

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN



GRADO EN INGENIERÍA DE ....

TRABAJO FIN DE GRADO

TÍTULO

NOMBRE

AÑO

**UNIVERSIDAD POLITÉCNICA DE MADRID**  
**ESCUELA TÉCNICA SUPERIOR**  
**DE INGENIEROS DE TELECOMUNICACIÓN**

Reunido el tribunal examinador en el día de la fecha, constituido por

Presidente: Dr. D. MIEMBRO 1

Vocal: Dr. D. MIEMBRO 2

Secretario: Dr. Dña. MIEMBRO 3

Suplente: D. SUPLENTE

para juzgar el Trabajo Fin de Titulación titulado:

**TÍTULO**

del alumno D. ALUMNO  
dirigido por Dr. D. DIRECTOR  
del Departamento de DEPARTAMENTO

Acuerdan otorgar la calificación de: \_\_\_\_\_

Y, para que conste, se extiende firmada por los componentes del tribunal, la presente diligencia

Madrid, DÍA de mes de AÑO

El Presidente

El Vocal

El Secretario

Fdo: \_\_\_\_\_ Fdo: \_\_\_\_\_ Fdo: \_\_\_\_\_

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN



GRADO EN INGENIERÍA DE ...

TRABAJO FIN DE GRADO

TÍTULO

NOMBRE

AÑO

## Resumen

## Abstract

## Agradecimientos

# Índice general

<b>Resumen</b>	<b>IV</b>
<b>Resumen</b>	<b>V</b>
<b>Agradecimientos</b>	<b>VI</b>
<b>Índice General</b>	<b>VII</b>
<b>Índice de Figuras</b>	<b>VIII</b>
<b>Índice de Tablas</b>	<b>X</b>
<b>Lista de acrónimos</b>	<b>XII</b>
<b>1. Introducción y Objetivos</b>	<b>1</b>
1.0.1. Tecnologías actuales . . . . .	1
1.0.2. Redes LPWAN . . . . .	2
1.0.3. LoRa . . . . .	4
1.0.4. Objetivos y motivaciones . . . . .	5
1.0.5. Metodología . . . . .	6
<b>2. Diseño software</b>	<b>7</b>
2.1. Diseño del protocolo . . . . .	7
2.1.1. Formato de trama . . . . .	8
2.1.1.1. Trama de conexión . . . . .	8
2.1.1.2. Trama de configuración . . . . .	9
2.1.1.3. Trama de datos . . . . .	10
2.1.2. Comportamiento del protocolo . . . . .	10
2.1.2.1. Conexión de un nodo a la red . . . . .	10
2.1.2.2. Envío y recepción de datos . . . . .	10
2.1.2.3. Tratamiento de errores . . . . .	11
2.2. Implementación . . . . .	12
2.2.1. Fundamentos teóricos . . . . .	12
2.2.1.1. Sistemas embebidos . . . . .	12
2.2.1.2. Sistemas operativos de tiempo real (RTOS) . . . . .	12
2.2.1.3. Máquinas de estados extendidas (xFSM) . . . . .	12
2.2.2. Descripción del entorno . . . . .	12
2.2.2.1. Drivers, Librerías y útiles . . . . .	12
2.2.2.2. Herramientas de edición, compilación y depuración . . . . .	13

2.2.3. Modelado del comportamiento de los nodos . . . . .	13
2.2.4. Modelado del comportamiento del nodo maestro . . . . .	13
2.2.5. Estructura del código . . . . .	13
2.2.6. Proceso de diseño e implementación . . . . .	13
2.2.7. Resultados obtenidos . . . . .	13
<b>3. Conclusiones y líneas futuras</b>	<b>15</b>
3.1. Conclusiones . . . . .	15
<b>Bibliografía</b>	<b>17</b>



# Índice de figuras

1.1. Parámetros fundamentales de un sistema inalámbrico . . . . .	2
1.2. Relación de consumo, velocidad y alcance de las diferentes tecnologías et al. (2016) . . . . .	3
2.1. Cronograma de conexión de un nodo . . . . .	10



# Índice de tablas

1.1. Ejemplo teórico para un canal con 125 KHz de ancho de banda . . . .	5
--	---

# Lista de Acrónimos

**IES:** Instituto de Energía Solar.

# Capítulo 1

## Introducción y Objetivos

Nos encontramos ante una autentica revolución en el sector de la ingeniería y las telecomunicaciones: la mejora y abaratamiento de las tecnologías y redes inalámbricas están cambiando la forma en la que interactuamos con nuestro entorno. Esto, unido a la aparición de nuevas tecnologías en el procesamiento de datos, ha propiciado una mayor demanda por parte de la industria.

