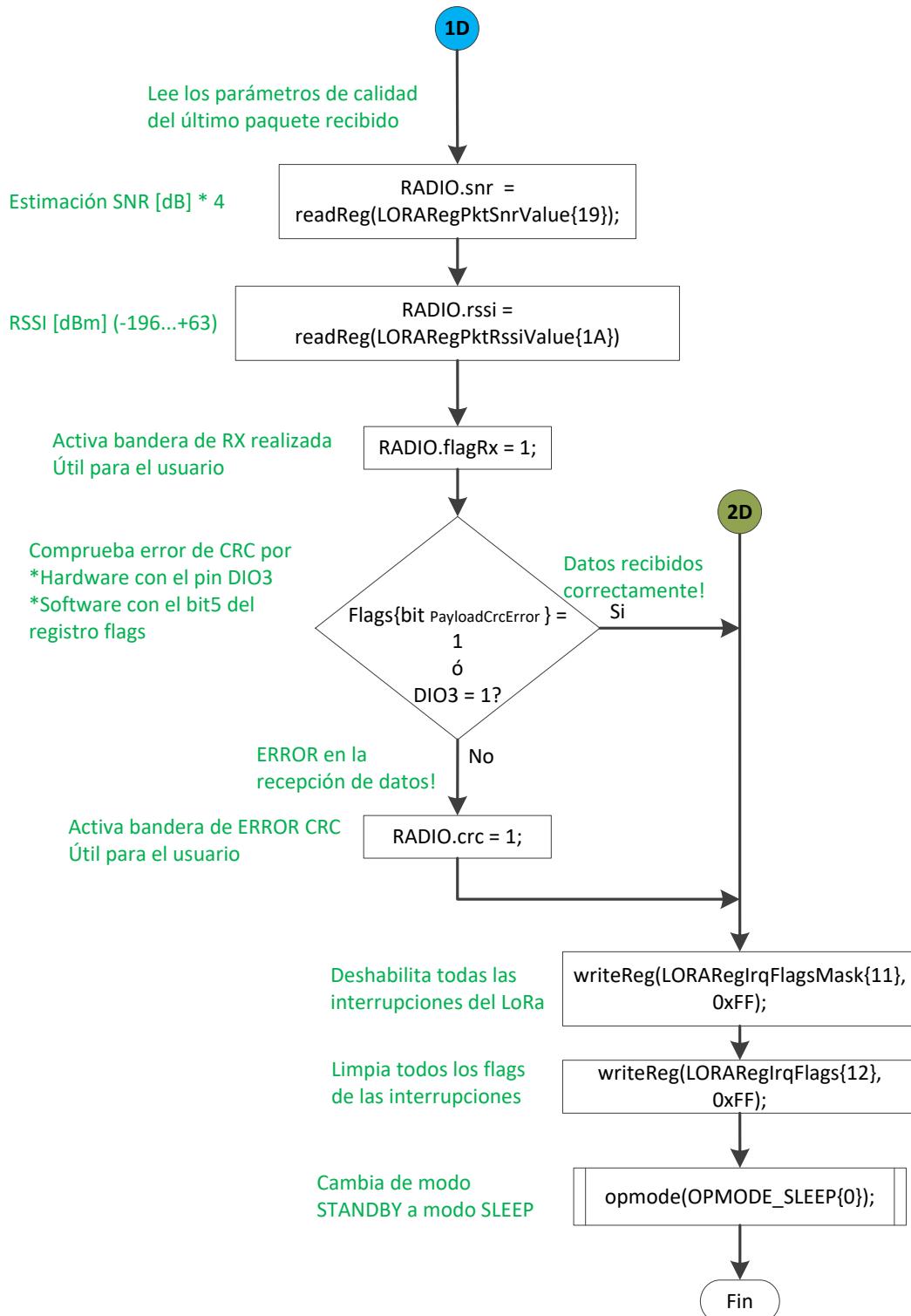


Figura 29. Esquema lógico de atención a la interrupción por recepción o transmisión de datos en el LoRa (parte II).

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

Es posible detectar la recepción de los datos comprobando el registro *RegIrqFlags_0x12* con el flag *RxDone* o por la detección de una interrupción de flanco ascendente por parte del pin DIO0 en el microcontrolador. Las *Figuras 28* y *29* presentan el esquema lógico propuesto para la detección de la interrupción generada por el pin DIO0 en el microcontrolador cuando se ha efectuado una transmisión correctamente o cuando se disponen de datos en la FIFO que deben ser gestionados por el microcontrolador.

Los parámetros generados por la interrupción corresponden a las variables:

- **RADIO.frameRX[..]:** vector de datos recibidos por el LoRa
- **RADIO.dataLenTX:** cantidad de datos recibidos
- **RADIO.flagTx:** Flag que le indica al usuario si se desarrolló correctamente una transmisión.
- **RADIO.flagRx:** Flag que le indica al usuario si se desarrolló la gestión de los datos recibidos.
- **RADIO.crc:** Flag que le indica al usuario si los datos recibidos están corruptos o contienen errores que debe gestión realizando nuevamente una petición.
- **RADIO.snr:** Calidad de la señal a ruido de los datos recibidos.
- **RADIO.rssi:** Indicador de fuerza de los datos recibidos.

Para leer datos de la FIFO del LoRa, se utiliza el registro *RegFifo_0x00* para indicar la posición de lectura de los datos. Para tener acceso a la FIFO, el chip radio debe estar en *modo STANDBY*. La *Figura 25* presenta el esquema lógico propuesto para extraer los datos de la FIFO luego de recibir datos. Para utilizar el esquema, se requiere que el usuario cree el vector *RADIO.frameRX* para el almacenamiento de los valores recibidos en la FIFO y cree la variable *RADIO.dataLenRX* que corresponderá a la cantidad o longitud de datos recibidos.

Utilizada para leer sobre la FIFO del LORA

RegFiFO → Dirección inicial FIFO

RADIO.frameRX → Vector referencia donde se escribirán los datos

RADIO.dataLenRX → Longitud de datos a escribir

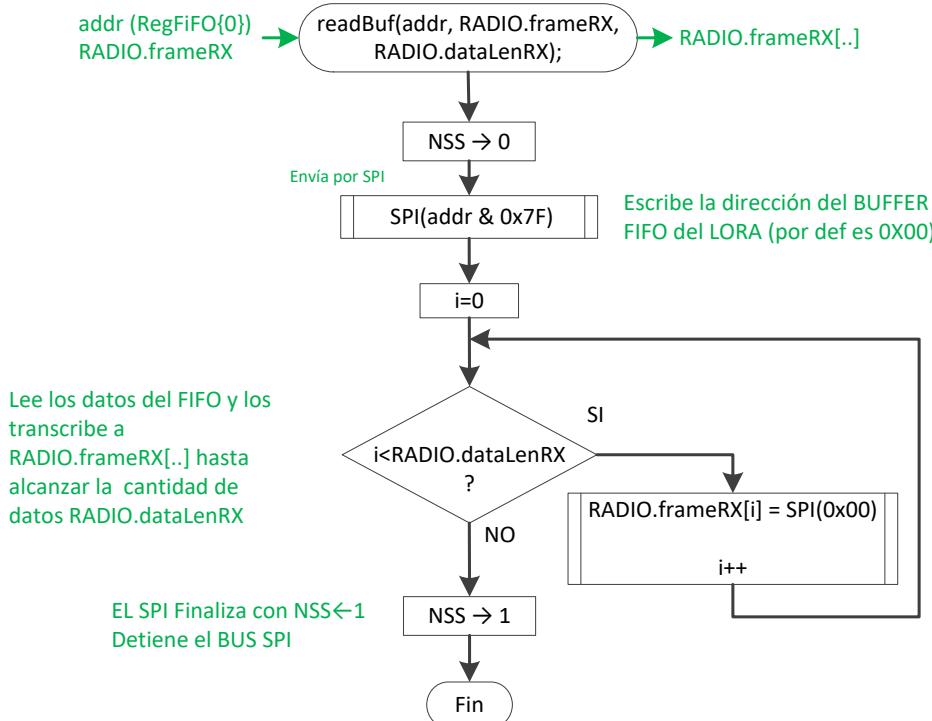


Figura 30. Esquema lógico de almacenamiento de datos sobre la FIFO del LoRa.

Fuente: Semtech SX1272 (2015) & Semtech SX1278 (2015). Desarrollado por el autor

Ejemplo de aplicación: red punto a punto (aplicación ping-pong)

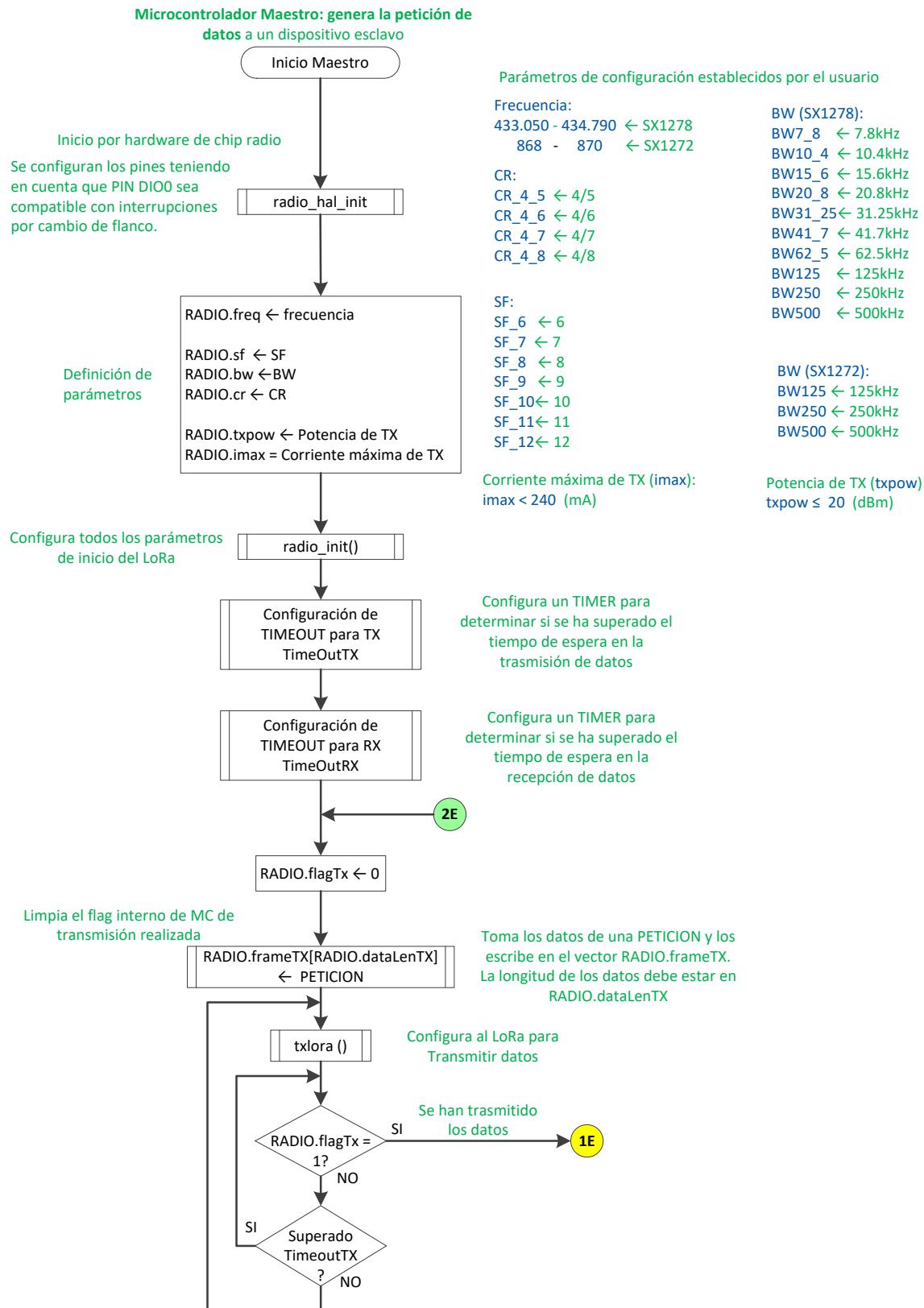


Figura 31. Esquema lógico para dispositivo maestro utilizando radio LoRa (parte I).

Fuente: Desarrollado por el autor

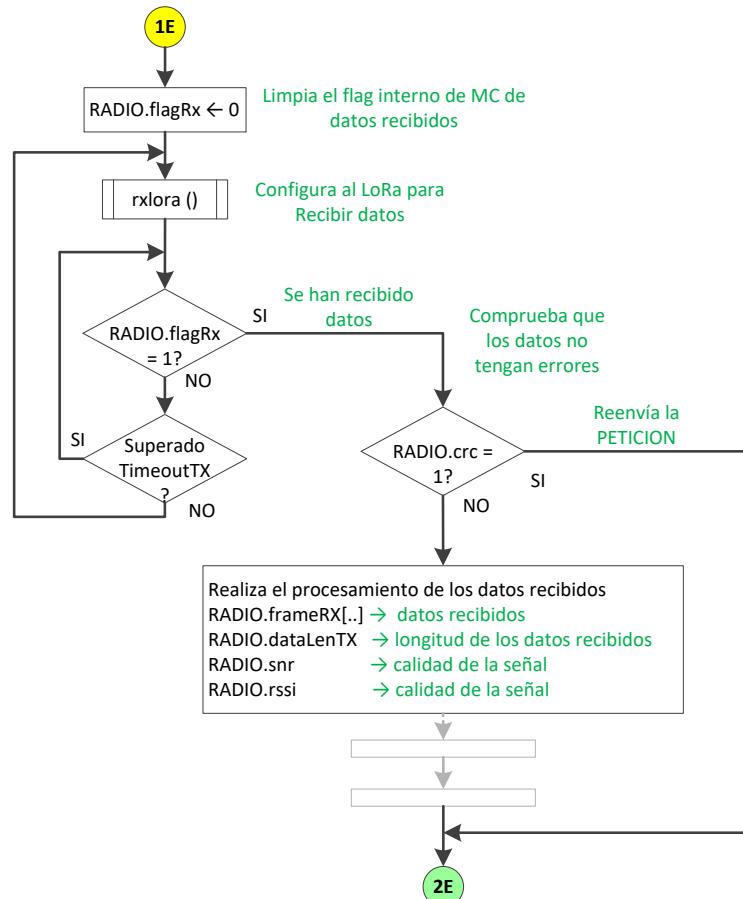


Figura 32. Esquema lógico para dispositivo maestro utilizando radio LoRa (parte II).
Fuente: Desarrollado por el autor

