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 Titulacin 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, DIA 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

Índice de figuras

Índice de tablas

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 procesado de datos, ha propiciado una mayor demanda por parte de la industria.

El término IoT (Internet of things hace referencia a este fenómeno y que consiste en la conexión de elementos cotidianos a la red. A día de hoy, se puede encontrar desde bombillas hasta extensas redes de farolas interconectas.

Según datos del IDC (International Data Corporation), para 2019 se prevé que la inversión mundial en IoT supere los $745 mil millones de dolares, lo que supone un 15,4 % más que el año pasado. Se espera que el ritmo de crecimiento continue 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, se expndrán brevemente las tecnologías utilizadas actualmente en la industria así como sus ventajas e inconvenientes. Seguidamente, se explicarán los conceptos de red WSN (Wireless Sensor Network) y de red LPWAN (Low Power Wide Area Network). Por último, se centrará en la tecnología que se utilizará para el desarrollo de este TFG, asi como en los objetivos planteados y la metodología aplicada.

1.1. Estado del arte

En el mercado actual podemos encontrar un amplio abanico de tecnologías de comunicación inalámbrica. Aunque en ocaciones estas tecnologías difieren mucho unas de otras, podemos caracterizarlas y compararlas mediante tres factores fundamentales: potencia, alcance y ancho de banda. Estos, tienen fuertes interdependencias entre si,

lo que hace complicado cumplirlos todos a la vez. Esta problemática se ve reflejada en la figura 1.1: si se requieren dos de los factores se tiene que renunciar al tercero.

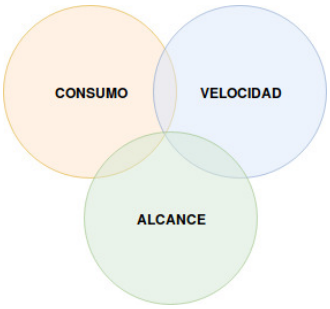


Figura 1.1: Parámetros fundamentales de un sistema inalambrico

Algunas de las tecnologías empleadas en IoT son:

- **NFC** (Near Field Communication): 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 alcanca velocidades de transmisión de hasta 848Kbps, dependiendo del entorno en el que se lleve acabo 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 mas 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): 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): Tecnología basada en el estándar IEEE 802.16, la cual puede dar servicio a redes de area 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.

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

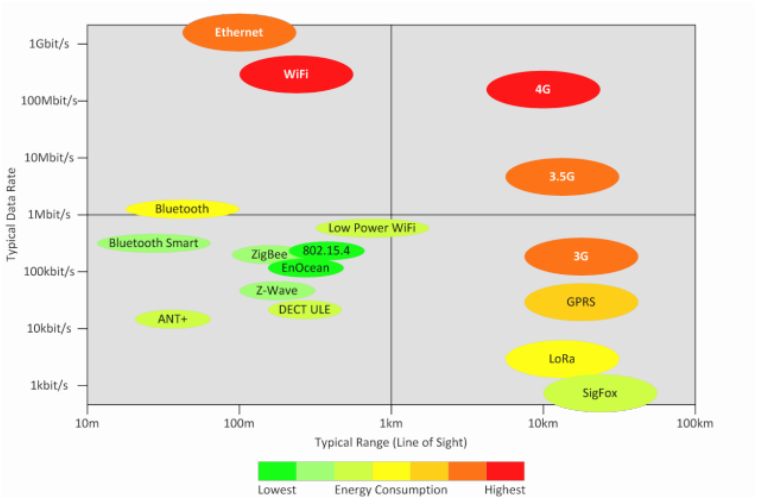


Figura 1.2: Relación de consumo, velocidad y alcance de las diferentes tecnologías [Friedli (2016)]

1.2. Redes WSN

Las redes inalámbricas de sensores o redes WSN (*Wireless Sensor Network* por sus siglas en inglés), se caracterizan por estar conformadas por dispositivos de bajo coste y bajo consumo (a los que nos referirémos como *nodos*) y que se utilizan para tareas de monitorización y control.

El modo en el que se conectan e interactuan los nodos, está determinado por su topología de red, la cual define el mapa físico o lógico de los nodos en la red. Existe una extensa variedad de topologías, las cuales están resumidas en la figura 1.3. La topología de una red, determinará en gran medida el consumo y coste de los nodos, la utilización de los canales de transmisión y el número máximo de nodos que pueden coexistir en una red.