El término *IoT* (Internet of things) es un concepto que hace referencia a este fenómeno y que consiste en la conexión de elementos cotidianos a la red. A día de hoy, podemos encontrar desde bombillas hasta extensas redes de farolas interconectadas.

Según datos del IDC (International Data Corporation), para 2019 se prevé que la inversión mundial en *IoT* supere los \$745 mil millones de dólares, lo que supone un 15,4% más que el año pasado. Se espera que el ritmo de crecimiento continúe los próximos años llegando a superar \$1 billón de dólares en 2022 Torchia (2019).

Ante este panorama, se hace evidente la necesidad de innovar y desarrollar soluciones más eficientes. El creciente aumento en el número de dispositivos conectados hace necesario el desarrollo de redes y tecnologías más robustas, que soporten el gran volumen de datos actual y el esperado en años venideros.

En esta primera parte, expondré brevemente las tecnologías utilizadas actualmente en la industria así como sus ventajas e inconvenientes. Finalmente me centraré en la tecnología utilizada en este proyecto, las motivaciones para usarla y los objetivos que busco alcanzar en este TFG.

### 1.0.1. Tecnologías actuales

En el mercado actual podemos encontrar un amplio abanico de tecnologías de comunicación inalámbrica. Aunque estas tecnologías distan, en ocasiones, mucho unas de otras, podemos caracterizarlas y compararlas mediante tres factores fundamentales: potencia, alcance y ancho de banda. Estos factores tienen fuertes interdependencias entre sí, lo que hace complicado cumplirlos todos a la vez. Esta problemática se ve reflejada en la figura 1.1: si queremos dos de los factores tenemos que renunciar al tercero.

Algunas de las tecnologías usadas para *IoT* son:

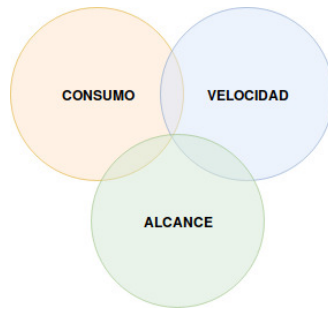


Figura 1.1: Parámetros fundamentales de un sistema inalámbrico

- **NFC:** (Near Field Communication) es un estándar de comunicaciones de corto alcance inalámbricas desarrollado por Sony y NPX. Trabaja en la frecuencia de 13.56MHz utilizando modulaciones OOK (on-off keying) y BPSK. Esta tecnología alcanza velocidades de transmisión de hasta 848Kbps, dependiendo del entorno en el que se lleve a cabo la comunicación Song and Issac (2011).
- **WiFi:** Tecnología de comunicación de corto alcance inalámbrica que utiliza el estándar IEEE 802.11x. Utiliza las bandas ISM de 2.4 GHz y 5.7 GHz, utilizando un amplio número de modulaciones (varían según la banda y versión del estándar). Es uno de los estándares más utilizados en la industria por su aplicación en redes locales (WLAN) y por las altas tasas binarias que alcanza (1.3 Gbps teóricos y hasta 400 Mbps reales) Minihold (2014).
- **BLE:** (Bluetooth Low Energy) Se trata de una tecnología de comunicaciones de corto alcance, diseñada por Bluetooth Special Interest Group. Utiliza la banda ISM de 2.4 GHz y alcanza tasa binarias de hasta 1.37 Mbps Cypress (2015). Este sistema destaca por su bajo consumo, pero tiene un alcance limitado y una tasa binaria no muy alta.
- **WiMAX:** (World Interoperability for Microwave Access) Es una tecnología basada en el estándar IEEE 802.16, la cual puede dar servicio a redes de área metropolitana. Utiliza las bandas entre 2.3 y 5.8 GHz y puede alcanzar tasas binarias de hasta 20 Mbps Minihold (2014). Destaca por su amplio alcance (hasta 70 Km), pero requiere mucha potencia para su funcionamiento.