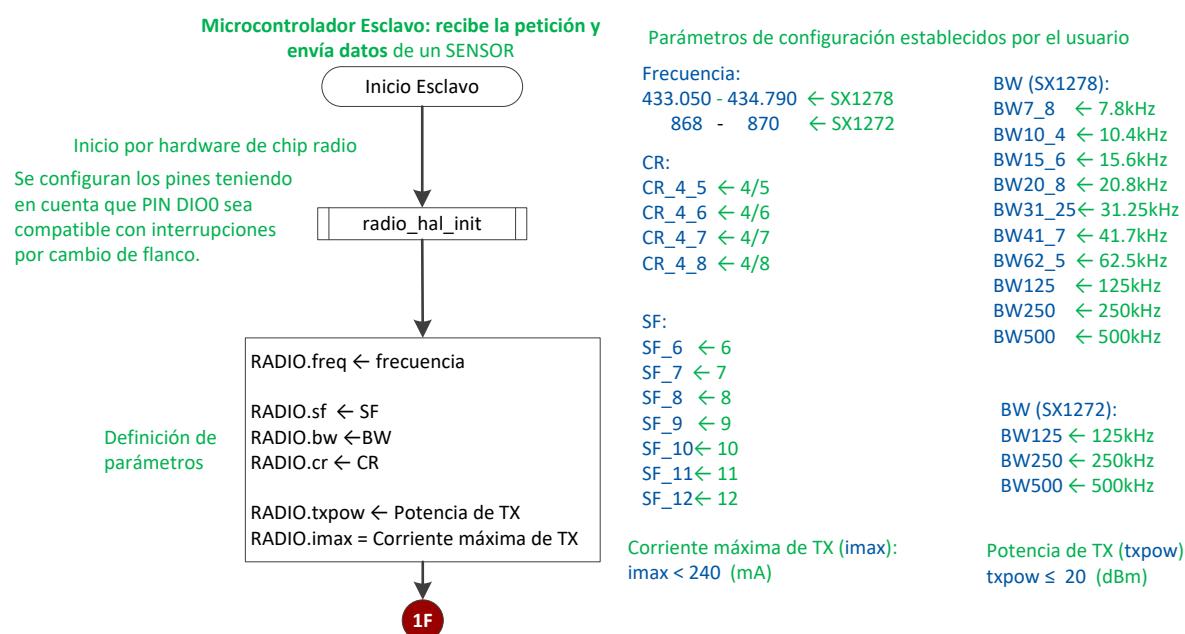


Figura 33. Esquema lógico para dispositivo esclavo utilizando radio LoRa (parte I).
Fuente: Desarrollado por el autor

Máster en Estrategias y Tecnologías para el Desarrollo

Título oficial interuniversitario de La Universidad Politécnica y Universidad Complutense de Madrid

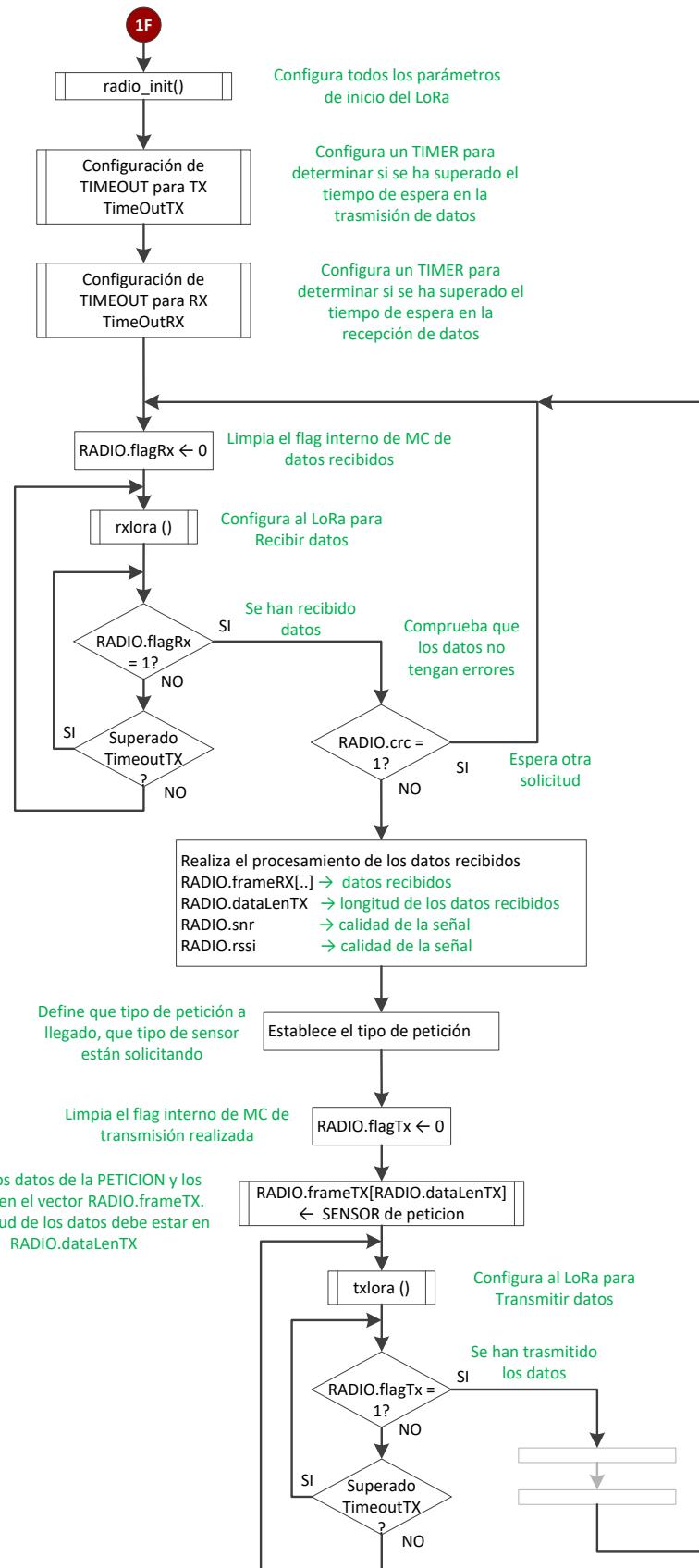


Figura 34. Esquema lógico para dispositivo esclavo utilizando radio LoRa (parte II).

Fuente: Desarrollado por el autor

Como ejemplo se plantea el uso de dos sistemas embebidos, uno utilizado como maestro y el otro utilizado como esclavo.

El dispositivo maestro se encarga de realizar la petición de datos a un dispositivo esclavo ubicado en un sitio remoto, el cual obtiene datos de sensores. El esclavo remite la información solicitada y continuamente se encuentra esperando peticiones del dispositivo esclavo. Las *Figuras 31* y *32* presentan el flujograma propuesto para la configuración del dispositivo maestro y las *Figuras 33* y *34* presentan el esquema propuesto de configuración para el dispositivo esclavo (sensor). Finalmente en la *Figura 35* se presenta un diagrama pictórico de la aplicación propuesta: maestro-esclavo (sensor) con la utilización de dispositivo LoRa.



Figura 35. Principio de funcionamiento “ping-pong” para la realización de pruebas de alcance en dispositivos LoRa.

Fuente: Desarrollado por autor

Ejemplo de aplicación: recepción de datos de varios dispositivos remotos (red punto - multipunto)

El ejemplo plantea un esquema de dispositivos conectados por una red punto-multipunto tipo estrella con un nodo central (circuito maestro) y 3 dispositivos esclavos. Como circuitos esclavos, se utilizan diversas versiones de sistemas embebidos.

La *Figura 36* presenta el esquema propuesto donde se aprecia la comunicación simultánea de 3 dispositivos hacia un nodo central. Se plantea el uso de esta aplicación con el uso del driver desarrollado para aplicaciones complejas con diversos sensores ubicados en sitios remotos. Cabe resaltar que los dispositivos LoRa deben ser de la misma referencia, estar configurados a la misma frecuencia y con los parámetros SF, BW y CR idénticos.

El dispositivo maestro se debe configurar como dispositivo “pasivo”, es decir continuamente recibir la información de los dispositivos esclavos. Una vez recibe la información de un dispositivo esclavo, remite la confirmación de recepción correcta de datos.

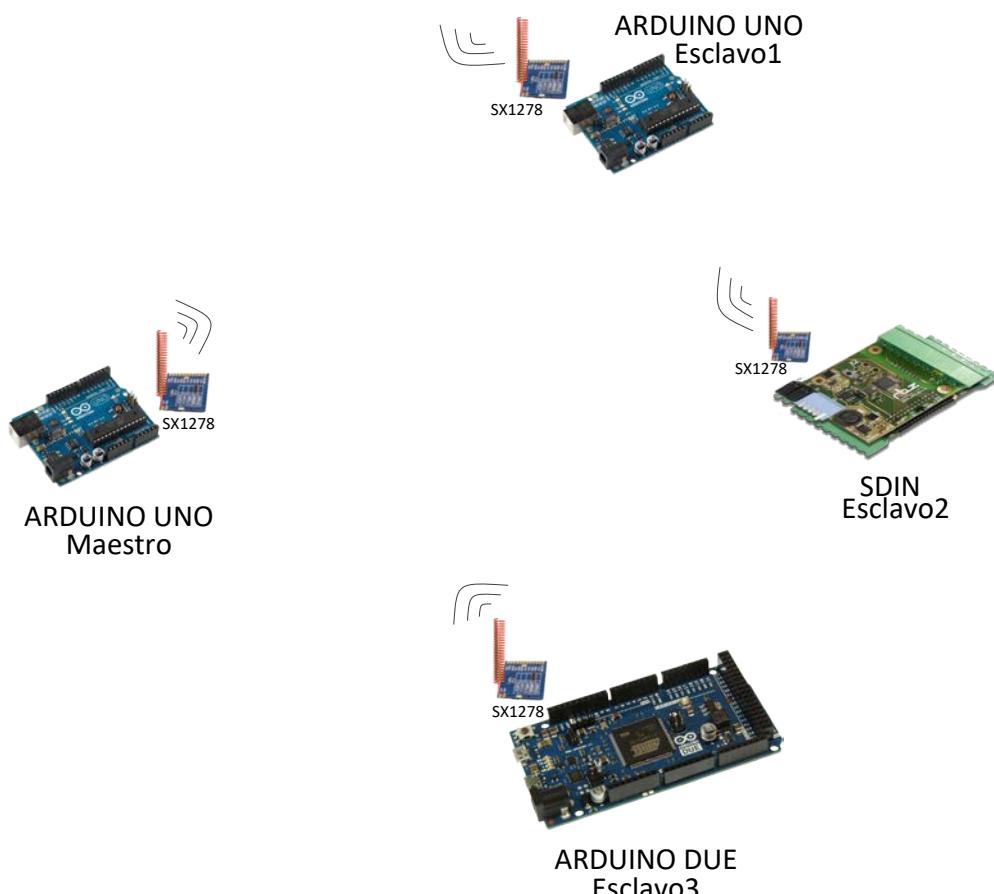


Figura 36. Tipología estrella para comunicación con diversos dispositivos.

Fuente: Esquema propuesto por el autor

El modem LoRa es *half dúplex*, es decir que los LoRa no puede trasmisir o recibir datos en el mismo instante; esto significa que los transceiver LoRa deben ser configurados para trasmisir o recibir datos. Cuando se incorpora más de dos dispositivos en un protocolo de comunicación *half dúplex*, se deben establecer estrategias que mitiguen los efectos adversos por colisión indeseada de datos cuando se reciben datos.

Si los dispositivos no cuentan con estrategias para evitar la transmisión de datos sobre el mismo canal de comunicación por parte de los dispositivos esclavos en un mismo instante, los datos llegarán errados al dispositivo maestro y la comunicación de datos será errada.

Este efecto se observa en las líneas de tiempo presentadas en la *Figura 37* donde los dispositivos esclavos tramiten información en un mismo instante y el dispositivo maestro no cuenta puede responder con la confirmación de comunicación efectuada, lo que ocasiona un consumo energético innecesario de energía en la transmisión de información por parte de los esclavos.

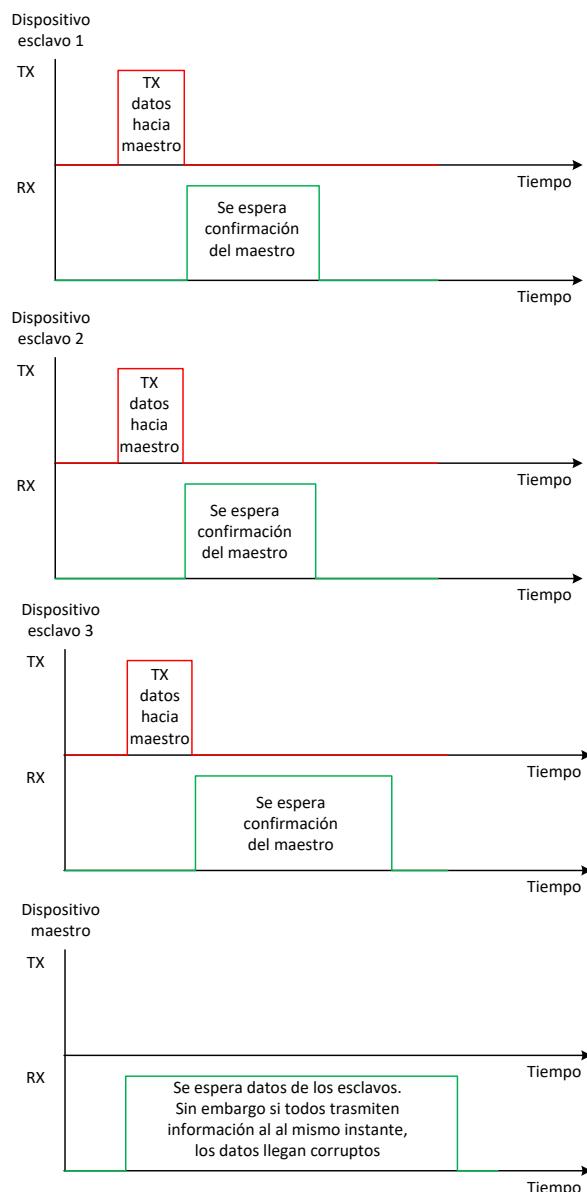


Figura 37. Líneas de tiempos para dispositivos esclavos y dispositivo maestro sin estrategias adecuadas para la transmisión de información.

Fuente: Esquema propuesto por el autor

La solución para evitar la colisión de datos, corresponde a establecer tiempos de muestreo y tiempos aleatorios de “des-sincronización” cuando se transmiten datos; es decir evitar con los tiempos adicionales de transmisión de datos la colisión de datos.

Una vez se establece el tiempo en el que no se presentan colisiones, se almacena dicho valor para ser utilizado consecutivamente en las próximas trasmisiones de datos y con lo anterior el sistema *half dúplex* se comporta sincronizado como se presenta en la Figura 38.