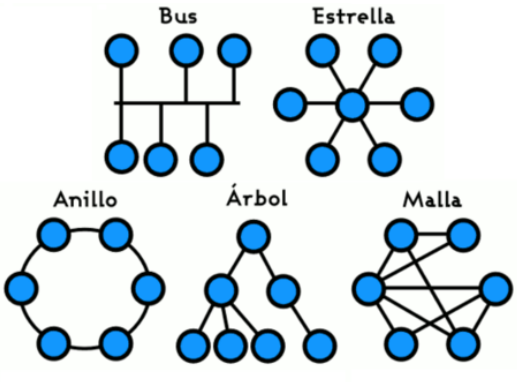


Figura 1.3: Topologías de red más usadas

Las redes WSN suelen estar conformadas por cientos o miles de nodos distribuidos

en extensas areas geográficas. Estos factores, hacen que que la manipulación directa de cada uno de los nodos sea un proceso costoso, largo y complicado. Para evitar esta problemática, los nodos suelen tener sistemas robustos frente a fallos y autonomías que llegan a durar meses o incluso años. Todo esto es posible gracias a la ulización de diferentes técnicas de transmisión, con las utilizadas en las redes LPWAN.

1.2.1. Redes LPWAN

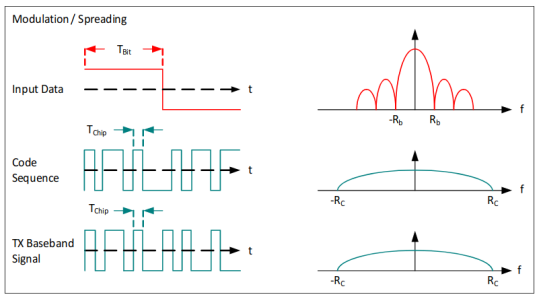
Se definen como redes de bajo consumo y largo alcance (LPWAN de sus siglas en inglés) al conjunto de redes inalámbricas caracterizadas por tener largos alcances, consumo mínimo y baja tasa binaría. Su bajo consumo posibilita implementarlo sobre dispositivos alimentados mediante baterías.

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 uso a sistemas con interfaces máquina a máquina (M2M de sus siglas en 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.

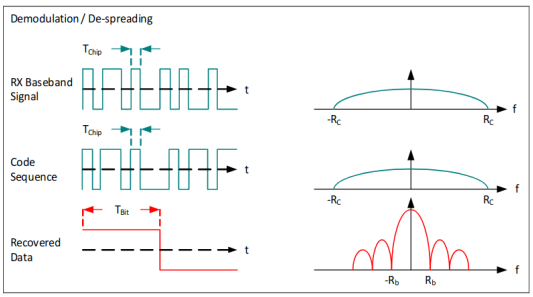
Para conseguir estas características, las redes LPWAN utilizan diferentes técnicas como:

- **Ultra Narrowband (UNB):** Traducido como "banda ultra-estrecha". Su funcionamiento se basa en la utilización de canales con poco ancho de banda para trasmitir.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.

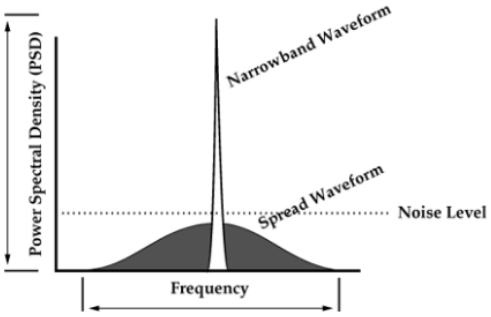
- **Spread Spectrum (SS):** Se trata de una modulación de espectro ensanchado en la cual, a cada canal, se le otorga un ancho de banda mayor al estrictamente necesario para funcionar. Al aumentar el ancho de banda, aumenta la energía por bit utilizada, mejorando el ratio señal a ruido en el receptor.