A parte de estas, existen muchas otras tecnologías, cuyas características se resumen en en la figura 1.2

### 1.0.2. Redes LPWAN

Se definen como redes de bajo consumo y largo alcance (LPWAN de sus siglas en inglés), al conjunto de redes inalámbricas caracterizadas por tener largos alcances, consumo mínimo y baja tasa binaria. Su bajo consumo posibilita implementarlo sobre dispositivos alimentados mediante baterías.

Uno de los principales inconvenientes de estas tecnologías es su baja tasa binaria, la cual hace imposible la transmisión de grandes volúmenes de datos y a su vez limita su utilización a sistemas con interfaces máquina a máquina (M2M de sus siglas en

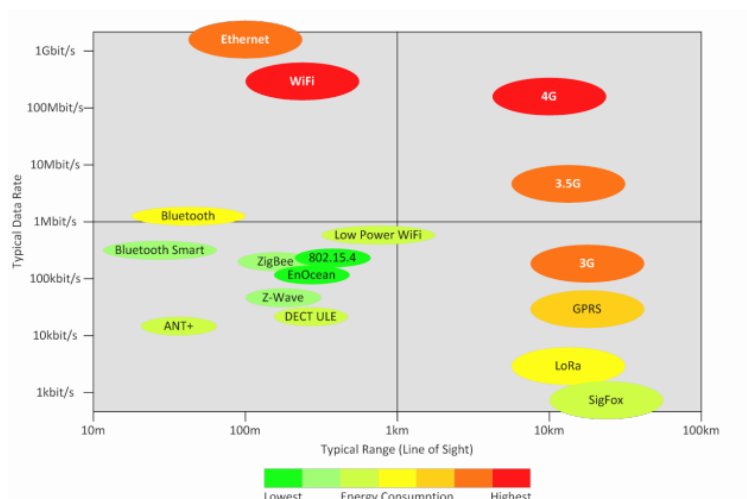
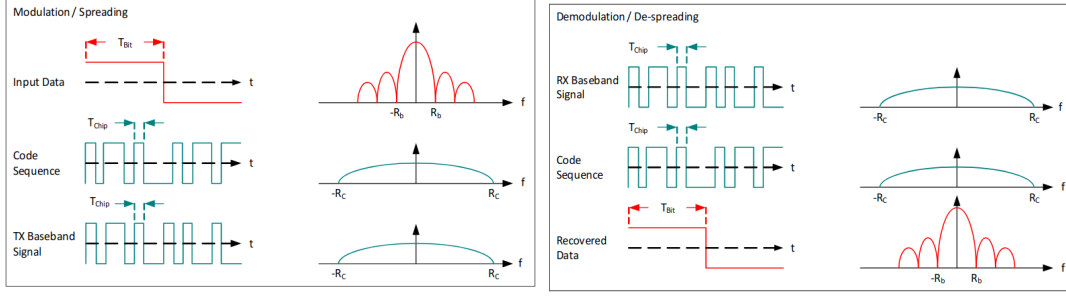


Figura 1.2: Relación de consumo, velocidad y alcance de las diferentes tecnologías et al. (2016)

inglés). Las tasas binarias con las que trabajan las redes LPWAN rondan los 0.3 Kbps y los 50 Kbps, dependiendo principalmente de las técnicas de transmisión y los estándares utilizados.