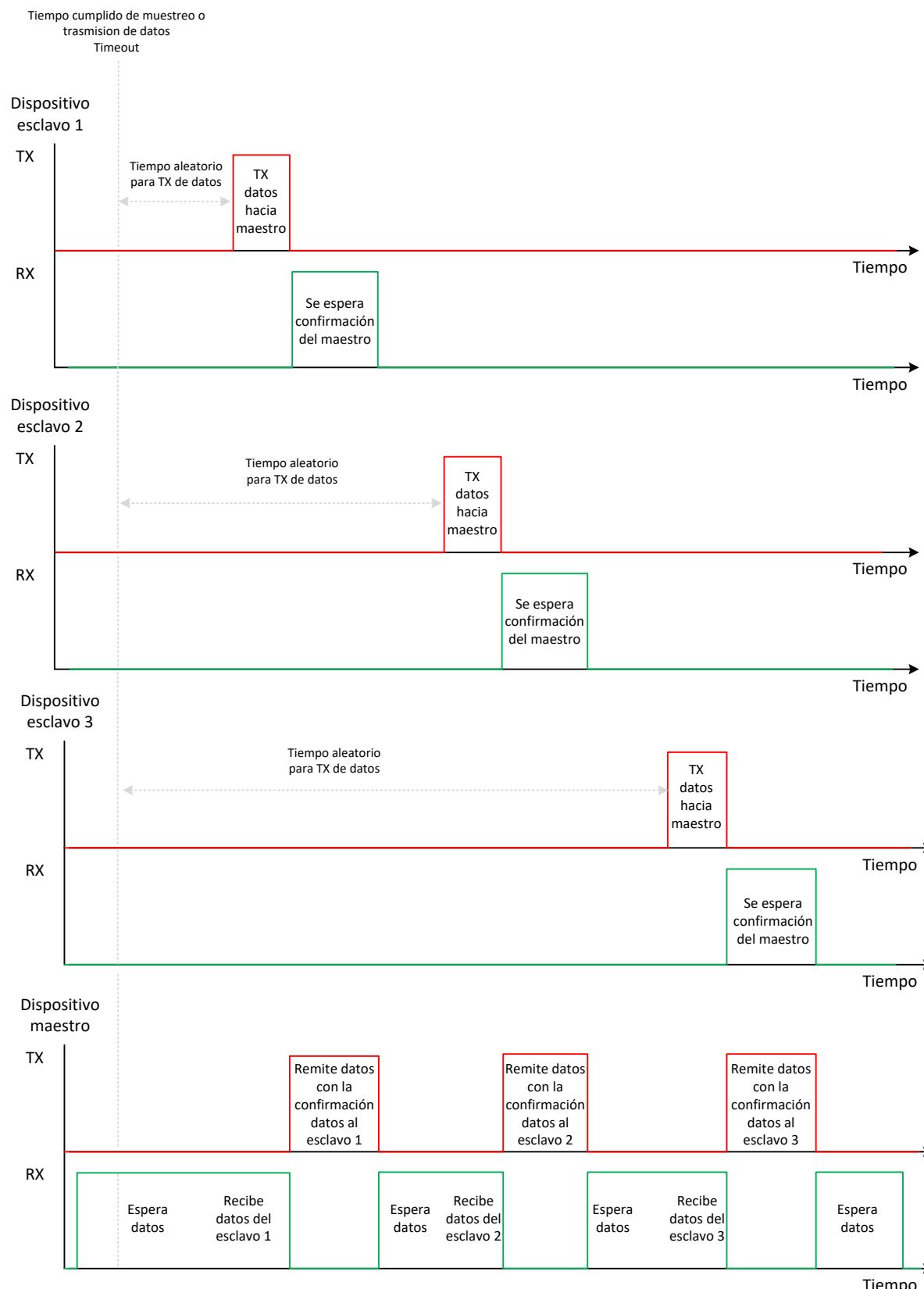


Figura 38. Líneas de tiempos para dispositivos esclavos y dispositivo maestro con tiempos aleatorios para la transmisión adecuada de datos.

Fuente: Esquema propuesto por el autor

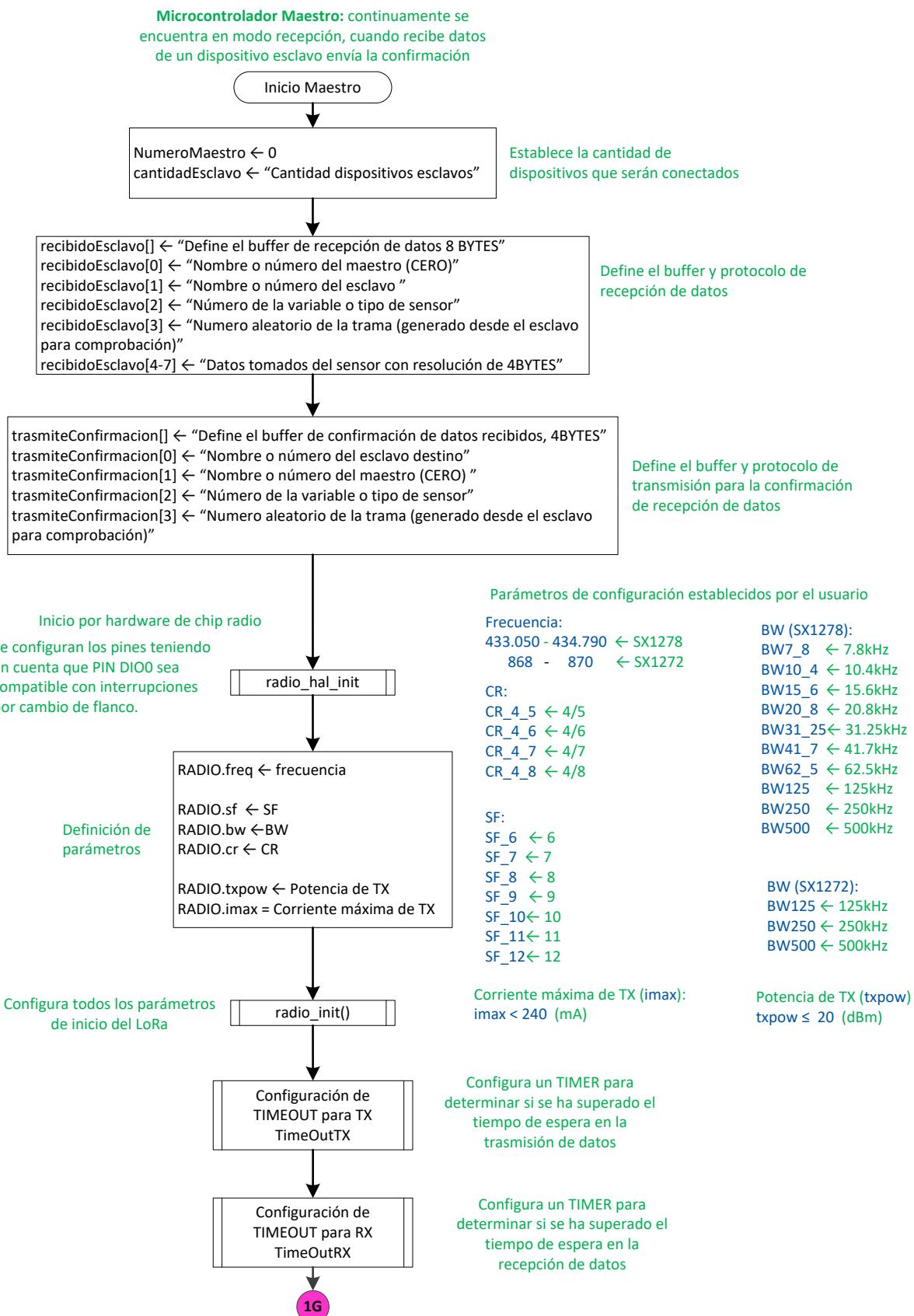


Figura 39. Esquema lógico para dispositivo maestro en aplicación punto-multipunto (parte I).

Fuente: Desarrollado por el autor

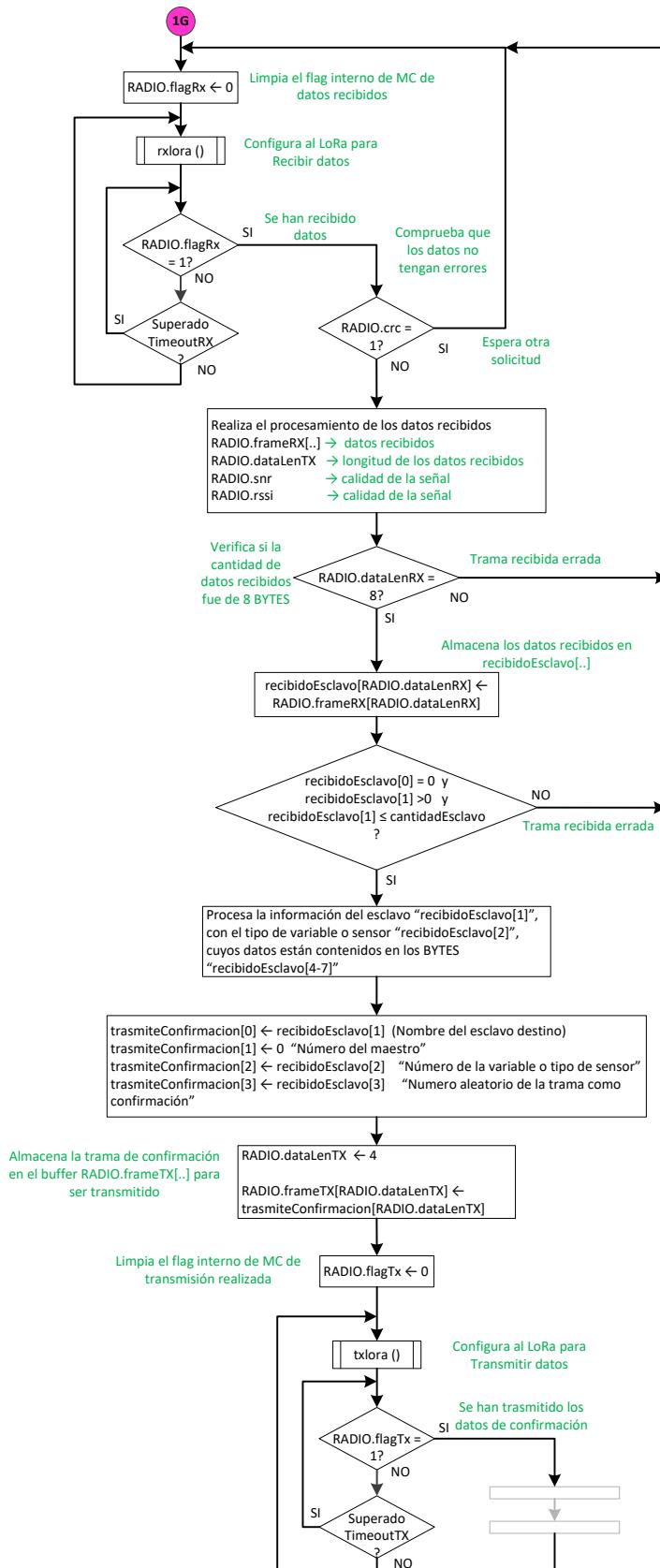


Figura 40. Esquema lógico para dispositivo maestro en aplicación punto-multipunto (parte II).

Fuente: Desarrollado por el autor

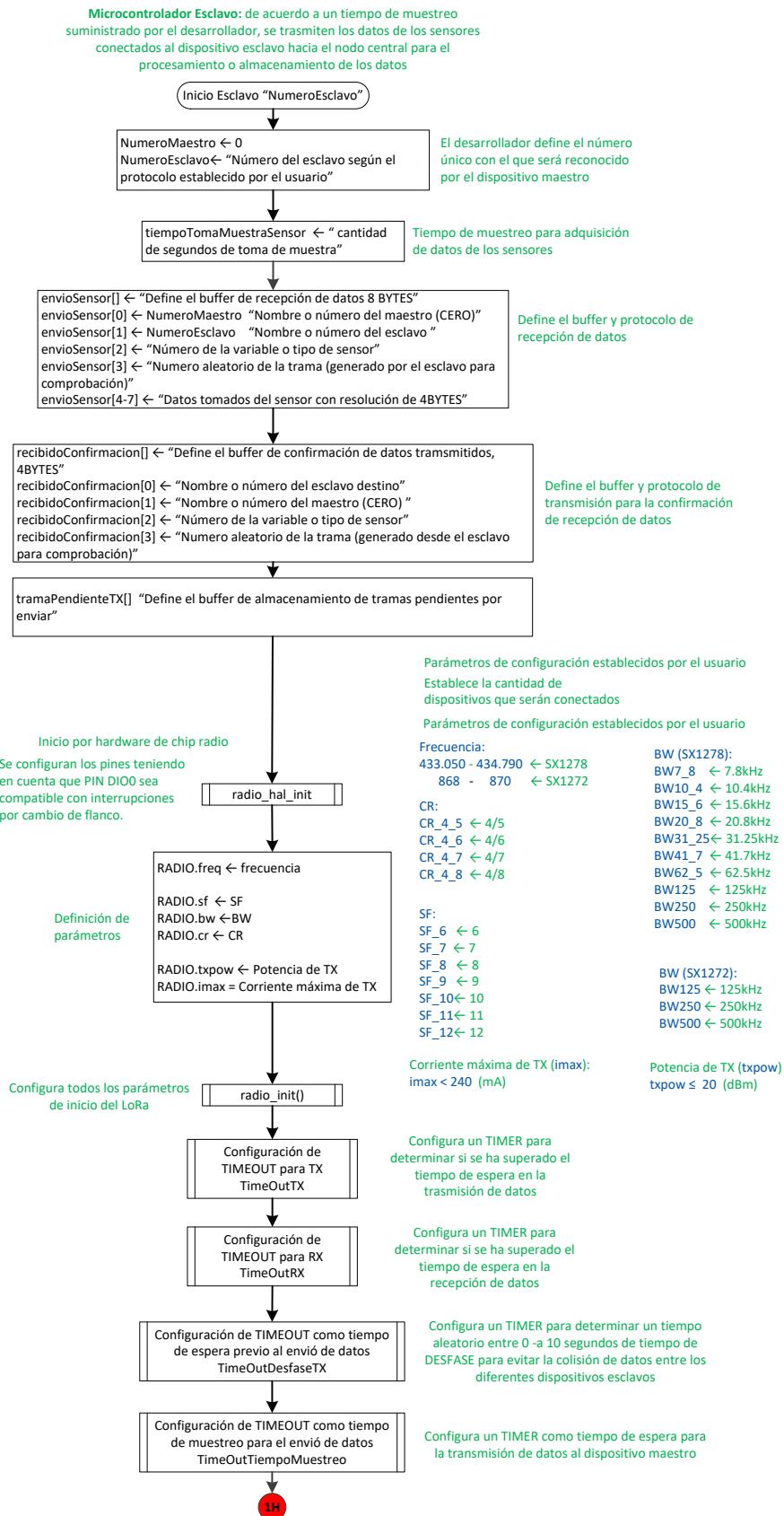


Figura 41. Esquema lógico para dispositivos esclavo en aplicación punto-multipunto (parte I).