(a) Modulación, ensanchado del espectro



(b) Demodulación, ensanchado del espectro



(c) Comparativa entre banda estrecha y espectro ensanchado

Figura 1.4: [(SEMTECH, 2015)]

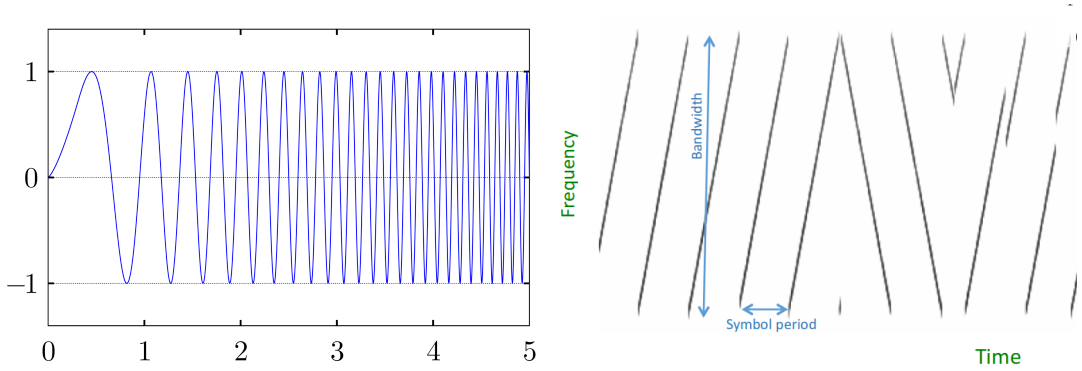
1.2.2. LoRa

LoRa (Long Rate) 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 for la empresa francesa Cycle y posteriormente adquirida por Semtech Corporation.

Un chirp es una señal sinusoidal cuya frecuencia varia con el tiempo. La modulación chirp de espectro ensanchado se basa en la utilización de pulsos chirp lineales para modular la señal. La utilización de estos 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 permite elegir ente 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:



(a) Representación de un chrip lineal (b) Representación de una señal modulada con LoRa

Figura 1.5: [Telkamp (2015)]

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

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

El SNR_{min} es la relación señal a ruido mínima que permite al receptor demodular la señal correctamente. Por otro lado, se define definir la sensibilidad del receptor como:

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

donde:

BW= Ancho de banda del canal
SNR_{min} = Relación señal a ruido mínima
NF = Figura de ruido del receptor (6bB)

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

1.3. 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 que eviten la utilización de hardware especifico y que, en futuras implementaciones, soporten un mayor número de arquitecturas de red.

Seguidamente, se propondrá una diseño hardware de nodos, enfocados a bajo consumo, que utilicen transmisores LoRa. El objetivo será crear un nodo de tamaño reducido, poco consumo y que pueda ser utilizado en múltiples aplicaciones. También se diseñarán placas con sensores que implementen estos nodos y que prueben su funcionamiento. Por último se realizará un estudio del funcionamiento del protocolo, alcance máximo y eiciencia 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.

1.4. 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 hardware

2.1. Herramientas de diseño

Para la implementación hardware del sistema se ha utilizado la herramienta *Altium Desinger*: Altium es un potente entorno de desarrollo que incluye todas las herramientas necesarias durante el proceso de diseño, prueba y fabricación de un prototipo hardware.

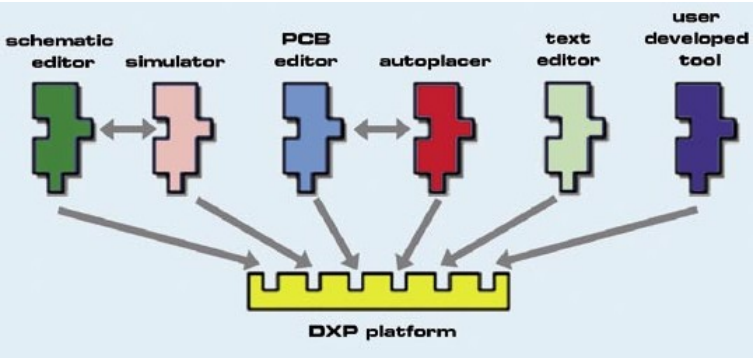


Figura 2.1: Componentes de Altium DPX

Altium es uno de los softwares de diseño hardware más utilizado actualmente por la industria, además cuenta con una gran cantidad de documentación, la cual facilita en uso y el aprendizaje. Todas estas ventajas han hecho que Altium sea la herramienta elegida para desarrollar este apartado del TFG.