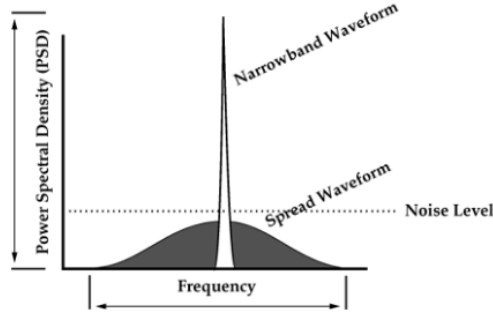
Para conseguir estas características, las redes LPWAN utilizan diferentes técnicas como:

- **Ultra Narrowband (UNB):** Se traduce como "banda ultra-estrecha". Su funcionamiento se basa en la utilización de canales con poco ancho de banda para transmitir. Su reducido ancho de banda conlleva menor consumo, un uso más eficiente del espectro y un aumento drástico del número de dispositivos que pueden operar en una misma red.
- **Spread Spectrum (SS):** Se trata de una modulación de espectro ensanchado en la cual, a cada canal, se le otorga un ancho de banda mayor al estrictamente necesario para funcionar. Al aumentar el ancho de banda, aumentamos la energía por bit utilizada, mejorando el ratio señal a ruido en el receptor.



(a) Modulación, ensanchado del espectro  
(SEMTECH, 2015)

(b) Caption



(c) Comparativa entre banda estrecha y espectro ensanchado

### 1.0.3. LoRa

LoRa (Long Range) es una técnica de modulación en espectro ensanchado, basada en la modulación Chirp de espectro ensanchado CSS (Chirp Spread Spectrum). LoRa es una modulación propietaria, desarrollada por la empresa francesa Cycle y posteriormente adquirida por Semtech Corporation.

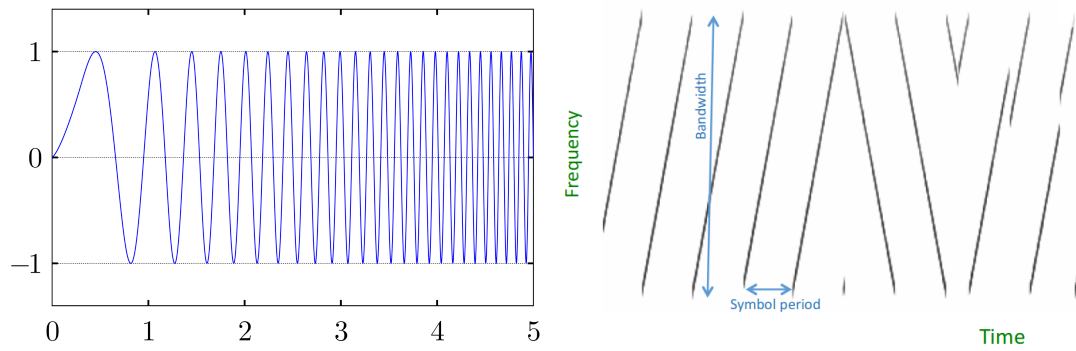
La modulación Chirp de espectro ensanchado se basa en la utilización de pulsos chirp lineales para modular la señal. Un chirp es una señal sinusoidal cuya frecuencia varía con el tiempo. La utilización de chirps para modular la señal hace al sistema robusto frente al desvanecimiento debido al multitrayecto, ruido pulsante en banda estrecha e interferencias debidas al efecto doppler.

Esta tecnología se caracteriza por su largo alcance, bajo consumo y bajo coste. También implementa tasas binaria variables, las cuales se configuran utilizando factores de ensanchado ortogonales (SF o Spreading Factor por sus siglas en inglés), los cuales están en escala logarítmica e indican el número de chirps por símbolo de la modulación. La utilización de estos factores permiten elegir entre mejorar el alcance o la tasa binaria, utilizando un ancho de banda constante.

La relación entre la tasa binaria, el factor de ensanchado y el ancho de banda se define mediante la siguiente ecuación:

$$R_b = SF \times \frac{1}{\left[\frac{2^{SF}}{BW}\right]} (\text{bit/sec}) \quad (1.1)$$





(a) Modulación, ensanchado del espectro (SEMTECH, 2015)

(b) Caption

Como podemos observar en la ecuación 1.1, al aumentar el valor de SF, disminuye la tasa binaria de la modulación. Lo contrario pasa con el alcance máximo, el cual aumenta al aumentar el valor de SF.