Máster en Estrategias y Tecnologías para el Desarrollo

Título oficial interuniversitario de La Universidad Politécnica y Universidad Complutense de Madrid



POLITÉCNICA

Fuente: Desarrollado por el autor

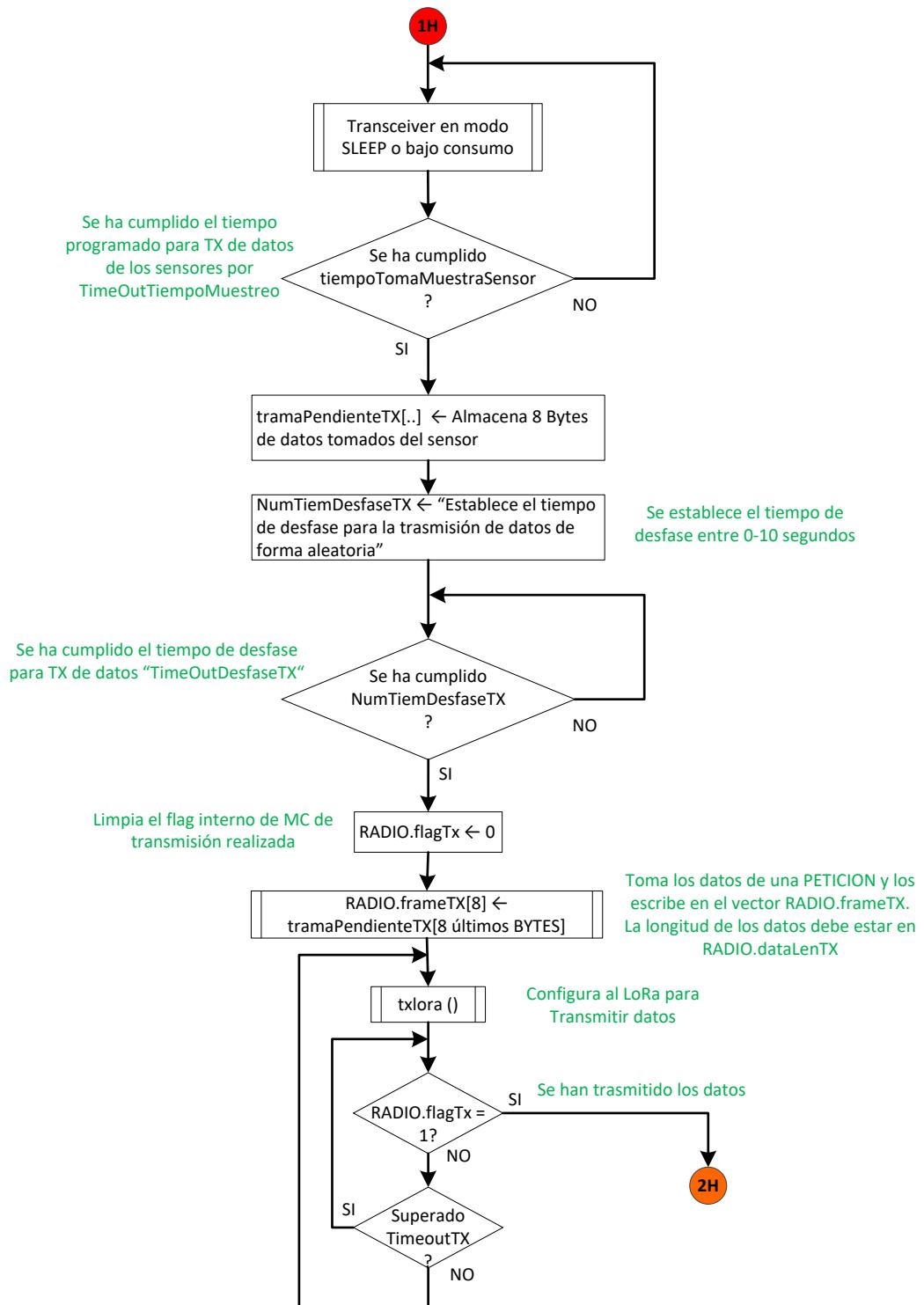


Figura 42. Esquema lógico para dispositivos esclavo en aplicación punto-multipunto (parte II).

Fuente: Desarrollado por el autor

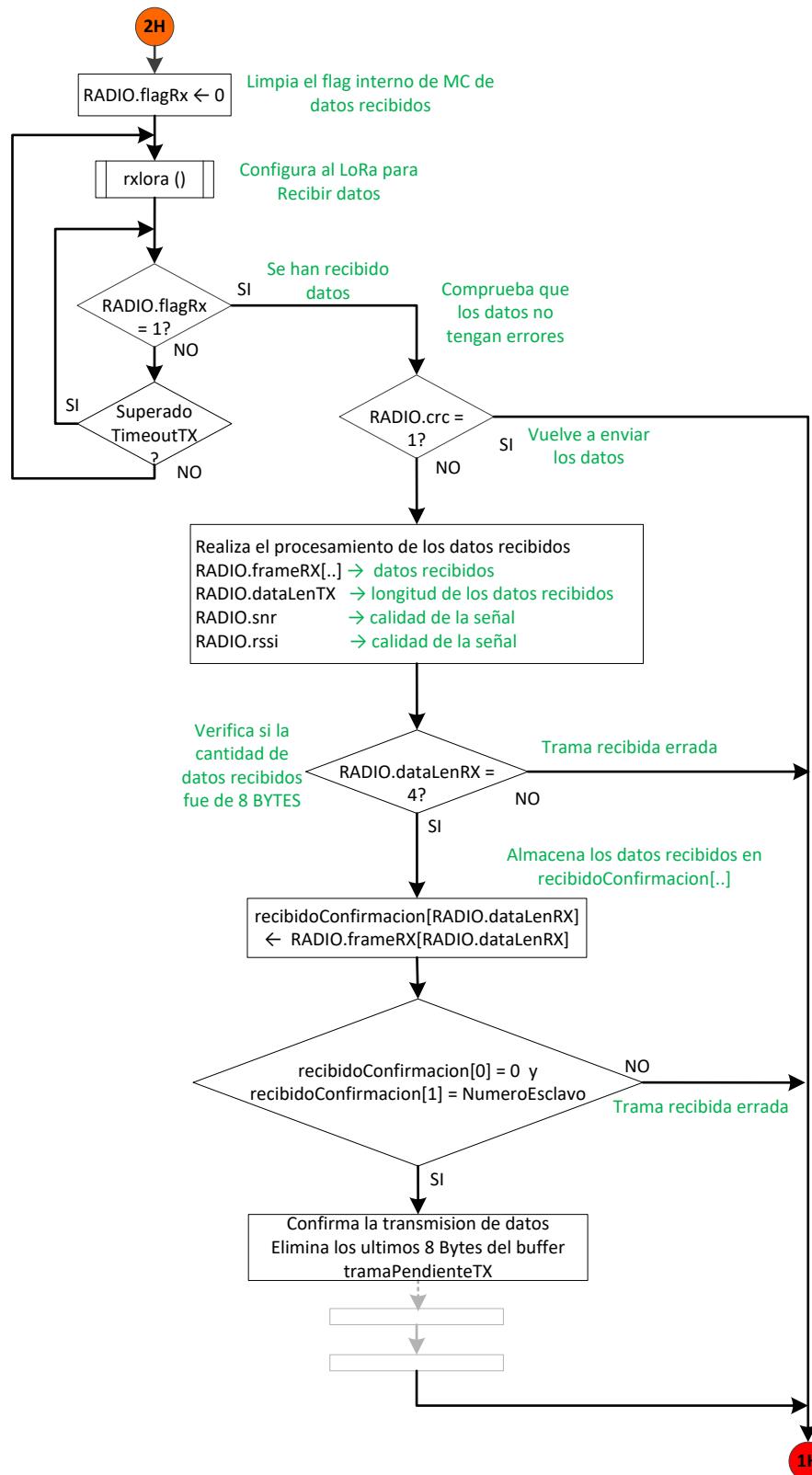


Figura 43. Esquema lógico para dispositivos esclavo en aplicación punto-multipunto (parte III).

Fuente: Desarrollado por el autor

El esquema presentado en las *Figuras 39 y 40* representan la lógica de programación planteada en el flujograma utilizado en sistema embebido maestro. Este dispositivo se encuentra continuamente en modo recepción de datos, cuando recibe la transmisión de datos de dispositivos esclavos, procesa la información y retorna la confirmación. El dispositivo esclavo se encarga de determinar los valores de los sensores remotos, una vez se cumple con el tiempo de muestreo o captura de los valores de los sensores, se tramite al dispositivo maestro. En caso que dos o más dispositivos esclavos transmitan al mismo instante, el dispositivo maestro no responderá la confirmación y en el próximo tiempo cumplido de muestreo se trasmitirán los valores pendientes por trasmitir. El flujograma propuesto del dispositivo esclavo se presentan en las *Figuras 41, 42 y 43*.

BIBLIOGRAFÍA

Dorji (2015). DRF1278F - 20dBm LoRa Long Range RF Front-end Module. Version 1.11. China.

NiceRF (2015). LoRa1278 100 mW 4 km larga distancia y alta sensibilidad (-139 dBm) 433 MHz módulo de transceptor inalámbrico. Recuperado de <http://es.aliexpress.com/item/2pcs-lot-LoRa1278-100mW-4km-Long-Distance-and-High-Sensitivity-139-dBm-433MHz-Wireless-Transceiver-Module/32461365864.html>, China. Consultado el 22 de junio de 2016.

Semtech (2013). SX1272 Development kit. User guide. Semtech Corporation. Estados Unidos

Semtech SX1272 (2015). SX1272/73 - 860 MHz to 1020 MHz Low Power Long Range Transceiver. Datasheet. Semtech Corporation. Estados Unidos.

Semtech SX1278 (2015). SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver. Datasheet. Semtech Corporation. Estados Unidos.

Texas Instruments (2005). SM-Band and Short Range Device Regulatory Compliance Overview. Estados Unidos.

Máster en Estrategias y Tecnologías para el Desarrollo

Título oficial interuniversitario de La Universidad Politécnica y Universidad Complutense de Madrid



ANEXO B

Instalación y uso del driver LoRa para sistemas embebidos.

JOSÉ DANIEL RODRÍGUEZ MUNCA

Tutor Académico: Manuel Sierra Castañer

Tutor Profesional: Álvaro Gutiérrez Martín

UNIVERSIDAD DE POLITÉCNICA DE MADRID
UNIVERSIDAD COMPLUTENSE DE MADRID
MASTER INTERUNIVERSITARIO EN ESTRATEGIAS Y TECNOLOGÍAS PARA
EL DESARROLLO
MADRID - ESPAÑA
2016

TABLA DE CONTENIDO

INSTALACIÓN Y USO DEL DRIVER LORA PARA SISTEMAS EMBEBIDOS ARDUINO	171
Estructura de archivos que componen el driver.....	171
Instalación de la librería para el uso con Arduino	173
Uso de librería para el control de dispositivos LoRa en aplicaciones o sketch de Arduino	173
Ejemplo de aplicación: red punto a punto (aplicación ping-pong).....	178
Ejemplo de aplicación: recepción de datos de varios dispositivos remotos (red punto - multipunto)	180
INSTALACIÓN Y USO DEL DRIVER LORA PARA SISTEMAS EMBEBIDOS SDIN .	183
Estructura de archivos que componen el driver.....	183
Uso de librería para el control de dispositivos LoRa en aplicaciones para SDIN.....	186
Ejemplo de aplicación: red punto a punto (aplicación ping-pong) con módulos SDIN....	188
Ejemplo de aplicación: recepción de datos de varios dispositivos remotos (red punto - multipunto) con módulos SDIN	190
BIBLIOGRAFÍA.....	193

LISTADO DE FIGURAS

<i>Figura 1. Estructura de archivos y carpetas del driver propuesto para el manejo de dispositivos LoRa utilizando Arduino.</i>	171
<i>Figura 2. Gestión entre archivos del driver propuesto para el manejo de dispositivos LoRa.</i>	172
<i>Figura 3. Base de aplicación o “sketch” para utilizar transceiver LoRa con IDE Arduino.</i>	177
<i>Figura 4. Principio de funcionamiento de aplicación “ping-pong”.</i>	179
<i>Figura 5. Esquema de conexión para LoRa1278 de NiceRF y Arduino UNO.</i>	179
<i>Figura 6. Tipología estrella para comunicación con diversos dispositivos Arduino.</i>	181
<i>Figura 7. Estructura de archivos y carpetas del driver propuesto para el manejo de dispositivos LoRa utilizando FreeRTOS –SDIN.</i>	183
<i>Figura 8. Gestión entre archivos del driver propuesto para el manejo de dispositivos LoRa para los módulos SDIN.</i>	185
<i>Figura 9. Principio de funcionamiento de aplicación “ping-pong” con SDIN.</i>	189
<i>Figura 10. Esquema de conexión para LoRa1278 de NiceRF y módulo SDIN.</i>	189
<i>Figura 11. Tipología estrella para comunicación con diversos dispositivos.</i>	191

LISTADO DE TABLAS

<i>Tabla 1. Instrucción para selección de transceiver LoRa en archivos radio.h.</i>	173
<i>Tabla 2. Banda de frecuencia ISM compatibles con los transceivers de radio LoRa</i>	175
<i>Tabla 3. Combinación de los principales parámetros configurables para los LoRa</i>	176
<i>Tabla 4. Comportamiento de los registros tipo flag o banderas de los estados finales de transmisión o recepción de datos para los LoRa</i>	178
<i>Tabla 5. Nomenclatura utilizada para la conexión entre la tarjeta LoRa1278 de NiceRF y la tarjeta Arduino UNO</i>	180
<i>Tabla 6. Nomenclatura utilizada para la conexión entre la tarjeta LoRa1278 de NiceRF y el módulo SDIN</i>	190

INSTALACIÓN Y USO DEL DRIVER LORA PARA SISTEMAS EMBEBIDOS ARDUINO

Estructura de archivos que componen el driver

El driver está compuesto por una serie de archivos con *código fuente* y archivos tipo *header* organizados en dos subcarpetas. La estructura organizada de archivos se presenta en la *Figura 1*.

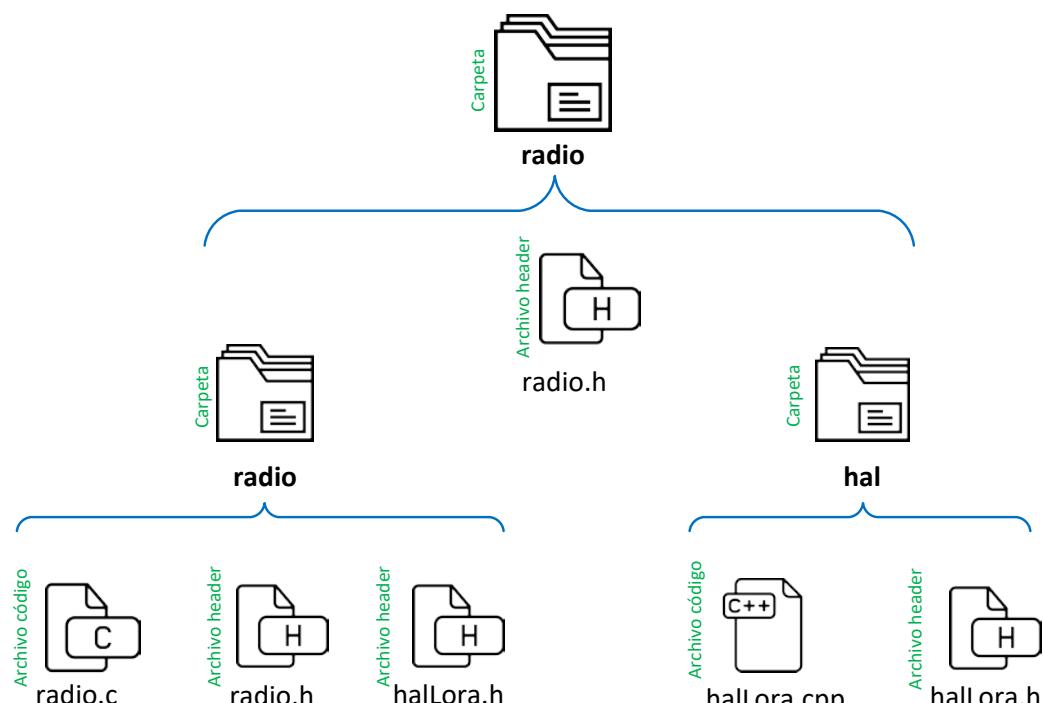


Figura 1. Estructura de archivos y carpetas del driver propuesto para el manejo de dispositivos LoRa utilizando Arduino.