Con Altium se han llevado a cabo las tareas de diseño de esquematicos, creación de componentes y footprints, diseño y rutado de la tarjeta y la generación de los ficheros de fabricación. Aunque Altium es un software propietario con un coste elevado, para la realización de los prototipos se han utilizado licencias de prueba del software.

2.2. Elección del hardware

2.2.1. Integrado CMWX1ZZABZ

El integrado CMWX1ZZABZ es un módulo diseñado por *Murata*, que incorpora un microcontrolador STM32L0 y un tranceptor de radio SX1276 en el mismo chip.

La forma en la que se encuentran conectados ambos módulo en el chip, se resume en la figura 2.2 .

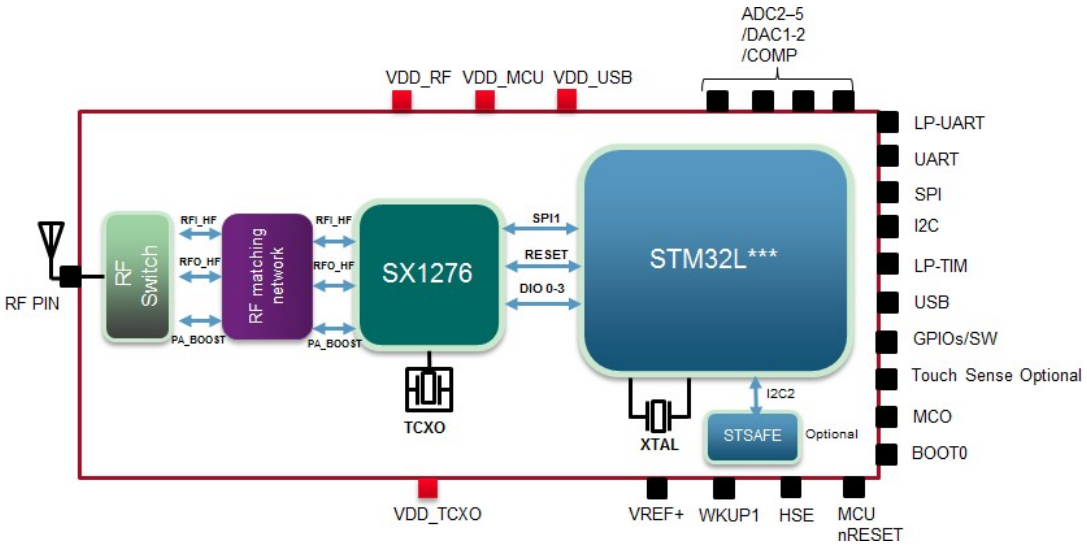


Figura 2.2: Esquema de bloques del chip CMWX1ZZABZ

La principal ventaja de este chip, es que reduce en gran medida en espacio utilizado, reduciendo el tamaño final de la tarjeta y simplificando las conexiones.

2.2.1.1. Microcontrolador STM32L072

El STM32l072, es un microcontrolador de ultrabajo consumo, diseñado por la empresa *STMicrocontrollers*. Cuenta con una arquitectura de Arm Cortex-M0+ de 32 bits funcionando a 32 MHz, además 192 KB de memoria flash, 6 KB de memoria de datos y 20 KB de memoria RAM. Este microcontralador tiene un amplio abanico de perifericos como SPI, I^2C, USART, USB, ADC, USB entre otros.

Lo más destacable de este microcontrolador es ultra-bajo consumo, y sus diferentes modos de funcionamiento, los cuales podemos ver resumido en la figura 2.3. El bajo consumo y altas prestaciones de este chip, lo hacen ideal para este proyecto.

2.2.1.2. Transceptor SX1276

El SX1276 es un tranceptor de radio, diseñado por *Semtech*, preparado para trabajar con la modulación lora en sistemas de largo alcance y bajo consumo. También puede funcionar como modulador FSK/OOK.