Definimos  $SNR_{min}$  como la relación señal a ruido mínima que permite al receptor demodular la señal correctamente. Por otro lado, podemos definir la sensibilidad del receptor como:

$$S_{re} = -174 + 10 * \log_{10} BW + SNR_{min} + NF \quad (1.2)$$

donde:

$BW$  = Ancho de banda del canal

$SNR_{min}$  = Relación señal a ruido mínima

$NF$  = Figura de ruido del receptor (6dB)

A modo de resumen, podemos ver en la tabla ?? un ejemplo con los valores teóricos de una implementación con un ancho de banda de 125 KHz.

SF	$SNR_{min}$ (dB)	Sensibilidad (dBm)	Tasa binaria (Kbps)
7	-7.5	-125	6.83
8	-10	-127	3.9
9	-12.5	-130	2.19
10	-15	-132	1.22
11	-17.5	-135	0.671
12	-20	-137	0.366

Tabla 1.1: Ejemplo teórico para un canal con 125 KHz de ancho de banda

#### 1.0.4. Objetivos y motivaciones

Con la popularización de LoRa, se ha extendido la utilización de LoRaWAN como protocolo de comunicación. LoRaWAN es un protocolo de red , creado por *LoRa Alliance*, que busca convertirse en un estándar.

El objetivo de este TFG será implementar un protocolo de comunicación, basado en LoRa, como alternativa a los ya existentes (LoRaWAN y SigFox). La principal motivación es el estudio de alternativas que impliquen un menor coste, eviten la utilización de hardware específico y que, en futuras implementaciones, soporten un mayor número de arquitecturas de red.

Tras esto, haré una diseño hardware de nodos, enfocados a bajo consumo, con transmisores LoRa. El objetivo será crear un nodo de tamaño reducido, poco consumo y que pueda ser utilizado en múltiples aplicaciones. También diseñaré placas con sensores que implementen estos nodos y que prueben su funcionamiento.

Por último haré un estudio del funcionamiento del protocolo, alcance máximo alcanzado y eficiencia energética de los nodos.

Cabe recalcar que la finalidad de este TFG no será crear un sistema que compita con LoRaWAN o con las redes que existen actualmente; lo que se busca, es aplicar los conocimientos sobre redes y sistemas de comunicación, adquiridos durante el grado, para implementar un protocolo de red y diseñar un hardware que lo soporte y probar el sistema final en un entorno real.

### **1.0.5. Metodología**

Con el fin de conseguir los objetivos planteados, se dividirá el trabajo en las siguientes fases:

- Estudio de las tecnologías inalámbricas actuales y sus protocolos.
- Estudio de LoRa y redes LPWAN
- Diseño teórico del protocolo y su funcionamiento.
- Elección de la plataforma hardware en la que se realizará el proyecto.
- Implementación del protocolo.
- Pruebas y depuración del protocolo mediante placas de desarrollo.
- Diseño y fabricación del hardware.
- Pruebas del sistema completo.

## Capítulo 2

# Diseño software

El protocolo de red es una de las piezas fundamentales en el desarrollo de este trabajo. En este apartado se realizará un análisis detallado de las especificaciones del protocolo, su formato de tramas y las máquinas de estados que gobiernan su funcionamiento.

Teniendo definidas las especificaciones, se describirá el proceso de implementación en la plataforma hardware diseñada: Herramientas de compilación y depuración, drivers utilizados y estructura del proyecto.

Por último, se realizarán pruebas de funcionamiento, con el fin de comprobar que se cumplen las especificaciones establecidas.

### 2.1. Diseño del protocolo

La idea principal es diseñar un protocolo poco pesado, diseñado especialmente para redes de sensores. Las siguientes características:

- Tramas de tamaños variables
- Configurable
- Extensible a diferentes arquitecturas de red
- Bidireccional
- Debe funcionar sin hardware específico
- Fácil utilizar

En esta primera versión del protocolo, implementaremos una red con una topología de estrella. Para esto, definiremos dos tipos o comportamientos diferentes para los dispositivos:

- **Nodos:** Son los dispositivos que contienen los sensores. Se conectan a la red administrada por un master.
- **Master:** Se encargan de administrar los nodos de la red y de redirigir los datos que le llegan a un servidor o base de datos.