Fuente: Esquema propuesto por el autor

La carpeta principal de denomiña **radio** y está compuesta por los elementos:

- **Carpeta radio:** contiene los archivos necesarios para el funcionamiento del driver en aplicaciones o “*sketch*” para Arduino.
 - **radio.h:** el *header file* contiene la definición o selección del transceiver LoRa utilizado por la aplicación de Arduino. Es suficiente con definir:

Instrucción	Comentario
#define CFG_1272_SEMTECH 1	Selecciona el CHIP SX1272 de SEMTECH
#define CFG_1278_DRF1278F 1	Selecciona el CHIP SX1278 de DORJI versión de tarjeta DRF1278F
#define CFG_1278_NICERF1278 1	Selecciona el CHIP SX1278 de NiceRF versión de tarjeta LORA1278

- **Carpeta radio:** la carpeta contiene el código fuente, junto la definición de variables y funciones necesarias para la configuración de los dispositivos LoRa.
 - **radio.h:** contiene la definición de variables y funciones para la configuración de los LoRa a nivel de software relacionado con el código presentado en **radio.c**.
 - **radio.c:** contiene las funciones para la configuración de los LoRa a nivel de software.
 - **halLora.h:** cuenta con la definición de funciones del archivo **halLoRa.cpp** (*HAL o hardware abstraction layer*).
- **Carpeta hal:** la carpeta contiene el código fuente relacionado con la capa de más bajo nivel que configura los periféricos internos de Arduino.
 - **halLora.h:** cuenta con la definición de la nomenclatura asignada a los pines o terminales de control de los dispositivos LoRa.
 - **halLora.cpp:** contiene el código fuente de la capa de más bajo nivel que configura los periféricos internos de arduino (interrupciones por cambio de flanco, comunicación SPI, comunicación UART, cambios de estados en terminales). A esta capa se le conoce como *HAL* o capa de abstracción de hardware.

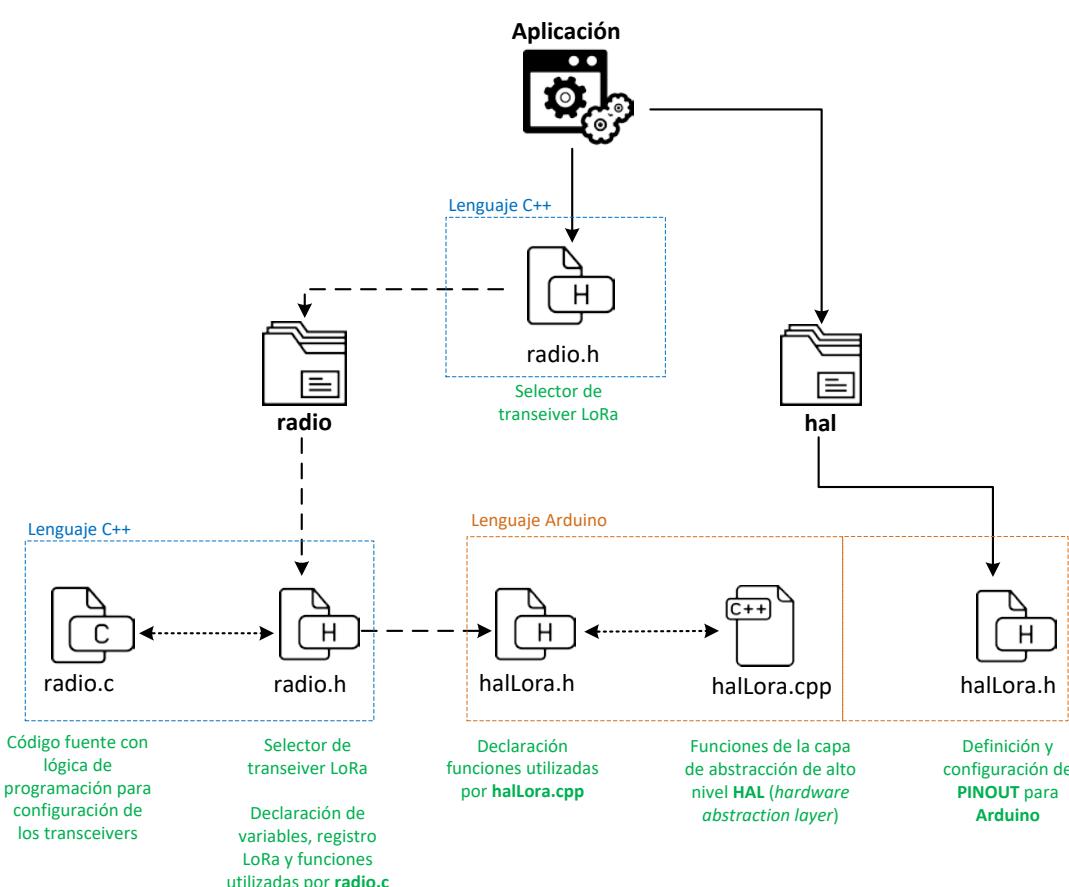


Figura 2. Gestión entre archivos del driver propuesto para el manejo de dispositivos LoRa.

Fuente: Esquema propuesto por el autor

La *Figura 2* presenta la gestión entre los diferentes archivos que componen el driver planteado.

Resulta estratégica la organización de los diferentes archivos debido a que el driver busca que el código presentado en **radio.c** se comporte como una capa de alto nivel, manteniendo el código invariante al uso de diferentes sistemas embebidos como Arduino, módulos SDIN o tarjetas de STMicroelectronics; mientras que el código presentado en **halLora.cpp** corresponde a una capa de alto bajo nivel de abstracción de hardware que de acuerdo al tipo de sistema embebido utilizado pueda ser ajustado para realizar la correcta homologación al microcontrolador utilizado.

Instalación de la librería para el uso con Arduino

La instalación del driver para el control de los dispositivos LoRa utilizando Arduino (se recomienda que el equipo de cómputo tenga instalado el IDE Arduino), se realiza copiando los archivos de la carpeta principal **radio** en la carpeta *libraries* de la instalación del *IDE arduino*:

<i>Ruta por defecto para Linux (ubuntu)</i>	<i>Ruta por defecto para Windows</i>
/usr/share/arduino/libraries	C:\Program Files (x86)\Arduino\libraries
	C:\Program Files\Arduino\libraries

Una vez copiado los archivos en la carpeta *libraries* de la *instalación del IDE arduino*, el usuario debe seleccionar el dispositivo LoRa que será utilizado por Arduino. Para seleccionar el transceiver, se debe habilitar una de las siguientes instrucciones en el archivo **radio/radio.h**:

Tabla 1. Instrucción para selección de transceiver LoRa en archivos radio.h.

Fuente: Planteamiento del autor

Instrucción	Selección de transeiver LoRa
#define CFG_1272_SEMTECH 1	SX1272 de SEMTECH
#define CFG_1278_DRF1278F 1	SX1278 de DORJI versión de tarjeta DRF1278F
#define CFG_1278_NICERF1278 1	SX1278 de NiceRF versión de tarjeta LORA1278

Uso de librería para el control de dispositivos LoRa en aplicaciones o sketch de Arduino

Una vez copiado los archivos del driver en la carpeta donde se ha instalado el IDE Arduino, el usuario debe crear una aplicación o sketch y seguir los siguientes pasos:

- Incluir la librerías para control por software (**radio**) y hardware (**halLora**):

```
#include <radio.h>           //Librería de alto nivel para configuración por software de los radio LoRa
#include <hal/halLora.h>       //Librería de bajo nivel para configuración de hardware radio LoRa (HAL)
#include <SPI.h>              //Librería para manejo de comunicación SPI
```

- Definir los pines de control de la tarjeta Arduino con respecto a los transceiver LoRa:

//Definición de los pines de control de la tarjeta Arduino

```
const radio_pinmap radio_pins = {
    .nss =      [número de pin],
    .rxtx = RADIO_UNUSED_PIN,
    .rst =      [número de pin],
    .dio0 =     [número de pin],
```

```
#if (defined CFG_1278_DRF1278F) || (defined CFG_1272_SEMTECH) //Selección la versión del SX1278
    .dio3 =     [número de pin],
#endif
#if defined(CFG_1278_NICERF1278)                                //Selecciona el CHIP SX1278 de NICERF
    .tx_en =    [número de pin],
    .rx_en =    [número de pin],
#endif
};
```

- Configurar los dispositivos LoRa de acuerdo a las necesidades del usuario. Esto se debe incorporar en la función propia de Arduino “**void setup()**”:

- Iniciar las variables tipo *flag* o *banderas*:

//Inicialización de variables del driver

```
RADIO.flagTx = 0;
RADIO.flagRx = 0;
RADIO.crc = 0;
```

- Definir la frecuencia de trabajo del transceiver:

RADIO.freq = [frecuencia de trabajo]; //Frecuencia de la banda ISM compatible con LoRa

Teniendo en cuenta que existen bandas de frecuencias de uso privativo (telefonía, internet, radio, entre otras), existen frecuencias de libre uso conocidas como ISM (Industrial, Scientific and Medical) que son frecuencias intencionalmente libres para uso no comercial en aplicaciones de tipo industrial, científico y médico. Se recomienda al usuario verificar la compatibilidad de la frecuencia de trabajo con la *Tabla 2* (bandas de uso ISM).

Tabla 2. Banda de frecuencia ISM compatibles con los transceivers de radio LoRa

Fuente: Texas Instruments (2005)

Banda ISM (MHz)	Frecuencia mínima (MHz)	Frecuencia máxima (MHz)	Ancho de Banda (MHz)	Usos por continente
433	433.050	434.790	1.84	Europa, África y parte del norte de Asia.
869	868	870	2	Europa, África, Asia y Oceanía.
900	902	928	26	Continente americano

- Definir la potencia de transmisión del transceiver:

//Configuración de potencia

RADIO.txpow = [Máxima potencia (dBm)]

//Máxima TX potencia

RADIO.imax = [Corriente máxima (mA)]

//Por defecto 100mA 45mA <=Imax <= 240mA

La potencia máxima se expresa en dBm y el valor definido debe estar entre 2 a 17.

La corriente máxima está dada en mA y el valor definido se debe encontrar entre 45mA y 240mA. El valor típico de corriente es de 100mA. Para valores significativos de corrientes (mayores a 125mA), se recomienda verificar la corriente máxima de suministro soportada por la fuente o batería de alimentación. Algunos sistemas embebidos como Arduino cuentan con reguladores con corriente máxima de suministro hasta de 120mA, por lo que al realizar una inadecuada configuración en el LoRa, podría destruirse el regulador y dañar la placa del sistema embebido.

- Definir las variables de configuración de la comunicación entre los transceivers LoRa:

//Variables de configuración de la comunicación LoRa

RADIO.sf = [Spread factor SF] //Configuración del Spread factor

RADIO.bw = [Ancho del canal BW] //Configuración del ancho de canal

RADIO.cr = [Coding rate CR] //Configuración de Coding rate

Los dispositivos LoRa cuentan con los parámetros *SF*, *BW* y *CR* que según las combinaciones entre estos, permite determinar el alcance o distancia máxima de recepción de datos, la velocidad de los datos y sobrecarga por corrección o detección de errores en los datos. Estos valores deben ser los mismos en todos los dispositivos radio LoRa de la red inalámbrica.

Los parámetros independientes de configuración se definen como (Semtech SX1272, 2015 & Semtech SX1278, 2015):

BW: Ancho de banda (entre más bajo sea este valor, el tiempo de la transmisión será mayor)

SF rate: Es el factor de alcance expresado como logaritmo de base 2. Cuanto mayor sea este valor, mejor rendimiento tendrá la transmisión de datos.

CR: Tasa de codificación de errores: Entre mayor sea este valor, mayor será la fiabilidad de los datos, pero con una sobrecarga en el tiempo de transmisión.

La *Tabla 3* presenta la combinación de los principales variables independientes para la configuración de la comunicación entre transceivers LoRa. Cabe aclarar que estos valores deben ser los mismos en todos los dispositivos radio LoRa de la red inalámbrica.

Tabla 3. Combinación de los principales parámetros configurables para los LoRa

Fuente: Semtech SX1278 (2015) & Semtech SX1272 (2015)

Ancho de banda BW (kHz)		Factor de alcance SF		Tasa de codificación de errores. CR	
Hardware	Software RADIO.BW	Hardware	Software RADIO.sf	Hardware	Software RADIO.cr
500	BW500	6 → 64 chips / symbol (modo utilizado para FSK)	SF_6	4/5	CR_4_5
250	BW250	7 → 128 chips / symbol	SF_7	4/6	CR_4_6
125	BW125	8 → 256 chips / symbol	SF_8	4/7	CR_4_7
62.5*	BW62_5*	9 → 512 chips / symbol	SF_9	4/8	CR_4_8
41.7*	BW41_7*	10 → 1024 chips / symbol	SF_10		
31.25*	BW31_25*	11 → 2048 chips / symbol	SF_11		
20.8*	BW20_8*	12 → 4096 chips / symbol	SF_12		
15.6*	BW15_6*				
10.4*	BW10_4*				
7.8*	BW7_8*				

*: Aplica solamente para los transceiver SX1278

- Configurar los pines I/O y la comunicación SPI:

```
hal_init() //Configuración de puertos de Arduino
```

- Configurar los pines I/O y la comunicación SPI:

```
radio_init(); //Configuración de inicio para los LoRa
```

Corresponde a la función general que configura todos los parámetros de inicio del transceiver LoRa.