Se carateriza por trabajar en un rango de frecuencias desde 137 MHz hasta los 1020 MHz, con un *"spreading factor"* o factor de enchanchado entre 6 y 12, un ancho de banda entr 7.5 KHz y 500 KHz y una sensibilidad entre -111 dBm y -148 dBm. Con todo esto, consigue velocidades binarias efectivas ente los 18 bps y los 37.5 Kbps.

Por otro lado, el trasmisor tiene un consumo típico de 10.8 mA en recepción, de entre 20 mA y 120 mA en recepción (dependiento de la configuración) y un consumo

Low power mode	Consumption	CPU	Flash / EEPROM	RAM	DMA & Peripherals	Clock	LCD	RTC
Sleep	41 µA/MHz (Range 1)	No	ON	ON	Active	Any	Available	
	36 µA / MHz (Range 2)							
	35 µA/MHz (Range 3)							
Low power run	8.55 µA (Flash OFF, 32 kHz)	Yes	ON or OFF	ON	Active	MSI	Available	
Low power sleep	4.65 µA (peripherals OFF)	No	OFF	ON	Active	MSI	Available	
Stop with RTC	0.82 µA (1.8 V)	No	OFF	ON	Frozen	LSE, LSI	OFF	ON
	1.0 µA (3 V)							
Stop	415 nA	No	OFF	ON	Frozen	-	OFF	OFF
Standby with RTC	655 nA (3 V)	OFF	OFF	OFF	OFF	LSE	OFF	ON
	845 nA (1.8 V)							
Standby	290 nA	OFF	OFF	OFF	OFF	-	OFF	OFF

Figura 2.3: Consumo y modos de funcionamiento del microcontrolador STM32L072

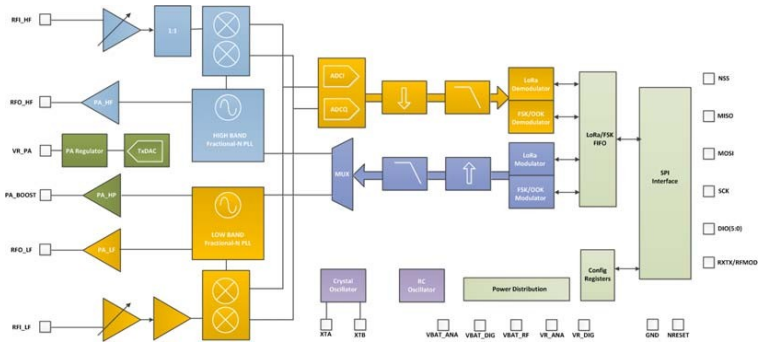


Figura 2.4: Esquema de bloques de la familia de tranceptores SX127x

típoco en modo *sleep* de 0.2 uA.

2.2.2. Conversor analógico/digital AD7194

El AD7194 es un conversor analógico/digital de bajo ruido diseñado para aplicaciones de alta presición. Creado por *Analog*, posee 16 canales de entrada analógicos, una resolución de 24 bits, un reloj interno de 4.92 MHz y un amplificador de entrada de ganacia programable. La comuicación se hace a través de la interfaz SPI del ADC y el SPI-2 del microcontrolador.

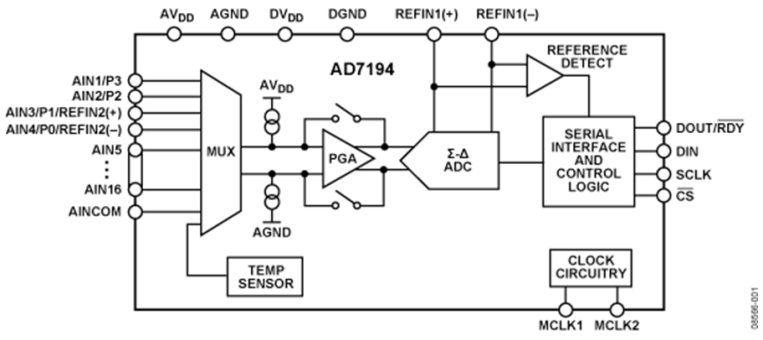


Figura 2.5: Diagrama funcional del AD7194

Entre sus opciones de configuración, permite la utilización de filtros digitales programables, rechazo de las bandas de 50 Hz y 60 Hz y la utilización de sus canales en modo diferencial (en cualquier configuración) o en modo pseudo-diferencial (tomando como canal negativo GND). Otra de las característica decisivas, es su consumo, que varia entre los 0.85 mA y los 5.3 mA en función de la ganancia de entrada elegida.