Como se verá durante el desarrollo, muchos de los formatos de tramas y características utilizados, están inspirados en otros protocolos como el de ethernet y el de LoRaWAN.

### 2.1.1. Formato de trama

La trama enviada por el sistema está compuesta por una cabecera de 7 bytes seguida de un payload de tamaño variable entre 1 y 256 bytes. El tamaño de la carga estará limitado por la configuración de sistema.

La representación gráfica de la trama es la siguiente:

Net address	Dest address	Dev address	Mac type	Flags	P. size	payload
8b	16b	16b	4b	4b	8b	0-256b

1. **Net address** (8b) Identificador de la red en la que se llevan acabo las transacciones. Puede tomar valores entr 0 y 254, el valor 255 está reservado para transacciones broadcast.
2. **Dest address** (16b) Identificador del nodo/master destino.
3. **Dev address** (16b) Identificador del nodo/master que realiza la transacción. Este valor puede estar definido por defecto o ser asignada por defecto por un master.
4. **Mac type** (4b) Identificado del tipo de trama que se envía. Este puede ser:

Tipo	Identificador	Descripción
TX	0	Trama de datos
JOIN	3	Trama de conexión
ACK	1	Trama de asentimiento
NACK	2	Trama de Asentimiento negativo

5. **Flags** (4b) Flags de la trama. aún no usadas.
6. **P. Size** (8b) Tamaño de la carga de la trama (expresado en bytes)
7. **Payload** (0-254) Carga de la trama. Los datos que contenga depenederán del tipo de trama (Mac type).

#### 2.1.1.1. Trama de conexión

Al iniciarse un nodo, este intentará conectarse a una red (conocida o cercana). Para hacerlo, enviará un paquete cuyo payload contendrá el identificador único de nodo así como el formato de trama que utilizará en la transacción. El formato es el siguiente:

1. **UUID** (16b) Identificador único del nodo.

UUID	Item 1	Size 1	Item 2	Size 2		Item N	Size N
16b	6b	2b	6b	2b	.....	6b	2b

2. **Item N** (6b) Identificador del tipo de dato a enviar. puede ser:

Tipo	Identificador	descripción
LIGHT	0x0	sensor lumínico
PRESURE	0x1	Sensor de presión
HUMIDITY	0x2	sensor de humedad
TEMPERATURE	0x3	Sensor de temperatura
.	.	.
.	.	.
.	.	.
CUSTOM 1	0x3C	Uso personalizado
CUSTOM 2	0x3D	Uso personalizado
CUSTOM 3	0x3E	Uso personalizado

3. **Size N** Identifica el tamaño del **Item**. puede tomar los siguientes valores:

Identificador	Tamaño
0x0	1b
0x1	8b
0x2	16b
0x3	24b

#### 2.1.1.2. Trama de configuración

Cuando es enviada por un master, contiene la configuración que deberá utilizar el nodo. El formato de trama es el siguiente:

UUID	SLEEP TIME	POWER	BW	SF
16b	16b	8b	8b	8b

1. **UUID** (16b) Identificador único del nodo (este no cambia, solo se utiliza en la verificación)
2. **SLEEP TIME** (16b) Tiempo entre transacciones (segundos). durante ese tiempo el nodo estará dormido
3. **POWER** (8b) Potencia de salida de transmisión (menor o igual a XdBm)
4. **BW** (4b) Identificador del ancho de banda del canal. Pude tomar los valores:

Identificador	Ancho de banda (KHz)
0x1	125
0x2	250
0x3	500

5. **SF** (4b) Identificador del spreading factor\* del transmisor. Puede tomar los valores:

Identificador	SF
0x1	SF7
0X2	SF8
0x3	SF9
0X4	SF10
0x5	SF11
0X6	SF12

6. **RXW** (8b) valor de la ventana de recepción (en segundos).