- Crear la aplicación con la lógica propuesta por el desarrollador:

```
void loop()
{
    [Lógica de control implementada por el usuario]
}
```

```

SKETCH

/*
* Definiciones necesarias para utilizar driver generado
***** */

#include <radio.h>                                //Librería de alto nivel para configuración por software de los radio LoRa
#include <hal/halLora.h>                            //Librería de bajo nivel para configuración de hardware radio LoRa (HAL)
#include <SPI.h>                                    //Librería para manejo de comunicación SPI

//Definición de los pines de control de la tarjeta Arduino
const radio_pinmap radio_pins = {
    .nss =      [número de pín],
    .rxtx = RADIO_UNUSED_PIN,
    .rst =       [número de pin],
    .dio0 =      [número de pin],
};

#if (defined CFG_1278_DRF1278F) || (defined CFG_1272_SEMTECH)           //Selección la versión/fabricante del SX1278
    .dio3 =      [número de pin],
#endif
#ifndef defined(CFG_1278_NICERF1278)                                     //Selecciona el CHIP SX1278 de NICERF
    .tx_en =     [número de pin],
    .rx_en =     [número de pin],
#endif
};

//Declaración de variables
[variables]

//Configuración de los dispositivos LoRa
void setup()                                         //Función de configuración inicial
{
    //Configuración de comunicación para realizar debugger
    Serial.begin([velocidad de comunicación del puerto USB(COM) para debugger]);

    //Inicialización de variables de aplicación general
    [variables = 0]

    //Inicialización de variables del driver
    RADIO.flagTx = 0;
    RADIO.flagRx =0;
    RADIO.crc =   0;

    //Configuración de frecuencia
    RADIO.freq = [frecuencia de trabajo (Hz)];

    //Configuración de potencia
    RADIO.txpow = [Máxima potencia (dBm): 2 – 17]          //Máxima TX potencia
    RADIO.imax =  [Corriente máxima (mA): 45 <=Imax <= 240] //Por defecto 100mA  45mA <=Imax <= 240mA

    //Variables de configuración de la comunicación LoRa
    RADIO.sf =  [Spread factor: SF_6, SF_7, SF_8, SF_9, SF_10, SF_11, SF_12] //Configuración del Spread factor
    RADIO.bw =  [Ancho del canal: BW7_8, BW10_4, BW15_6, BW20_8, BW31_25, BW41_7, BW62_5, BW125, BW250, BW500]
    RADIO.cr =  [Coding rate: CR_4_5, CR_4_6, CR_4_7, CR_4_8]                //Configuración de Coding rate

    hal_init();        //Configuración de puertos de Arduino
    radio_init();     //Configuración de inicio para los LoRa

    Serial.flush();
}

void loop()
{
    [Lógica de control implementada por el usuario]
}

```

Figura 3. Base de aplicación o “sketch” para utilizar transceiver LoRa con IDE Arduino.

Fuente: Código propuesto por el autor

La *Figura 3* presenta el código propuesto para el uso del driver generado para aplicaciones con dispositivos LoRa y sistemas embebidos Arduino.

Cabe aclarar que de acuerdo al modo de configuración de los dispositivos LoRa, los registros tipo *flags* o *banderas* se comportan de acuerdo a la lógica presentada en la *Tabla 4*.

Tabla 4. Comportamiento de los registros tipo flag o banderas de los estados finales de transmisión o recepción de datos para los LoRa.

Fuente: Propuesta del autor

Registro tipo flag o bandera	Estado	Comentario
<i>RADIO.flagTx</i>	0	En modo transmisión, indica que aún no se ha trasmisido la información del buffer <i>RADIO.frameTX</i> de cantidad <i>RADIO.dataLenTX Bytes</i> .
	1	En modo transmisión, indica que se ha trasmisido la información del buffer <i>RADIO.frameTX</i> de cantidad <i>RADIO.dataLenTX Bytes</i> .
<i>RADIO.flagRx</i>	0	En modo recepción, indica que aún no se ha recibido información.
	1	En modo recepción, indica que se ha recibido <i>RADIO.dataLenRX Bytes</i> en el buffer <i>RADIO.frameRX</i> y que el usuario debe gestionar la información.
<i>RADIO.crc</i>	0	En modo recepción, indica que si <i>RADIO.flagRx</i> = 1, la información recibida en el buffer <i>RADIO.frameRX</i> es correcta .
	1	En modo recepción, indica que si <i>RADIO.flagRx</i> = 1, la información recibida en el buffer <i>RADIO.frameRX</i> es errada .

El driver también generan las variables:

- ***RADIO.snr***: Calidad de la señal a ruido de los datos recibidos.
- ***RADIO.rssi***: Indicador de fuerza de los datos recibidos.

Ejemplo de aplicación: red punto a punto (aplicación ping-pong)

Como ejemplo se plantea el uso de dos sistemas embebidos, uno utilizado como maestro y el otro utilizado como esclavo.

El dispositivo maestro se encarga de realizar la petición de datos a un dispositivo esclavo ubicado en un sitio remoto, el cual obtiene datos de sensores. El esclavo remite la información solicitada y continuamente se encuentra esperando peticiones del dispositivo esclavo. La

Máster en Estrategias y Tecnologías para el Desarrollo

Título oficial interuniversitario de La Universidad Politécnica y Universidad Complutense de Madrid



POLITÉCNICA



Figura 4 se presenta un diagrama pictórico de la aplicación propuesta: maestro-esclavo (sensor) con la utilización de dispositivo LoRa.



Figura 4. Principio de funcionamiento de aplicación “ping-pong”.

Fuente: Desarrollado por el autor

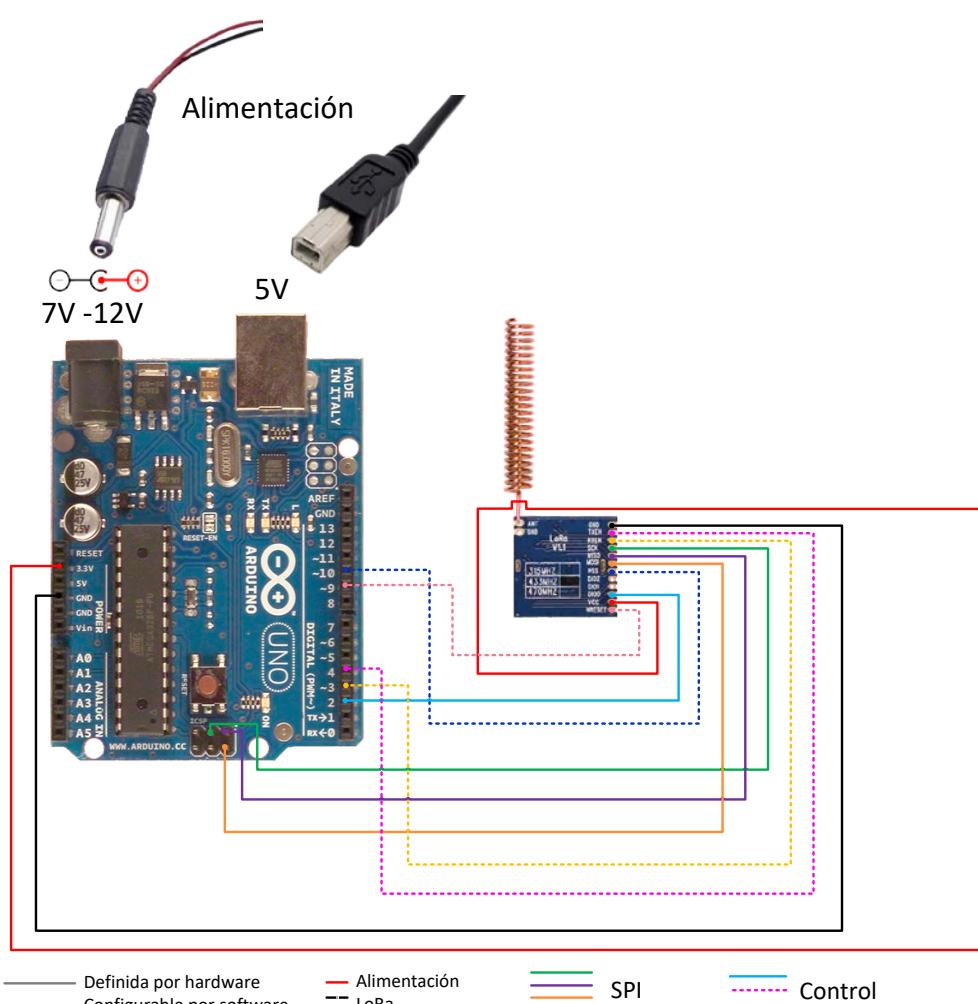


Figura 5. Esquema de conexión para LoRa1278 de NiceRF y Arduino UNO.

Fuente: Esquema propuesto por el autor

La *Figura 5* presenta el esquema de conexión utilizado entre los dispositivos LoRa SX1278 y el sistema embebido Arduino Uno. La *Tabla 5* presenta la nomenclatura utilizada en el transceiver SX1278 y la tarjeta Arduino Uno.

Tabla 5. Nomenclatura utilizada para la conexión entre la tarjeta LoRa1278 de NiceRF y la tarjeta Arduino UNO
Fuente: NiceRF (2015)

Tipo	Nomenclatura estándar	Notación en tarjetas	
		LoRa1278	Arduino UNO
Alimentación	VCC	VCC	3.3V
	GND	GND	GND
SPI	NSS	NSS	10
	MOSI	MOSI	11
	MISO	MISO	12
	SCK	SCK	13
CONTROL	RESET	NRESET	9
	DIO0	DIO0	2
	TXEN	TXEN	3
	RXEN	RXEN	4

El flujograma del código fuente empleado para la prueba se encuentra en el “ANEXO A Desarrollo del driver para el uso de los dispositivos de radio LoRa con sistemas embebidos” y los códigos fuente se presentan en la carpeta *radio/ejemplos/ping_pong* que acompaña al driver.

El sketch *Master_Sx1278_PingPong.ino* corresponde al archivo maestro y el sketch *Esclavo_Sx1278_PingPong.ino* corresponde al código del sistema embebido esclavo.

Ejemplo de aplicación: recepción de datos de varios dispositivos remotos (red punto - multipunto)

El ejemplo plantea un esquema de dispositivos conectados por una red punto-multipunto tipo estrella con un nodo central (circuito maestro) y 3 dispositivos esclavos.

La *Figura 6* presenta el esquema propuesto donde se aprecia la comunicación simultánea de 3 dispositivos hacia un nodo central. Se plantea el uso de esta aplicación con el uso del driver desarrollado para aplicaciones complejas con diversos sensores ubicados en sitios remotos. Cabe resaltar que los dispositivos LoRa deben ser de la misma referencia, estar configurados a la misma frecuencia y con los parámetros SF, BW y CR idénticos.

El dispositivo maestro se debe configurar como dispositivo “pasivo”, es decir continuamente recibir la información de los dispositivos esclavos. Una vez recibe la información de un dispositivo esclavo, remite la confirmación de recepción correcta de datos.

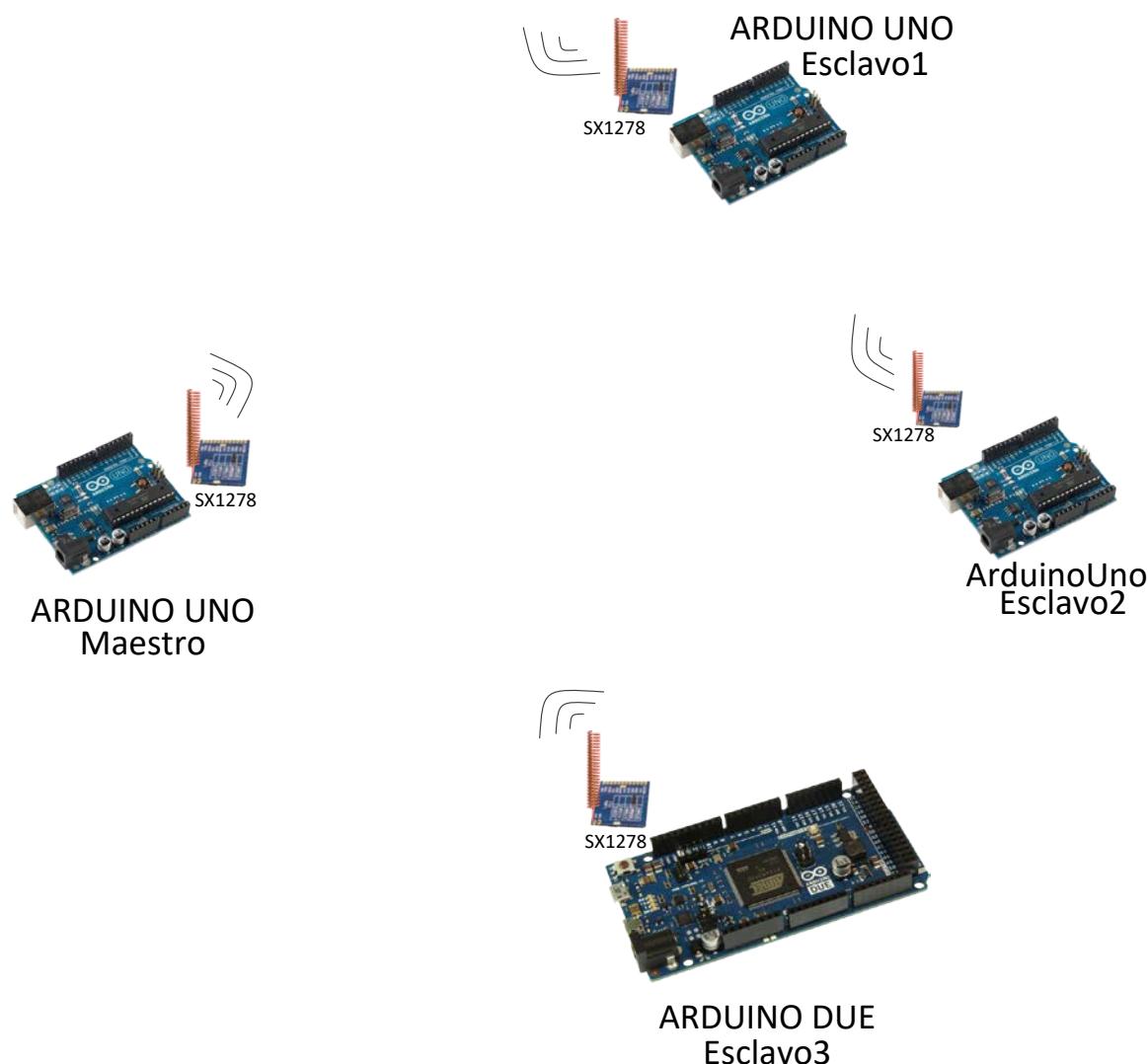


Figura 6. Tipología estrella para comunicación con diversos dispositivos Arduino.

Fuente: Esquema propuesto por el autor

El modem LoRa es *half dúplex*, es decir que los LoRa no puede trasmisir o recibir datos en el mismo instante; esto significa que los transceiver LoRa deben ser configurados para trasmisir o recibir datos. Cuando se incorpora más de dos dispositivos en un protocolo de comunicación *half dúplex*, se deben establecer estrategias que mitiguen los efectos adversos por colisión indeseada de datos cuando se reciben datos.

Si los dispositivos no cuentan con estrategias para evitar la transmisión de datos sobre el mismo canal de comunicación por parte de los dispositivos esclavos en un mismo instante, los datos llegarán errados al dispositivo maestro y la comunicación de datos será errada y

ocasionará un consumo energético innecesario de energía en la transmisión de información por parte de los esclavos.

La solución para evitar la colisión de datos, corresponde a establecer tiempos de muestreo y tiempos aleatorios de “des-sincronización” cuando se trasmiten datos.

En el documento Anexo A: “Desarrollo del driver para el uso de los dispositivos de radio LoRa con sistemas embebidos” – apartado “*Ejemplo de aplicación: recepción de datos de varios dispositivos remotos (red punto – multipunto)*”, se describe de manera detallada el proceso de desarrollo de la prueba desarrollada para crear la red punto-multipunto con dispositivos LoRa.

La *Figura 5* presenta el esquema de conexión utilizado entre los dispositivos LoRa SX1278 y el sistema embebido Arduino Uno. La *Tabla 5* presenta la nomenclatura utilizada en el transceiver SX1278 y la tarjeta Arduino Uno.

El código fuente utilizado por los dispositivos maestro y esclavos se presenta en la carpeta *radio/ejemplos/multipunto* que acompaña al driver.