La motivación de usar este ADC, es su aplicación: La medida de termopares. En esta aplicación, se hace necesario una gran presición (las variaciones de tensión en los termopares es muy pequeña) y el bajo ruido de cuantificación. Además, al tener 16 de canales, se pueden utilizar un mayor número de termopares.

2.2.3. Sensor de temperatura MCP9808T

El MCP9808T es un sensor de temperatura digital, que permite medir termperaturas entre -20 ºC y 100 ºC con una presición de $\pm 0{,}25^{o}C$. Entre su características destaca su comunicación a través de I^2C, sus diferentes modos de operación y su bajo consumo, el cual ronda los 200 uA en funcionamiento.

Su finalidad en el proyecto, será aproximar la temperatura de unión de los conectores a los termopares, la cual es necesaria para realizar una correcta medida (la union de cobre + estaño de los conectores genera una diferencia de potencial parácita que influye en las medidas).

2.2.4. Componentes activos

- **Regulador TLV755P:**
- **Regulador AD1582BRTZ:**
- **Operacional MCP6001T:**

2.2.5. Componentes pasivos

2.3. Diseño las tarjetas

2.3.1. Fase 1: nodo básico

En esta parte se diseñará un nodo general, sin aplicación específica que contendrá lo necesario para un correcto funcionamiento del microcontrolador y la radio. Se procurará minimizar el tamaño y número de componentes.

2.3.1.1. Esquemáticos

2.3.1.2. Layout

2.3.2. Fase 2: Tarjeta de expación

2.3.2.1. Esquemáticos

2.3.2.2. Layout

2.4. Lista de componentes

2.5. Fabricación y montaje

Capítulo 3

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.

3.1. 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:

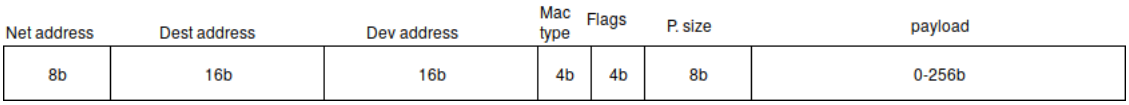
- **Nodos:** Son los dispositivos que contienen los sensores. Se conectan a la red administrada por un master.
- **Master:** Se encargan de administrar los nodos de la red y de redirigir los datos que le llegan a un servidor o base de datos.

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.

3.1.1. 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:



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.

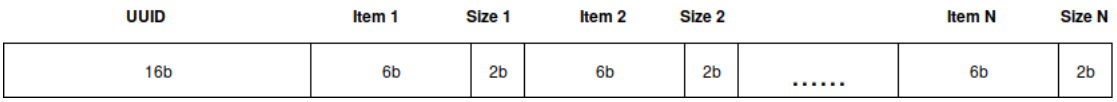
6. **P. Size** (8b) Tamaño de la carga de la trama (expresado en bytes)

7. **Payload** (0-254) Carga de la trama. Los datos que contenga depenederán del tipo de trama (Mac type).

3.1.1.1. 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.



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

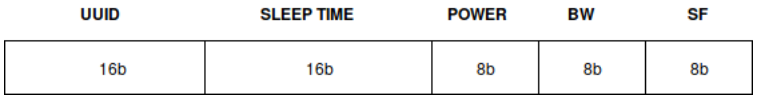
Tipo	Identificador	descripción
LIGTH	0x0	sensor lumínico
PRESURE	0x1	Sensor de presión
HUMIDITY	0x2	sensor de humedad
TEMPERATURE	0x3	Sensor de temperatura
.	.	.
.	.	.
.	.	.
CUSTOM 1	0x3C	Uso personalizado
CUSTOM 2	0x3D	Uso personalizado
CUSTOM 3	0x3E	Uso personalizado

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

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

3.1.1.2. Trama de configuración

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



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 trasmisión (menor o igual a XdBm)
4. **BW** (4b) Identificador del ancho de banda del canal. Pude tomar los valoes:

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

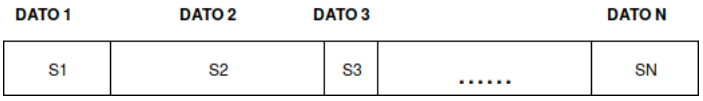
5. **SF** (4b) Identificador del spreading factor* del transmisor. Puede tomar los valores:

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).