### 2.1.1.3. Trama de datos

Los datos tendrán el orden y el tamaño definido en la primera conexión. Se puede expresar mediante el siguiente esquema:

DATO 1	DATO 2	DATO 3		DATO N
S1	S2	S3	.....	SN

### 2.1.2. Comportamiento del protocolo

En este apartado se definirá el comportamiento del sistema ante diferentes eventos. Entre ellos podemos encontrar:

#### 2.1.2.1. Conexión de un nodo a la red

Cuando un nodo pretenda conectarse a un red, enviará una trama de conexión con la información referente a los sensores que tiene implementados y el tamaño del dato. Este paquete se puede enviar en modo broadcast (si se desconoce la red a la que se pretende conectar) o especificando una dirección de red.

El master, al recibir una trama de conexión, responderá con una trama de configuración con los parámetros que usará el nodo durante su funcionamiento. El funcionamiento se puede resumir en el siguiente cronograma:

Figura 2.1: Cronograma de conexión de un nodo

#### 2.1.2.2. Envío y recepción de datos

Estando el nodo conectado a la red, podrá enviar datos al master siguiendo el formato de datos especificado en el proceso de conexión. El nodo podrá enviar tramas con una separación temporal mínima entre ellas. El tiempo de espera entre tramas la especifica el master en la trama de configuración (valor de *Sleep Time*).

Tras enviar un dato, el nodo abrirá una ventana de recepción, cuya duración se especifica en la trama de configuración (valor de  $RXW$ ). Durante ese tiempo, estará a la espera de datos del master. Existe la opción de hacer transacciones con asentimiento (ACK o NACK), pero no es recomendable en redes con muchos sensores debido a las limitaciones de uso del canal.

### 2.1.2.3. Tratamiento de errores

El sistema deberá ser robusto frente a errores. La forma en la que el sistema responderá a errores será la siguiente:

Nombre	descripción	solución
RX_ERROR	Paquete recibido dañado o con formato desconocido	Se incrementará el contador de errores. Nodo: Si se supera el número de máximo de errores volverá al proceso de JOIN. Master: descarta el paquete
RX_NACK	El nodo master ha recibido un paquete dañado o con formato desconocido	Se incrementará el contador de errores. Si se supera el número de máximo de errores volverá al proceso de JOIN
TX_TIMEOUT	Se ha superado el tiempo máximo para enviar un paquete	Se reconfigura la radio.
RX_TIMEOUT	Se ha superado el tiempo máximo para recibir un paquete	Si el sistema está configurado para recibir ACK, el sistema pasará a dormir un tiempo aleatorio antes de intentar enviar un nuevo paquete
CONF_ERROR	Existe un error en el paquete de configuración	Se carga la configuración inicial en el nodo y vuelve a start
PR_ERROR	Error al procesar la trama: Formato desconocido o datos incoherentes	Se descarta la trama y se carga un mensaje tipo NACK en el buffer de salida

## 2.2. Implementación

### 2.2.1. Fundamentos teóricos

#### 2.2.1.1. Sistemas embebidos

#### 2.2.1.2. Sistemas operativos de tiempo real (RTOS)

#### 2.2.1.3. Máquinas de estados extendidas (xFSM)

### 2.2.2. Descripción del entorno

#### 2.2.2.1. Drivers, Librerías y útiles

Para la implementación del sistema, se han utilizado diferentes drivers para facilitar el proceso de implementación. Estos drivers son los encargados de controlar y gestionar los diferentes periféricos del microcontrolador y el transceptor de LoRa. Los drivers utilizados han sido diseñados por STM y Semtech para sus dispositivos, entre ellos encontramos:

- **HAL drivers:** la *Hardware Abstraction Layer* es una iniciativa de STMicroelectronics creada con el fin de reducir el tiempo y dificultad de desarrollo de sistemas. Está compuesta por un conjunto de APIs (*Application Programing Interfaces* por sus siglas en inglés) que permiten inicializar, configurar y manipular los diferentes periféricos de los microcontroladores de STM. Estos drivers están accesibles a través de la red bajo una licencia de software libre BSD.
- **LoRa drivers:** Diseñados por Semtech, permiten y control de sus trancceptores de LoRa. Son accesibles a través de la red y, al igual que los drivers de STM, están bajo una licencia BSD.