El sketch *Master_Sx1278_Multipunto.ino* corresponde al archivo maestro y el sketch *Esclavo_Sx1278_Multipunto.ino* corresponde al código del sistema embebido esclavo.

INSTALACIÓN Y USO DEL DRIVER LORA PARA SISTEMAS EMBEBIDOS SDIN

Estructura de archivos que componen el driver

El driver está compuesto por una serie de archivos con *código fuente* y archivos tipo *header* organizados en dos subcarpetas. La estructura organizada de archivos se presenta en la *Figura 7*.

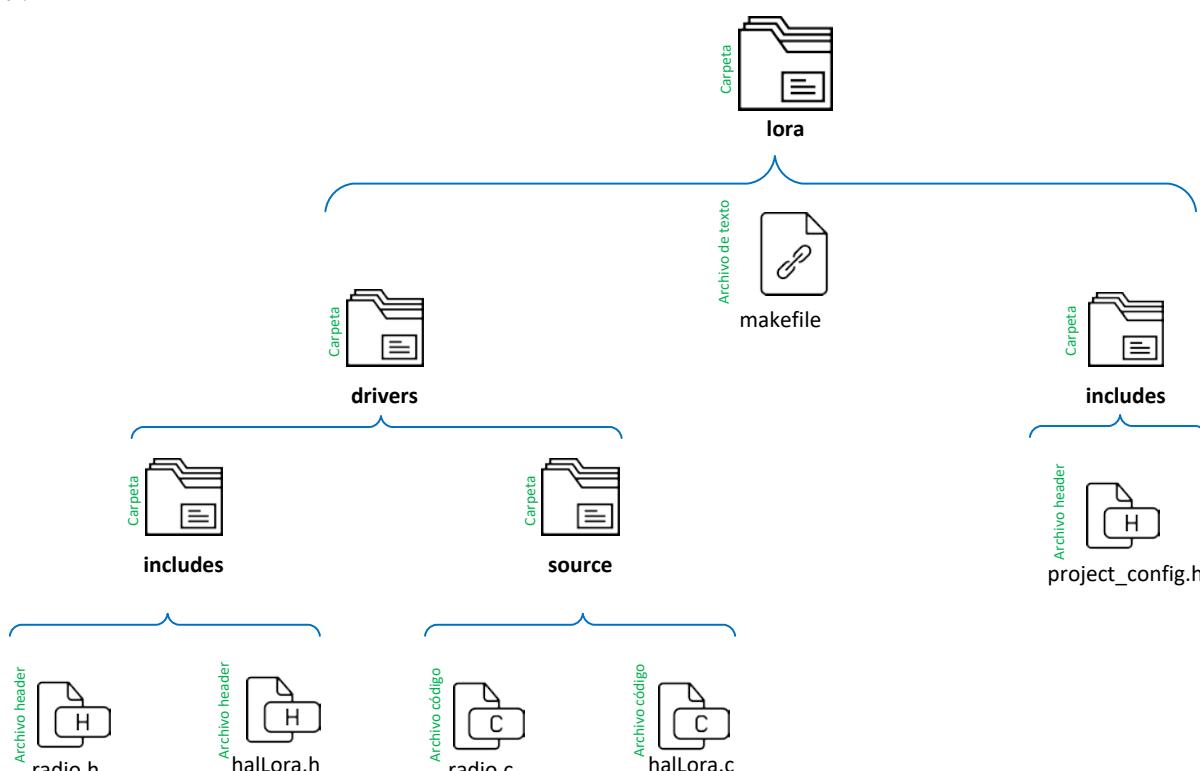


Figura 7. Estructura de archivos y carpetas del driver propuesto para el manejo de dispositivos LoRa utilizando FreeRTOS –SDIN.

Fuente: Esquema propuesto por el autor

La carpeta principal de denomina **lora** y está compuesta por los elementos:

- **Makefile:** es el archivo que cuenta con la información del proyecto para el uso de FreeRTOS. Con respecto al driver, el apartado “**DRIVERS_SOURCE= **” debe contener la ruta de las librerías relacionadas con el control de los LoRa:

```
DRIVERS_SOURCE= \
$(DRIVERS_SOURCE_DIR)/source/halLora.c \
$(DRIVERS_SOURCE_DIR)/source/radio.c \

```

- **Carpeta drivers:**

- Carpeta **includes**:

- **radio.h:** contiene la definición de variables y funciones para la configuración de los LoRa a nivel de software relacionado con el código presentado en **radio.c**.
- **halLora.h:** cuenta con la definición de funciones del archivo **halLoRa.c** (*HAL o hardware abstraction layer*). Además contiene la definición de la nomenclatura asignada a los pines o terminales de control de los dispositivos LoRa.
- Carpeta source:
 - **radio.c:** contiene las funciones para la configuración de los LoRa a nivel de software.
 - **halLora.c:** contiene el código fuente de la capa de más bajo nivel que configura los periféricos internos de SDIN (interrupciones por cambio de flanco, comunicación SPI, comunicación UART, cambios de estados en terminales). A esta capa se le conoce como *HAL* o capa de abstracción de hardware.
- Carpeta includes:
 - **project_config.h:** el *header file* contiene la definición o selección del transceiver LoRa utilizado por la aplicación de SDIN. Es suficiente con definir:

Instrucción	Comentario
#define CFG_1272_SEMTECH 1	Selecciona el CHIP SX1272 de SEMTECH
#define CFG_1278_DRF1278F 1	Selecciona el CHIP SX1278 de DORJI versión de tarjeta DRF1278F
#define CFG_1278_NICERF1278 1	Selecciona el CHIP SX1278 de NiceRF versión de tarjeta LORA1278

○

La *Figura 8* presenta la gestión entre los diferentes archivos que componen el driver planteado.

Resulta estratégica la organización de los diferentes archivos debido a que el driver busca que el código presentado en **radio.c** se comporte como una capa de alto nivel, manteniendo el código invariante al uso de diferentes sistemas embebidos como el módulo SDIN, módulos SDIN (microcontrolador de 32 bits STM32f10x) o tarjetas de STMicroelectronics; mientras que el código presentado en **halLora.c** corresponde a una capa de alto bajo nivel de abstracción de hardware que de acuerdo al tipo de sistema embebido utilizado pueda ser ajustado para realizar la correcta homologación al microcontrolador utilizado.

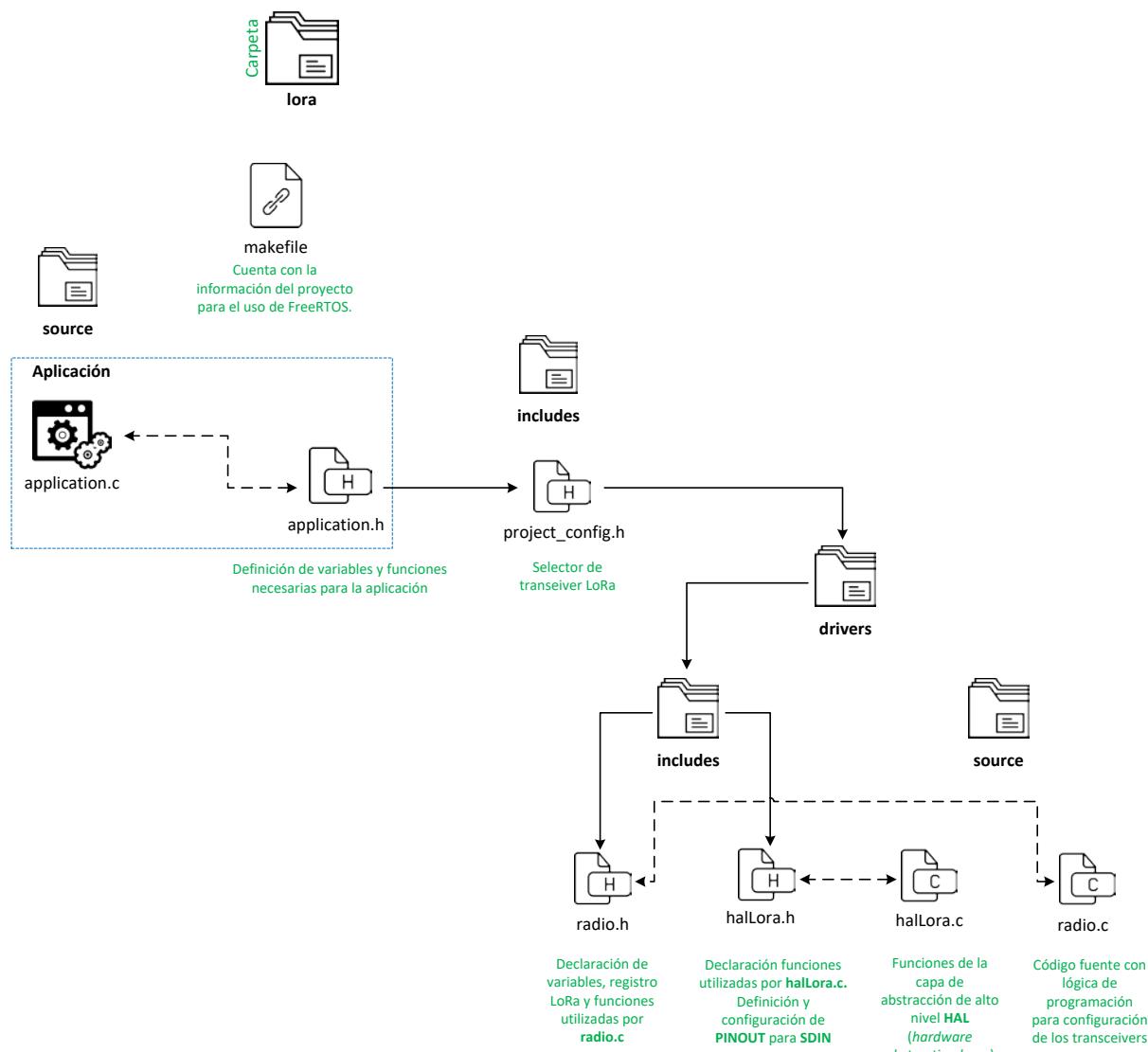


Figura 8. Gestión entre archivos del driver propuesto para el manejo de dispositivos LoRa para los módulos SDIN.

Fuente: Esquema propuesto por el autor

La Figura 8 presenta la gestión entre los diferentes archivos que componen el driver planteado.

Resulta estratégica la organización de los diferentes archivos debido a que el driver busca que el código presentado en **radio.c** se comporte como una capa de alto nivel, manteniendo el código invariante al uso de diferentes sistemas embebidos como Arduino, módulos SDIN o tarjetas de STMicroelectronics; mientras que el código presentado en **halLora.c** corresponde a una capa de alto bajo nivel de abstracción de hardware que de acuerdo al tipo de sistema embebido utilizado pueda ser ajustado para realizar la correcta homologación al microcontrolador utilizado.

Uso de librería para el control de dispositivos LoRa en aplicaciones para SDIN

La instalación del driver para el control de los dispositivos LoRa utilizando SDIN (se recomienda que el equipo de cómputo tenga instalado el IDE Eclipse), se realiza copiando los archivos del fichero principal **lora** junto con la carpeta donde se encuentra instalado los archivos del S.O. FreeRTOS:

- Incluir las librerías para control por software (**radio**) y hardware (**halLora**) en el header file **application.h**:

```
#include "radio.h"           //Librería de alto nivel para configuración por software de los radio LoRa
#include "halLora.h"          //Librería de bajo nivel para configuración de hardware radio LoRa (HAL)
#include "project_config.h"    //Librería para la selección del transceiver LoRa
```

- Definir los pines de control de la tarjeta SDIN con respecto a los transceiver LoRa:

```
//Definición de los pines de control de la tarjeta SDIN
```

#define LORA_SPI	SPI2
#define LORA_SPI_CLK	RCC_APB1Periph_SPI2
#define LORA_SPI_IRQn	SPI2_IRQn
#define LORA_SPI_PORT	GPIO [letra del puerto]

```
// SPI CS/NSS
```

#define LORA_NSS_PIN	GPIO_Pin_[número del Pin]
#define LORA_NSS	LORA_SPI_PORT, LORA_NSS_PIN

```
// SPI SCK
```

#define LORA_SCK_PIN	GPIO_Pin_[número del Pin]
#define LORA_SCK	LORA_SPI_PORT, LORA_SCK_PIN

```
// SPI MISO
```

#define LORA_MISO_PIN	GPIO_Pin_[número del Pin]
#define LORA_MISO	LORA_SPI_PORT, LORA_MISO_PIN

```
// SPI MOSI
```

#define LORA_MOSI_PIN	GPIO_Pin_[número del Pin]
#define LORA_MOSI	LORA_SPI_PORT, LORA_MOSI_PIN

```
#define LORA_RST_PORT
```

#define LORA_RST_PIN	GPIOA
#define LORA_RST	GPIO_Pin_[número del Pin]
#define LORA_RST_MODE	LORA_RST_PORT, LORA_RST_PIN
	GPIO_Mode_Out_PP

- Configurar los dispositivos LoRa de acuerdo a las necesidades del usuario. Esto se debe incorporar en la función propia de **application.c** (ejemplo “TestLora_Task()”):

- Iniciar las variables tipo *flag* o *banderas*:

```
//Inicialización de variables del driver
```

```
RADIO.flagTx = 0;
```

```
RADIO.flagRx = 0;
```

RADIO.crc = 0;

- Definir la frecuencia de trabajo del transceiver:

RADIO.freq = [frecuencia de trabajo]; //Frecuencia de la banda ISM compatible con LoRa

Teniendo en cuenta que existen bandas de frecuencias de uso privativo (telefonía, internet, radio, entre otras), existen frecuencias de libre uso conocidas como ISM (Industrial, Scientific and Medical) que son frecuencias intencionalmente libres para uso no comercial en aplicaciones de tipo industrial, científico y médico. Se recomienda al usuario verificar la compatibilidad de la frecuencia de trabajo con la *Tabla 2* (bandas de uso ISM).

- Definir la potencia de transmisión del transceiver:

//Configuración de potencia

RADIO.txpow = [Máxima potencia (dBm)] //Máxima TX potencia

RADIO.imax = [Corriente máxima (mA)] //Por defecto 100mA 45mA <=Imax <= 240mA

La potencia máxima se expresa en dBm y el valor definido debe estar entre 2 a 17.

La corriente máxima está dada en mA y el valor definido se debe encontrar entre 45mA y 240mA. El valor típico de corriente es de 100mA. Para valores significativos de corrientes (mayores a 125mA), se recomienda verificar la corriente máxima de suministro soportada por la fuente o batería de alimentación.

- Definir las variables de configuración de la comunicación entre los transceivers LoRa:

//Variables de configuración de la comunicación LoRa

RADIO.sf = [Spread factor SF] //Configuración del Spread factor

RADIO.bw = [Ancho del canal BW] //Configuración del ancho de canal

RADIO.cr = [Coding rate CR] //Configuración de Coding rate

Los dispositivos LoRa cuentan con los parámetros *SF*, *BW* y *CR* que según las combinaciones entre estos, permite determinar el alcance o distancia máxima de recepción de datos, la velocidad de los datos y sobrecarga por corrección o detección de errores en los datos. Estos valores deben ser los mismos en todos los dispositivos radio LoRa de la red inalámbrica.