3.1.1.3. Trama de datos

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



3.1.2. Comportamiento del protocolo

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

3.1.2.1. 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 3.1: Cronograma de conexión de un nodo

3.1.2.2. 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 un 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.

3.1.2.3. Tratamiento de errores

El sistema deberá se robusto frente a errores. La forma en la que el sistema responderá a erroes 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 numero 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 numero 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 el 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

3.2. Implementación

3.2.1. Fundamentos teóricos

3.2.1.1. Sistemas embebidos

3.2.1.2. Sistemas operativos de tiempo real (RTOs

3.2.1.3. Máquinas de estados extendidas (xFSM)

3.2.1.4. Colas circulares

3.2.2. Descripción del entorno

3.2.2.1. 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 (*Aplication Programin Interfaces* por sus siglas en inglés) que permiten inicializar, configurar y manipular los diferentes periferícos 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 tranceptores 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 tranceptor, 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 de 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.

3.2.2.2. 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.

3.2.3. Modelado del comportamiento de los nodos

El sistema se ha modualdo mediante máquinas de estado extendidad cuyo comportamiento de los nodos se puede ver resumido en la figura 3.2. La función principal de los nodos será enviar datos referentes a los sensores de forma periódica al nodo maestro. El comportamiento del nodo cuenta con los siguientes 8 estados:

- **JOIN:** Los nodos intentarán establecer una conexión con un master. Para ello, enviaran un paquete de tipo JOIN como el especificado en la sección 3.1.1.1. Al llegar un paquete con estas características, el master responderá con un paquete tipo CONFIG con la configuración del nodo, siguiendo el formato visto en la sección 3.1.1.2 . Si el paquete de configuración contiene información válida, el sistema iniciará su funcionamiento habitual. El sistema solo volverá a JOIN si se reinicia ó, si la función esta habilitada, si supera el número máximo de paquetes tipo NACK definido.