Como base para el proyecto, se ha utilizado uno de los ejemplos elaborados por STM para su placa de desarrollo "B-L072Z-LRWAN1", la cual se ha utilizado durante el proceso de implementación. Para utilizar este ejemplo, ha sido necesario modificar el driver de transceptor, así como las secciones de configuración del hardware. La finalidad de las modificaciones era eliminar las dependencias internas con el middleware de LoRaWAN, el cual se encontraba implementado por defecto. También se modificó la estructura del proyecto y el sistema de compilación.

A parte de estas herramientas, también se han utilizado otras herramientas provenientes de terceros, para la implementación de las máquinas de estados y para el sistema operativo de tiempo real. Las herramientas son:

- **FreeRTOS:** Es una implementación del kernel de un sistema operativo de tiempo real, diseñado para sistemas embebidos. Su uso se ha popularizado por la cantidad de plataformas con la que es compatible, lo bien documentado que está y por estar bajo una licencia de software libre MIT.
- **FSM:** Es una implementación de una maquina de estados, realizada por — e inspirada en la maquina propuesta por —. Es una implementación sencilla y fácilmente escalable.



El middleware de FreeRTOS ha sido implementado utilizando la configuración propuesta por STM para su microcontrolador. Por otro lado, la implementación del FSM ha sido modificada para comprobar bits de estado y no funciones en las transiciones de estados.

#### 2.2.2.2. Herramientas de edición, compilación y depuración

El software de este TFG se ha desarrollado completamente sobre un sistema operativo linux pero, procurando su compatibilidad con otros sistemas operativos. Para ello, se han utilizado herramientas software libres y multiplataforma.

Para configurar el proyecto se ha utilizado *Cmake*. Cmake es una herramienta libre y multiplataforma diseñada para compilar y probar software. Esta herramienta permite generar ficheros de configuración con independencia del sistema operativo y el compilador utilizado.

Como compilador se ha utilizado el conjunto de herramientas *GCC ARM Embedded*. GCC ( GNU Compiler Collection por sus siglas en inglés), es un conjunto de compiladores y librerías, creados por el proyecto GNU, que permiten la compilación de código en C, C++, Ada y Fortran. Concretamente, se ha utilizado la variante para arquitecturas ARM, la cual permite la compilación cruzada del código.

Para depurar el código, se han utilizado las herramientas GDB y STLINK. STLINK es un circuito integrado que permite la depuración y programación de microcontroladores STM32. Por otro lado, GDB (GNU Debugger) es un depurador de lenguajes como C y C++ que permite ejecutar instrucción a instrucción el programa y conocer, en cada momento, el estado de los registros y memoria.

#### 2.2.3. Modelado del comportamiento de los nodos

#### 2.2.4. Modelado del comportamiento del nodo maestro

#### 2.2.5. Estructura del código

#### 2.2.6. Proceso de diseño e implementación

#### 2.2.7. Resultados obtenidos



## Capítulo 3

# Conclusiones y líneas futuras

### 3.1. Conclusiones



# Bibliografía

- Cypress (2015). Bluetooth low energy (ble). Technical report, Cypress Semiconductor Corporation.
- et al., M. F. (2016). Energy efficiency of the internet of things. Technical report, IEA 4E Electronic Devices and Networks Annex.
- Minihold, R. (2014). Analisis of wifi and wimax and wireless network coexistence. Technical report, School of Computing, Teesside University.
- SEMTECH (2015). Lora modulation basics. Technical Report 2, Semtech Corporation.
- Song, S. and Issac, B. (2011). Near field communication (nfc) technology and measurements. Technical report, Rohde and Schwarz.
- Torchia, M. (2019). Idc forecasts worldwide spending on the internet of things to reach 745 billion in 2019, led by the manufacturing, consumer, transportation, and utilities sectors. *IDC Research Reports*.