Los parámetros independientes de configuración se definen como (Semtech SX1272, 2015 & Semtech SX1278, 2015):

BW: Ancho de banda (entre más bajo sea este valor, el tiempo de durante la transmisión será mayor)

SF rate: Es el factor de alcance expresado como logaritmo de base 2. Entre mayor sea este valor, se tendrá un mejor rendimiento en la transmisión de datos.

CR: Tasa de codificación de errores: Entre mayor sea este valor, mayor será la fiabilidad de los datos, pero con una sobrecarga en el tiempo de transmisión.

La *Tabla 3* presenta la combinación de los principales variables independientes para la configuración de la comunicación entre transceivers LoRa. Cabe aclarar que estos valores deben ser los mismos en todos los dispositivos radio LoRa de la red inalámbrica.

- Configurar los pines I/O y la comunicación SPI:

```
hal_init() //Configuración de puertos de SDIN
```

- Configurar los pines I/O y la comunicación SPI:

```
radio_init(); //Configuración de inicio para los LoRa
```

Corresponde a la función general que configura todos los parámetros de inicio del transceiver LoRa.

- Crear la aplicación con la lógica propuesta por el desarrollador:

```
while(1)
{
    [Lógica de control implementada por el usuario]
}
```

- En la función **Application_Start()**, crear la tarea relacionada con la aplicación de uso de los LoRa (ejemplo: TestLora_Task):

```
xTaskCreate(TestLora_Task, "TestLora_Task", TEST_LORA_TASK_STACK, NULL,
TEST_LORA_TASK_PRIORITY, &TestLoraTask_Handle); //TestLora_Task->
PRUEBA UNO A UNO - Esclavo
```

Cabe aclarar que de acuerdo al modo de configuración de los dispositivos LoRa, los registros tipo *flags* o *banderas* se comportan de acuerdo a la lógica presentada en la *Tabla 4*.

El driver también generan las variables:

- **RADIO.snr:** Calidad de la señal a ruido de los datos recibidos.
- **RADIO.rssi:** Indicador de fuerza de los datos recibidos.

Ejemplo de aplicación: red punto a punto (aplicación ping-pong) con módulos SDIN

Como ejemplo se plantea el uso de dos sistemas embebidos, uno utilizado como maestro y el otro utilizado como esclavo.

El dispositivo maestro se encarga de realizar la petición de datos a un dispositivo esclavo ubicado en un sitio remoto, el cual obtiene datos de sensores. El esclavo remite la información solicitada y continuamente se encuentra esperando peticiones del dispositivo esclavo. La

Figura 9 se presenta un diagrama pictórico de la aplicación propuesta: maestro-esclavo (sensor) con la utilización de dispositivo LoRa.



Figura 9. Principio de funcionamiento de aplicación “ping-pong” con SDIN.

Fuente: Desarrollado por el autor

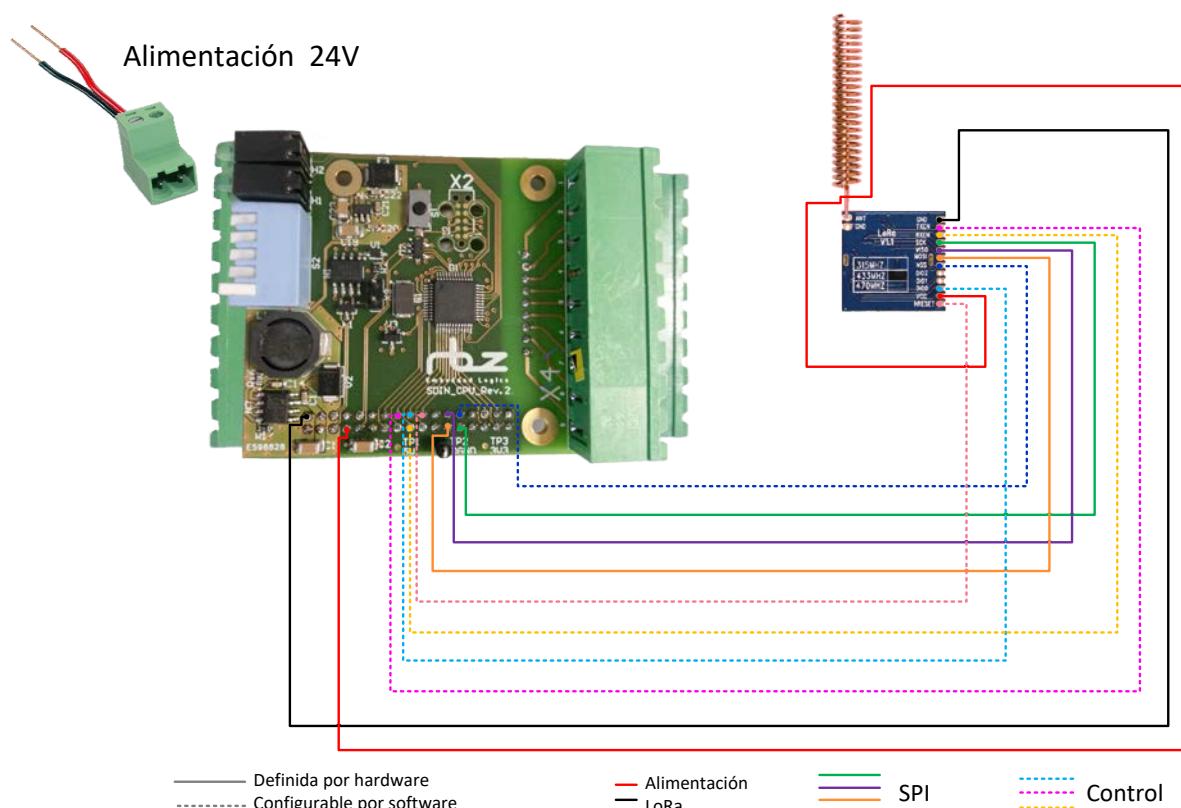


Figura 10. Esquema de conexión para LoRa1278 de NiceRF y módulo SDIN.

Fuente: Esquema propuesto por el autor

La *Figura 10* presenta el esquema de conexión utilizado entre los dispositivos LoRa SX1278 y el sistema embebido SDIN. La *Tabla 6* presenta la nomenclatura utilizada en el transceiver SX1278 y la tarjeta SDIN.

Tabla 6. Nomenclatura utilizada para la conexión entre la tarjeta LoRa1278 de NiceRF y el módulo SDIN

Fuente: NiceRF (2015)

Tipo	Nomenclatura estándar	Notación en tarjetas	
		LoRa1278	SDIN
Alimentación	VCC	VCC	3.3V
	GND	GND	GND
SPI	NSS	NSS	PB12
	MOSI	MOSI	PB15
	MISO	MISO	PB14
	SCK	SCK	PB13
CONTROL	RESET	NRESET	PA7
	DIO0	DIO0	PA5
	TXEN	TXEN	PA3
	RXEN	RXEN	PA4

El flujograma del código fuente empleado para la prueba se encuentra en el “ANEXO A Desarrollo del driver para el uso de los dispositivos de radio LoRa con sistemas embebidos” y los códigos fuente se presenta en la carpeta *lora/source/application.c* que acompaña al driver.

La tarea se denomina *TestLora_Task()* corresponde al código fuente del dispositivo maestro del sistema embebido SDIN.

Ejemplo de aplicación: recepción de datos de varios dispositivos remotos (red punto - multipunto) con módulos SDIN

El ejemplo plantea un esquema de dispositivos conectados por una red punto-multipunto tipo estrella con un nodo central (circuito maestro) y 3 dispositivos esclavos.

La *Figura 11* presenta el esquema propuesto donde se aprecia la comunicación simultánea de 3 dispositivos hacia un nodo central. Se plantea el uso de esta aplicación con el uso del driver desarrollado para aplicaciones complejas con diversos sensores ubicados en sitios remotos. Cabe resaltar que los dispositivos LoRa deben ser de la misma referencia, estar configurados a la misma frecuencia y con los parámetros SF, BW y CR idénticos.

El dispositivo maestro se debe configurar como dispositivo “pasivo”, es decir continuamente recibir la información de los dispositivos esclavos. Una vez recibe la información de un dispositivo esclavo, remite la confirmación de recepción correcta de datos.

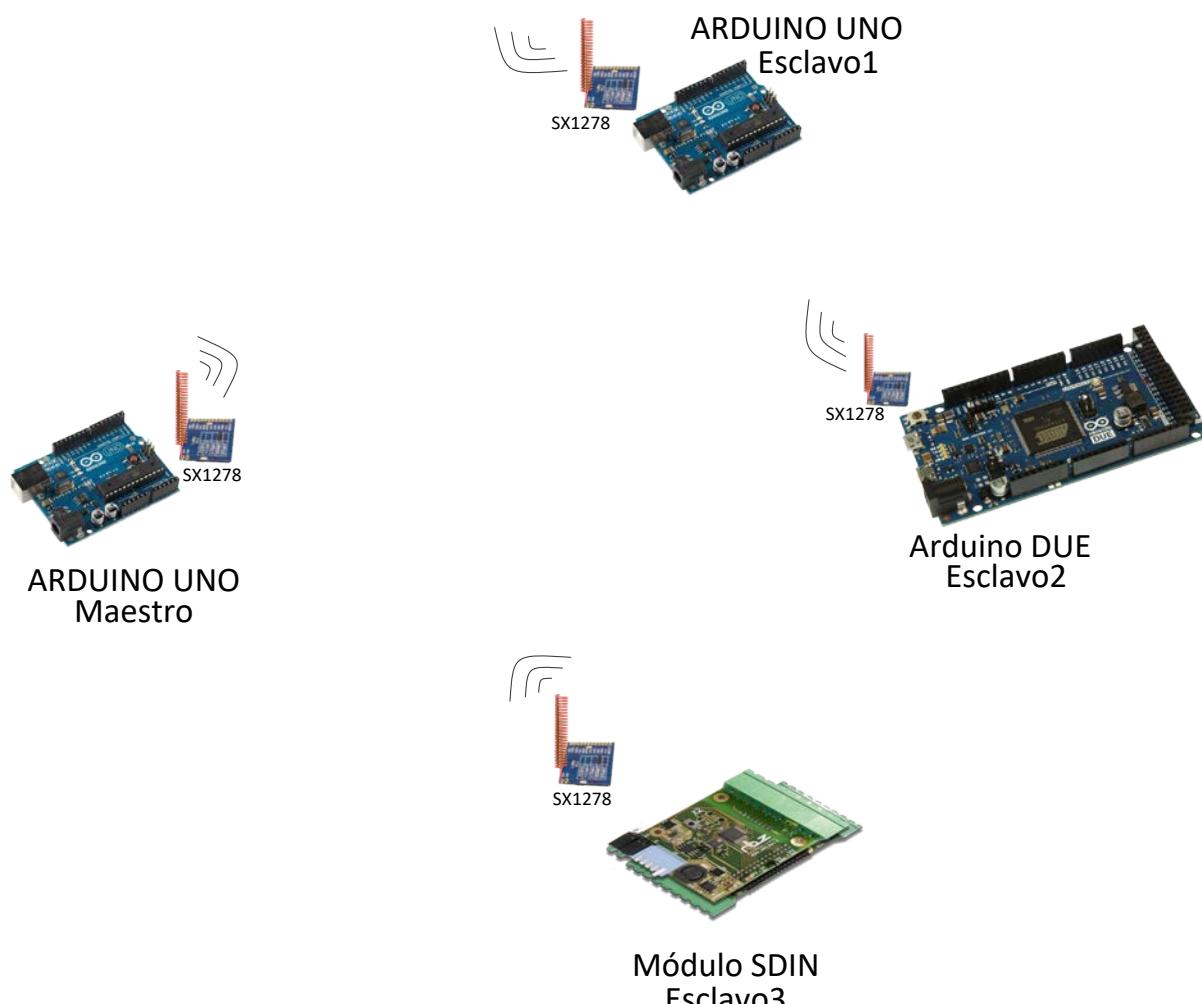


Figura 11. Tipología estrella para comunicación con diversos dispositivos.

Fuente: Esquema propuesto por el autor

El modem LoRa es *half dúplex*, es decir que los LoRa no puede trasmisir o recibir datos en el mismo instante; esto significa que los transceiver LoRa deben ser configurados para trasmisir o recibir datos. Cuando se incorporan más de dos dispositivos en un protocolo de comunicación *half dúplex*, se deben establecer estrategias que mitiguen los efectos adversos por colisión indeseada de datos cuando se reciben datos.

Si los dispositivos no cuentan con estrategias para evitar la transmisión de datos sobre el mismo canal de comunicación por parte de los dispositivos esclavos en un mismo instante, los datos llegarán errados al dispositivo maestro y la comunicación de datos será errada y ocasiona un consumo energético innecesario de energía en la transmisión de información por parte de los esclavos.

La solución para evitar la colisión de datos, corresponde a establecer tiempos de muestreos y tiempos aleatorios de “des-sincronización” cuando se transmiten datos; es decir evitar con los tiempos adicionales de transmisión de datos la colisión de datos.

En el documento Anexo A: “Desarrollo del driver para el uso de los dispositivos de radio LoRa con sistemas embebidos” – apartado “*Ejemplo de aplicación: recepción de datos de varios dispositivos remotos (red punto – multipunto)*”, se describe de manera detallada el proceso de desarrollo de la prueba desarrollada para crear la red punto-multipunto con dispositivos LoRa.

La *Figura 10* presenta el esquema de conexión utilizado entre los dispositivos LoRa SX1278 y el sistema embebido Arduino Uno. La *Tabla 6* presenta la nomenclatura utilizada en el transceiver SX1278 y la tarjeta SDIN.

El código fuente utilizado por el dispositivo esclavo se presenta en la carpeta *lora/source/application.c* que acompaña al driver.

La tarea se denomina *TestLoraMultiple_Task()* corresponde al código fuente del dispositivo esclavo del sistema embebido SDIN.

BIBLIOGRAFÍA

NiceRF (2015). LoRa1278 100 mW 4 km larga distancia y alta sensibilidad (-139 dBm) 433 MHz módulo de transceptor inalámbrico. Recuperado de <http://es.aliexpress.com/item/2pcs-lot-LoRa1278-100mW-4km-Long-Distance-and-High-Sensitivity-139-dBm-433MHz-Wireless-Transceiver-Module/32461365864.html>, China. Consultado el 22 de junio de 2016

Semtech SX1272 (2015). SX1272/73 - 860 MHz to 1020 MHz Low Power Long Range Transceiver. Datasheet. Semtech Corporation. Estados Unidos.

Semtech SX1278 (2015). SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver. Datasheet. Semtech Corporation. Estados Unidos.

Texas Instruments (2005). SM-Band and Short Range Device Regulatory Compliance Overview. Estados Unidos.

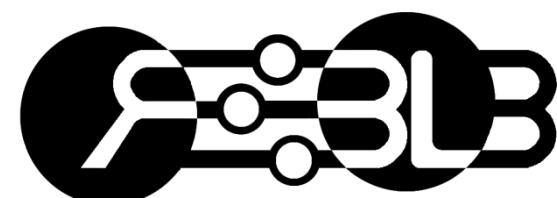
Máster en

Estrategias y Tecnologías para el Desarrollo

Título oficial interuniversitario de la Universidad Politécnica y Universidad Complutense de Madrid



POLITÉCNICA



2016