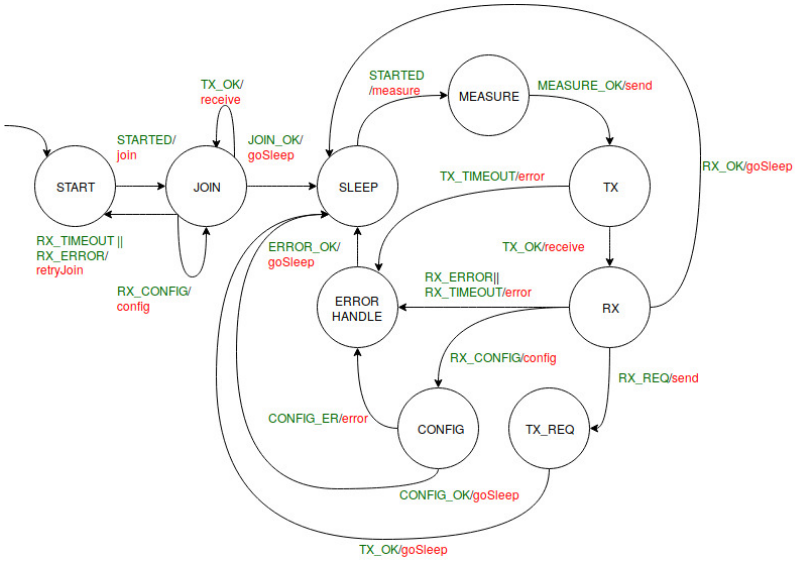


Figura 3.2: Esquema de modelado de los nodos mediante máquinas de estado

- **SLEEP:** El sistema entrará en bajo consumo durante el tiempo definido. En este estado, se deshabilitaran los recursos no esenciales con el fin de minimizar el consumo del sistema.
- **MEASURE:** El sistema procesará las entradas de los sensores y cargará los datos en el buffer de salida. El formato de trama tendrá que coincidir con el especificado en proceso de JOIN, de lo contrario el nodo master responderá con un mensaje tipo NACK.
- **TX:** El sistema enviará los elementos del buffer de salida. En caso de error, este será tratado.
- **RX:** Tras enviar los datos, se abrirá una ventana de recepción a la espera de un paquete de datos. En caso de error, este será tratado.
- **CONFIG:** Si el el paquete recibido es de tipo configuración (ver sección 3.1.1.2), esta se cargará en el sistema y será utilizada en la siguiente transacción.
- **TX_REQ:** Si el paquete recibido es del tipo request info, se cargará la información del sistema en un paquete tipo configuración y se enviará al nodo master. (TODO -¿revisar)
- **ERROR_HANDLE:** Si ocurriese algún error durante la ejecución de alguno de los procesos, este se capturara y se intentará buscarle una solución. La forma en la que se tratarán los erroes, se encuentra especificada en la sección 3.1.2.3.

3.2.4. Modelado del comportamiento del nodo maestro

A diferencia de los nodos vistos en el apartadado 3.2.3, el comportamiento de los nodos maestros se ha modelado mediante dos máquinas de estados. La principal motivación de este cambio, será separar la funciones de recepción y transmisión de paquetes de las funciones de procesado de datos, de esta forma, evitamos la perdida de

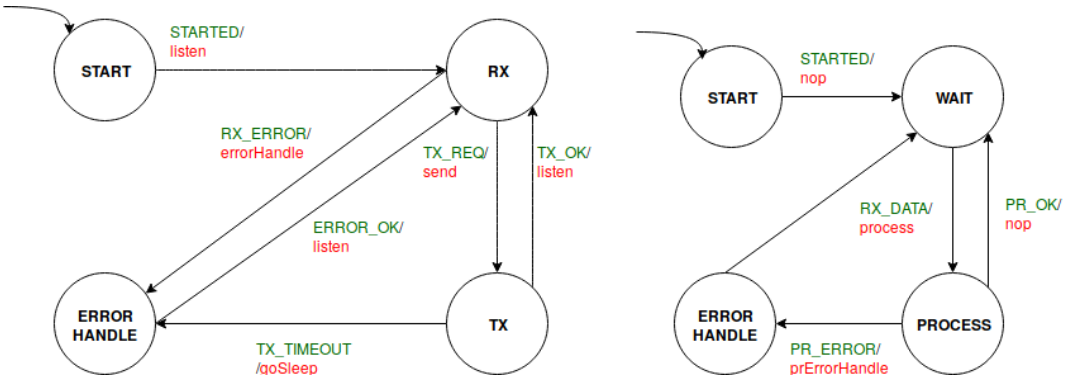


Figura 3.3: Modelado del comportamiento mediantes maquinas de estado extendidas

datos durante procedentes de los nodos. Todo esto es posible gracias a las posibilidades que nos ofrecese FreeRTOs y las máquinas de estados extendidas.

los estados de la máquina de recepción de datos son los siguientes:

- **RX:** Este es el estado por defecto del receptor. Se encuentra siempre escuchando el canal a la espera de datos. Cuando recibe datos los encola en una cola circular y avisa a la segunda máquina de la prescencia de datos.
- **TX:** Cuando la segunda máquina notifica la existencia de datos en el buffer de salida, esta envía los datos y vuelve a su estado por defecto.
- **ERROR_HANDLE:** Recupera al sistema de los errores notificados por otros estados.

Por otro lado, los estados asociados al procesado de datos son los siguientes:

- **WAIT:** Se matiene a la espera de datos.
- **PROCESS:** Cuando existen datos encolados, los procesa y genera un respuesta.
- **ERROR_HANDLE:** Recupera el sistema de un error.

Los estados y las transiciones asociadas de cada máquina se pueden ver resumidos en la figura 3.3.

3.2.5. Estructura del código

3.2.6. Proceso de diseño e implementación

3.2.7. Resultados obtenidos

Capítulo 4

Conclusiones y líneas futuras

4.1. Conclusiones

Bibliografía

Cypress (2015). Bluetooth low energy (ble). Technical report, Cypress Semiconductor Corporation.

Friedli, M. (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.

Telkamp, T. (2015). Lora, lorawan, and the challenges of long-range networking in shared spectrum. Technical report, Cognitive Radio Platform NL,.

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*.